vizrt

Viz Artist
User's Guide

Product Version 3.5

December 10, 2013

Disclaimer

Vizrt provides this publication "as is" without warranty of any kind, either expressed or implied.

This publication may contain technical inaccuracies or typographical errors. While every precaution has been taken in the preparation of this document to ensure that it contains accurate and up-to-date information, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained in this document.

Vizrt's policy is one of continual development, so the content of this document is periodically subject to be modified without notice. These changes will be incorporated in new editions of the publication. Vizrt may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

Vizrt may have patents or pending patent applications covering subject matters in this document. The furnishing of this document does not give you any license to these patents.

Technical Support

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

Last Updated

December 10, 2013

Contents

1 Introduction

This guide will help you to understand the Viz Artist user interface, and how to perform certain tasks.

This section contains information on the following topics:

- Document Structure
- Related Documents
- Contact Vizrt

1.1 Document Structure

Sections 1 and 2 introduce Viz Artist and its basic concepts. The Workbench section gives an overview of the Main Window and Main Menu in Viz Artist, and also a description of the Scene Editor and the Scene Editor Views that display the scene from different camera angles.

Section 3 describes how to Manage Work and Viz Graphics Hub.

Section 4 describes the Viz scene tree, and gives you the Tree Knowledge and Skills needed in order to familiarize yourself with the scene tree and its workings.

Section 5 describes how to perform global tasks such as working with the Scene Settings, Implementing Video, Searching, Importing, Archiving and Post Rendering of graphics to video clips and/or still images.

Section 6 describes how to work with Lights and Materials.

Sections 7 and 8 describe how to work with the Animation Stage when Creating Animations.

Section 9 describes how to work with Cameras in scenes used in 2D and 3D (stereoscopy) broadcast production.

Section 10 describes all configuration properties and parameters and how to use them when Configuring Viz Artist and Viz Engine.

Sections 11, 12, 13, 14, 15, 16 and 17 describe all the built-in plug-ins. Most of the plug-in sections also have small how to procedures with illustrations that will help you to get started with your scene design.

Section 18 describes the On Air Mode that is used for testing and to performance analyze Viz scenes.

Section 19 describes the core aspects of Transition Logic scene design with a small tutorial at the end to help you get started with your amazing transition logic designs.

Section 20 gives an extensive Glossary with simple and to the point explanations of keywords and acronyms referenced throughout the document.

1.2 Related Documents

1. Viz Artist Script reference: Contains information on how to create scripts for a scene.
2. Viz Engine Administrator’s Guide: Contains information on how to install the Viz Engine software and supported hardware.
3. Viz Engine Plug-in SDK reference: Contains information on how to create a customized Viz plug-in.

1.3 Contact Vizrt

We encourage your suggestions and feedback regarding the product and this documentation. Our support E-mail address is support@vizrt.com. You are also welcome to use the web-based support at www.vizrt.com, or to contact your local Vizrt support person by telephone.

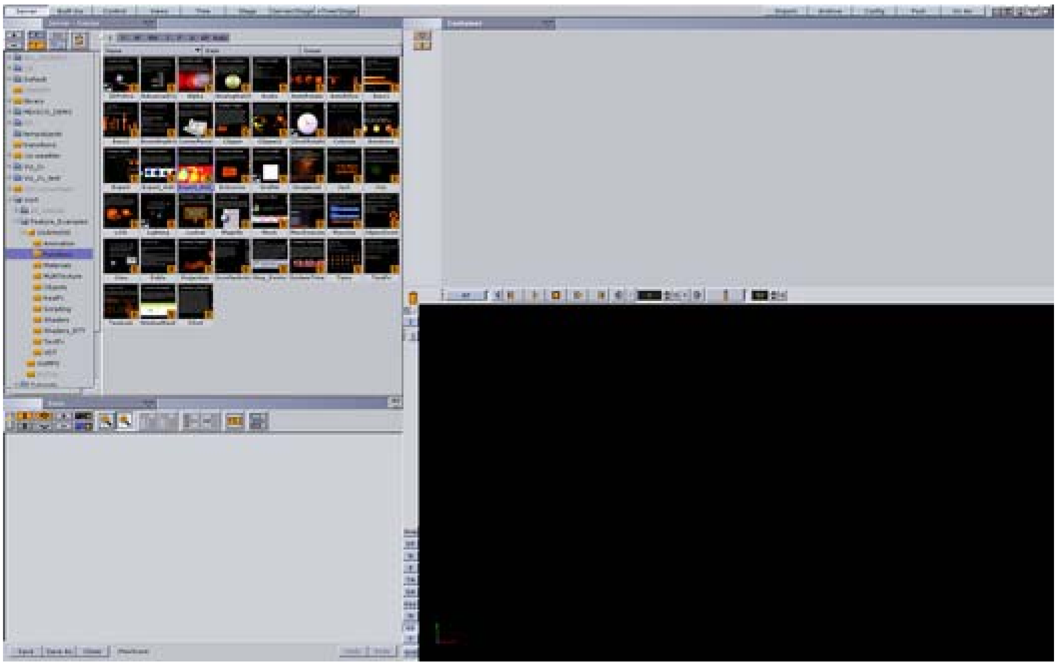
2 Workbench Description

This section contains information on the following topics:

- Main Window
- Main Menu
- Built-in Menu
- Scene Editor
- Working with the Scene Editor
- Scene Editor Views
- Script Editor
- Keyboard and Mouse Shortcuts

2.1 Main Window

After a successful database login, the Main window appears.



At the top of the Main window is the Main menu. Below the Main menu is the Display area. While the Main menu is always visible at the top, the content of the Display area vary according to the selection in the Main menu. At system startup (and when Server has been selected from the Main menu) the area is divided into four panes; Server (upper left corner), Scene Tree (lower left corner), Property (upper right corner), and Output (lower right corner). From the Server area it is possible to control database elements such as materials, images, and fonts. The elements in the database can be added to the scene tree, which is the logical visualization of the scene. The scene tree is built up of containers, which again can hold items such as materials or plug-ins. The containers can be controlled from the Property area, where size, position and other settings can be modified. The actual visualization of the scene is available in the Output area.

2.2 Main Menu

The buttons in the Main menu makes it possible to change the view of the Display area.

- **Server** – Shows the following panes; Server (top left), Scene Tree (bottom left), Property (top right), and Output (bottom right). From the Server area it is possible to control the database content such as scenes, materials, and images. The elements in the database, which is common for all users connected to the database, can be added to the scene. Server is the default view that appears at system startup. For more information, see:
  - Manage Work and Viz Graphics Hub
  - Server Menu
  - Server Tree Menu

- **Built Ins** – Shows the following panes; Plug-ins (top left), Scene Tree (bottom left), Property (top right), and Output (bottom right). From the Plug-ins area it is possible to control the built in plug-ins such as geometry, container, shader, texture and scene plug-ins. The features vary according to which plug-in packages that are licensed on each machine. For more information, see:
  - Built-in Menu
  - Adding Plug-ins
  - Plug-in

- **Control** – Shows the following panes; Controls (top left), Scene Tree (bottom left), Property (top right), and Output (bottom right). The Control area shows control objects and channels. For more information, see:
  - Control

- **Views** – Shows the following panes; Views (left), Property (top right), and Output (bottom right). The Views area shows three configurable camera views, in addition to the one in the Output area. The additional camera views make it possible to monitor the scene from up to four various angles simultaneously. For more information, see:
  - Scene Editor Views.

- **Tree** – Shows the following panes; Scene Tree (left), Property (top right), and Output (bottom right). The Tree view is similar to the Server view, but without the database appearance. This option is practical when the scene tree is complex and holds many containers. If the scene tree is still not big enough in the regular Tree view, it is possible to click the Zoom Out button above the scene tree. This will shrink the size of all the containers, so that even more containers fit in. For more information, see:
  - Tree Knowledge and Skills

- **Stage** – Shows the following panes; Stage (top), Scene Tree (bottom left), and Output (bottom right). From the Stage it is possible to control the animations in the scene. The Stage shows the timeline from animation start (left side) to stop (right side). For more information, see:
  - Animation Stage
  - Creating Animations

- **Server/Stage** – Shows the following panes; Stage (top), Server (bottom left), and Output (bottom right). The Server/Stage view is similar to the regular Stage view, but here the scene tree has been replaced with the database content.

- **+Tree/Stage** – Shows the following panes; Stage (top), Server (bottom left), Scene Tree (bottom center), and Output (bottom right). The +Tree/Stage view is only available on machines with a screen resolution of 1600x1200 or higher, and shows both the database content and the scene tree.

- **Import** – Opens the Import pane. From this pane it is possible to transfer archives and various files into the database. For more information, see:
  - Importing

- **Archive** – Opens the Archive pane. From this pane it is possible to archive multiple elements from the database to a single file. For more information, see:
  - Archiving

- **Config** – Opens the Configuration pane. From the Configuration pane it is possible to control various options regarding the program functionality when Configuring Viz.

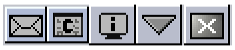
Note: The appearance of the Configuration interface varies according to the platform type. A Viz Artist 3 installation on a machine that contains a video card will run the software in Video mode, while a machine with no video card will run the software in Standard PC mode. To find out which platform type that applies to a specific installation, select the License Information button from the Main menu, and see the platform information in the window that appears. In the Configuration, an installation in Video mode has a few additional sections compared to the Standard PC mode, for example Video Board and Video Input.

Tip: Alternatively, the Configuration can be launched as a separate application from *Start > All Programs > Vizrt > Viz 3 > Config. Note that after altering the configuration settings, the system should always be restarted in order for the changes to take effect.*

- **Post** – Opens the Post Rendering pane. From this pane it is possible to export image chains or scene clips. For more information, see:
  - Post Rendering.

- **On Air – Opens the On Air panel. This panel** switches the application from the modelling tool to a plain rendering engine. The application waits for control commands through its communication port, or it can be controlled manually from the control buttons when Viz is set in On Air Mode.

Note: By default, the port number is set to *6100.* The port number and other communication settings can be modified from the Communication section of the Configuration.



At the right side of the Main menu are five additional icons.

- **Message Box** - Opens the Message window. This window displays server warnings, and chat messages can be sent/received to/from other users connected to the database.

- **Show Commands** - Opens the Viz Engine Console window. This window displays the commands that are used for the communication between the user interface and the rendering Viz Engine. The console window is mainly used for debugging purposes.

Tip: Alternatively, the Viz Engine Console window can be opened by pressing the keyboard shortcut ALT+C. Note that the pointer must be over certain areas in the GUI (for example the Server area) for this to work.

- **License Information** - Opens the License Information window. This window shows the status of the current license such as license expiration date, platform type (Standard PC or Video), licensed plug-in packages, Viz version number, etc. This is also where the Viz Artist 3 license can be renewed. For details on how to renew the license, see the *Viz Engine Manual*.

- **Minimize** - Minimizes the Viz Artist 3 window.

- **Close** - Closes the Viz Artist 3 window.

Tip: Alternatively, use the keyboard shortcut double-ESC to close the window.

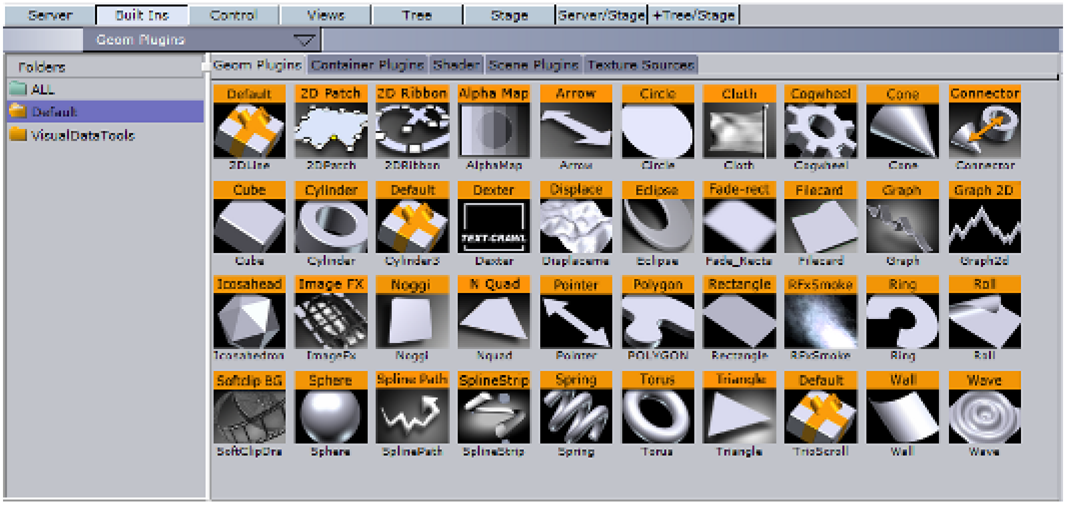
2.3 Built-in Menu

The Built-in button on the main menu provides access to the various plug-ins. These are:

- Geometry Plugins
- Container Plugins
- Shader Plugins
- Scene Plugins
- Texture Sources

See Also

- Built-in View Shortcuts



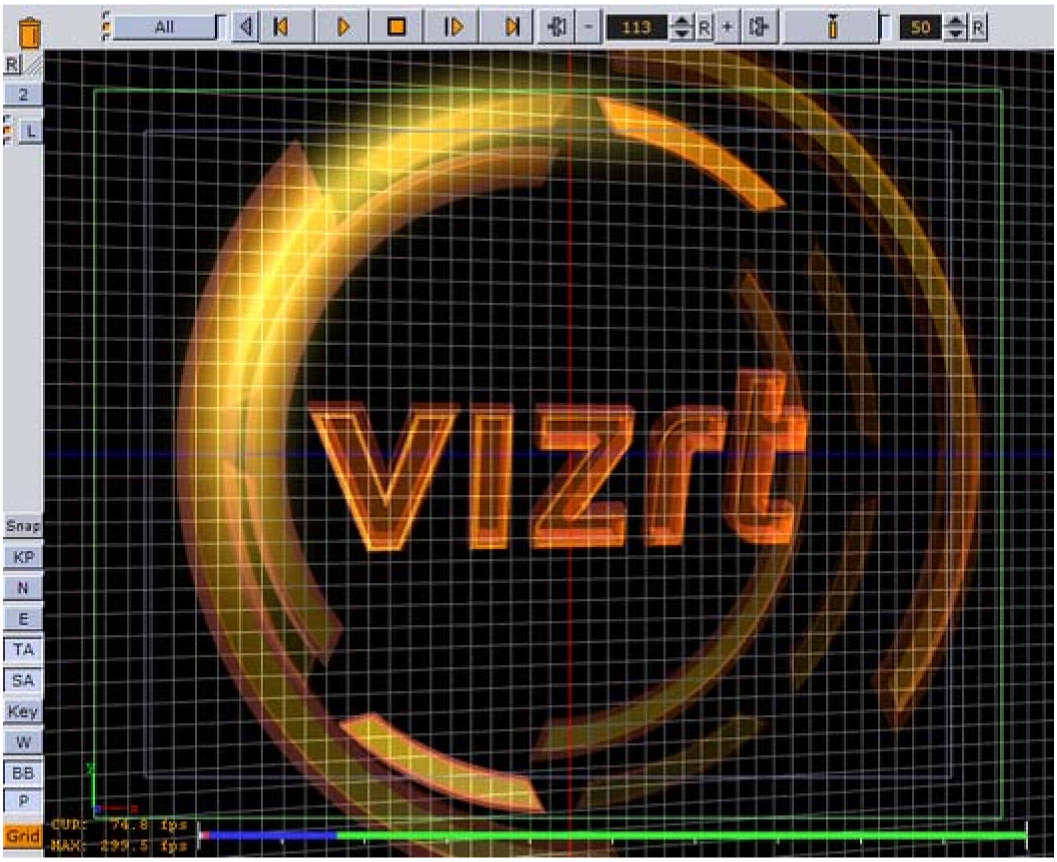
Note the following hints and tips for the built-in menu:

- Clicking the **All** folder in left pane shows all plug-ins, without any categorization. Alternatively, click the various folders to browse according to the categories that have been set.
- Press a letter on the keyboard to jump to a plug-in starting with that letter.
- Double-clicking a folder tab takes you to the **Default** folder for that plug-in category.
- You can drag all plug-ins straight into the scene tree (except scene plug-ins).

To add a plug-in to your favorites

- Right-click the plug-in and select **Add as favorite**.
  Favorites are saved in the Favorites Bar.

2.4 Scene Editor



```
CUR:  74.8 fps
MAX: 299.5 fps
```

The scene editor has a set of useful functions that enables the designer to see how the graphics will look when keyed, seen from a different angle in virtual sets, positioned on screen, performance of the scene and more.

At the left side of the scene editor is a vertical bar. This bar makes it possible to display additional information in the scene editor, as well as providing more options while working.

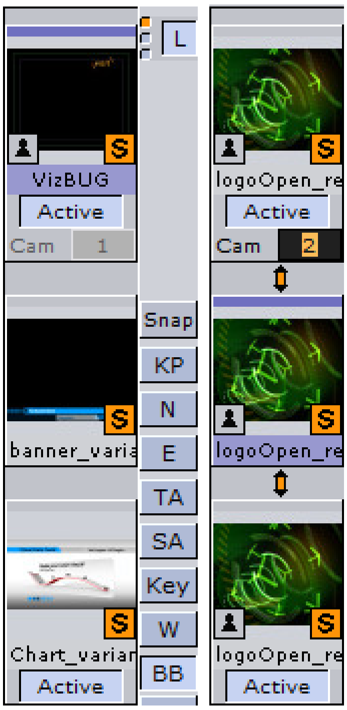
This section contains information on the following topics:

- Layer Manager
- Buttons

See Also

- Working with the Scene Editor
- Scene Editor General Shortcuts
- Scene Tree Sort View Shortcuts
- Scene Editor Camera Controls and Shortcuts
- Scene Editor Orthogonal View Controls and Shortcuts
- Scene Editor Animation Controls and Shortcuts

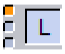
2.4.1 Layer Manager



All scenes are by default designed to run in the main layer, but can be played out in the front and back layer as well. The layer manager can be used to test combinations of scenes in the three layers (front, main and back).

When working with the layer manager you are not able to move and edit objects in the scene using the Scene Editor; however, you can edit the scene by opening it (see Context Menu). Picking in the Scene Editor is done on the loaded layer. The Grid Toolbar also works for the current layer.

You can also set the same scene to the front and back layer, creating what is known as self layer(s). This will allow you to link the scenes and adjust the camera settings for the front layer. When saving a scene, any self layer settings will be preserved.



The three small indicators show which layer is active in the scene tree, and are referred to as traffic lights.

Context Menu

- **Open <scene>**: Opens the scene.
- **Set Inactive/Active**: Enables you to display the layer or not.
- **Reset layer**: Removes the scene from the layer (the same as drag and drop into the bin).
- **Database Search**: Opens the location of the scene.

To add scenes to the layer manager

- Simply drag and drop the scenes onto the layer placeholder(s).

See Also

- Buttons
- Working with the Scene Editor
- Timeline Editor

2.4.2 Buttons

Click the scene editor buttons once to enable the features, click them again to disable.

- **Snap**: Takes a snapshot of the current content in the render output. For more information, see Taking a Snapshot.
- **KP**: Activates the preview of the key signal.
- **N**: Shows the normal vertices. For example, it is practical to use this feature when the lighting is not reflected in the desired manner. Enabling this feature shows the vertices as purple handles. These handles makes it easier to predict how the light will be reflected.
- **E**: Enables the handling of interactive script/plug-in events. Interactive scripts and plug-ins are those related to mouse or keyboard actions.
- **TA**: Displays the title area as the outline of a light blue rectangle in the render output.

Tip: The size of the title area can be defined under the Rendering scene settings.

- **SA**: Displays the safe area as the outline of a green rectangle in the render output.

Tip: The size of the safe area can be defined under the Rendering scene settings.

- **Key**: Shows what the key output will look like.

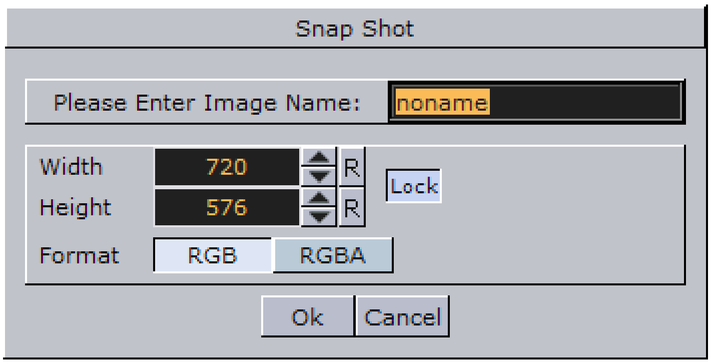
Note: Scene editing is disabled while Key is displayed.

- **W**: Renders the scene in wire frame mode.
- **BB**: Enables the bounding box visualization for the selected container.?
- **P**: Displays the Performance Bar.
- **Grid**: Opens the Grid Toolbar.

This section also contains information on the following topics and procedures:

- Taking a Snapshot
- Performance Bar
- To enable the performance bar in the scene editor
- Grid Toolbar
- To enable the Grid toolbar

Taking a Snapshot



Snap Shot

Please Enter Image Name: noname

Width 720

Height 576

Format RGB RGBA

A snapshot can be taken of the current content in the render output.

When clicking the Snap button, the Server view will automatically be selected from the main menu, and the database content will appear. The image will be placed in the folder/project that was last visited.

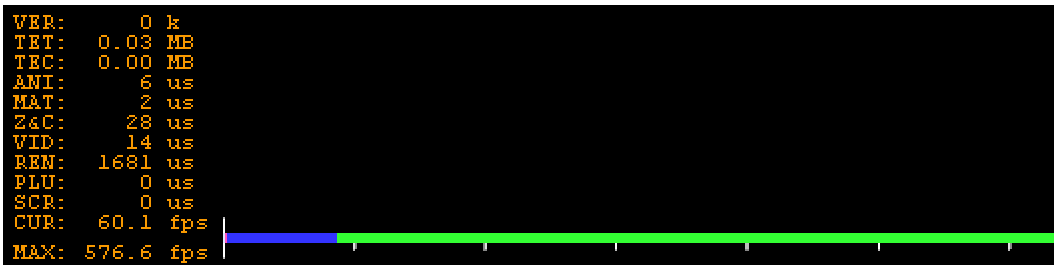
To take a snapshot

1. Click the **Snap** button to create an RGB image, or
2. Press the **SHIFT** key and click the **Snap** button to create an RGBA image.
3. In the Snapshot dialog box that appears, enter the **name**, **size**, and **format** for the image.
4. Click **Ok** to save the image to the database.

Note: An RGBA image will use the key information as alpha.

Note: To save the image in another folder/project, select the destination, and then click the Snap button again.

Performance Bar



```
VER:      0 k
TET:   0.03 MB
TEC:   0.00 MB
ANI:      6 us
MAT:      2 us
Z&C:     28 us
VID:     14 us
REN:   1681 us
PLU:      0 us
SCR:      0 us
CUR:   60.1 fps
MAX:  576.6 fps
```

The Performance Bar can, in addition to being displayed in On Air Mode, be displayed in the lower part of the scene editor. The performance bar gives an idea of the performance (frames per second) the scene is rendering in. Each indicator shows how many microseconds each rendering step takes.

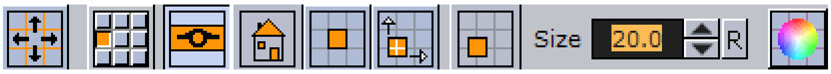
Blue color shows occupied memory, while green shows available. They are also linked to the bars with different colors:

To enable the performance bar in the scene editor

- Click the scene editor's **P** button. If enabling the Performance bar by middle-clicking the P button, all performance indicators appear.

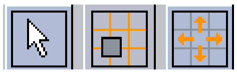
Grid Toolbar

The Grid toolbar can be displayed in the lower part of the Render pane. Grids are used to align containers.



- **Plane Type**: Shows which plane the grid should lie in:
    - **Off**: Indicates that the grid is switched off.
    - **Align X/Y**: Aligns the grid to the X/Y-plane.
    - **Align X/Z**: Aligns the grid to the X/Z-plane.
    - **Align Y/Z**: Aligns the grid to the Y/Z-plane.
    - **Free Grid**: Makes it possible to use a free grid (for example aligned to a container).
- **Select Grid**: Sets a predefined grid. 9 grids can be selected and manually configured as presets for the scene.
- **Show/Hide Grid**: Shows or hides the grid in the render output.
- **Home**: Sets the grid back to its default position.
- **Snap to Grid**: If this button is enabled, and the position of a container is modified by dragging it around in the render output, the position of the object will be snapped to the grid (i.e. the center of the X and Y axis).
- **Align Object on Plane**: To align the axes of a container to the corresponding axes of the grid, first enable this option and next left click the container you want to align and drag it around. As long as this option is switched on, the container will move only along the plane if you drag it around in the render output.
- **Move Object to Plane**: If you select a container and next click this icon, the container will be moved so its center is on the plane.
- **Size**: Defines the raster size of the grid.
- **Change Color of Grid**: Changes the color of the grid.

The following three buttons are only available if the grid type is set to Free Grid:



- **Picking On/Off**: Aligns the grid to the object. Enable picking and click the object to align the grid. The grid will be aligned to the vertex you click on.
- **Move Grid to Object**: Moves the grid so the center of the selected container will be on it.
- **Free Plane Value Editor**: Switches the free plane value editor on and off. In this editor you can adjust the grid's plane.

To enable the Grid toolbar

- Click the **Grid** button seen as one of the display options of the render window.

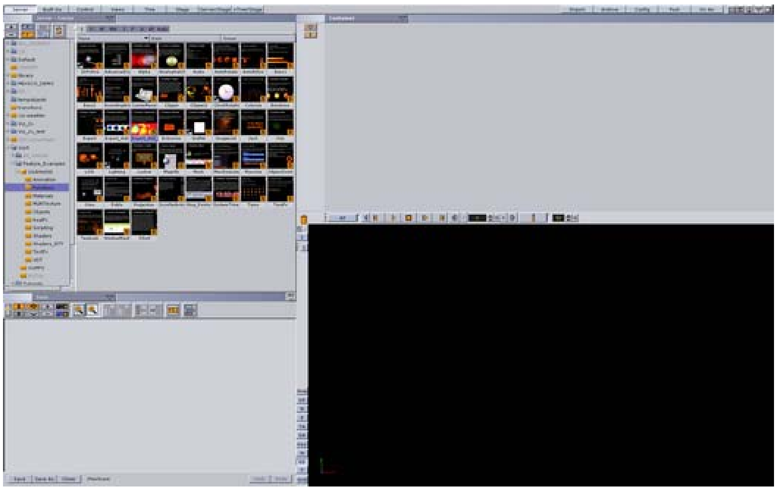
See Also

- Layer Manager
- Working with the Scene Editor
- Timeline Editor

2.5 Working with the Scene Editor

In the scene editor you can edit some transformation parameters of the containers in your scene. Furthermore you can switch some properties of a container on and off.

Figure 1: Scene Editor



This section contains information of the following topics:

- Scene Editor Context Menu
- To select an option from the context menu
- To select an object
- To change an object’s position
- To rotate an object
- To select multiple objects
- To scale an object
- To move the axis center of an object
- To copy an object

See Also

- Working with the Scene Editor
- Scene Editor General Shortcuts
- Scene Tree Sort View Shortcuts
- Scene Editor Camera Controls and Shortcuts
- Scene Editor Orthogonal View Controls and Shortcuts
- Scene Editor Animation Controls and Shortcuts

2.5.1 Scene Editor Context Menu

In the scene editor you can also open a shortcut menu by a right-click. This shortcut menu gives you a set of options:

- **Position**: Switches into the position mode.
- **Rotation**: Switches into the rotation mode.
- **Scale**: Switches into scale mode.
- **Center**: Switches into center editing mode.
- **Space**: Switches between Local and World coordinate systems. For more information, see Transformation Principles.
- **Properties**: (only applicable if a container is selected).
  - **Hide** lets you hide the container in the render output.
  - **Lock** enables you to lock the container. If you do so you cannot change the position of a container by dragging it in the render output.
  - **Transformation** opens the transformation editor for the container.
  - **Normal vectors** toggle the display of the normal vectors at the bottom of the properties you will find a list of all properties applied to the container with the possibility to switch them **on** and **off**.
  - **Add material** allows you to add a material to all objects selected.
  - Any other properties of the container. You can turn them on and off (except materials).
- **Animation**: (only applicable if a container is selected and it has an animation defined) displays the animation path of when Working with Position Keyframes.

Note: When you right-click to select the context menu, you need to hold down the right mouse button when selecting submenu items.

To select an option from the context menu

1. Right-click and hold the right button.
2. Move the cursor to the option you want to select and release the button.

To select an object

1. Select an object in the scene editor by **left-clicking** it. Note that this is done on a per pixel basis.
   - The appearing bounding box indicates which object has been selected if the Bounding Box option is enabled.
   - The entry for the container in the scene tree is highlighted.
   - By default the transformation editor will be opened in the Container editor pane. If other editors are selected (e.g. image or text editor), those editors will also be opened if available for other objects.
2. Select another object by either using the mouse to **left-click** it, or using the **arrow keys**. Using the cursor keys will enable you to navigate through the structure of the scene tree.
3. Unselect an object by pressing **Space** or click in an empty space.

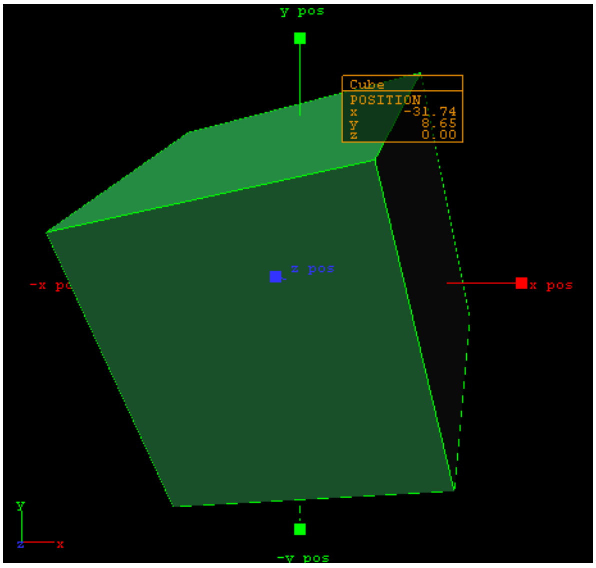
To select multiple objects

- Hold down the CTRL key and click all the objects in the scene editor you wish to select.

To delete an object

- Select the object and press the CTRL + Delete keys.

To change an object’s position

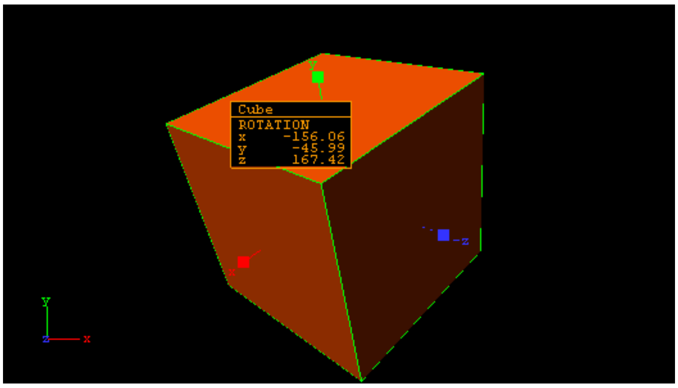


```
Cube
POSITION
x     -31.74
y       8.65
z       0.00
```

1. Method 1:
   - Left-click the object and hold the left button while moving the cursor around.
   - Hold **Shift** while moving the cursor to lock the axes so you can change the position of the container only along one axis at a time.
2. Method 2:
   - Right-click the object and select **position** from the context menu.
   - Grab one of the handles and change the position by dragging them around the scene editor.
3. To move the object in the z space, click and hold down both the left and center mouse buttons, and then drag within the scene editor.

   The container will follow the movement along the axes shown in the selected camera view. During this operation a status window is shown, giving you information about the position.

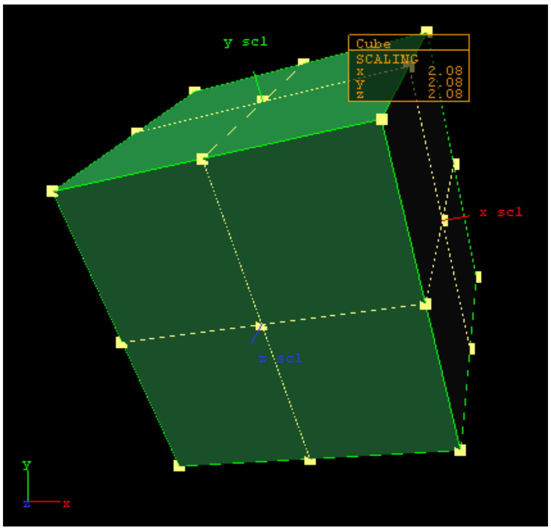
To rotate an object



Cube
ROTATION
x -156.06
y -45.99
z 167.42

1. Rotate an object by holding the middle mouse button down and moving the mouse around to rotate on all the axes, or
2. Right-click to open the context menu and select **rotation** and drag the handles side ways to rotate on all the axes.

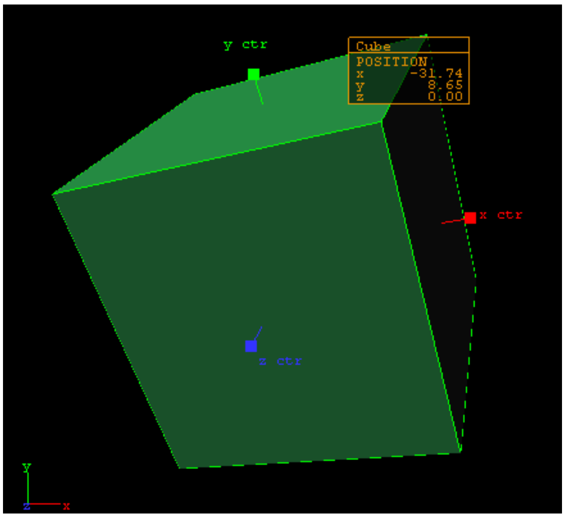
To scale an object



Cube
SCALING
x 2.08
y 2.08
z 2.08

1. Right-click to open the context menu and select **scale**.
2. Drag the handles to scale the object.

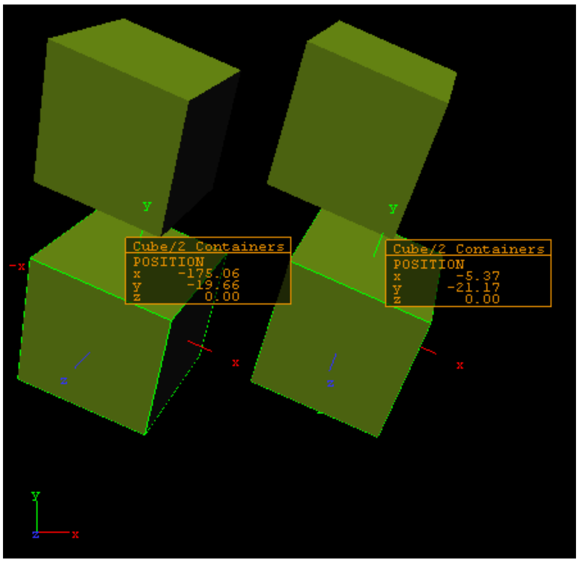
To move the axis center of an object



```
Cube
POSITION
x   -31.74
y     8.65
z     0.00
```

1. Right-click to open the context menu and select **center** in the shortcut menu. This will display the axis of the selected container with handles on their ends.
2. Grab one of the handles and change the center position by dragging them around the scene editor.

To copy an object



```
Cube/2 Containers
POSITION
x  -175.06
y   -19.66
z     0.00
```

```
Cube/2 Containers
POSITION
x    -5.37
y   -21.17
z     0.00
```

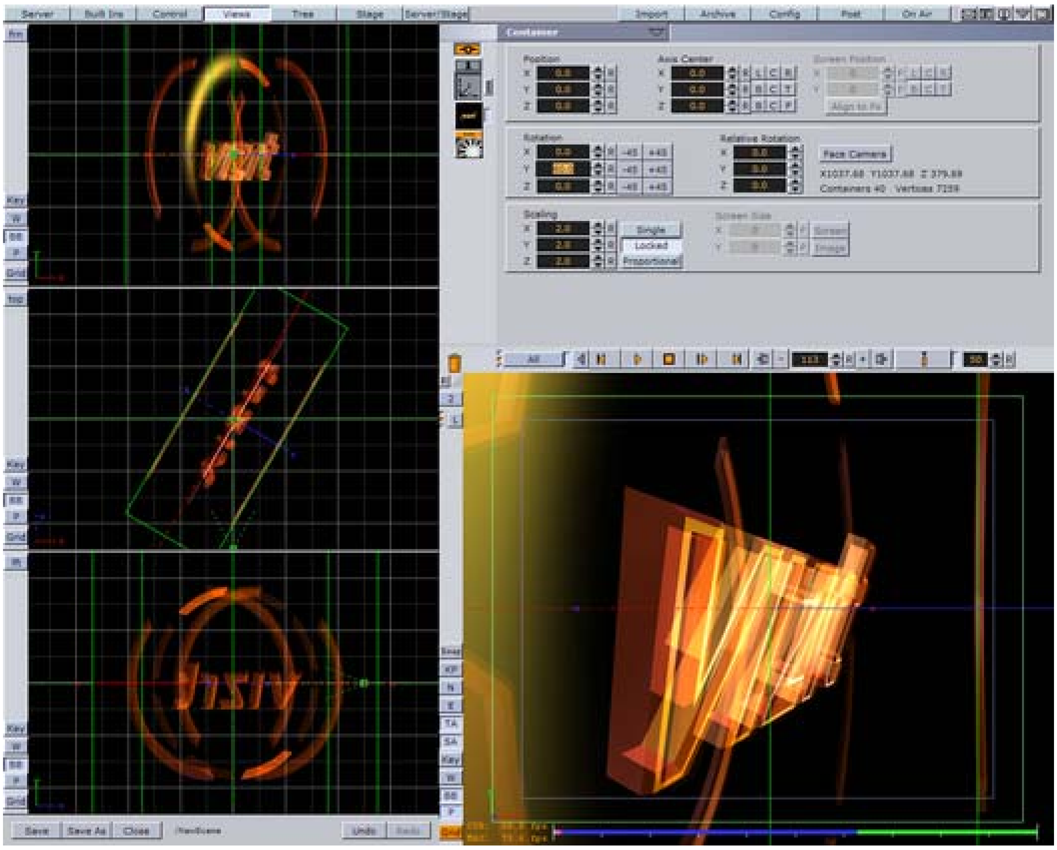
- While holding the **CTRL** key click and hold the **left mouse button**, and do a drag and drop operation to the new location of one or more objects.

Note: You can use multi drag to copy a container in the render output. This is done by holding CTRL while dragging the container around, but instead of releasing the left button, do a right-click while holding CTRL and the left mouse button pressed.

See Also

- Buttons
- Layer Manager
- Timeline Editor

2.6 Scene Editor Views



By clicking the **Views** button in the main menu, three extra scene editor views open on the left side of the main scene editor.

These are a good aid for getting an overview of the objects in the scene and visualize their position in relation to lights and cameras. They have in principle the same functionality as the main scene editor and lets you perform the same operations on objects as you would in the main scene editor. As well as the main scene editor, the side views may be set to view through one of the 16 editable cameras or through one of the six predefined orthogonal views. Setting them to one of the predefined orthogonal views allows you to change position and direction of lights and cameras.

If you set the scene editor to display one of the predefined orthogonal side views there are some special shortcuts that may be used to pan and zoom:

- **Pan**: By clicking the left button in the background (not the objects).
- **Zoom in (Z) and out (X)**: Zoom in or out by holding down the **Z** or **X** keys and draw a rectangle with the left button pressed, or press the **Z** or **X** keys multiple to zoom in or out in a stationary position.

In the six predefined orthogonal views, the currently selected camera, and in addition all cameras that have been given the status “vis” or “inf”, will appear as symbols. By clicking on them in the side views, their focus will be set in the main scene editor and handles for editing position and direction will be visible (drag the handles to change position and direction). The same is the case with lights. The one currently selected and those which have the “vis” status will be visible as symbols and position and direction can be edited with the cursor.

See Also

- Lights and Materials
- Cameras

2.7 Script Editor

To launch the script editor

1. In the main menu, click **Server**.
2. In the right pane, select **Scene Settings**.
3. Click the **Script** tab.

Figure 2: Script Editor Launch

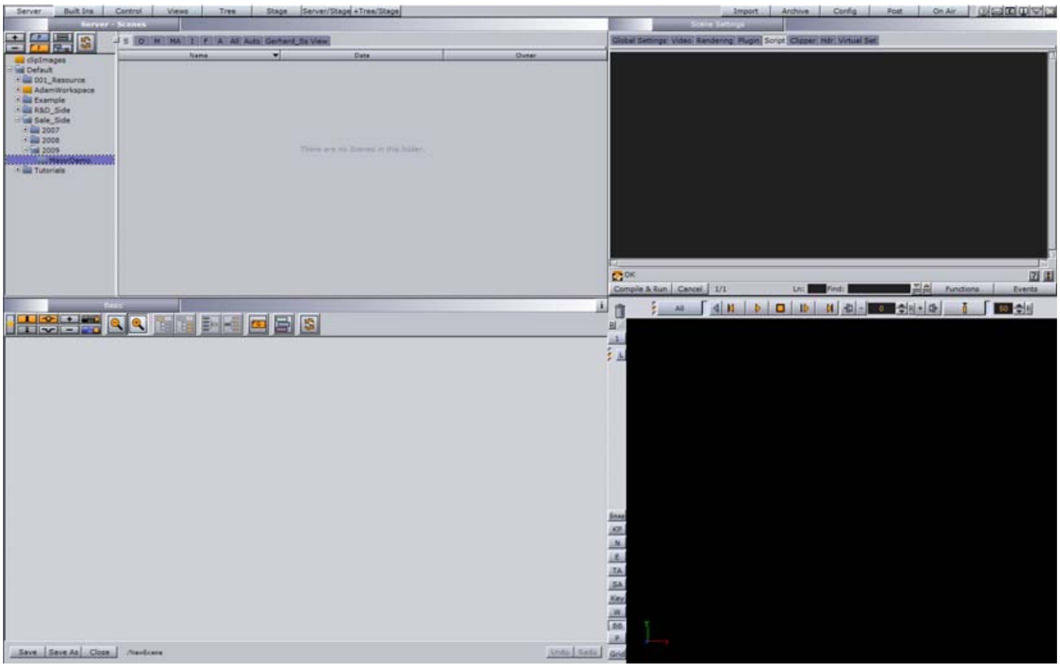
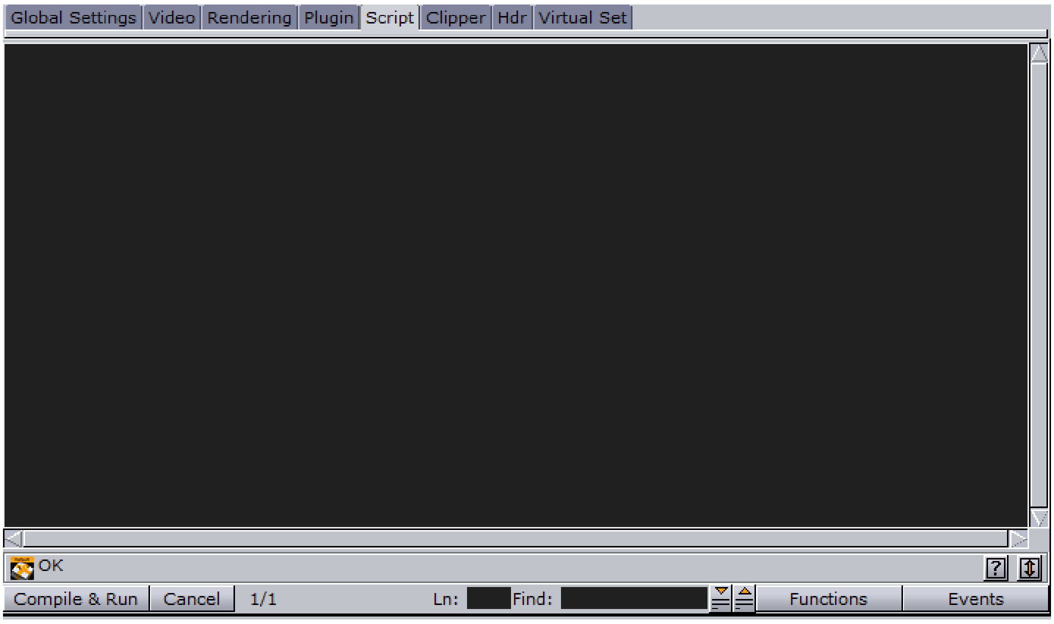


Figure 3: Script Editor



- Enter scripts according to the proper syntax in the black area. See the Script documentation for details.
- **OK**: Shows that all parts of the script entered are correct. If there is an error, it will indicate where.
- **Compile & Run**: Compiles and runs the script.
- The **Cancel**: Cancels all changes.
- **Ln**: Jump to the line number entered in the field.
- **Find**: Free text search. The up-arrow will search above the cursor’s location. The down-arrow will search below the cursor’s location.

- **Functions**: List of all the functions defined. See the Script documentation for details.
- **Events**: List of all built-in events, such as mouse-click, mouse-move, keyboard button press, etc.
- ?: Launches the script documentation.
- Up/Down arrows: Pops-out/docks script editor.

See Also

- Script Editor Shortcuts

2.8 Keyboard and Mouse Shortcuts

This section contains information on the following topics:

- Application Controls and Shortcuts
- Integer/Float Controls
- Server View Shortcuts
- Built-in View Shortcuts
- General Scene Tree Shortcuts
- Scene Tree Basic View Controls and Shortcuts
- Scene Tree Sort View Shortcuts
- Scene Editor General Shortcuts
- Scene Editor Camera Controls and Shortcuts
- Scene Editor Orthogonal View Controls and Shortcuts
- Scene Editor Animation Controls and Shortcuts
- Stage Controls and Shortcuts
- Polygon Plug-in Editor Shortcuts
- Script Editor Shortcuts

2.8.1 Application Controls and Shortcuts

Table 1: Application Controls and Shortcuts

Function	Key	Mouse
Display Context Menu		Right mouse button click
Quit Viz Artist	Double-Esc Alt+q Alt+F4	
Show/Hide Viz Command Console	Alt+c	
Show Video Output Configuration Window	Alt+v	
Show Performance Analyzer Window	Ctrl	Right mouse click on Exit button
Quit Viz Engine	Double-Esc Alt+q Alt+F4	

2.8.2 Integer/Float Controls

Table 2: Integer/Float Controls

Function	Key	Mouse
Increase/Decrease very large steps (100)	Shift+Alt	Hold left mouse button and move
Increase/Decrease large steps (10)	Alt	Hold left mouse button and move
Increase/Decrease normal steps (1)		Hold left mouse button and move
Increase/Decrease small steps (0.1)	Ctrl	Hold left mouse button and move
Increase/Decrease very small steps (0.01)	Shift+Ctrl	Hold left mouse button and move
Reset whole group		Double click one of the Rs (Reset Button)

2.8.3 Server View Shortcuts

Table 3: Server View Shortcuts

Function	Key	Mouse	Comments
Open Graphics Hub Search Window	Ctrl+f		
Switch to Scenes	Ctrl+1		
Switch to Geometries	Ctrl+2		
Switch to Materials	Ctrl+3		
Switch to Advanced Materials	Ctrl+4		
Switch to Images	Ctrl+5		
Switch to Fonts	Ctrl+6		
Switch to Audio Files	Ctrl+7		
Switch to All	Ctrl+8		
Switch to Auto Selection	Ctrl+9		
Show/Hide Properties of a File	Ctrl+p		
Toggle Detail View	Ctrl+y		
Create new Material or new Scene	Ctrl+a		This works only in Material or Scene View.
Copy File(s)	Ctrl+c		
Cut File(s)	Ctrl+x		
Paste File(s)	Ctrl+v		This will create a link. File will not be duplicated.
Paste File(s)	Ctrl+Shift+v		This will duplicate the file(s).
Export	Ctrl+e		This works only in Image View.
Rename File	F2		

Table 3: Server View Shortcuts

Function	Key	Mouse	Comments
Duplicate Selected File(s)	Ctrl	Left mouse button click and drag	
Link File(s) to a different Folder/ Project	Alt	Left mouse button click and drag	
Multidrop		Hold left mouse button and move; middle mouse button click	

2.8.4 Built-in View Shortcuts

Table 4: Built-in View Shortcuts

Function	Key	Mouse
Jump to plug-in starting with a given letter	Press the first letter of the plug-in name.	
Jump to Default folder for a category of plug-ins		Double-clicking the plug-in folder tab.

2.8.5 General Scene Tree Shortcuts

Table 5: General Scene Tree Shortcuts

Function	Key
Switch to Basic View	Ctrl+1
Switch to Sort View	Ctrl+2
Switch to Search View	Ctrl+f
Show Scene Log Window	Ctrl+4
Show Scene Information Window	Ctrl+5
Refresh	Ctrl+a
Jump to Top Container	Ctrl+Home
Jump to Last Container	Ctrl+End

2.8.6 Scene Tree Basic View Controls and Shortcuts

Table 6: Scene Tree Basic View Controls and Shortcuts

Function	Key	Mouse	Comments
Insert New Container	Ctrl+Insert		The new Container will be created above the first selected Container.
Delete Selected Container(s)	Ctrl+Delete		
Open Tree	Ctrl+o		
Collapse Tree	Ctrl+i		
Lock Selected Container(s)	Ctrl+l		
Unlock Selected Containers(s)	Ctrl+Shift+l		

Table 6: Scene Tree Basic View Controls and Shortcuts

Function	Key	Mouse	Comments
Group Selected Container(s)	Ctrl+g		
Ungroup Selected Container(s)	Ctrl+Shift+g		
Hide Selected Container(s)	Ctrl+h		
Show Selected Container(s)	Ctrl+Shift+h		
Undo	Ctrl+z		
Redo	Ctrl+y		
Save the Scene	Ctrl+s		
Copy Selected Container(s)	Ctrl	Left mouse click on container name and drag	
Copy Selected Container(s) to Clipboard	Ctrl+c		
Paste Containers from Clipboard	Ctrl+v		They will be created one hierarchical level underneath selected container.
Cut Selected Container(s)	Ctrl+x		
Move Selected Container(s)		Left mouse click on container name and drag	
Move Selected Container with Magic Move	Alt	Left mouse click on container name and drag	
Copy Properties of Container		Left mouse click on container Properties and drag	
Merge Group	Ctrl+m		
Split Container	Ctrl+Shift+m		

2.8.7 Scene Tree Sort View Shortcuts

Table 7: Scene Tree Sort View Shortcuts

Function	Key
Sort by Vertices	Ctrl+q
Sort by Render Time	Ctrl+w
Sort by Texture Size	Ctrl+e
Toggle Unique for Texture Size	Ctrl+t

2.8.8 Scene Editor General Shortcuts

Table 8: Application Shortcuts

Function	Key	Mouse
Unselect container	Space	
Delete container	CTRL + Delete	
Copy container(s)	CTRL	Left mouse click
Rotate container		Middle mouse click and hold
Toggle bounding box on/off	b	
Select next container in tree	Cursor down	
Select previous container in tree	Cursor up	
Select next container down in tree hierarchy	Cursor right	
Select parent container	Cursor left	
Switch additional performance information on		Middle mouse click on “P”
Create Snapshot in RGBA	Shift	Right mouse click on “Snap”
Add new keyframe in stage of selected container	Return	
Enable expansive Z-Sort mode	s	

2.8.9 Scene Editor Camera Controls and Shortcuts

Table 9: Scene Editor Camera Controls and Shortcuts

Function	Key	Mouse
Switch Camera from 1 to 10	1 - 0	

Table 9: Scene Editor Camera Controls and Shortcuts

Function	Key	Mouse
Switch to Front Camera	Numeric Keypad Home	
Switch to Back Camera	Numeric Keypad Ins	
Switch to Top Camera	Numeric Keypad 8	
Switch to Bottom Camera	Numeric Keypad 2	
Switch to Left Camera	Numeric Keypad 4	
Switch to Right Camera	Numeric Keypad 6	
Change Position of Camera in x	p	Left mouse button click and drag (left-right)
Change Position of Camera in y	p	Middle mouse button click and drag (up-down)
Change Position of Camera in z	p	Right mouse button click and drag (up-down)
Zoom Camera	u	Left mouse button click and drag (left-right)
Pan Camera	i	Left mouse button click and drag (left-right)
Tilt Camera	i	Middle mouse button click and drag (up-down)
Twist Camera	i	Right mouse button click and drag (left-right)
Orbit in x (Rotate Camera around Selected Object in x)	o	Middle mouse button click and drag (up-down)
Orbit in y (Rotate Camera around Selected Object in y)	o	Left mouse button click and drag (left-right)
Orbit in z (Rotate Camera around Selected Object in z)	o	Middle mouse button click and drag (left-right)
Orient Camera against Selected Container	t	
Reset Camera Transformation	r	

2.8.10 Scene Editor Orthogonal View Controls and Shortcuts

Table 10: Scene Editor Orthogonal View Controls and Shortcuts

Function	Key	Mouse
Pan		Left mouse button click and drag on background
Rubberband Zoom In	z	Left mouse button click and drag
Rubberband Zoom Out	x	Left mouse button click and drag

2.8.11 Scene Editor Animation Controls and Shortcuts

Table 11: Scene Editor Animation Controls and Shortcuts

Function	Key	Mouse
Reset handle of selected keyframe in animation path	d	
Manipulate single handle	Ctrl	Left mouse button click handle and drag
Mirror Handles	Shift	Left mouse button click handle and drag
Insert unlocked path point	Ins	Left mouse click on path

2.8.12 Stage Controls and Shortcuts

Table 12: Stage Controls and Shortcuts

Function	Key	Mouse
Start/Continue Animation Forward	Enter	
Start Animation Backwards	Shift+Enter	
Set Timeline Marker/Value to Start	Ctrl+Home	
Set Timeline Marker/Value to End	Ctrl+End	
Set View to Start	Home	
Set View to End	End	
Move Timeline Marker/Value in large steps (5)	Left/Right Cursor	

Table 12: Stage Controls and Shortcuts

Function	Key	Mouse
Move Timeline Marker/Value in small steps (1)	Ctrl+Left /Right Cursor	
Animation View (without Spline View)	Ctrl+2, Shift+PageUp	
Spline View (with Animation View)	Ctrl+3, Shift+PageDown	
Animation/Spline View	Ctrl+4	
Delete Selected Keyframe(s)	Delete	
Pan View		Middle mouse click and drag
Zoom Time Axis In	z, y	Mouse wheel
Zoom Time Axis Out	x	Mouse wheel
Zoom Time Axis to Selected Channel	Shift+z, Shift+y	
Zoom Time Axis to whole Animation	Shift+x	
Zoom Spline View In	a	
Zoom Spline View Out	s	
Zoom Spline View to whole Spline	Shift+a	
Zoom Time Axis and Spline View to view all keyframes of selected Channel	Shift+q	
Zoom Time and Value Axis to View All Keyframes	Shift+w	
Zoom Time Axis In/Out	z	Middle mouse click and drag
Zoom Value Axis in Spline Editor In/Out	a	Middle mouse click and drag
Copy Selected Keyframe	c	Left mouse click and drag
Bring View to Timeline	m	
Reverse the Order of Selected Keyframes	Ctrl+r	
Snap to Grid on/off	Ctrl+g	
Set Time Axis to Seconds	Ctrl+8	
Set Time Axis to Frames	Ctrl+9	

Table 12: Stage Controls and Shortcuts

Function	Key	Mouse
Set Time Axis to Fields	Ctrl+0	
Move Timeline Marker to nearest Keyframe	j	
Scale a Portion of Selected Director	t	Left mouse click on Director Bar *(This indicates a region where keyframes can be scaled.)*
Rubberband Selection	s	Left mouse click and drag
Zoom in onto specific region	z	Left mouse click and drag
Insert a new Stop Point	s	Left mouse click on Director Bar
View Spline 1	Ctrl+z	
View Spline 2 (if available)	Ctrl+x	
View Spline 3 (if available)	Ctrl+c	
Jump to next Keyframe	Tab	
Jump to previous Keyframe	Shift+Tab	
Escape from moving a keyframe	Esc	

2.8.13 Polygon Plug-in Editor Shortcuts

Table 13: Polygon Plug-in Editor Shortcuts

Function	Key	Mouse
Reset Viewport Offset	Home	
Pan		Middle mouse click and drag
Zoom In/Out		Mouse wheel
Rubberband Zoom In	Shift+	Left mouse click and drag
Zoom In	z	
Zoom Out	x	
Set Zoom 1	Ctrl+Home, Ctrl+z	
Set Zoom 1 to 9	Shift+Ctrl+1 to 9	
Sharpen Edge of Selected Point(s)	Esc	
Delete Selected Point(s)	Delete	
Show/Hide Bounding Box	Space	
Invert Screen Colors	b	
Insert New Point on Spline	Insert+	Left mouse click on Spline
Show Handles only on Selected Point On/Off	a	
Create New Spline	Insert	Left mouse click
Close/Open Selected Spline	c	
Set/Hide Anchor at Cursor Position	Ctrl+a	*or*: Double click middle mouse button
Enable/Disable Spline Info	i	
Switch between Curve and Straight Lines	l	
Split Spline at Cursor	s, Shift+Mouseover	
Switch to Cutting Mode	Shift+s	
Merge Left		Double click left mouse button
Move Active Spline	Alt+	Left mouse click and drag

Table 13: Polygon Plug-in Editor Shortcuts

Function	Key	Mouse
Make Spline Active and Move	Shift+	Click and drag middle mouse button *(Cursor must be inside Bounding Box)*
Copy	Ctrl+c	
Cut	Ctrl+x	
Paste	Ctrl+v	
Scale Spline		Click and drag left mouse button on Bounding Box handle
Rotate Spline		Click and drag left mouse button next to vertex of Bounding Box handle

2.8.14 Script Editor Shortcuts

Table 14: Script Editor Shortcuts

Function	Key
Copy	Ctrl-c
Cut	Ctrl-x
Paste	Ctrl-v, Shift-Insert
Search for Search String	F3
Search for Selected Text	F4
Comment	F5
Uncomment	F6
Increase Indent	Tab
Decrease Indent	Shift-Tab
Undo	Ctrl-z

3 Tree Knowledge and Skills

Scenes are root elements that hold other elements. Existing scenes are stored in the database and available through the Server area, see To view scenes in a project/folder. For general information about scenes and procedures on how to create, import, rename, and delete scenes, see Scenes.

This section contains information on the following topics:

- Opening a Scene
- Working with the Scene Tree
- Modifying Container Items

3.1 Opening a Scene

To open a scene in the database

1. Select **Server** from the main menu.
2. From the Server tree, select the folder/project where the scene is saved.
3. **Double-click** the scene to open it.

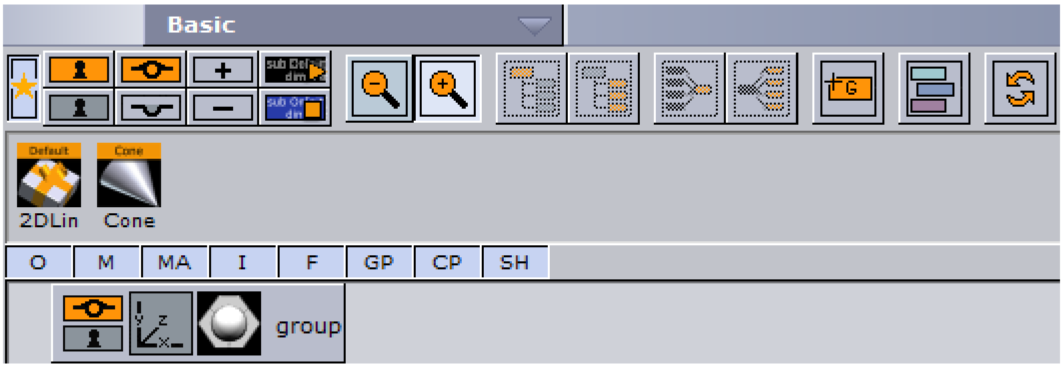
The content of the scene will be displayed as containers in the scene tree, and visualized in the renderer window.

Note: Active Elements or scenes are marked with a Keyhole icon.

Tip: If the folder/project's path is unknown, try Searching for it.

Tip: The Status bar in the lower left part of the window, between the Close and Undo buttons, shows the name and path of the current scene. Clicking on the path will select the scene in the server view.

3.2 Working with the Scene Tree



The scene tree is the logical visualization of all the elements in a scene. The tree consists of containers that hold properties. The elements can be geometry, images, materials, texts, transparency functions, key functions, and many more.

Scenes are built by dragging construction elements, like images or built-in geometry plug-ins, onto the scene tree (it is also possible to drag them directly into the Output area). The containers in the scene tree are then organized in a hierarchy and grouped in logical divisions. Containers and groups will be explained in more detail in the following paragraphs.

This section contains information on the following topics and procedures:

- To view the Scene Tree
- Scene Tree Menu
- Display Area
- Favorites Bar
- Containers
- Groups

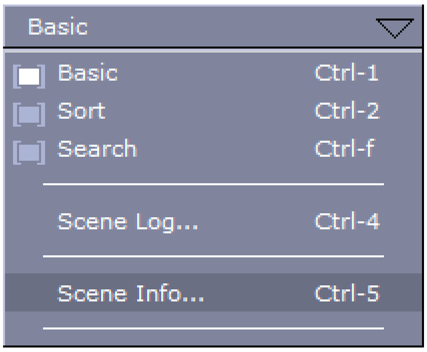
To view the Scene Tree

- Select Server or Tree from the Main menu.

See Also

- General Scene Tree Shortcuts

3.2.1 Scene Tree Menu



At the top of the Scene Tree area is the Scene Tree menu. The Scene Tree menu changes the way the scene tree is displayed.

- **Basic**: Shows an overview of all the containers in the scene. For more information, see Basic.
- **Sort**: Sorts the elements in the scene tree based upon criteria that are set in the Display area. For more information, see Sort.
- **Search**: Allows searching within the scene tree. For more information, see Search.

- **Scene Log**: Opens the Log window, which provides information about the loading process of the scene, such as errors, for example if a plug-in is not found.
  - If an error occurs (e.g. missing plug-ins) the exclamation mark icon will be shown, and the log window can be opened by clicking it.
  - If the log window is closed, but not cleared, the exclamation mark icon will still be shown in gray.
- **Scene Info**: Opens the Scene Information window, which provides information about the scene entered by the designer.

- **Scene Information (button)**: Opens the Scene Information window, which provides information about the scene entered by the designer.

3.2.2 Display Area

Below the menu is the Display area, which changes appearance according to the selection in the Scene Tree menu.

This section contains information on the following topics:

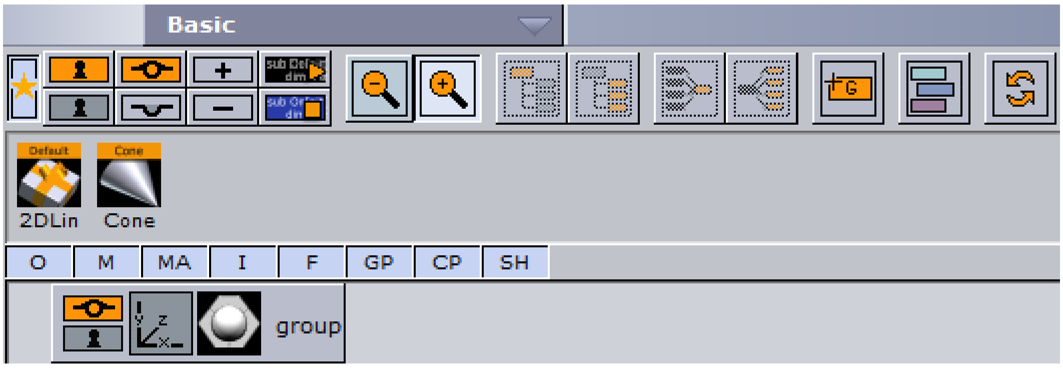
- Basic
- Sort
- Search

See Also

- Scene Tree Basic View Controls and Shortcuts

- Scene Tree Sort View Shortcuts

Basic



If Basic is selected from the Scene Tree menu, all containers are shown in the tree hierarchy.

- **Show/Hide Favorites**: Shows/hides the Favorites Bar.

- **Lock/Unlock**: Enables or disables the ability to modify the transformation of the container in the Scene Editor.

- **Show/Hide**: Makes the container visible/invisible in the scene editor, and eventually in the final rendered output.

- **Expand/Collapse**: Expands/collapses the scene tree.

- **Start/Stop**: Starts (compiles)/stops all container scripts.

- **Zoom In/Zoom Out**: Zooms the scene tree in/out. Zooming out applies smaller icons to the containers, providing a better overview, especially in complex scenes.

- **Group/Ungroup**: Creates a new group (see Groups), and places the selected containers as sub containers. The Ungroup button moves the sub containers up one level.

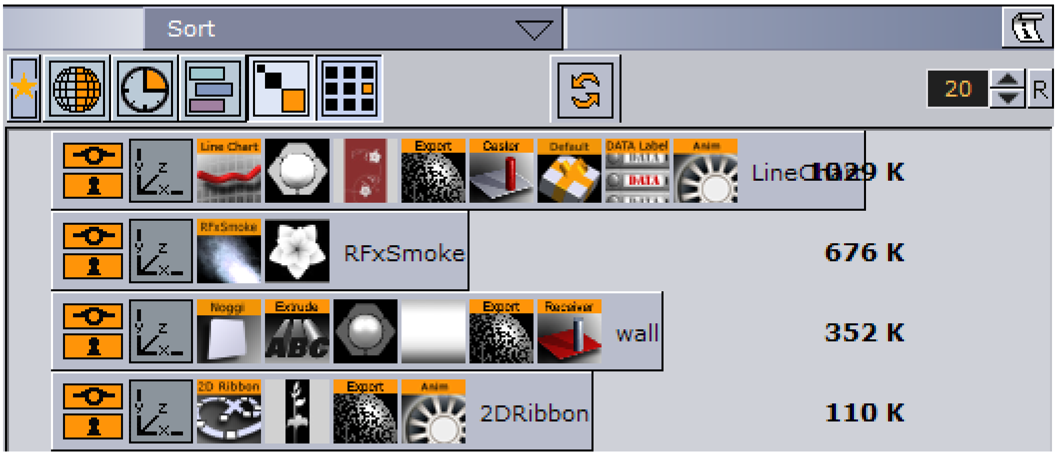
- **Merge/Split**: Merges the selected containers to one compound object, or splits an existing compound object. For more information, see Merging Containers.

- **Create Group**: Creates a new empty group at the top of the scene tree. To place the new group at a specific place in the scene tree, drag the icon. For more information, see Adding Root or Sub containers.

- **Colorize**: Makes it possible to colorize the selected containers. Text can be added to the colors to make them more descriptive. Text for scene specific colors can be modified in the Scene Settings editor (see Global Settings), and text for global colors can be modified in the Configuration’s User Interface section.

- **Refresh**: Refreshes the scene tree. Use this button if scripts have been deployed to change the scene tree.

Sort



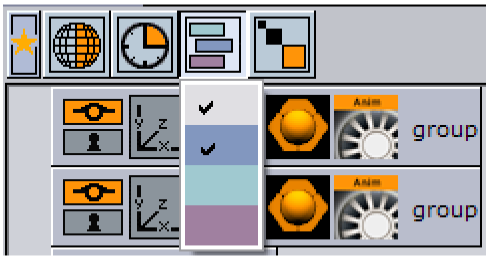
If Sort is selected from the Scene Tree menu, the elements in the scene tree can be sorted based upon various criteria. Sorting the scene tree makes it possible to analyze the scene and optimize the performance.

Note: Sorting the scene tree does not take the logical tree structure into notice. Select Basic from the Scene Tree menu to show the “normal” tree structure again.

- **Show/Hide Favorites**: Shows/hides the Favorites Bar.

- **Vector**: Sorts the scene tree by the number of vectors the containers have. The container with most vectors is displayed on top.

- **Render Time**: Sorts the scene tree by the time it takes to render the containers. Each container is displayed with the percentage of total time the scene takes in order to be rendered. The container with the highest percentage is displayed on top.



- **Sort by Colors**: Facilitates filtering only a few colorized containers that should be subject to change.

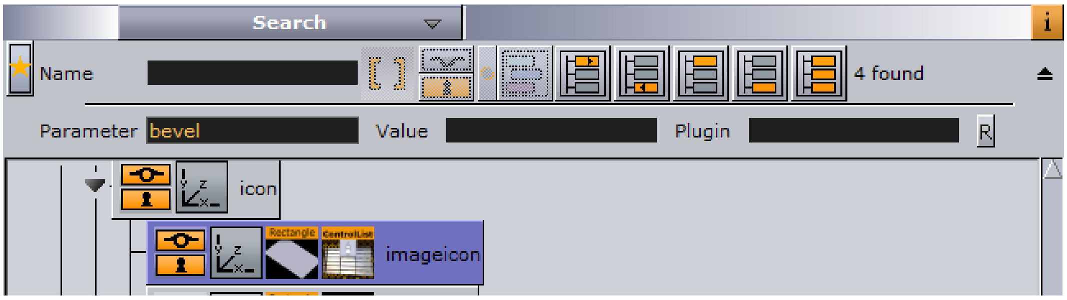
- **Texture Size**: Sorts the scene tree by the size of the container texture. The container with the largest texture size is displayed on top.

- **Unique**: Shows unique group objects with the same texture only once. This option is only available if the scene tree containers are sorted by vector size (click the Texture Size button).

- **Refresh**: Refreshes the scene tree.

The input box to the right defines the number of search results. Changing this value reduces or increases the number, showing only the containers with the highest values.

Search



Parameter bevel

If Search is selected from the Scene Tree menu, a search can be performed within the scene tree. You can also expand the search by clicking the expand button (to the right). This will give you the option to search for values, and parameter and plugin names within your scene.

Search results are highlighted in the scene tree, and the number of results are displayed in the notification area at the right side of the Display area.

The different buttons can be used to search for specific properties of the scene or highlight search results. For example, dragging a plugin e.g. *Extrude* to the *Dragsite for Properties* will allow you to click the *Show all* button and highlight all occurances of that plugin in your scene tree. Further you can choose to change properties for all containers or click show next/previous to change individual container properties. Also, if you have added your favorite plugins to the Favorites Bar you can easily add plugins to all or individual containers.

- **Show/Hide Favorites**: Shows/hides the Favorites Bar.

Name: To?u*

- **Name**: Searches for containers with the specified name.

Tip: Wildcards such as * or may be used.

Note: The search is case sensitive.

- **Dragsite for Properties**: Drag an element from the Server area or a property from a container in the scene tree to search for its occurrence within the scene. An icon from the Favorites bar can also be dragged.

- **Hidden**: Searches for all hidden containers.

- **Locked**: Searches for all locked containers.

- **Inactive Properties**: Searches for containers that contain properties that are switched to inactive.

- **Color**: Searches for all containers with the assigned color tile. For more information, see Container Menu.

- **Show Next**: Shows the next result.

- **Show Previous**: Shows the previous result.

- **Show First**: Shows the first result.

- **Show Last**: Shows the last result.

- **Show All** shows all results.
  - Showing all results is especially practical during scene cleanup. For example, if a specific plug-in should no longer be used, search for its occurrence, highlight all results, and then drag them onto the trash can.
  - If you click Show All the Next and Previous buttons will enable you to step through the entire scene tree. The currently selected container will be highlighted above the search result.

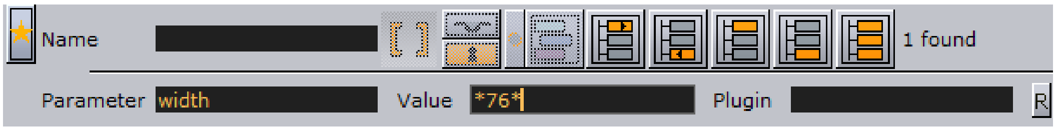
Note: While jumping through the search results, the container that is currently highlighted in the scene tree will also be selected in the Output area, and opened in the belonging editor.

- **Parameter**: Allows you to search for a parameter name
- **Value**: Allows you to search for the a parameter value
- **Plugin**: Allows you to search for a plugin name

To perform an extended search

1. Start Viz Artist

2. **Load** a scene



Name

1 found

Parameter width

Value *76*

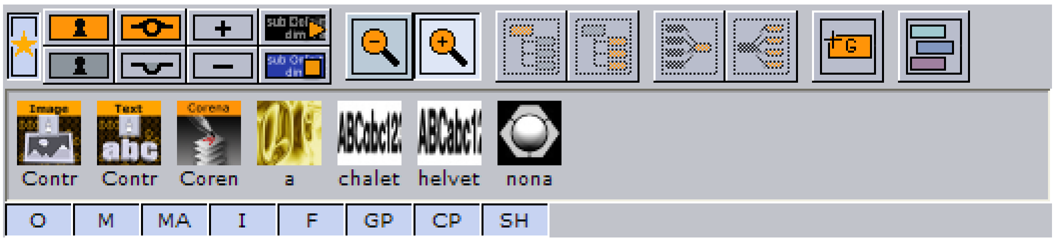
Plugin

3. Go to the Scene Tree and select **Search** (CTRL+f) from the drop-list **expand the search options**
4. **Enter** your search criteria and press Enter. Your search critera can be a combination of parameter name, value and plugin name. Below are some examples:
   - Parameter: tesselation, width, *show*
   - Value: 20, *10.0*
   - Plugin: *rect*

5. Use the **“Show ...” buttons to navigate** the Scene Tree

Tip: Use the label names in the plugin editors as your search criteria. If you do not get a hit, use the asterisk (*) symbol.

3.2.3 Favorites Bar



Selecting the star icon in the Display area shows the Favorites bar at the top of the scene tree.

For ease of use, elements can be added as favorites. This provides fast access to elements that are often used. All element types can be added as favorites, except scenes and audio files. Geometry and function plug-ins can also be added to the favorites. All favorites are also saved as part of the user settings to the database.

The buttons below the Favorites bar’s display area enables and disables the display of favorites from the following categories:

- **O**: Shows favorites that are objects/geometries.
- **M**: Shows favorites that are materials.
- **MA**: Shows favorites that are materials advanced (merged object).
- **I**: Shows favorites that are images.
- **F**: Shows favorites that are fonts.
- **GP**: Shows favorites that are geometry plug-ins.
- **CP**: Shows favorites that are container plug-ins.
- **SH**: Shows favorites that are shader plug-ins.

This section also contains information on the following procedures:

- To add an element as a favorite
- To remove an element as a favorite

To add an element as a favorite

1. Right-click it in the Server area, and then select Add as Favorite, or
2. Drag it from the Server area onto the Favorites bar.

To remove an element as a favorite

1. Right-click the element in the Server area and select Delete from Favorites, or
2. Drag it from the Favorites bar onto the trash can.

Tip: Elements can also be added as favorites by dragging them from the result window after a database search onto the Favorites bar.

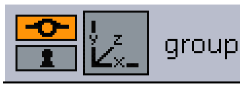
3.2.4 Containers

A container combines a set of elements and properties to one block used in the scene.

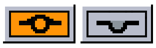
This section contains information on the following topics:

- Container Icons
- Container Menu
- Adding Containers
- Adding Root or Sub containers
- Copying Containers
- Moving Containers
- Renaming Containers
- Editing Container Properties
- Deleting Containers
- Merging Containers

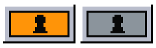
Container Icons



Every container is shown with a name and three basic icons:



- **Show/Hide**: Indicates if the container and all its sub containers are visible/ invisible in the Output area. Click the icon to change the state.



- **Locked/Unlocked**: Indicates if the container is locked so that properties cannot be manipulated in the Output area, or unlocked so that properties can be manipulated. Click the icon to change the state.

Example: An unlocked element can be resized and repositioned in the Output area.

- **Transformation Editor**: Opens the Transformation Editor in the Container view.

A container that has sub containers is marked with a black arrow to the left of the container.

▶ ▼

- **Collapse/Expand**: If the arrow points to the right, the container is collapsed and the sub containers are not displayed in the scene tree. If the arrow points downwards, the container is expanded and the sub containers are displayed in the scene tree. Click the icon to change the state.

The following icons are only available if the respective elements are applied to the container.

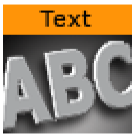
Note: Materials and objects can be dragged from a container to the Database area. This will create new independent instances of these elements in the database. Dragging an image from a container to the Database area will only create a linked element.

- **Object**: Shows the geometry held by the container (e.g. a merged or imported object). The thumbnail icon reflects the object. Click the icon to open the geometry editor.

- **Material**: Indicates that a material has been applied to the container. The color of the icon reflects the color of the material and its properties. If there is no material icon, a standard white base color is used for the container. Click the icon to open the material editor (see Working with Materials).

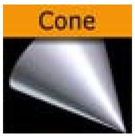
Note: A sub container can inherit color from a parent container.

- **Image**: Indicates that an image has been applied to the container. The thumbnail on the icon reflects the image. There are two ways to apply an image to a container. If the container holds geometry, the image is used for texturing the geometry. If there is no geometry, the image is displayed in the render output. Click the icon to open the Image Editor.

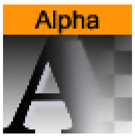


- **Font**: Indicates that a font has been applied to the container. Click the icon to open the Text Editor.

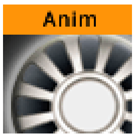
The following icons are only available if the respective plug-ins are applied to the container:



- **Geometry**: Shows the geometry held by the container. The icon reflects the built-in Geometry Plugins, in this example a cone object.

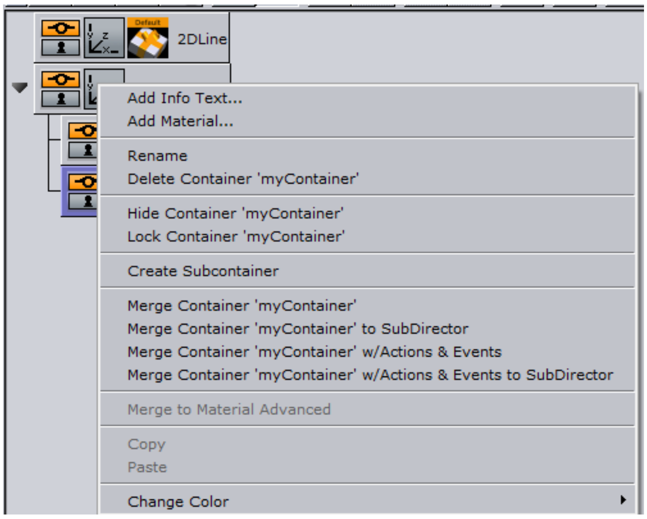


- **Containers**: Indicates that the container holds a container function. The icon reflects the Container Plugins, in this example an alpha object.



- **Animation**: Indicates that the container contains animated elements. For more information, see Creating Animations.

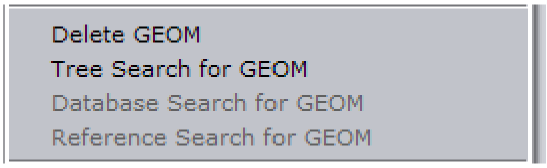
Container Menu


2DLine
Add Info Text...
Add Material...
Rename
Delete Container 'myContainer'
Hide Container 'myContainer'
Lock Container 'myContainer'
Create Subcontainer
Merge Container 'myContainer'
Merge Container 'myContainer' to SubDirector
Merge Container 'myContainer' w/Actions & Events
Merge Container 'myContainer' w/Actions & Events to SubDirector
Merge to Material Advanced
Copy
Paste
Change Color

Right-clicking a container in the scene tree opens the Container menu. When right-clicking the container name, the Container menu holds the following options:

- **Add Info Text**: Makes it possible to add text-based information on a container. Also available as a plug-in by clicking **Built-ins** and then double-clicking the **Container Plug-ins** tab.
- **Add Material**: Adds a Material to the selected container. Working with Materials.
- **Rename**: Makes it possible to change the name of the selected container.
- **Delete Container [name]**: Deletes the selected container.
- **Show/Hide Container [name]**: Indicates if the selected container and all its sub containers are visible/invisible in the Output area.
- **Lock/Unlock Container [name]**: Disables or enables the ability to modify the transformation of the selected container in the Output area.
- **Create Sub Container**: Creates a new empty child container for the selected container.
- **Merge Container [name]**: Merges the selected container and all its sub containers into one compound object, or extracts an already merged object. For more information, see Merging Containers.
- **Merge Container [name] to Sub Container**: Merges the selected container(s) and all the respective sub containers into one compound object and creates a new sub director for the merged animation.
- **Merge Container [name] w/ Actions & Events to Sub Director**: Merges the selected container(s) and all the respective sub containers into one compound object, including actions and events and creates a new sub director for the merged animation..
- **Merge Container [name] w/ Actions & Events to Sub Container**: Merges the selected container(s) and all the respective sub containers into one compound object, including actions and events.
- **Merge to Material Advanced**: Merges multiple objects to one, creating advanced material(s) (see Working with Advanced Materials).
- **Change Color**: Makes it possible to apply a color to the selected container. Colorized containers make the scene tree more organized. For example, all font containers can be colored red, while all containers that hold images can be colored blue. Four active colors are configured by default. Color can also be applied from the Favorites Bar. how to configure the Colors in the User Interface configuration section.

When right-clicking a container property, the Container menu holds the following additional options:



- **Delete [property]**: Deletes the property from the container.
- **Tree Search for [property]**: Selects Search from the Favorites menu, and searches the scene tree for instances of this property. For more information, see.
- **Database Search for [property]**: Searches for the property in the database, and displays the result in the Server area.

Note: This menu option is only available for images.

- **Reference Search for [property]**: Searches for the property in the database, and displays the result in the Search window. For more information, see References.

Note: This menu option is only available for images.

Adding Containers

Containers can be added to the scene by dragging an element onto the scene tree, render output or property editor.

By dragging an element to the render output, the container will be added at the top of the scene tree. Also note that elements can be added to a scene by dragging them from the result window after a database search.

By selecting multiple elements from the Server area and dragging them onto the Scene Tree, all of them will be added.

Adding Root or Sub containers

Elements that are dragged onto the scene tree can be positioned in various ways, and create either root or sub containers.

- **Sub container** Dropping an element at the right side of a container in the scene tree will create a sub container.

- **Root Container Below/Above** Dropping the element at the left side of a container in the scene tree will create a root container at the same hierarchy level, either below or above the selected container.

Adding Empty Containers

Clicking the Create Group button (CTRL+Insert) will add an empty container above the selected container. If a container is not selected it will add the new container to the root level. The Create Group button is only available if Basic is selected from the Scene Tree menu.

To add an empty container

- Click the Create Group button from the Display area.

To add a container at a specific location in the scene tree

- Drag the Create Group icon and place it as a above or below of an existing container, or as a sub container.

Copying Containers

To copy a container within a scene

- Hold CTRL while dragging the container to its new position, or use CTRL+C and CTRL+V to copy and paste the container.
    - Copied containers will be placed as sub container of the selected container.

Note: When copying a container with sub containers, all sub containers are copied as part of the same operation.

To copy multiple containers at once

- Hold CTRL and the left mouse button while dragging the containers to their new position, and then click the right mouse button to release.

Moving Containers

To move a container within a scene

- Drag it to the new position, or hold the ALT key down while dragging the container in order to recalculate transformations such that global coordinates are preserved, or use CTRL+X and CTRL+V to cut and paste the container(s) to the new position.

Renaming Containers

To rename a container

1. Double-click the container name, or
2. Right-click a container, and then in the Container menu that appears, click Rename.

To rename multiple containers

- Use the Corena plug-in.

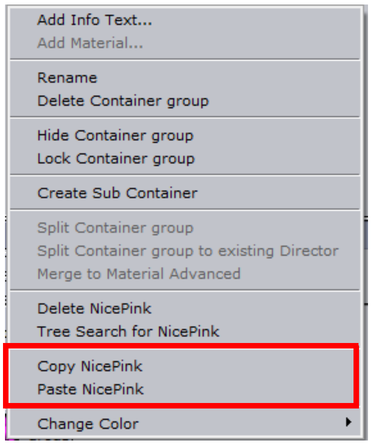
Editing Container Properties

To edit the properties of a container

- Select the container either in the scene tree, or in the render output. The belonging editor will then be displayed in the Property area.

To copy a property from one container to another

- Drag the property from the source container to the target container, or use the context menu.



Deleting Containers

To delete a container from a scene

1. Drag it onto the trash can, or
2. Right-click the container, and in the context menu select Delete Container [Name], or
3. Use the shortcut keys CTRL+Delete.

Note: If deleting a parent container, all sub containers will be deleted too.

Merging Containers

Containers can be merged into compound objects and saved as reusable components to the Viz Graphics Hub database. Any animations will also be merged and added to the new compound object's container.

A saved compound object can be used in other scenes and will per default have an active reference between the stored object and all scenes where it has been used. Through use of stored objects with active references, changes can be applied to many scenes at once, which will typically be a meaningful way to work when designing a set of scenes that share many design elements.

Note: The geometry of a compound object cannot be edited. A compound object must be split in order for the geometry to be edited again.

Note: Splitting a referenced compound object disables the reference.

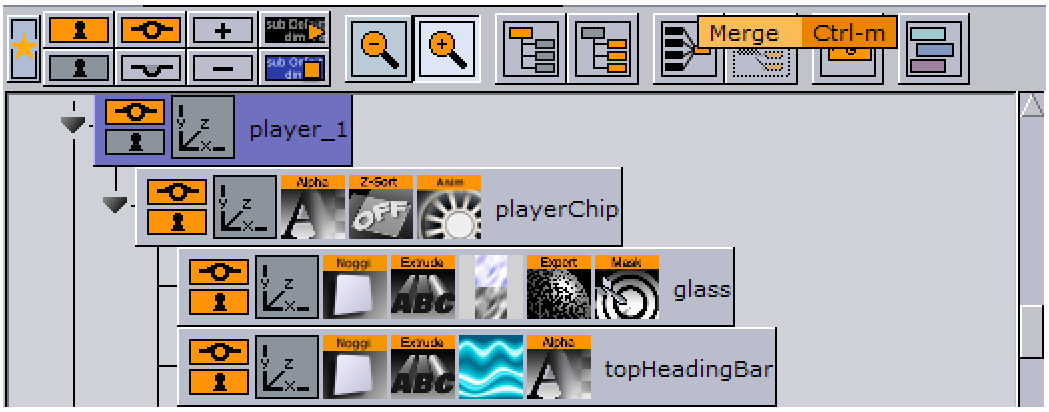
Sometimes, when opening a scene with merged objects, a message displaying "*Old merge style, convert to new merge style*" appears. This is a well-known Autofollow plug-in in combination with merged objects issue.

In merged objects the ordering of containers is reversed which leads to broken autofollow functionality (for first and next reference modes); hence, this merge-style conversion corrects the ordering. Designers that worked around this problem in their scenes do not need to convert; hence, the conversion is not automated.

This behavior is not backward compatible because older versions assume reversed sort order. Ungrouping and grouping with older Viz versions on new scenes will not solve this issue.

To overcome this issue it is recommended to update all Viz Artist and Viz Engine versions to the official 3.2.2 release and later by opening the scene and converting it to the new merge style.

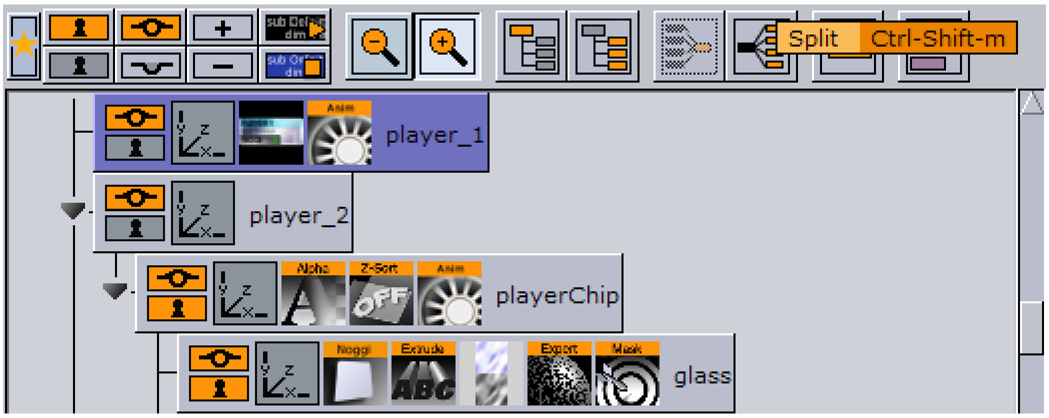
To merge containers into a compound object



1. Select a container and all its sub containers.
2. Click the **Merge** button (CTRL+M), or right-click a container, and from the appearing context menu, select **Merge Container [Name]**.

Note: The Merge button is only available if Basic is selected from the Scene Tree menu.

To split a merged compound object



1. Select a merged object.
2. Click the Split button (CTRL+SHIFT+M) from the Display area, or right-click the container that holds the compound object, and from the appearing context menu, select **Split Container [Name]**.

Note: The Split button is only available if Basic is selected from the Scene Tree menu.

To save a compound object to Viz Graphics Hub

1. Select and right-click a container (with or without sub containers), and from the appearing context menu select **Merge Container [name]**.
2. **Drag** the compound object's icon from the property editor or scene tree and **drop** it onto a folder/project in the **Server Area**.
3. **Type** a descriptive name for the object in the appearing dialog box, and click **Ok**. The object will now be visible in under the *Server Objects tab*.

Tip: Any animations or scripts used within the object will be kept when saving.

Note: Make sure to drag only the object icon, not the whole container.

See Also

- Groups

3.2.5 Groups

The containers in the scene tree can be organized in a hierarchy and grouped in logical divisions. The hierarchy depth is practically unlimited. Grouping containers can be compared to creating a parent/child hierarchy. Such a hierarchy provides a better overview, especially in complex scenes, and is very practical since changing the properties of the group (for example adding a new material) also applies the changes to all the sub containers.

Note: Changing the properties of one of the sub containers (for example adding a material) will override the settings for this particular sub container.

Moving one group is also much easier than moving a lot of single containers.

▼ ▶

A parent container, that has sub containers, is marked with a black arrow to the left of the container. If the arrow points to the right, the container is collapsed and the sub containers are not displayed in the scene tree. If the arrow points downwards, the container is expanded and the next level of sub containers are displayed.

Any container can basically be used as parent, although group containers (that hold transformation only) are best suited.

This section also contains information on the following procedures:

- To group existing containers
- To ungroup containers
- To detach a single container from a group

To group existing containers

- **Select** the **containers** that should be grouped, and click the **Group** (CTRL+G) button from the Display area. This will add an empty group container one level higher than the original containers.

Note: The Group button is only available if Basic is selected from the Scene Tree menu.

To ungroup containers

- **Select** the **parent container**, and click the **Ungroup** (CTRL+SHIFT+G) button from the Display area.

Note: The Ungroup button is only available if Basic is selected from the Scene Tree menu. Also, tree structure further down in the hierarchy will remain.

To detach a single container from a group

- Drag it from the group and place it somewhere in the scene tree.

See Also

- Adding Empty Containers

3.3 Modifying Container Items

If a container is selected from the Scene Setup menu, the selected container property, for example the transformation properties, can be modified in the property area.

Elements that are stored in the database can be added to scenes. Elements that are of type object or font will create a new container using the element as geometry. Materials and audio clips can only be applied to existing containers. Images can be added as geometry of a container, or applied on existing containers as texture. Built-in geometry plug-ins can also be added to the scene.

This section contains information on the following topics:

- Manipulating Properties
- Transformation Editor
- Editing Properties in the Render Output
- Copying Properties
- Geometry Editor
- Text Editor

- Texture Editor
- Adding Plug-ins

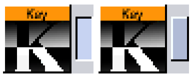
3.3.1 Manipulating Properties

There are a few operations that apply to the container items:

- **Replacing Items**: In order to replace an item applied to a container, simply drag the new item onto the container.
- **Removing Items**: In order to remove an item from a container, drag it from the container or from the Property editor onto the trash can. Alternatively, right-click the container item in the scene tree, and from the shortcut menu that opens, select Delete <Name of the Property>.

Note: Every container must contain transformation, so the transformation property can not be removed.

- **Saving Items in the Database**: If a scene specific item has been created, and it is likely that the item should be reused in another scene, it is possible to save the item in the database. To do so, simply drag the item from the container onto the desired folder/project in the Server area.



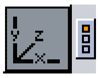
- **Enabling/Disabling Items**: Items are automatically enabled when added to a container. Viz Artist has an easy way of enabling or disabling the items that are applied to a container. This is a practical tool when multiple items are applied to a container. To enable or disable an item, click the switch next to the icon representing the item in the Property editor.

See Also

- Containers

3.3.2 Transformation Editor

Every container in a scene contains transformation properties. Transformation holds information about the most basic properties of a container, such as position and rotation. The transformation properties can be edited in the Transformation editor.



The transformation editor is also able To stick transformation properties to the Container view. This feature allows Position, Rotation, Scaling and Axis Center parameters to stick to the Container view while opening other editors. This enables designers to adjust transformation properties while working on other objects (e.g. text, geometry).

This section contains information on the following topic and procedures:

- Transformation Principles

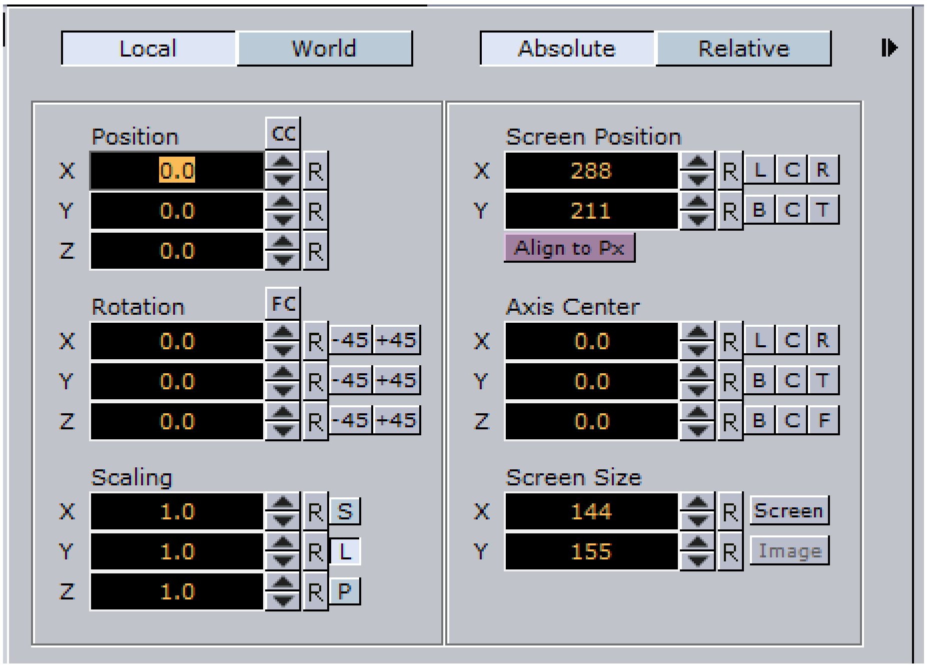
- Properties and Parameters
- To open the transformation editor
- To stick transformation properties to the Container view

Transformation Principles

Containers can be manipulated in several contexts:

- **Local** refers to changes that can be made to the container in its local coordinate system.
- **World** refers to changes that can be made to containers in their world coordinate system.
- **Absolute** refers to changes that can be made to containers in absolute values.
- **Relative** refers to changes that can be made to containers relative to current values.

Properties and Parameters

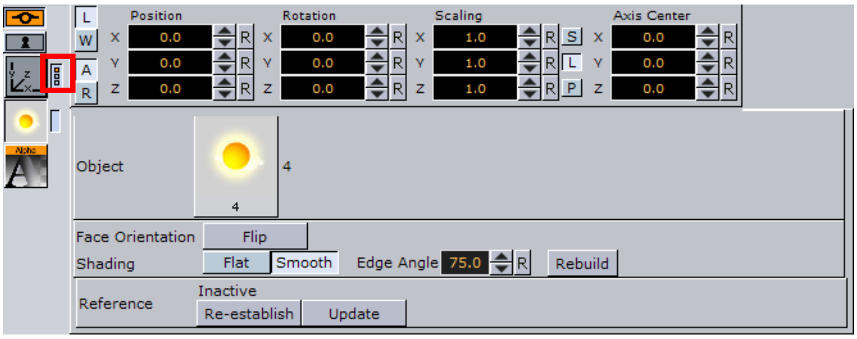


- **Position**: Sets the position of the object along the X, Y, and Z axis.
- **CC** (Center Camera): Centers the container to the camera.
- **Rotation**: Rotates the object according to the X, Y, and Z axes. With the bounding box option enabled, the rotation axes will be visible as three lines; X is red, Y is green, and Z is blue. To rotate in steps of 45 degrees, use the buttons labeled 45 and +45.
- **FC** (Face Camera): Changes the rotation in such a way that the object faces the camera.
- **Screen Position**: Sets the screen position along the X and Y axis for images and some objects by specifying pixels instead of the regular positioning that does not use pixels as input values. The Screen Position option is only available if the image/object faces the camera. To achieve this, click the Face Camera button. When changing the Screen Position values, the values for the regular

positioning will be changed too. The screen position can be set to left, center, or right by clicking L, C, or R, or bottom, center, or top by clicking B, C, or T. The screen positioning is performed based on the current Z value for the container and camera view. If you for instance enter the screen position values X=0 and Y=0, that will place the object in the upper left corner. Thereafter the Z position will be changed, and the object will not align the viewport corner anymore. Then the pixel values must be set to zero again to reposition the image/object.

- **Axis Center**: Sets the axis origin of the object. The rotation axis of the object will be visible in the render output if the Bounding Box option is enabled.
  - The origin can be set to left, center, or right on the object's X axis by clicking L, C, or R.
  - The origin can be set to bottom, center, or top on the object's Y axis by clicking B, C, or T.
  - The origin can be set to back, center, or front on the object's Z axis by clicking B, C, or F.
  - When a group is selected but certain containers in it are not visible, clicking C *with the mouse's middle button* will define the axis center only for those containers in the group which are visible.
  - **Align to Pixel**: When using the normal positioning editor to position an image/object that is facing the camera head on, it can occur that the image/object is off by fractions of a pixel. This can happen because of rounding errors in the screen size and face camera functions. This can produce visual artifacts in some cases on some graphics cards. When the image/object is not aligned to the screen pixels, Align to Px button turns red. Clicking the button adjusts the image/object to be moved to the nearest matching whole pixel.
- **Scaling**: Before changing the scale values of the container, make sure to set the preferred kind of scaling:
  - **S** (Single): Scales each axis by itself.
  - **L** (Locked): Scales all axes at once, so that X, Y, and Z have the same value.
  - **P** (Proportional): Scales all axes at once, so that the values remain proportional.
- **Screen Size**: Makes it possible to scale an object/image based on pixels. The Screen Size option is only available if the object/image faces the camera. To achieve this, click the Face Camera button.
  - **Screen**: Scales the object/image to the size of the output format.
  - **Image**: Scales the image to the original pixel value.
- **Additional information on the selected container**
  - Clicking the Expand button displays this information to the right of the transformation editor. This information includes parent containers plus all *visible* child containers.
  - **Width, Height, and Depth**: Shows the real size of the object in units (not pixels).
  - **Containers**: Shows the number of containers that are selected. If the selected container holds sub containers, all sub-containers are counted in addition to the main container. A container that holds geometry is counted as two.
  - **Vertices**: Shows the number of vertices that are rendered within the selected container, including the vertices in all sub containers.
  - **Primitives**: Shows the number of primitives in the container including the primitives in all sub containers.

- **Docking**: Clicking the docking button keeps a compact version of the transformation editor always open while you work with additional editors.



	Position	Rotation	Scaling	Axis Center
X	0.0	0.0	1.0	0.0
Y	0.0	0.0	1.0	0.0
Z	0.0	0.0	1.0	0.0

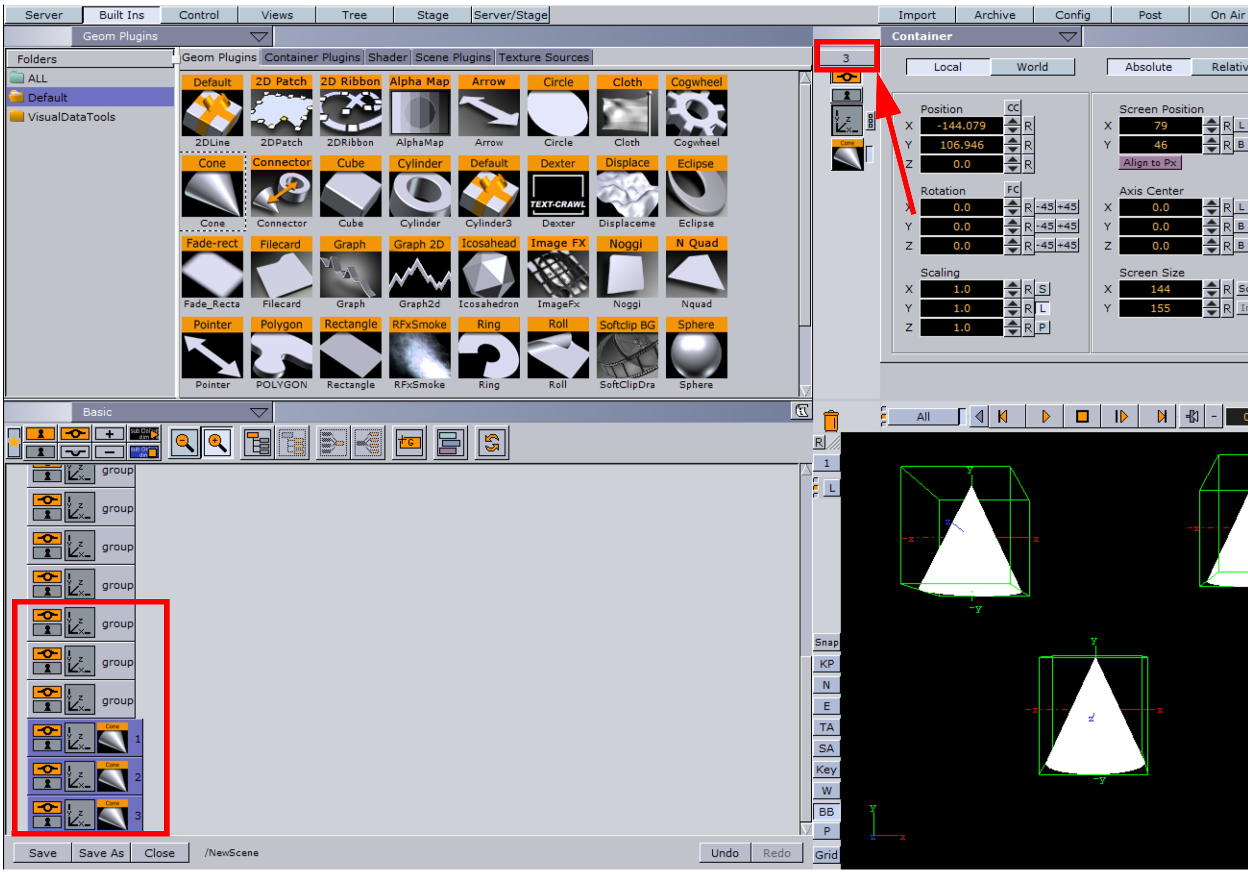
Object 4

Face Orientation: Flip

Shading: Flat, Smooth; Edge Angle 75.0; Rebuild

Reference: Inactive; Re-establish, Update

- **Multiple select**: You can select multiple containers in the tree and then all changes made in the transformation editor will be on everything selected.
    - When multiple containers are selected, the mouse pointer will have an ‘M’ on it when inside the transformation editor.
    - The number of containers selected is shown in the box to the left of the transformation editor.
    - Mousing over that box will indicate the last container selected.
    - All values shown are of the last selected container.
    - Clicking the box enables you to select another container’s values to be displayed.



Container: Local, World; Absolute, Relativ

	Position	Rotation	Scaling
X	-144.079	0.0	1.0
Y	106.946	0.0	1.0
Z	0.0	0.0	1.0

	Screen Position	Axis Center	Screen Size
X	79	0.0	144
Y	46	0.0	155
Z		0.0	

To open the transformation editor

1. Click the Transformation icon on the container, or

2. Select the container from the scene tree, and then click the icon located on the left side of the Property area.

To stick transformation properties to the Container view

- Click the button next to the transformation editor button in the container view.

3.3.3 Editing Properties in the Render Output

Some transformation parameters may be changed directly in the render output through the manipulation of handles in the Scene Editor.

3.3.4 Copying Properties

If a container has certain settings, for example position or rotation, and another container in the scene should have identical settings, it is possible to drag transformation properties from one container to the other.

The distribution of transformation properties can be done with position, rotation, scaling, and axis center.

To copy properties from one container to another

1. **Open** the Transformation Editor of the source container.
2. **Select** the container in the scene tree that should be given new properties.
3. **Drag** the property that should be distributed onto the **target container**, by either transferring the whole property (drag the property name label) or just a sub property (drag the X, Y, or Z label).

Note: If copying properties are done correctly, an icon with the property name will be displayed while dragging it.

3.3.5 Geometry Editor

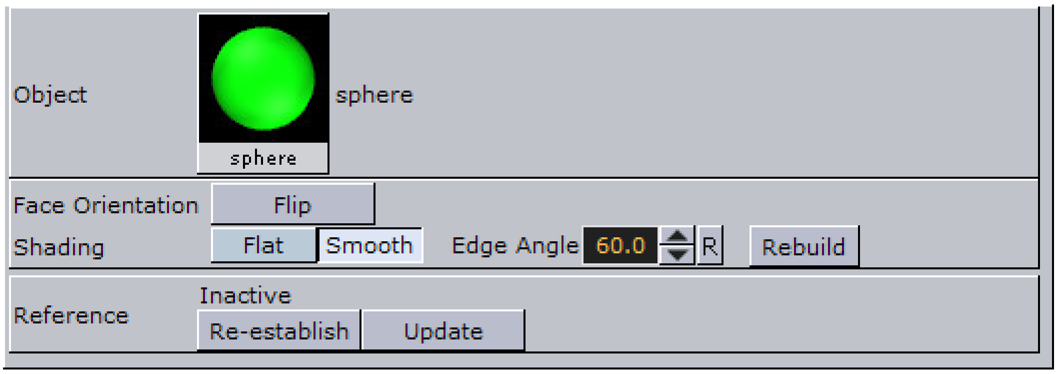
An object holds information about its geometry. When adding an object to a scene, a new container will be created using the object as geometry. Objects that have been created with external 3D programs and imported into Viz Artist can be modified in the Geometry editor.

Note: Face Orientation and Shading are only applicable if the reference is deactivated.

This section contains information on the following topic and procedure:

- Properties and Parameters
- To open the Geometry editor

Properties and Parameters



- **Object**: Shows a thumbnail of the object, and also the object name. By clicking the thumbnail, the object will be selected in the database. Be aware that this can only be done when the reference to the database is active.
- **Face Orientation**: After importing objects that have been created by external 3D programs, it might happen that the polygons face orientation is wrong. By clicking the Flip button, the polygons are set correctly.
- **Shading**: Imported polygon objects normally use shading for creating a smooth surface on the object, which is set together by a planar polygon mesh. The shading value in the Geometry editor sets a threshold between which edges are to be smoothed by shading and which are to be kept as sharp edges. All edges with an angle smaller than the defined value are smoothed by shading. Shading in Viz Artist is performed in the following way: A shade value is produced at each vertex of the polygonal mesh representing the surface of the object. This is done by creating a surface normal on each polygonal facet. The surface normal at each vertex is created by averaging the surface normals for the surrounding facets. The shade at each vertex can then be calculated. Once the shade at the vertices of the polygonal mesh are known, the shade at points interior to the polygonal facets are interpolated from the values at the vertices. This technique makes curved surfaces look “smooth shaded” even though based on a representation of planar facets. The interpolation of shade values is incorporated into the polygon scan conversion routine. Hence an increase in realism is obtained at far less performance cost than carrying out a pixel-by-pixel shading calculation over the whole original surface. Shading may be set to Flat, which means that all shading is disabled, or Smooth, which enables the Edge Angle option. After adjusting the Edge Angle, make sure to click the Rebuild button in order for the changes to take effect.
- **Reference**: If using an imported or compound object in a scene, there is a reference between the object in the scene and the object in the database. If changes are applied to the object in the scene, the object in the database including any scenes where this object is used with an established reference, will reflect those changes, as it is the same object.
  - **Deactivate**: In order to create a unique copy of the object for the scene, the reference link must be deactivated. Changes that are applied to a deactivated object do not lead to changes in the original object in the database.
  - **Re-establish**: Reactivates the reference and overwrites the object in the scene with the object in the database. An active reference is practical when creating a set of scenes with the same object. If for instance all the scenes should have the same background and the background element is created

as an object with active references, changes can easily be applied to all the scenes that use that object.

- **Update**: Reactivates the reference and overwrites the object in the database with the object in the scene.
- **Status**: Shows the current status of the reference.

Note: The reference is automatically deactivated if splitting the object. If the object is split, it must first be merged before the Re-establish and Update operations can be performed.

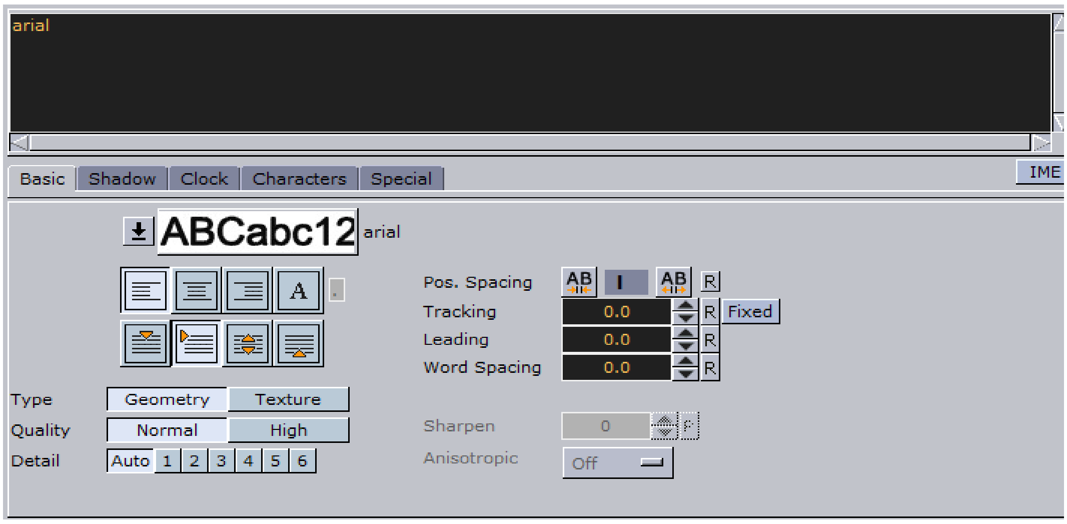
To open the Geometry editor

- Click the icon representing the object either on the container or in the Properties editor.

See Also

- Adding Containers
- Objects

3.3.6 Text Editor


arial

ABCabc12 arial

Pos. Spacing

Tracking 0.0

Leading 0.0

Word Spacing 0.0

Sharpen 0

Anisotropic Off

In the Text editor, text can be added to the scene and the visualization of the scene specific text can be modified by altering properties and adding special effects. More advanced modifications can be performed in the Fontstyle Editor. The modifications in the Fontstyle editor applies to all instances of the font, not just the one used in this scene.

At the top of the Text editor is an input box. The characters in this input box is displayed in the render output. By default, the text equals the font name.

The Text editor contains five panels. Click the tabs at the top of the Text editor to open the respective panels.

This section contains information on the following procedures and topics:

- To open the Text editor
- Basic
- Shadow
- Clock
- Characters
- Special
- Special Fonts

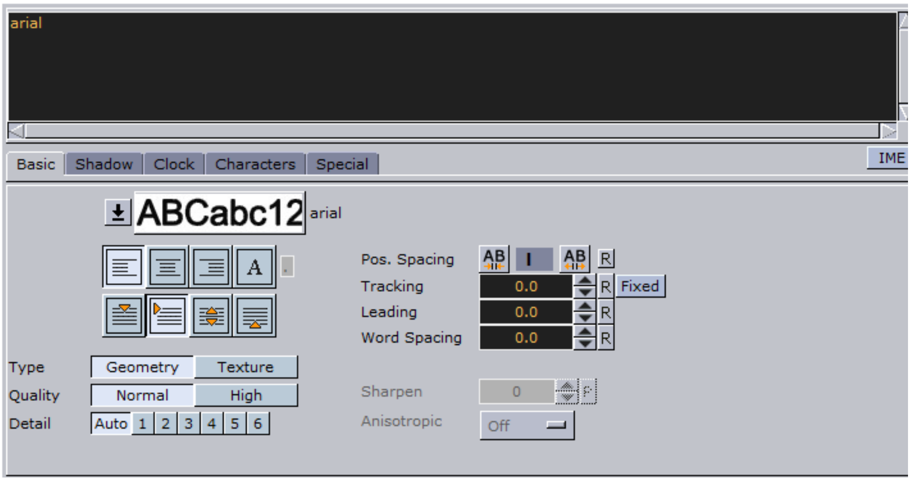
To open the Text editor

- Click the icon representing the text either on the container or in the Properties editor.

Note: After dragging a font onto the scene, the Text editor will automatically open, and a new container will be created using the font as geometry.

Basic

Click the Basic tab in the Text editor to open the Basic panel.



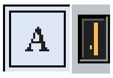
Pos. Spacing
Tracking 0.0
Leading 0.0
Word Spacing 0.0
Type: Geometry | Texture
Quality: Normal | High
Detail: Auto 1 2 3 4 5 6
Sharpen 0
Anisotropic Off

At the top of the Basic panel is the default preview of the text that is being visualized in the scene. It also shows the name of the font.

Clicking the arrow opens a list that displays all available fonts created from the same base font. To use another font, select it from the list. To use a totally different font in the scene, drag the new font from the Server area onto the visualization of the text.

- **Horizontal Justification**

  - Sets the text justification to left, center, or right.



  - Makes it possible to define a character, typically a decimal separator like comma, and justify the text aligned after this character. Enter the justification character in the small input box to the right of the button.

- **Vertical Justification**: Sets the vertical justification on the top of the whole text block, below the first line, center of the middle line in the text block and at the bottom of the text block.
- **Type**
  - **Geometry**: Geometry type fonts are rendered as polygons.
  - **Texture**: Texture creates the font as a texture instead of a polygon. This can to some degree create a poorer look, but it reduces the number of polygons in the scene. The texture resources will of course get an increased demand, so to render the fonts as texture is a trade off between polygon and texture performance. For half height mode rendering, texture fonts are normally the better choice.
- **Quality**: Sets the quality of the font to either Normal or High. The quality of a texture font can be changed in the Fontstyle Editor.

Note: The Quality option is only applicable if the font type is set to Geometry.

- **Sharpen**: Makes it possible to sharpen the text.

Note: The Sharpen option is only applicable if the font type is set to Texture.

- **Detail** Sets the level of detail, or tessellation of the font, to either automatic level or one out of six fixed settings. The fonts are rendered as polygons, so by switching to wireframe mode, the difference becomes obvious as the detail level changes. One is the highest level of detail, six is the lowest. If setting a fixed level of detail, the automatic level will be overwritten and rendered with the same level of detail independent of the font size.

Note: The Detail option is only applicable if the font type is set to Geometry.

- **Pos. Spacing**: Local kerning of the font style. Clicking the left button will decrease the position spacing between characters. Clicking the right button will increase the position spacing between characters. Fonts can be permanently modified using the Fontstyle Editor.
- **Tracking**: Changes the tracking between all characters by the same amount. Clicking the Fixed button changes the text from true type to one where each character uses the same space.

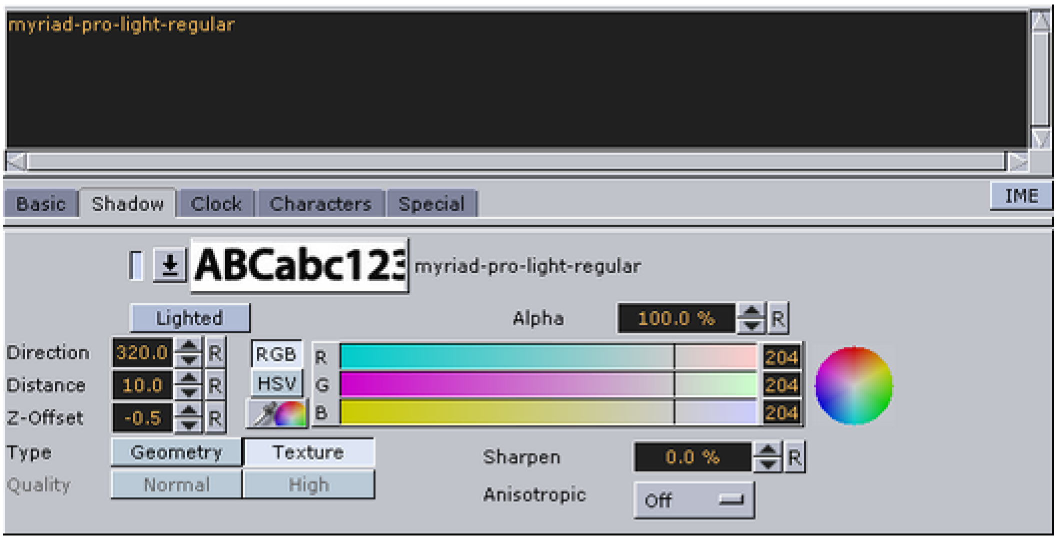
Note: To change the kerning between two individual characters only, place the cursor between those characters, and then use ARROW UP/DOWN while holding ALT to increase/decrease the kerning. Press ALT and SHIFT to increase the size of the steps.

- **Leading**: Modifies the distance between the text lines.
- **Word Spacing**: Changes the size of the blanks between the words in the text.

- **Anisotropic**: Sets the quality of textures applied to the surfaces of 3D objects when drawn at a sharp angle. Choose either to let the application determine the anisotropic filtering settings (*Off*), or select from a number of available settings. Higher values gives better image quality at the expense of some performance.

Shadow

Click the Shadow tab in the Text editor to open the Shadow panel.


myriad-pro-light-regular

ABCabc123 myriad-pro-light-regular

Alpha 100.0 %

Direction 320.0

Distance 10.0

Z-Offset -0.5

R 204

G 204

B 204

Type: Geometry / Texture

Quality: Normal / High

Sharpen 0.0 %

Anisotropic Off

From the Shadow panel, shadow can be added to the text. Click the small switch left of the font visualization to enable text shadowing. Be aware that all the modifiers that are applied to the text in the Fontstyle editor are applied to the shadow too.

- **Lighted**: Lights the font with the current light settings (see Working with Light).
- **Alpha**: Changes the alpha value of the shadow.
- **Direction**: Sets the direction of the shadow from 0 to 360°.
- **Distance**: Sets the distance between text and shadow.
- **Z-Offset**: Sets the displacement of the shadow on the Z axis.
- **Color**: Sets the color of the shadow.

Note: The Color option is only applicable if the color modifier has not been applied to the font.

- **Type**: Sets the shadow type to either geometry or texture.
- **Quality**: Sets the quality of the font to either Normal or High.

Note: The Quality option is only applicable if the font type is set to Geometry.

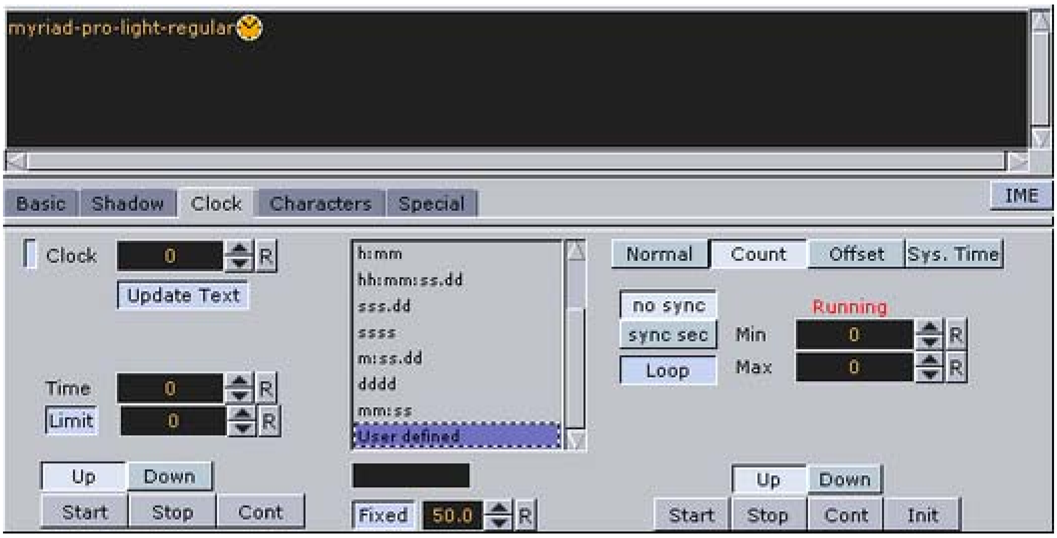
- **Sharpen**: Makes it possible to sharpen the shadow.

Note: The Sharpen option is only applicable if the font type is set to Texture.

- **Anisotropic**: Sets the quality of textures applied to the surfaces of 3D objects when drawn at a sharp angle (see Basic).

Clock

Click the Clock tab in the Text editor to open the Clock panel.



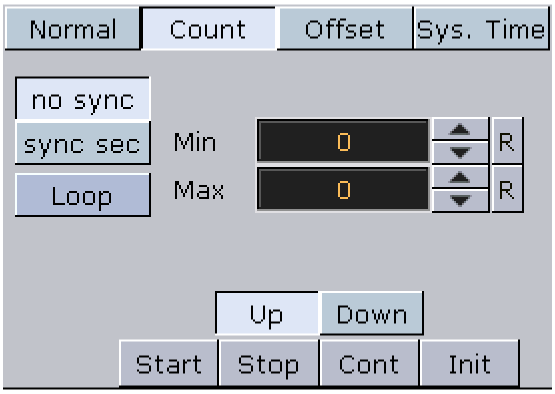
In the Clock panel, the display of a clock can be inserted and edited. Click the small switch in the upper left corner of the panel to enable the clock. The clock will be added to the text string, if any, at the place where the cursor is.

- **Clock**: Sets the ID of the clock. Up to ten different clock IDs can be used in a scene. Clocks with the same ID are linked together, which means they are synchronized and all start when starting one of them. This makes it possible to for example display different time zones.
- **Time**: Defines the starting time in seconds. If the clock for example should start to count at 2 minutes, the Time should be set to 120.
- **Limit**: Click the Limit button to enable the Limit option, which makes it possible to set a maximum count limit in seconds. If the clock for example should stop to count at 2 minutes, the Limit should be set to 120.

Note: If counting down to 0, the Limit must be set to 0 seconds, or the clock will continue to count below 0. If counting upwards starting at 0, make sure that the Limit is not 0, as the clock will not count at all.

- **Up and Down**: Specifies if the clock should count upwards or perform a countdown.
- **Start, Stop and Continue**: Makes the clock start, stop and continue counting.
- **Format**: Sets the format of the clock display. More formats (20) can be pre-defined in the configuration section under Clock Formats when Configuring Viz.
  - **H**: Stands for hours.
  - **M**: Stands for minutes.
  - **S**: Stands for seconds.
  - **D**: Stands for digits of a second. One digit shows a tenth of a second, the second digit shows a hundredth of a second, the third digit shows a millisecond, and so on.
  - **User Defined**: Select User Defined from the list to enable the User Defined text box. In this box, type a non-standardized clock format. Make sure to add correct separation characters.

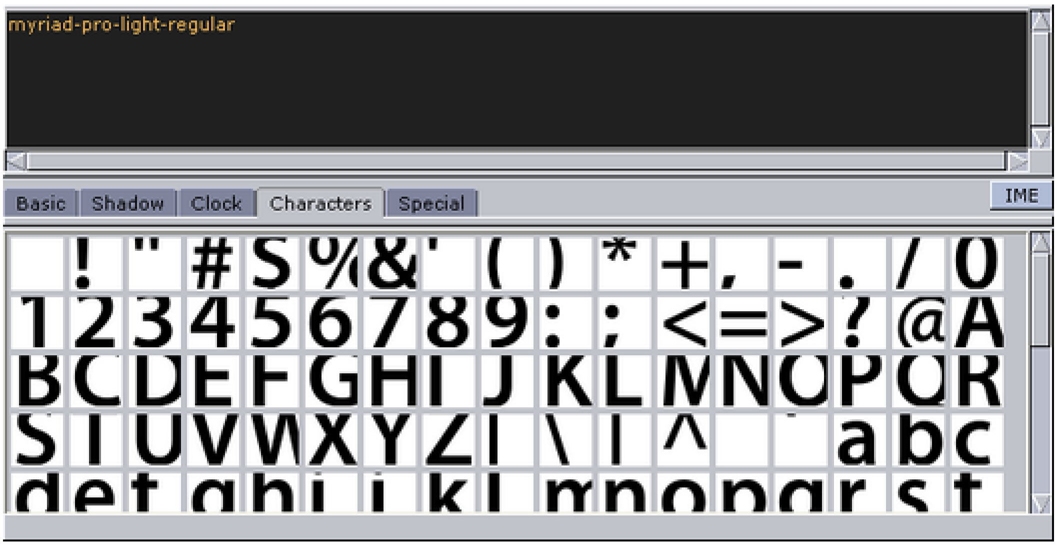
- **Fixed**: Sets a fixed text box for each letter. The kerning of non-proportional fonts vary for the various letters. For example, 1 has less kerning than 6. To avoid the letters from “jumping” around while counting, the Fixed option can be enabled. The Fixed value refers to the size of the text box around each letter. Note that the problem with changeable kerning can be avoided by using a proportional font.



- **Normal**: Shows a normal main clock view.
- **Count**: Allows the creation of a “time count” version of a standard clock, which counts a defined time lap in relation to one of the running clocks. For example, if a clock is running, another container with the same clock can be created. The state of the second clock should be set to Count, and a time range defined. When clicking the Start button under the Count section, the count clock will start counting the time defined. It is related to the main clock, so if stopping the main clock, the count clock will also stop. A typical use for this function is for counting penalty times in sport graphics. In a sport game running on efficient time, a penalty count down should stop when the main game time stops.
  - **No Sync**: With this option enabled, the count clock will run with no second synchronization to the main clock. It is almost impossible to start the count clock exactly at a whole second, so the count clock will most likely run unevenly compared to the main clock. This will in many cases not create the desired view.
  - **Sync Sec**: The count clock synchronizes its view with the main clock.
  - **Loop**: Loops the count clock within the defined count range.
  - **Up and Down**: Specifies if the clock should count upwards or perform a countdown.
  - **Start, Stop and Continue**: Makes the clock start, stop and continue counting.
  - **Init**: Resets the clock’s counter.
- **Offset**: Enables an input box where the offset value in seconds can be entered. For example, if the value is set to 20, after clicking the Start button, 20 seconds will elapse before the clock starts running.
- **Sys. Time**: Uses the system time to set the time.

Characters

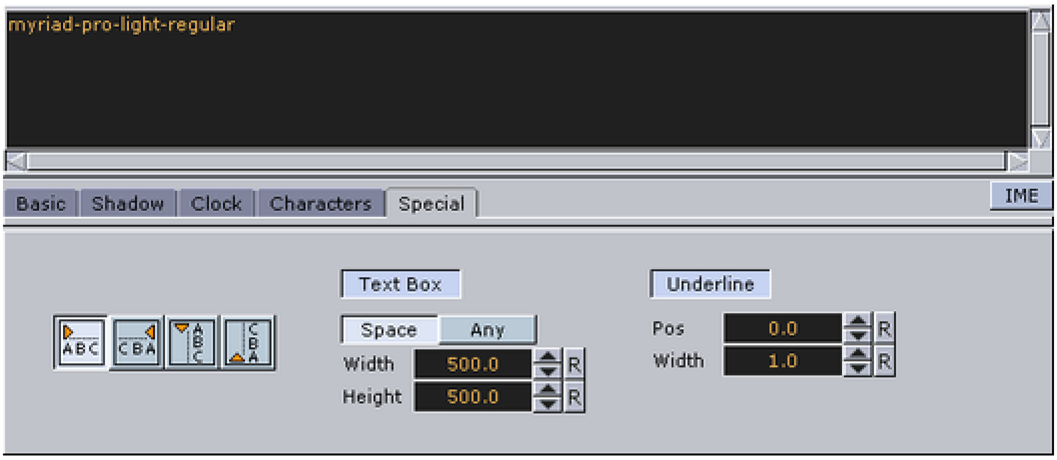
Click the Characters tab in the Text editor to open the Characters panel.


myriad-pro-light-regular

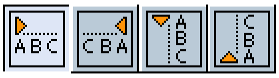
The Characters panel shows the character table in the fonts library. As an alternative to using the keyboard, the symbols from the table can be dragged onto the input box at the top of the Text editor. This is a helpful tool when using special characters that do not exist on the keyboard, or is hard to find/use.

Special

Click the Special tab in the Text editor to open the Special panel.


myriad-pro-light-regular

Width 500.0
Height 500.0
Pos 0.0
Width 1.0



- **ABC**: Switches the text orientation from left to right, right to left, downwards, and upwards.
- **Text Box**: Click the Text Box button to enable the Text Box option. The text box defines the total size of the text, and appears as an orange outline in the Output area if enabling the Bounding Box option in the Scene Editor. Note that the Text Box option is only available if the Text Orientation is set from left to right. Without the text box, the text is displayed on one line. With the text box, the text can be divided into multiple lines, and the text on each line automatically gets the selected justification. Note that there is no hyphenation function. Only the text that fits within the text box definition is shown in the Output area.

  - **Width**: Defines the width of the text box.
  - **Height**: Defines the height of the text box.
- **Underline**: Click the Underline button to enable the Underline option, which makes it possible to underline the text.
  - **Pos**: Defines the position of the underline, from-20 to 20.
  - **Width**: Defines the width of the underline, from 1 to 10.

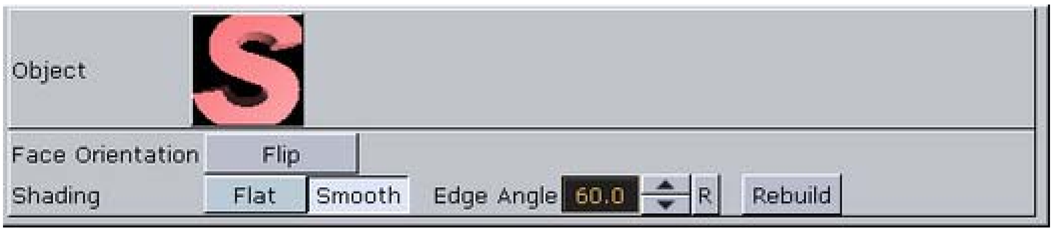
Special Fonts

Special options apply for multibyte and geometry fonts.

- **Multibyte**: Click the IME tab to open a Script editor at the top of the Text editor. The Script editor will replace the regular black input box. The Script editor makes it possible to use complex script fonts.

Note: The IME tab is only available if the current font is of type multibyte (more than 4 bytes).

- **Geometry**: If adding a geometry font, the text will be treated like a compound object created from characters. The text can be split to become a group of single character objects (not text!). Splitting of text makes it possible to edit the properties of single character objects.



Object

Face Orientation: Flip

Shading: Flat, Smooth; Edge Angle: 60.0

Note: Be aware that it is a one way operation. It is not possible to merge characters into text again, so by performing the splitting operation, the text cannot be edited anymore.

See Also

- Adding Containers
- Fonts
- Fontstyle Editor

3.3.7 Texture Editor

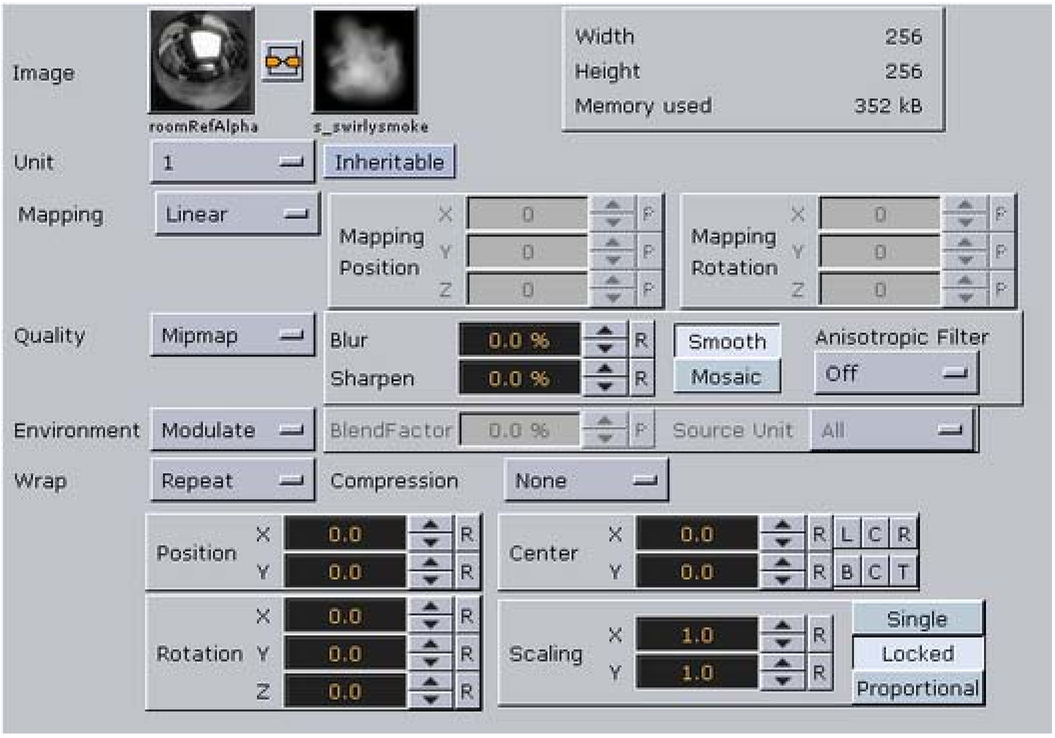


Image: roomRefAlpha, s_swirlysmoke

Width 256
Height 256
Memory used 352 kB

Unit: 1
Inheritable

Mapping: Linear
Mapping Position X 0, Y 0, Z 0
Mapping Rotation X 0, Y 0, Z 0

Quality: Mipmap
Blur 0.0 %
Sharpen 0.0 %
Smooth
Mosaic
Anisotropic Filter: Off

Environment: Modulate
BlendFactor 0.0 %
Source Unit: All

Wrap: Repeat
Compression: None

Position X 0.0, Y 0.0
Center X 0.0, Y 0.0

Rotation X 0.0, Y 0.0, Z 0.0
Scaling X 1.0, Y 1.0

Textures are applied to a container by simply dragging textures (e.g. an image) from the Server area onto a container. In the container, an icon will appear, showing a thumbnail of the texture. A texture can also be added to a scene without applying it to a container making the texture available. Geometries can be added later on, or the texture can be left as it is in the scene.

Tip: Global modifications to a texture can be performed in the editor. Modifications apply to all instances of the texture, and not just the one used in the scene.

This section contains information on the following topic and procedure:

- Properties and Parameters
- To open the texture editor

Properties and Parameters

- **Image**: Two textures can be used in one container.
  - **Default**: Shows a thumbnail of the primary texture, the same thumbnail that is shown on the container. The texture can be replaced by dragging another texture onto the drop zone.
  - **Alpha**: Shows the thumbnail of a secondary texture, if any. This texture is used as alpha channel for the default texture.
  - **Information**: Shows the original width, height, and file size of the primary texture.

Tip: To merge the two textures into one, click the little button between the textures. In the dialog box that opens, select a name for the new texture, and then click the Ok button. This will save the new texture to the database.

- **Unit**: Texture units take a texture and map them to a piece of geometry.
  - **1-4**: If the graphics card supports it is possible to have up to 4 available texture units which can be used in parallel to do multi-texturing.
  - **Inheritable**: Units can be inherited onto the textures of child containers. Multiple inherit is possible, but be sure to use different texture units for each texture that should be inherited to not overwrite a unit.
- **Mapping**: The mapping option adjusts the way texture mapping is done.
  - **Vertex**: Means that the texture mapping is done using the object texture, or UV, coordinates. These coordinates index a texture image, and are interpolated across the object to determine, at each of the object pixels, a texture image value. In most cases on 3D objects, this will give the best looking result. The built-in geometry plug-ins in Viz Artist have defined their texture coordinates. Imported objects can have texture coordinates if they have been generated by an external 3D modelling program. If for example adding a picture to a cube, all sides will look the same, and the whole image will be displayed on each side.
  - **Linear**: Is a “straight forward” mapping where the image so to say is laid over the object directly on the frontal plane. The object is then moved through the image and every pixel on the object inherits the pixel it “touches” on the image. On 3D objects, this will cause a repetition of the pixels on the surfaces in the Z axis. If for example adding a picture to a cube, the front and back side will look good, but the remaining sides will be a repetition of the pixels laying on the edge of the front and the back side.
  - **Reflection**: Maps the texture as if it where a reflection from a surrounding object. The reflection simulates that the texture image is an all-embracing object that casts its reflection on the object. The object acts in a way like a mirror.
  - **Spherical**: Spherically texture projection onto a geometry. The mapping position and rotation can be adjusted with the corresponding parameters.
  - **Cylindrical**: Cylindrical texture projection onto a geometry.
  - **Mapping Position**: Sets the texture projection origin for Spherical and Cylindrical mapping.
  - **Mapping Rotation**: Sets the texture projection rotation for Spherical and Cylindrical mapping.

Note: Mapping position and rotation are only available for Spherical and Cylindrical mapping types.

  - **Cube**: Cube texture projection onto a geometry. This option needs a vertical cross cubemap which specifies how each of the 6 side projections should look like.

Note: In order to use the Cube mapping method for texture it is not possible to use the same texture on another object with a different mapping method than Cube.

- **Quality**: The quality parameters specify a trade between rendering speed and texture quality.

  - **Pixel**: Shows the texture “as it is” with no smoothing and anti aliasing. It takes little resources to render, but it can often create a poor visualization. This can be visible as “toothed” lines and transitions, especially when close to the texture. A motion of either object or camera, especially in the Z axis, will create a “blurring motion” on the texture, as the texture scales to fit the object.
  - **Linear**: Performs a linear interpolation to smooth the texture when being magnified or shrunk. The texture looks good, but some distortions can be visible when the textured object is animated further away on the Z axis. As the object then gets smaller and smaller, the shrinking and interpolation of the texture creates a lot of “noise” on the texture. As a consequence, the linear quality is appropriate when the objects that have the texture do not change their size much.
  - **Mipmap**: Performs a linear interpolation to smooth the texture. In addition, it offers a solution to the problem that appears on the two other qualities when the object is being moved away along the Z axis or shrunk. To avoid the “noise” that we see when a texture constantly scales to try to fit onto a shrinking/magnifying, the mipmap quality creates a set of prefiltered images in a decreasing scale down to almost on pixel. These are used as the object scales up and down and a more static look of the texture is thereby achieved. Mipmap quality absorbs more memory performance than the two other qualities.

Note: The following quality options are only available for the Mipmap quality.

  - **Blur**: Makes the image blurry. The blur filter can be set to either Smooth, which blurs the image smoothly, or Mosaic, which makes the blur look like tiles.
  - **Sharpen**: This option is useful when the rendering window is scaled to half height mode.
  - **Anisotropic Filter**: The Anisotropic Filter can be set to either Off or 2-16x. Anisotropic filtering is used, when textured objects need to be rendered in z-direction, where textures usually get blurred.
- **Environment**: The environment parameters specify the way the texture RGBA information is combined with the object RGBA information.
  - **Blend**: Blends the RGBA parameters of texture and object.
  - **Decal**: Uses only the RGBA information of the texture and disregards the RGBA of the object.
  - **Modulate**: Uses the RGBA information of the texture and enables in addition use of lighting on the textured object.
  - **Add**: Adds source unit 1 and source unit 2 together.
  - **Add Signed**: Adds source unit 1 and source unit 2 then subtracts 0.5.
  - **Interpolate**: Linear interpolation between source unit 1 and source unit 2.
  - **Blend Factor**: Defines in interpolation mode how much of source unit 1 and unit 2 is used (0.0%: shows current unit, 100%: shows source unit).
  - **Source Unit**: Sets the second source unit for interpolation mode (can be all other units together or just a specific unit).

Note: Blend Factor and Source Unit environment options are only available for the Interpolated environment.

- **Wrap**: Wraps the object by repeating or clamping (no repeat - the texture’s edge color is used to fill the rest of the object) the texture onto the object’s surface. This setting comes into operation if the texture mapping cannot fill the whole object (e.g. the texture was scaled below 0.0).

  - **Repeat**: Means that the whole texture is repeated over again to fill out the object.
  - **Clamp**: Means that the pixels on the end of the image are repeated to fill out the object.
- **Compression**: Texture compression can be used to save texture memory on the graphics card.
  - **None**: No compression.
  - **DXT1**: Compresses to an RGB image with a simple on/off alpha value. This format has a 6:1 compression ratio.
  - **DXT3**: Compresses to an RGBA image with a 4:1 compression ratio.
  - **DXT5**: Compresses to an alternative RGBA image that treats the alpha channel different than DXT3 (alpha looks smoother). DXT5 has the same compression ratio as DXT3.
- **Position**: Defines the position of the texture relative to the one stored in the picture.
- **Rotation**: Defines the rotation of the texture relative to the one stored in the picture.
- **Center**: Shifts the center of the image.
- **Scaling**: Scales the image that is used as texture. Before changing the scale values of the image, make sure to set the preferred kind of scaling:
  - **Single**: Scales each axis by itself.
  - **Locked**: Scales all axes at once, so that X, Y, and Z have the same value.
  - **Proportional**: Scales all axes at once, so that the values remain proportional.

To open the texture editor

- Click the icon representing the texture either on the container or in the Properties editor.

See Also

- Adding Containers
- Images
- Image Editor

3.3.8 Adding Plug-ins

A standard Viz Artist installation provides a set of plug-ins. Most plug-ins can be added to containers, except for example scene plug-ins. As Viz Artist uses an open interface for the plug-ins, new geometries can either be obtained or created and then installed with Viz Artist.

The plug-in becomes available by selecting Built Ins from the main menu, and then selecting the appropriate category from the drop-down menu that appears above the Plugins view, or alternatively one of the tabs (e.g. GP, CP and so on).

A built in plug-in is applied to the scene by drag and drop onto the scene tree. In the container, an icon representing the plug-in will appear.

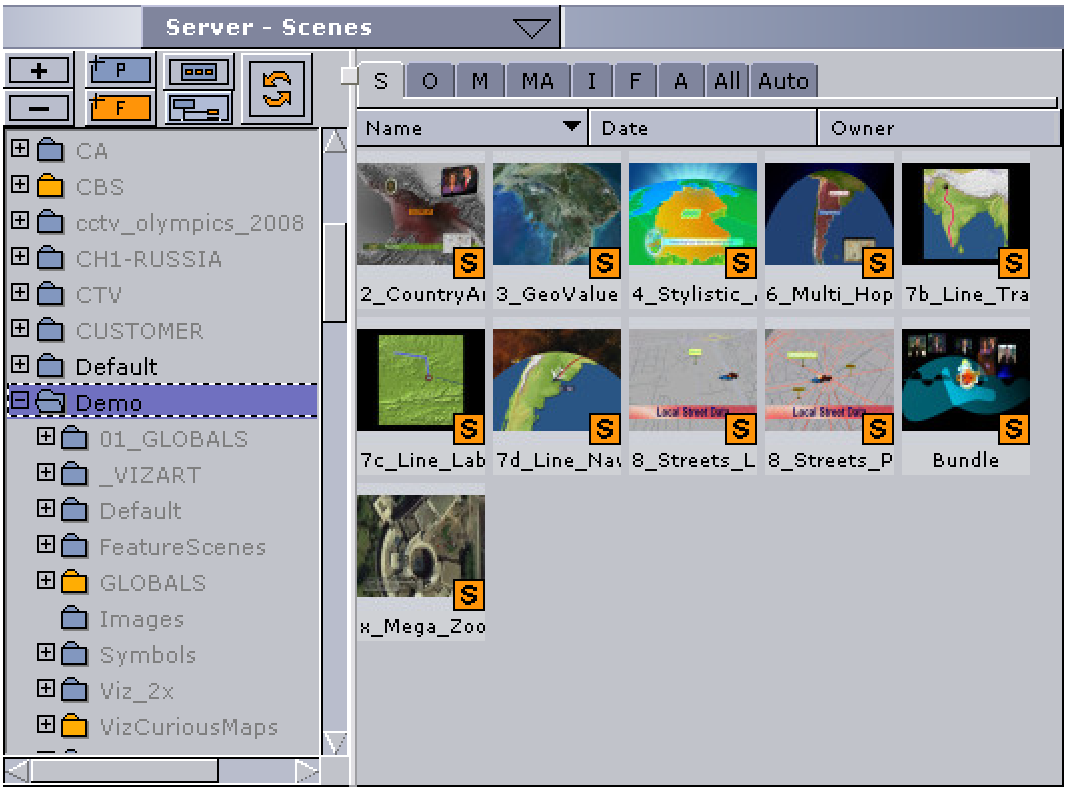
Most plug-ins have a belonging editor with properties that can be changed. To open one of the editors, click the icon representing it, either on the container or in the properties editor.

See Also

- Geometry Plugins
- Container Plugins
- Shader Plugins
- Scene Plugins
- Texture Sources

4 Manage Work and Viz Graphics Hub

Figure 4: Server area



The Server area shows an overview of all available elements in the database. The Server area is divided in two; the Tree area (left side) and Element area (right side). The Tree area shows the logical tree of projects and folders that build up the database. The Element area shows information of the elements in the selected project or folder.

Tip: To resize the two areas, drag the vertical separator.

This section contains information on the following topics:

- Viz Graphics Hub
- Tree Area
- Element Area
- Properties Pane
- Elements
- Working with Elements
- Editing Elements in the Database

To display the server area

1. Select **Server** from the **Main menu, or**

- This will display the server area together with the scene tree and render output.

2. Select **Server/Stage** from the **Main menu**, or
   - This will display the server area together with the Stage and render output.
3. Select +**Tree/Stage** from the **Main menu.**
   - This will display the server area together with the stage, scene tree, and render output.

4.1 Viz Graphics Hub

Viz Graphics Hub is the database solution where all Viz Artist 3 items are taken care of. Items can be either scenes, geometries, images, materials, fonts, and so on.

In order to start Viz Artist 3 successfully, the user must log in to a running Viz Graphics Hub.

The database can either be a local instance, where only one user can log in. Such a database can be installed on the computer where Viz Artist is used. In a multi user environment, the database is installed and run on a centralized server, providing shared access to various users at the same time.

The individual items are stored in the Viz Graphics Hub data directory as file objects. The database itself manages the items in terms of properties and Universally Unique Identifiers (Uuids).

Clients can work with and organize items through the Viz Artist GUI in a logical project and folder structure provided by the database. Although each item can be listed/referenced from more than one project/folder, it will reside in the database only once.

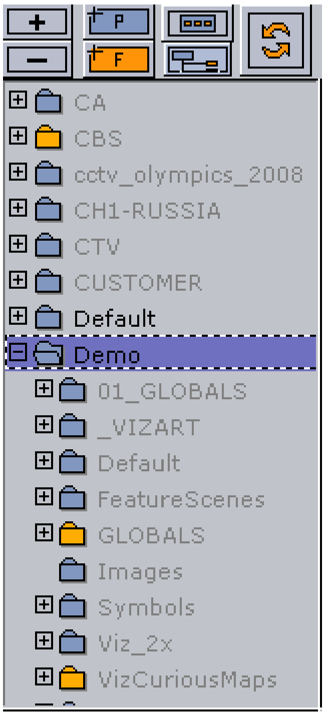
See Also

- Viz Graphics Hub User's Guide

4.2 Tree Area

The Tree area creates a structured storage of elements, and is located at the left side of the Server area.

Figure 5: Tree area



Right-clicking a project/folder in the server tree opens the Server Tree Menu.

From the Database section of the Configuration, Viz Artist can be configured to activate only the folders in the tree that contain the selected element type. The labels of projects/folders that contain the selected type are colored black, while other labels (recursive down) are grayed out.

This section contains information on the following topics:

- Folders and Projects
- Server Tree Bar
- Server Tree Menu

4.2.1 Folders and Projects

The server tree can contain two types of folders; regular folders and projects. The difference between a folder and a project is that by defining a working project, the tree can be collapsed so that only the currently selected project and its subfolders and subprojects are shown. Folders and projects can be dragged around after being added to the server tree.

This section contains information on the following topics:

- To create a folder
- To create a project
- To create a subfolder
- To create a subproject
- To rename a project/folder
- To delete a project/folder
- To expand a project/folder branch

- To expand a project/folder branch including all its subbranches
- To collapse a project/folder branch, but not its subbranches
- To collapse a project/folder branch including all its subbranches
- To add a project/folder bookmark
- To delete a project/folder bookmark

To create a folder

F

1. Drag the **Add New Folder** icon in the Server Tree Bar to a project or folder in the server tree.

To create a folder at the same hierarchy level as the target project/folder, make sure to drop the icon at the left part of the project/folder label in the server tree. Dropping it at the lower left part of the label will place the folder under the target project/folder, while dropping it at the upper left side will place it above.

2. In the dialog box that opens, type a descriptive name for the folder, and then click the **Ok** button.

Note: Special characters, such as space, are not allowed in folder names.

To create a project

P

1. Drag the **Add New Project** icon in the Server Tree Bar to a project or folder in the server tree.

To create a project at the same hierarchy level as the target project/folder container, make sure to drop the icon at the left part of the project/folder label in the server tree. Dropping it at the lower left part of the label will place the project under the target project/folder, while dropping it at the upper left side will place it above.

2. In the dialog box that opens, type a descriptive name for the project, and then click the **Ok** button.

Note: Special characters, such as space, are not allowed in project names.

To create a subfolder

1. Right-click a folder or project in the server tree.
2. In the Server Tree Menu that appears, select **Create Sub Folder**.
3. In the dialog box that opens, type a descriptive name for the subfolder, and then click the **Ok** button.

Note: Special characters, such as space, are not allowed in folder names.

Alternatively, perform the following steps:

1. Select a project or folder in the server tree.
2. Click the **Add New Folder** button in the Server Tree Bar.
3. In the dialog box that opens, type a descriptive name for the subfolder, and then click the **Ok** button.

Note: Special characters, such as space, are not allowed in folder names.

Alternatively, perform the following steps:

1. Drag the **Add New Folder** icon in the Server Tree Bar to a project or folder in the server tree.

To create a subfolder, make sure to drop the icon at the right part of the project/folder label in the server tree. Dropping the icon at the left part of the label will create a folder at the same hierarchy level.

2. In the dialog box that opens, type a descriptive name for the subfolder, and then click the **Ok** button.

Note: Special characters, such as space, are not allowed in folder names.

To create a subproject

1. Right-click a project in the server tree.

Note: It is not possible to add a subproject to a folder.

2. In the Server Tree Menu that appears, select **Create Sub Project**.
3. In the dialog box that opens, type a descriptive name for the subproject, and then click the **Ok** button.

Note: Special characters, such as space, are not allowed in project names.

Alternatively, perform the following steps:

1. Select a project or folder in the server tree.
2. Click the **Add New Project** button in the Server Tree Bar.

If a project has been selected in the server tree, the new project will be created a subproject. If a folder has been selected in the server tree, the new project will be created at the server tree root level, as it is not possible to add a subproject to a folder.

3. In the dialog box that opens, type a descriptive name for the subproject, and then click the **Ok** button.

Note: Special characters, such as space, are not allowed in project names.

Alternatively, perform the following steps:

1. Drag the **Add New Project** icon in the Server Tree Bar to a project in the server tree.

To create a subproject, make sure to drop the icon at the right part of the project label in the server tree. Dropping the icon at the left part of the label will create a project at the same hierarchy level.

Note: It is not possible to add a subproject to a folder.

2. In the dialog box that opens, type a descriptive name for the subproject, and then click the **Ok** button.

Note: Special characters, such as space, are not allowed in project names.

To rename a project/folder

1. Right-click a project/folder in the server tree.
2. In the Server Tree Menu that appears, select **Rename**.
3. In the text box that appears, type a new descriptive name, and then press **ENTER**.

Alternatively, perform the following steps:

1. Select a project/folder in the server tree.
2. Press **F2**.
3. In the text box that appears, type a new descriptive name, and then press **ENTER**.

To delete a project/folder

1. Right-click a project/folder in the server tree.
2. In the Server Tree Menu that appears, select **Delete**.

Note: If the project/folder is added as a bookmark, the bookmark will also be deleted.

Alternatively, perform the following step:

- Drag a project/folder to the trash can.

To expand a project/folder branch

⊞

- Click the **Expand** icon to the left of the project/folder name.

To expand a project/folder branch including all its subbranches

1. Select a project/folder in the server tree.

   +

2. Click the **Expand** button in the Server Tree Bar.

To collapse a project/folder branch, but not its subbranches

⊟

- Click the **Collapse** icon to the left of the project/folder name.

To collapse a project/folder branch including all its subbranches

1. Select a project/folder in the server tree.

   −

2. Click the **Collapse** button in the Server Tree Bar.

To add a project/folder bookmark

1. Right-click a project/folder in the server tree.
2. In the Server Tree Menu that appears, select **Add as Bookmark**.

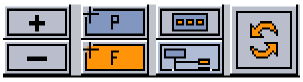
To delete a project/folder bookmark

1. Right-click a project/folder in the server tree.
2. In the Server Tree Menu that appears, select **Delete from Bookmarks**.

4.2.2 Server Tree Bar

The Tree bar consists of various buttons that makes it possible to create projects/folders and modify the appearance of the server tree. It is located at the top left side of the Server area.

Figure 6: Tree bar



The Tree bar contains the following options:

+

- **Expand**: Expands the currently selected branch and all its subbranches, see To expand a project/folder branch.

−

- **Collapse**: Collapses the currently selected branch and all its subbranches, see To collapse a project/folder branch, but not its subbranches.

- **Add New Project**: Makes it possible to add a new project to the server tree, see To create a project and To create a subproject.

- **Add New Folder**: Makes it possible to add a new folder to the server tree, see To create a folder and To create a subfolder.

- **Show Current Project**: Narrows down the tree, so that only the selected project with its subfolders and subprojects are shown.

Note: This option is only available if a project is selected in the server tree.

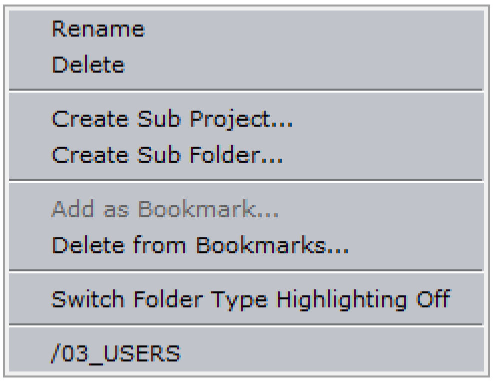
- **Show All Folders/Projects**: If the tree is narrowed down to only one project, the tree is expanded to show all folders and projects again.

- **Refresh**: Refreshes the server tree and adds updates.

4.2.3 Server Tree Menu

Right-clicking a project/folder in the server tree opens the Server Tree menu.

Figure 7: Server Tree menu



Rename
Delete
Create Sub Project...
Create Sub Folder...
Add as Bookmark...
Delete from Bookmarks...
Switch Folder Type Highlighting Off
/03_USERS

- **Rename**: Makes it possible to change the name of the selected project/folder, see To rename a project/folder.
- **Delete**: Deletes the selected project/folder, see To delete a project/folder.
- **Create Sub Project**: Makes it possible to create a subproject under the currently selected project, see To create a subproject.

Note: It is not possible to add a subproject to a folder.

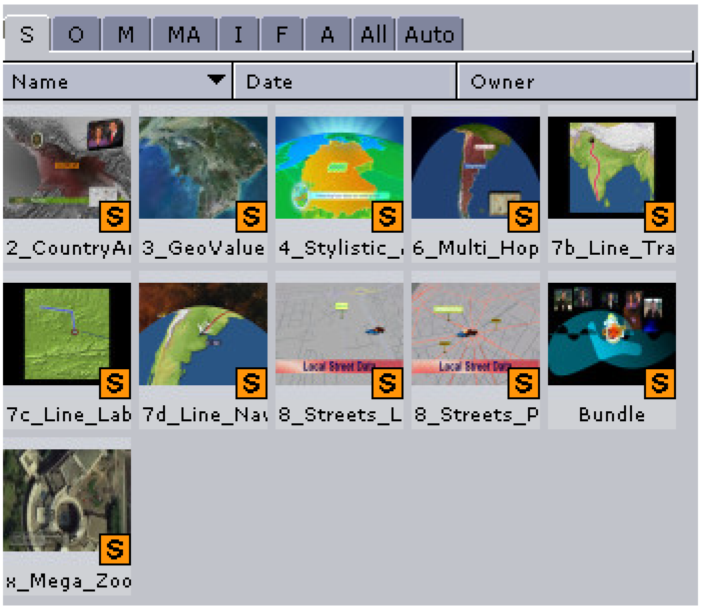
- **Create Sub Folder**: Makes it possible to create a subfolder under the currently selected project/folder, see To create a subfolder.
- **Add as Bookmark**: Adds the selected project/folder to the list of bookmarks, see To add a project/folder bookmark.
- **Delete from Bookmarks**: Deletes the selected project/folder from the list of bookmarks, see To delete a project/folder bookmark.

- **Switch Folder Type Highlighting On/Off**: Enables you to switch on or off highlighting of folder types. For example when the Scene tab is selected the folder will be highlighted if the folder contains scenes.

At the bottom of the Server Tree menu, below the command separator, is the list of available bookmarks. Projects/folders that are used often should be added to the list of bookmarks. Clicking a bookmark in the list will open the corresponding project/folder.

4.3 Element Area

Figure 8: Element area



The Element area shows the elements that are available for the project/folder selected in the server tree. The Element area is located at the right side of the Server area.

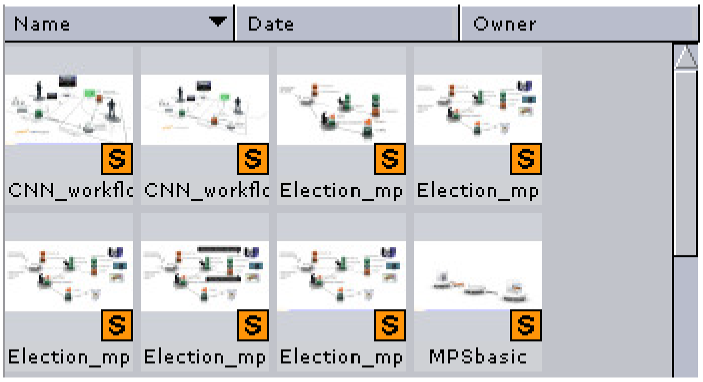
This section contains information on the following topics:

- Element List
- Server File Context Menu
- Content Switcher
- Server Menu
- Properties Pane

4.3.1 Element List

The Element list shows all elements in the selected project/folder.

Figure 9: Element list showing all elements in selected project/folder



Elements are by default displayed with thumbnail and element name, and sorted alphabetically by name. To display additional information about the elements, see Adding Details and Adding Properties. Right-clicking an item in the Element List opens a the Server File Context Menu context menu.

Tip: Holding the pointer over a thumbnail will display the whole element name as a tool tip.

If details are added to the Element list, see Adding Details, the following columns are available:

- **Name**: Shows the element name.
- **I**: If this column heading is enabled, the elements in the list will be displayed with thumbnails.
- **Date**: Shows the last date and time when the element was modified.
- **Owner**: Shows the user name of the element owner.

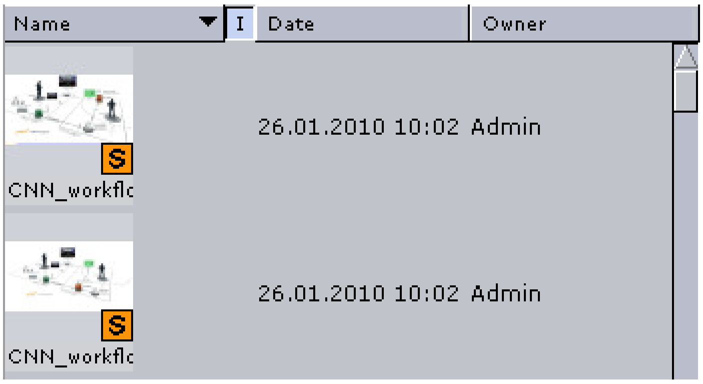
This section contains information on the following topics:

- Adding Details
- Adding Properties
- Sorting Elements

Adding Details

The elements in the list can be displayed with additional information about owner and creation date.

Figure 10: Element list showing elements with detailed information



To display the elements in the list with additional information, perform the following step:

- Select **Detail** from the Server Menu.

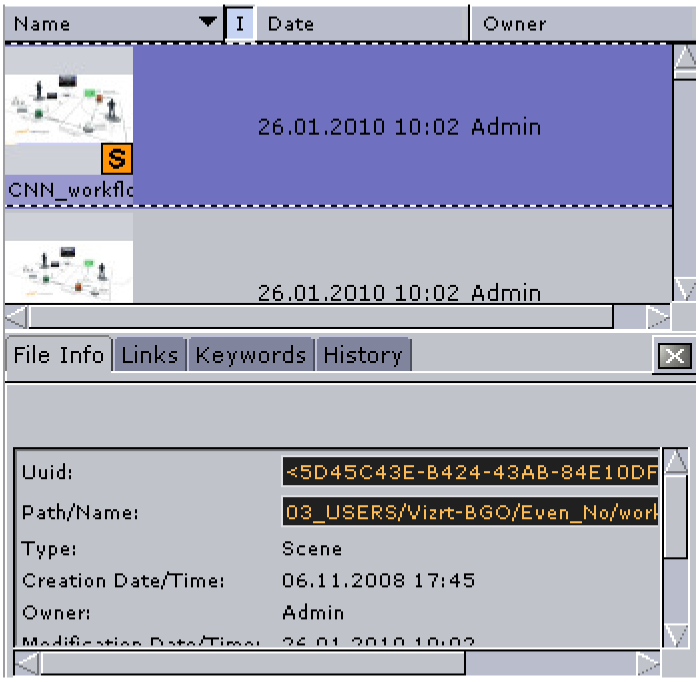
Alternatively, perform the following step:

- Press the keyboard shortcut **CTRL+Y** while holding the pointer over the Server area.

Adding Properties

Enabling the Properties Pane provides detailed information about file, links, keywords, and history for the selected element in the list.

Figure 11: Element list showing elements with detailed information and properties



Uuid: <5D45C43E-B424-43AB-84E10DF

Path/Name: 03_USERS/Vizrt-BGO/Even_No/work

Type: Scene

Creation Date/Time: 06.11.2008 17:45

Owner: Admin

Modification Date/Time: 26.01.2010 10:02

To display the Properties Pane and provide detailed information about file, links, keywords, and history for the selected element in the list, perform the following step:

- Select **Show Properties** from the Server Menu.

Alternatively, perform the following step:

- Press the keyboard shortcut **CTRL+P** while holding the pointer over the Server area.

Alternatively, perform the following steps:

1. Right-click an element in the Element List.
2. In the menu that appears, select **Show Properties**.

Sorting Elements

Note: Details must be added to the Element list in order to perform the following procedures, see Adding Details.

By default, elements in the Element List are sorted alphabetically by name.

To sort the elements by name from A-Z, and making sure that the A elements are placed at the top of the list, perform the following step:

- Click the **Name** column heading, making sure that the arrow next to the heading label points downward.

To sort the elements by name from A-Z, and making sure that the Z elements are placed at the top of the list, perform the following step:

- Click the **Name** column heading, making sure that the arrow next to the heading label points upward.

To sort the elements by date, and making sure that the most recently modified elements are placed at the top of the list, perform the following step:

- Click the **Date** column heading, making sure that the arrow next to the heading label points upward.

To sort the elements by date, and making sure that the most recently modified elements are placed at the bottom of the list, perform the following step:

- Click the **Date** column heading, making sure that the arrow next to the heading label points downward.

To sort the elements by owner from A-Z, and making sure that the A owners are placed at the top of the list, perform the following step:

- Click the **Owner** column heading, making sure that the arrow next to the heading label points downward.

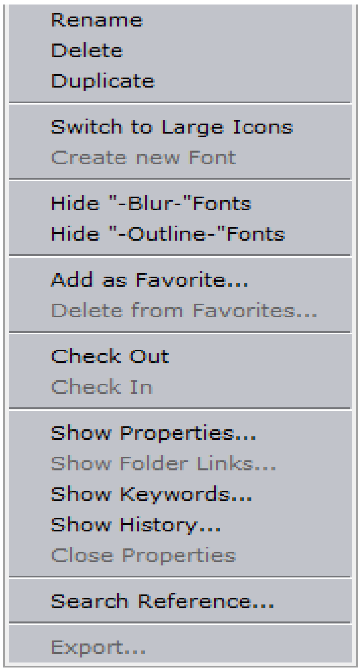
To sort the elements by owner from A-Z, and making sure that the Z owners are placed at the top of the list, perform the following step:

- Click the **Owner** column heading, making sure that the arrow next to the heading label points upward.

4.3.2 Server File Context Menu

Right-clicking an element in the Element List opens a menu, which contains the following options:

Figure 12: Server file context menu



- **Rename**: Makes it possible to change the name of the selected element. :
  - To rename a scene
  - To rename an object
  - To rename an Advanced Material
  - To rename an image
  - To rename a font
  - To rename an audio clip.
- **Delete**: Makes it possible to delete the selected element.
  - To delete a scene
  - To delete an object
  - To delete an Advanced Material
  - To delete an image
  - To delete a font
  - To delete an audio clip
- **Delete with all references**: Deletes the selected element and all its referenced elements. Referenced elements are not deleted if they are referenced by elements you are not deleting.
- **Duplicate**: Creates an independent copy of the selected element. The copy will be placed in the same folder as the original element. For more information, see To copy an element.
- **Switch to Small/Large Icons**: Displays small or large element icons in the Element List.
- **Create New [Element]**: Makes it possible to add a new element:
  - In the dialog box that opens, type a descriptive element name.
  - Click the **Ok** button.

Note: The create new element option is only available for scenes and materials.

- **Show/Hide "Blur" Fonts**: Shows (or hides) the blur fonts in the Server area. The default is configured in Various section of the User Interface portion of Viz Config.
- **Show/Hide "Outline" Fonts**: Shows (or hides) the outline fonts in the Server area. The default is configured in Various section of the User Interface portion of Viz Config.
- **Add as Favorite**: Adds the selected element to the Favorites Bar in the Scene Tree area.

Note: It is not possible to add scenes as favorites.

- **Delete from Favorites**: Deletes the selected element from the Favorites Bar.
- **Check Out**: Checks out the selected element, so that only the user that checks out the file can modify it. For more information, see Active Elements and To check out an element.

Tip: Other users can see the name of the user that has checked out the file from the Properties Pane.

- **Check In**: Checks in the selected element, so that all users can modify it. For more information, see Active Elements and To check in an element.

Note: The check in option is only available for elements that have been checked out.

- **Show Properties**: Displays the Properties Pane at the bottom of the Element area, with the File Info tab selected.
- **Show Folder Links**: Displays the Properties Pane at the bottom of the Element area, with the Links tab selected.

Note: The folder links option is only available for elements that are linked. For more information, see To link an element.

- **Show Keywords**: Displays the Properties Pane at the bottom of the Element area, with the Keywords tab selected.
- **Show History**: Displays the Properties Pane at the bottom of the Element area, with the History tab selected.
- **Close [Properties]**: Closes the Properties Pane.
- **Search Reference**: Opens the Search window and searches the database for other elements that use the selected element. For more information, see References.

Note: It is not possible to perform a reference search for materials.

- **Export**: Opens the Select Export Directory dialog box, where the selected element can be exported. For more information, see To export an image.

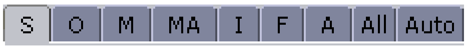
Note: This option is only available for images and audio.

- **Export as XML**: Allows you to export an XML representation of a scene and other objects into a file. XML files can be used to compare two versions of a scene using a file comparison utility (aka diff tool).
- **Deploy (with Prefix...)**: Allows you to deploy scenes and other elements onto a different Viz Graphics Hub(s). See the Deploy configuration section for more information.

4.3.3 Content Switcher

Projects/folders can contain various types of elements. Selecting an element type from the Content switcher narrows down the list of elements. Selecting the All tab displays all elements in the selected project/folder, regardless of type.

Figure 13: Content switcher



- **S**: Shows only the Scenes in the selected project/folder.
- **O**: Shows only the Objects in the selected project/folder.
- **M**: Shows only the Materials in the selected project/folder.
- **MA**: Shows only the Materials Advanced in the selected project/folder.
- **I**: Shows only the Images in the selected project/folder.
- **F**: Shows only the Fonts in the selected project/folder.
- **A**: Shows only the Audio Clips in the selected project/folder.
- **All**: Shows all types of elements in the selected project/folder.
- **Auto**: Shows the element type with the most instances in the selected project/folder. For example, if there are five materials in a project/folder, but only one instance of each of the other element types, the materials will be shown in the Element List.

This section contains information on the following topics:

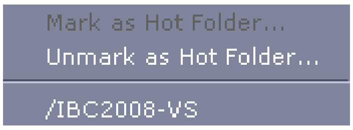
- Content Switcher Menu
- Viewing History List
- To show the History list

Content Switcher Menu

Right-clicking one of the tabs in the Content Switcher opens the Content Switcher menu. At the bottom of the menu, below the command separator, is the list of hot folders for the selected element type. Hot folders are favorite folders that the user often uses. Selecting a hot folder in the list will open the corresponding project/folder and switch to the selected element type.

The Content switcher menu contains the following options:

Figure 14: Content Switcher menu



- **Mark as Hot Folder**: Adds the currently selected project/folder to the list of favorites.

Note: Each element type has a separate list of hot folders.

- **Unmark as Hot Folder**: Removes the currently selected project/folder from the list of favorites.

Viewing History List

The History list shows an overview of previously used projects/folders and is saved independent of type. Each element type has a separate History list, which can contain up to 20 hits.

Note: A project/folder is added to the History list when an element has been dragged, regardless if the element was actually used in a scene.

To show the History list

- Press CTRL while right-clicking and holding the pointer over one of the tabs in the Content Switcher, or CTRL while left clicking to directly switch to the type and project/folder of earliest used folder in history.

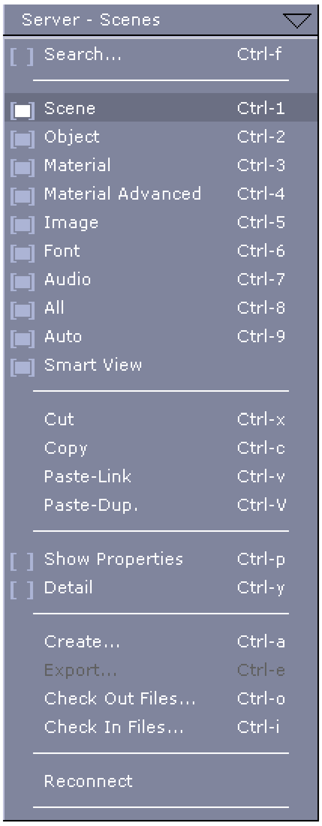
See Also

- Lights and Materials

4.3.4 Server Menu

At the top of the Server area is the Server menu.

Figure 15: Server menu



- **Search**: Opens the Search window, see Searching.
- **Scene**: Shows only the Scenes in the selected project/folder.
- **Object**: Shows only the Objects in the selected project/folder.
- **Material**: Shows only the Materials in the selected project/folder.
- **Material Advanced**: Shows only the Materials Advanced in the selected project/folder.
- **Image**: Shows only the Images in the selected project/folder.
- **Font**: Shows only the Fonts in the selected project/folder.
- **Audio**: Shows only the Audio Clips in the selected project/folder.
- **All**: Shows all types of elements in the selected project/folder.
- **Auto**: Shows the element type with the most instances in the selected project/folder. For example, if there are five materials in a project/folder, but only one instance of each of the other element types, the materials will be shown in the Element List.
- **Smart View**: Opens the SmartView Search functionality.
- **Cut**: Cuts the currently selected element, so that it can be pasted in another project/folder. For more information, see To move an element.
- **Copy**: Copies the currently selected element, so that it can be pasted in another project/folder. For more information, see To copy an element.
- **Paste-Link**: Pastes a cut or copied element as a linked file. Since the new element is linked, the element thumbnail will contain a Link icon in the lower left corner. For more information, see To link an element.
- **Paste-Dup**: Pastes a cut or copied element as an independent file. For more information, see To copy an element.

- **Show Properties**: Displays the Properties Pane at the bottom of the Element area, with the File Info tab selected. For more information, see Adding Properties.
- **Detail**: By default, elements are presented in the Element list with thumbnail and name. By selecting the Detail option, additional information about owner and creation date is added. For more information, see Adding Details.
- **Create**: Makes it possible to add a new element. For more information, see Opening a Scene and Creating Materials.

Note: The create option is only available for scenes and materials.

- **Export**: Makes it possible to export an element. For more information, see To export an image.

Note: The export option is only available for images.

- **Export as XML**: Allows you to export an XML representation of a scene and other objects into a file. XML files can be used to compare two versions of a scene using a file comparison utility (aka diff tool).
- **Check Out Files**: Checks out the selected elements, so that only the user that checks out the files can modify them. For more information, see Active Elements and To check out an element.

Tip: Other users can see the name of the user that has checked out the files from the Properties Pane.

- **Check In Files**: Checks in the selected elements, so that all users can modify them. For more information, see Active Elements and To check in an element.

Note: The check in option is only available for elements that have been checked out.

- **Reconnect**: Reestablishes the connection to Viz Graphics Hub.

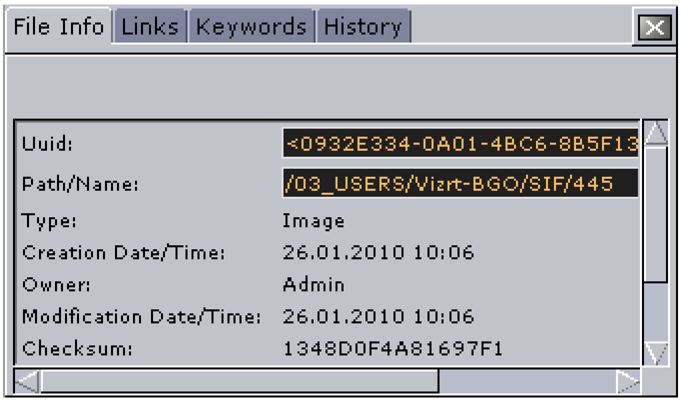
See Also

- Lights and Materials
- Server View Shortcuts

4.3.5 Properties Pane

The Properties pane shows detailed information about the selected element in the Element List. It is displayed below the Element list.

Figure 16: Properties pane with File Info tab selected

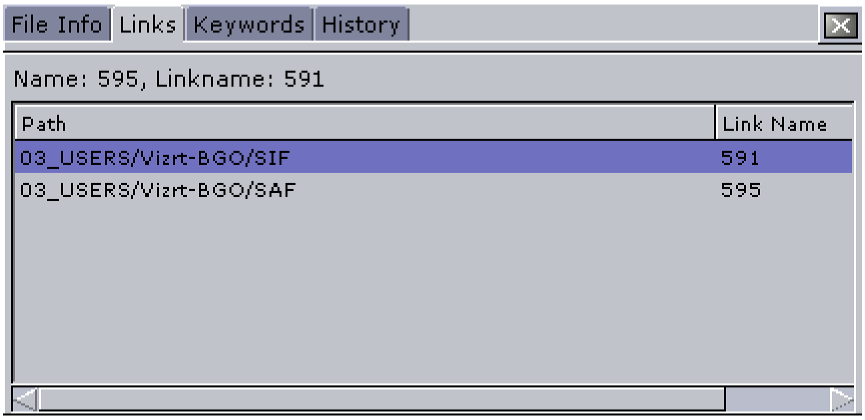


Uid: <0932E334-0A01-4BC6-8B5F13
Path/Name: /03_USERS/Vizrt-BGO/SIF/445
Type: Image
Creation Date/Time: 26.01.2010 10:06
Owner: Admin
Modification Date/Time: 26.01.2010 10:06
Checksum: 1348D0F4A81697F1

The File Info tab shows various information about the selected element:

- **UUID**: Shows the universally unique element identifier.
- **Path/Name**: Shows the path to where the element is located and the name of the element.
- **Type**: Shows the element type, for example font, image, or material.
- **Creation Date/Time**: Shows the date and time when the element was created.
- **Owner**: Shows the name of the user that owns the element.
- **Modification Date/Time**: Shows the date and time when the element was last modified.
- **Checksum**: Displays the file's checksum in case you need to check its validity.
- **File size**: Shows the size of the element file.
- **Checked Out By**: If the element has been checked out, it shows the name of the user that has performed the checkout operation. For more information, see Active Elements.
- **Locked By**: If the element has been locked, it shows the name of the user that has performed the lock operation. For more information, see Active Elements.

Figure 17: Properties pane with Links tab selected



Name: 595, Linkname: 591

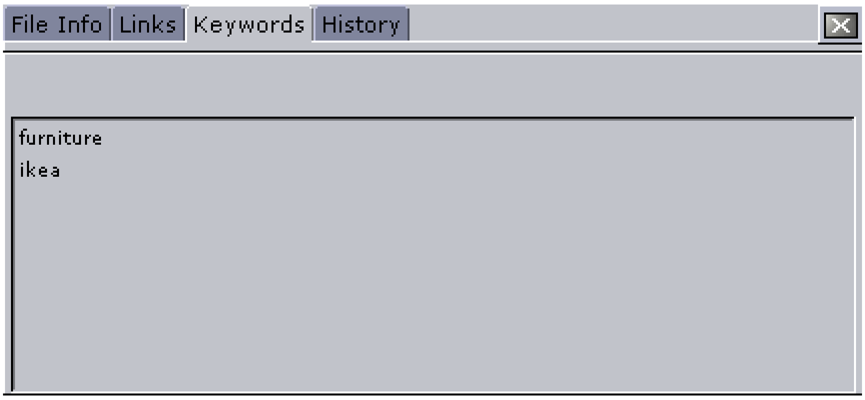
Path	Link Name
03_USERS/Vizrt-BGO/SIF	591
03_USERS/Vizrt-BGO/SAF	595

The Links tab shows the links that are assigned to the selected element:

- **Path**: Shows the path to where the linked elements are located.
- **Link Name**: Shows the names of the linked elements.

Tip: Click one of the links to select the project/folder where the linked file is placed, and to display the element in the Element List.

Figure 18: Properties pane with Keywords tab selected

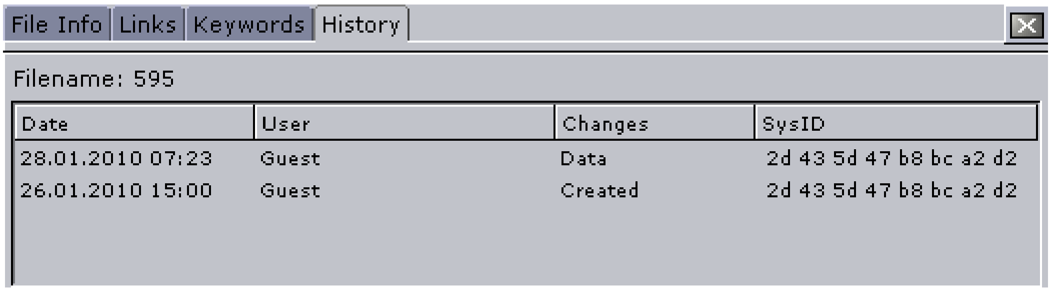


furniture
ikea

The Keywords tab shows the keywords that are assigned to the selected element. Keywords can be assigned for example when importing elements to Viz Artist. See Assigning Keywords.

Note: If multiple elements have been selected, and some keywords are grayed out, the gray keywords are assigned to at least one of the selected elements, but not all.

Figure 19: Properties pane with History tab selected



Filename: 595

Date	User	Changes	SysID
28.01.2010 07:23	Guest	Data	2d 43 5d 47 b8 bc a2 d2
26.01.2010 15:00	Guest	Created	2d 43 5d 47 b8 bc a2 d2

The History tab shows the history of the selected element.

- **Date**: Shows the date and time when the element was created or modified.
- **User**: Shows the name of the user that created or modified the element.
- **Changes**: Shows the operation that was performed on the element, for example if the document was created.
- **SysID**: Shows the Viz Artist license number on the machine that performed the operation.

This section contains information on the following topics:

- Displaying the Properties Pane
- Closing the Properties Pane

Displaying the Properties Pane

To show the Properties pane with the File Info tab selected, see Adding Properties.

To show the Properties pane with the Links tab selected, perform the following step:

- Select **Show Folder Links** from the Server File Context Menu.

Note: The show folder links option is only available for elements that are linked. For more information, see To link an element.

To show the Properties pane with the Keywords tab selected, perform the following step:

- Select **Show Keywords** from the Server File Context Menu.

To show the Properties pane with the History tab selected, perform the following step:

- Select **Show History** from the Server File Context Menu.

Closing the Properties Pane

To close the Properties pane, perform the following step:

- Select **Close [Properties]** from the Server File Context Menu.

Alternatively, perform the following step:

- Click the **Close** icon in the upper right corner of the Properties Pane.

4.4 Elements

Elements can be items that are used to build up a scene, or they can be scenes themselves.

This section contains information on the following topics:

- Scenes
- Objects
- Images
- Fonts
- Audio Clips

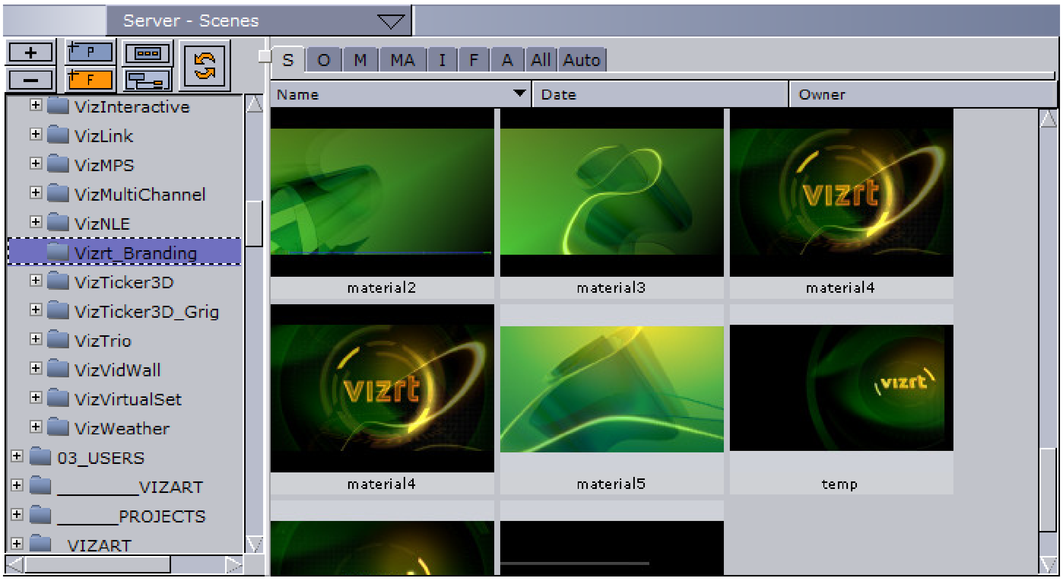
See Also

- Lights and Materials

4.4.1 Scenes

Scenes are root elements that hold other elements. Existing scenes are stored in the database and available through the Server area, see To view scenes in a project/folder.

Figure 20: Server area showing all scenes in selected folder



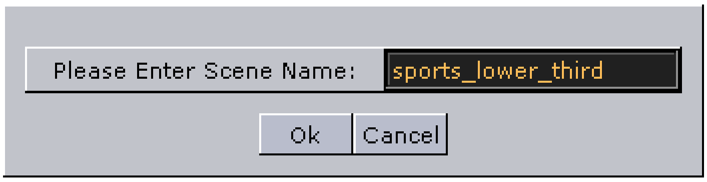
If a scene is selected, the Scene Tree area shows the logical visualization of all the elements within the scene. The scene tree consists of containers that hold elements, organized in a hierarchy and grouped in logical divisions. The elements can be geometries, images, materials, texts, transparency functions, key functions, and many more. For more information, see Working with the Scene Tree.

This section contains information on the following topics:

- To create a Scene
- To import a Scene
- To rename a scene
- To search for scene references
- To delete a scene
- To save a scene
- To undo/redo operations in a scene
- To view scenes in a project/folder

To create a Scene

1. Select the project/folder where the scene should be placed.
2. Make sure to open the scene view of the selected folder. See To view scenes in a project/folder.
3. Right-click the Element List.
4. In the menu that appears, select **Create new Scene**.


Please Enter Scene Name: sports_lower_third

5. In the dialog box that opens, type a descriptive name for the scene.

Note: Special characters, such as space, are not allowed in element names.

6. Click the **Ok** button.

Alternatively, perform the following steps:

1. Select the project/folder where the scene should be placed.
2. Make sure to open the scene view of the selected folder, see To view scenes in a project/folder.
3. Select **Create** from the Server Menu.
4. In the dialog box that opens, type a descriptive name for the scene.
5. Click the **Ok** button.

Alternatively, perform the following steps:

1. Select the project/folder where the scene should be placed.
2. Make sure to open the scene view of the selected folder, see To view scenes in a project/folder.
3. Press the keyboard shortcut **CTRL+A** while holding the pointer over the Server area.
4. In the dialog box that opens, type a descriptive name for the scene.
5. Click the **Ok** button.

To import a Scene

- For information about how to import a scene, or an archive, from a hard disk or network share to the database, see Importing.

To rename a scene

1. Select the project/folder where the existing scene is placed.
2. Make sure to open the scene view of the selected folder, see To view scenes in a project/folder.
3. Select the scene that should be renamed.
4. Press **F2**.
5. In the text field that appears, type a new descriptive name.

Note: Special characters, such as space, are not allowed in element names. Also, if renaming a linked element, only the name of the selected element will change, not the linked one. For more information, see To link an element.

Alternatively, perform the following steps:

1. Select the project/folder where the existing scene is placed.

2. Make sure to open the scene view of the selected folder, see To view scenes in a project/folder.
3. Right-click the scene that should be renamed.
4. In the menu that appears, select **Rename**.
5. In the text field that appears, type a new descriptive name.

To search for scene references

1. Select the project/folder where the existing scene is placed.
2. Make sure to open the scene view of the selected folder, see To view scenes in a project/folder.
3. Right-click the scene that should be used to search for References.
4. In the menu that appears, select **Search Reference**.

To delete a scene

To delete a scene, perform the following steps:

Tip: Multiple scenes can be deleted at once.

1. Select the project/folder where the existing scene is placed.
2. Make sure to open the scene view of the selected folder, see To view scenes in a project/folder.
3. Drag the scene that should be deleted to the trash can.
4. In the dialog box that opens, confirm the delete operation.

Note: If deleting a linked element, only the link to the project/folder will be removed. The element itself will remain in the database. For more information, see To link an element. Also, referenced elements cannot be deleted. Referenced elements are used by other elements.

Alternatively, perform the following steps:

1. Select the project/folder where the existing scene is placed.
2. Make sure to open the scene view of the selected folder, see To view scenes in a project/folder.
3. Right-click the scene that should be deleted.
4. In the menu that appears, select **Delete**.
5. In the dialog box that opens, confirm the delete operation.

To save a scene

- Click the **Save** button in the lower left corner of the window.

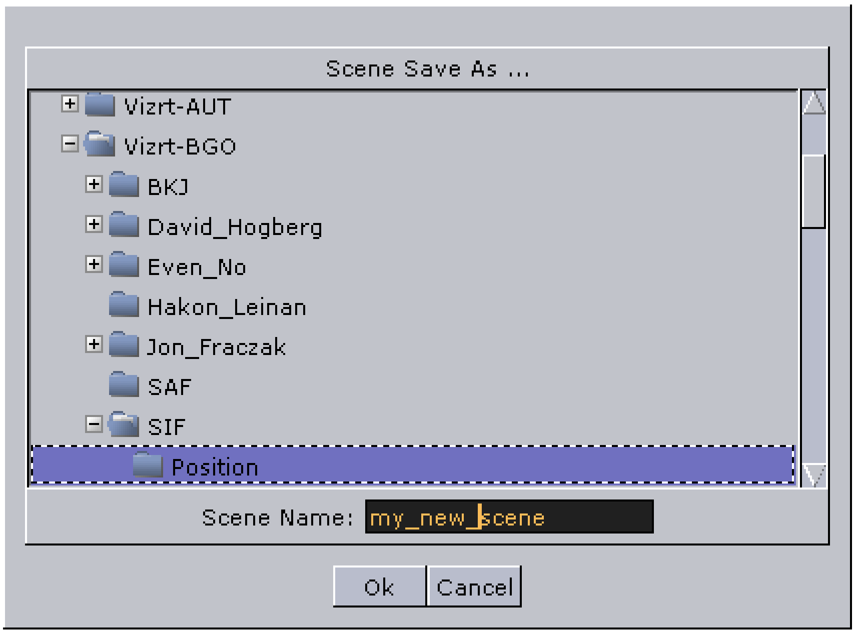
Alternatively, perform the following step:

- Press the keyboard shortcut **CTRL+S** while holding the pointer over the Scene Tree area.

To save a scene with a new name

1. Click the **Save As** button in the lower left corner of the window.

Figure 21: Scene Save As dialog box



Scene Name: my_new_scene

2. In the **Scene Save As** dialog box that opens, type a descriptive name in the **Scene Name** text box, and then click the **Ok** button.

To undo/redo operations in a scene

- Click the **Undo**/**Redo** button in the lower left corner of the window.

Tip: Holding the pointer over the **Undo**/**Redo** buttons will display a text in the status bar describing the action that is about to be performed. Also, operations can be undone/redone although the scene has been saved. However, this is not possible after the scene has been closed and reopened.

Alternatively, perform the following step:

- Press the keyboard shortcut **CTRL+Z/Y** while holding the pointer over the Scene Tree area.

To view scenes in a project/folder

1. Select a project/folder in the server tree.
2. Select the **S** tab of the Content Switcher.

Alternatively, perform the following steps:

1. Select a project/folder in the server tree.
2. Select **Scene** from the Server Menu.

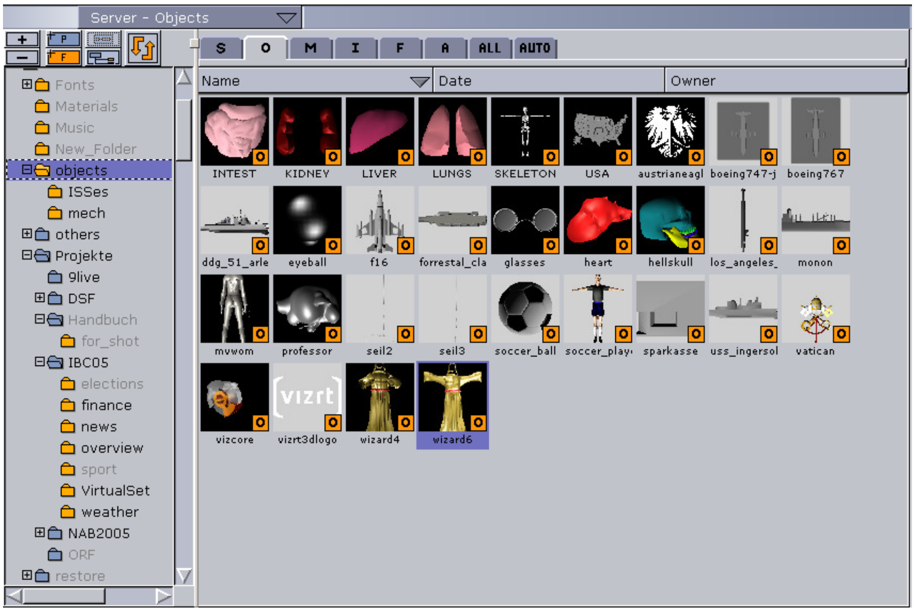
Alternatively, perform the following steps:

1. Select a project/folder in the server tree.
2. Press the keyboard shortcut **CTRL+1** while holding the pointer over the Server area.

4.4.2 Objects

An object holds information about geometry. Existing objects are stored in the database and available through the Server area, see To view objects in a project/ folder.

Figure 22: Server area showing all objects in selected folder



Objects can be created in Viz Artist or imported from an external modelling program. Existing objects are stored in the database and available through the Server area. For more information about importing objects, see Importing.

Viz Artist also provides a set of objects that are so-called built-in Geometry Plugins. These geometries are common plug-ins that can be used in scenes.

An object can be added to a scene as a standalone container. To add an object as a container item (part of another container), drag the object from the Element List onto the container in the scene tree. Containers in the scene tree, including animations, can be merged into compound objects and saved as reusable components.

This section contains information on the following procedures:

- To rename an object
- To search for object references
- To delete an object
- To view objects in a project/folder

To rename an object

1. Select the project/folder where the existing object is placed.

2. Make sure to open the object view of the selected folder, see To view objects in a project/folder.
3. Select the object that should be renamed.
4. Press **F2**.
5. In the text field that appears, type a new descriptive name.

Note: Special characters, such as space, are not allowed in element names. Also, if renaming a linked element, only the name of the selected element will change, not the linked one. For more information, see To link an element.

Alternatively, perform the following steps:

1. Select the project/folder where the existing object is placed.
2. Make sure to open the object view of the selected folder, see To view objects in a project/folder.
3. Right-click the object that should be renamed.
4. In the menu that appears, select **Rename**.
5. In the text field that appears, type a new descriptive name.

To search for object references

1. Select the project/folder where the existing object is placed.
2. Make sure to open the object view of the selected folder, see To view objects in a project/folder.
3. Right-click the object that should be searched.
4. In the menu that appears, select **Search Reference**.

, References.

To delete an object

Tip: Multiple objects can be deleted at once.

1. Select the project/folder where the existing object is placed.
2. Make sure to open the object view of the selected folder, see To view objects in a project/folder.
3. Drag the object that should be deleted to the trash can.
4. In the dialog box that opens, confirm the delete operation.

Note: If deleting a linked element, only the link to the project/folder will be removed. The element itself will remain in the database. For more information, see To link an element. Also, referenced elements cannot be deleted. Referenced elements are used by other elements.

Alternatively, perform the following steps:

1. Select the project/folder where the existing object is placed.
2. Make sure to open the object view of the selected folder, see To view objects in a project/folder.
3. Right-click the object that should be deleted.

4. In the menu that appears, select **Delete**.
5. In the dialog box that opens, confirm the delete operation.

To view objects in a project/folder

To view all objects available in a specific project/folder, perform the following steps:

1. Select a project/folder in the server tree.
2. Select the **O** tab of the Content Switcher.

Alternatively, perform the following steps:

1. Select a project/folder in the server tree.
2. Select **Object** from the Server Menu.

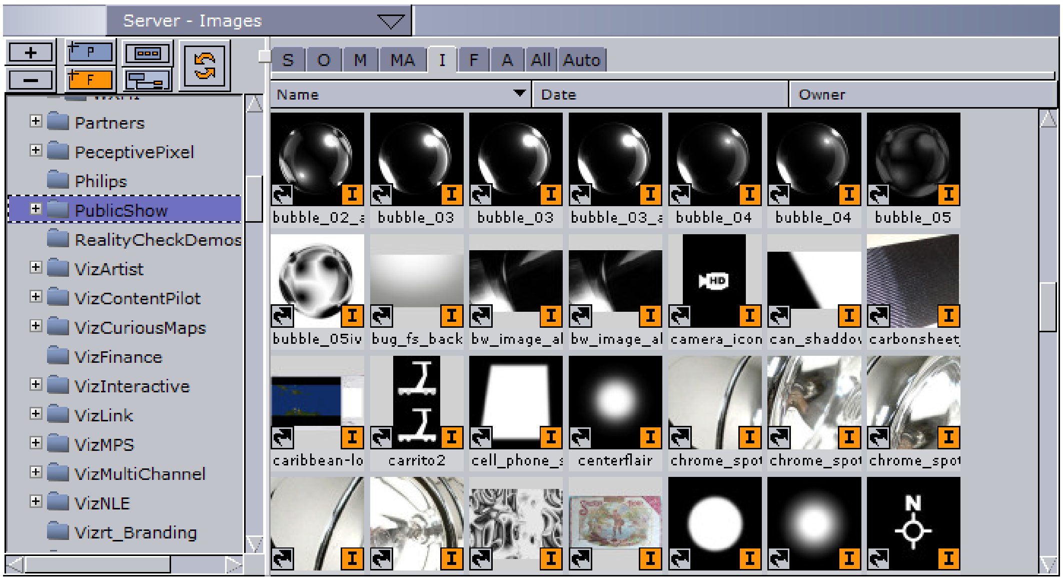
Alternatively, perform the following steps:

1. Select a project/folder in the server tree.
2. Press the keyboard shortcut **CTRL+2** while holding the pointer over the Server area.

4.4.3 Images

Viz Artist support still images in formats TIFF, BMP, SGI, JPG, TGA, PSD, XPM, HDR, PNG. Existing images are stored in the database and available through the Server area.

Figure 23: Server area showing all images in selected folder



An image can be added to a scene as a standalone container, see Adding Containers. To add an image as a container item (part of another container), drag the image from the Element List onto the container in the scene tree. For more information, see Manipulating Properties.

This section contains information on the following procedures:

- To add an Image
- To edit an image
- To rename an image
- To search image references
- To delete an image
- To view images in a project/folder
- To export an image

To add an Image

There are two main ways to add images to the database:

- **Importing Images**: Images can be imported from a hard disk or network share to the database, see Importing.

Tip: Multiple images can be imported at once.

- **Taking Snapshots**: Images can be created through taking a snapshot of the render output.

To edit an image

1. Open the Image Editor, which makes it possible to edit the settings of the image.
2. Select the project/folder where the existing image is placed.
3. Make sure to open the image view of the selected folder, see To view images in a project/folder.
4. Double-click the image that should be modified.
5. In the Image Editor that opens, define the new settings.
6. Click the **Save** button.

To rename an image

1. Select the project/folder where the existing image is placed.
2. Make sure to open the image view of the selected folder, see To view images in a project/folder.
3. Select the image that should be renamed.
4. Press **F2**.
5. In the text field that appears, type a new descriptive name.

Note: Special characters, such as space, are not allowed in element names. Also, if renaming a linked element, only the name of the selected element will change, not the linked one. For more information, see To link an element.

Alternatively, perform the following steps:

1. Select the project/folder where the existing image is placed.
2. Make sure to open the image view of the selected folder, see To view images in a project/folder.
3. Right-click the image that should be renamed.
4. In the menu that appears, select **Rename**.

5. In the text field that appears, type a new descriptive name.

To search image references

1. Select the project/folder where the existing image is placed.
2. Make sure to open the image view of the selected folder, see To view images in a project/folder.
3. Right-click the image that should be used to search for References.
4. In the menu that appears, select **Search Reference** (To search by reference).

To delete an image

Tip: Multiple images can be deleted at once.

1. Select the project/folder where the existing image is placed.
2. Make sure to open the image view of the selected folder, see To view images in a project/folder.
3. Drag the image that should be deleted to the trash can.
4. In the dialog box that opens, confirm the delete operation.

Note: If deleting a linked element, only the link to the project/folder will be removed. The element itself will remain in the database. For more information, see To link an element. Also, referenced elements cannot be deleted. Referenced elements are used by other elements.

Alternatively, perform the following steps:

1. Select the project/folder where the existing image is placed.
2. Make sure to open the image view of the selected folder, see To view images in a project/folder.
3. Right-click the image that should be deleted.
4. In the menu that appears, select **Delete**.
5. In the dialog box that opens, confirm the delete operation.

To view images in a project/folder

1. Select a project/folder in the server tree.
2. Select the **I** tab of the Content Switcher.

Alternatively, perform the following steps:

1. Select a project/folder in the server tree.
2. Select **Image** from the Server Menu.

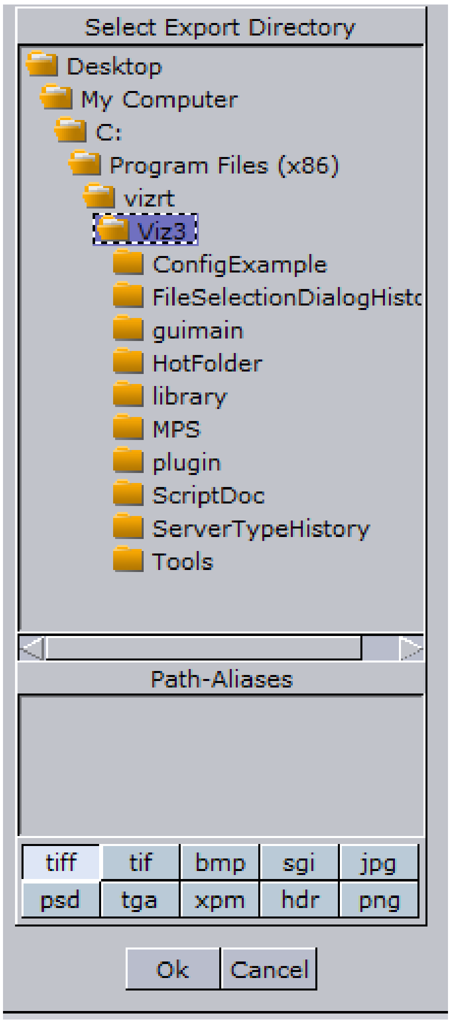
Alternatively, perform the following steps:

1. Select a project/folder in the server tree.
2. Press the keyboard shortcut **CTRL+5** while holding the pointer over the Server area.

To export an image

1. Select the project/folder where the existing image is placed.
2. Make sure to open the image view of the selected folder, see To view images in a project/folder.
3. Right-click the image that should be exported.
4. In the menu that appears, select **Export**.

Figure 24: Select Export Directory dialog box



5. In the Select Export Directory dialog box that opens, select the folder where the image should be exported to.
6. Select the image format.

Tip: The format can be set to; TIFF, BMP, SGI, JPG, PSD, TGA, XPM, HDR or PNG. When exporting an RGB image with alpha, an RGBA image will be exported.

7. Click the **Ok** button.
   The name of the exported image will be equal to the name of the original image.

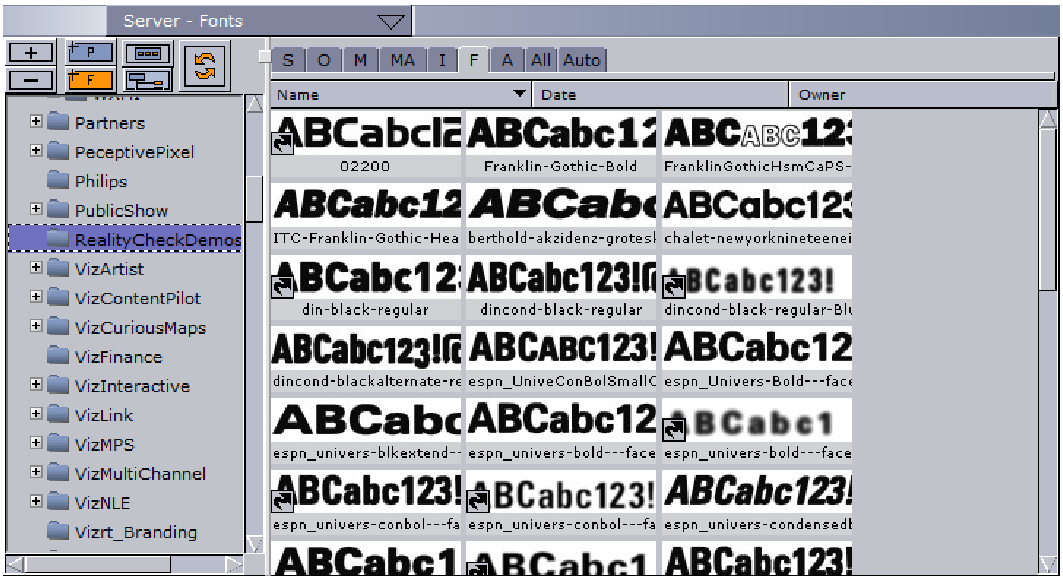
Alternatively, perform the following steps to export one or more images from the database to a hard disk or network share:

1. Select the project/folder where the existing images are placed.
2. Make sure to open the image view of the selected folder, see To view images in a project/folder.
3. Select the images that should be exported from the Element List.
4. Select **Export** from the Server Menu.
5. In the Select Export Directory dialog box that opens, select the folder where the images should be exported to.
6. Select the image format.
7. Click the **Ok** button.

4.4.4 Fonts

Fonts are complete character sets within a unique style. Existing materials are stored in the database and available through the Server area, see To view fonts in a project/folder.

Figure 25: Server area showing all fonts in selected folder



A font can be added to a scene as a standalone container, see Adding Containers. To add a font as a container item (part of another container), drag the font from the Element List onto a the container in the scene tree. For more information, see Manipulating Properties.

This section contains information on the following procedures:

- To import a Font
- To edit a font
- To create a font
- To rename a font
- To search font references
- To delete a font
- To view fonts in a project/folder

To import a Font

- For information about how to import a font, or an archive, from a hard disk or network share to the database, see Importing.

To edit a font

1. Open the Fontstyle Editor, which makes it possible to edit the settings of a font.
2. Select the project/folder where the existing font is placed.
3. Make sure to open the font view of the selected folder, see To view fonts in a project/folder.
4. Double-click the font that should be modified.
5. In the Fontstyle Editor that opens, define the new settings.
6. Click the **Save** button.

To create a font

1. Open the Fontstyle Editor, which makes it possible to create a new font based on an existing font, perform the following steps:
2. Select the project/folder where the existing font is placed.
3. Make sure to open the font view of the selected folder, see To view fonts in a project/folder.
4. Double-click the font that should be modified.
5. In the Fontstyle Editor that opens, define the new settings.
6. Click the **Save As** button.
7. In the Font Save As dialog box that opens, type a descriptive name for the new font in the text box.
8. Click the **Ok** button.

To rename a font

1. Select the project/folder where the existing font is placed.
2. Make sure to open the font view of the selected folder, see To view fonts in a project/folder.
3. Select the font that should be renamed.
4. Press **F2**.
5. In the text field that appears, type a new descriptive name.

Note: Special characters, such as space, are not allowed in element names. Also, if renaming a linked element, only the name of the selected element will change, not the linked one. For more information, see To link an element.

Alternatively, perform the following steps:

1. Select the project/folder where the existing font is placed.
2. Make sure to open the font view of the selected folder, see To view fonts in a project/folder.
3. Right-click the font that should be renamed.
4. In the menu that appears, select **Rename**.
5. In the text field that appears, type a new descriptive name.

To search font references

1. Select the project/folder where the existing font is placed.
2. Make sure to open the font view of the selected folder, see To view fonts in a project/folder.
3. Right-click the font that should be used to search for References.
4. In the menu that appears, select **Search Reference** (To search by reference).

To delete a font

Tip: Multiple fonts can be deleted at once.

1. Select the project/folder where the existing font is placed.
2. Make sure to open the font view of the selected folder, see To view fonts in a project/folder.
3. Drag the font that should be deleted to the trash can.
4. In the dialog box that opens, confirm the delete operation.

Note: If deleting a linked element, only the link to the project/folder will be removed. The element itself will remain in the database. For more information, see To link an element. Also, referenced elements cannot be deleted. Referenced elements are used by other elements.

Alternatively, perform the following steps:

1. Select the project/folder where the existing font is placed.
2. Make sure to open the font view of the selected folder, see To view fonts in a project/folder.
3. Right-click the font that should be deleted.
4. In the menu that appears, select **Delete**.
5. In the dialog box that opens, confirm the delete operation.

To view fonts in a project/folder

1. Select a project/folder in the server tree.
2. Select the **F** tab of the Content Switcher.

Alternatively, perform the following steps:

1. Select a project/folder in the server tree.
2. Select **Font** from the Server Menu.

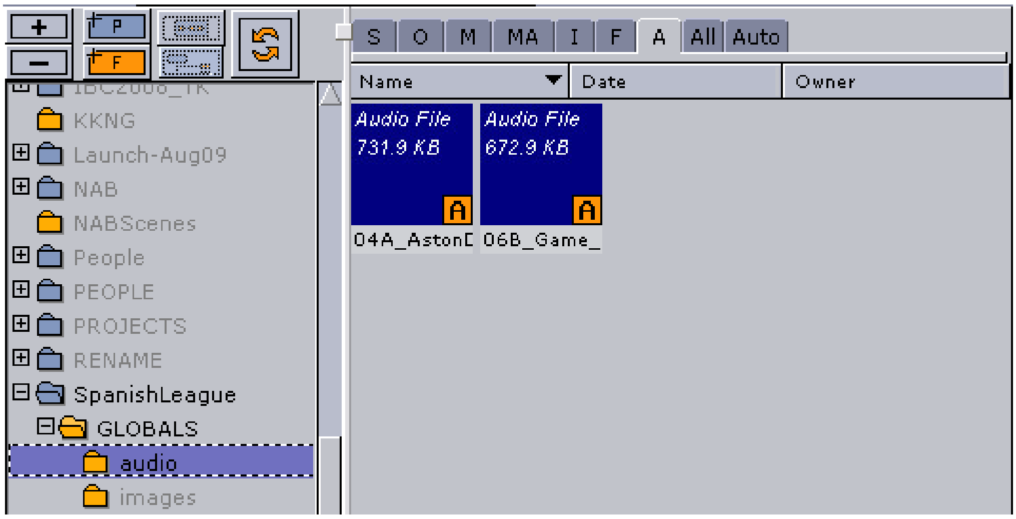
Alternatively, perform the following steps:

1. Select a project/folder in the server tree.
2. Press the keyboard shortcut **CTRL+6** while holding the pointer over the Server area.

4.4.5 Audio Clips

Viz Artist support audio clips in formats WAVE, MP3, and Ogg Vorbis. Audio clips are stored in the database and available through the Server area.

Figure 26: Server area showing all audio clips in selected folder



An audio clip can be added to a scene as a standalone container, see Adding Containers. To add an image as a container item (part of another container), drag the audio clip from the Element List onto the container in the scene tree. Audio clips can be added to the animation of a scene, see Animating Audio.

This section contains information on the following procedures:

- To import an Audio Clip
- To rename an audio clip
- To search audio clip references
- To delete an audio clip
- To view audio clips in a project/folder

To import an Audio Clip

- For information about how to import an audio clip, or an archive, from a hard disk or network share to the database, see Importing.

To rename an audio clip

1. Select the project/folder where the existing audio clip is placed.
2. Make sure to open the audio clip view of the selected folder, see To view audio clips in a project/folder.
3. Select the audio clip that should be renamed.
4. Press F2.
5. In the text field that appears, type a new descriptive name.

Note: Special characters, such as space, are not allowed in element names. Also, if renaming a linked element, only the name of the selected element will change, not the linked one. For more information, see To link an element.

Alternatively, perform the following steps:

1. Select the project/folder where the existing audio clip is placed.

2. Make sure to open the audio clip view of the selected folder, see To view audio clips in a project/folder.
3. Select the audio clip that should be renamed.
4. In the menu that appears, select **Rename**.
5. In the text field that appears, type a new descriptive name.

To search audio clip references

1. Select the project/folder where the existing audio clip is placed.
2. Make sure to open the audio clip view of the selected folder, see To view audio clips in a project/folder.
3. Right-click the audio clip that should be used to search for References.
4. In the menu that appears, select **Search Reference** (To search by reference).

To delete an audio clip

Tip: Multiple audio clips can be deleted at once.

1. Select the project/folder where the existing audio clip is placed.
2. Make sure to open the audio clip view of the selected folder, see To view audio clips in a project/folder.
3. Drag the audio clip that should be deleted to the trash can.
4. In the dialog box that opens, confirm the delete operation.

Note: If deleting a linked element, only the link to the project/folder will be removed. The element itself will remain in the database. For more information, see To link an element. Also, referenced elements cannot be deleted. Referenced elements are used by other elements.

Alternatively, perform the following steps:

1. Select the project/folder where the existing audio clip is placed.
2. Make sure to open the audio clip view of the selected folder, see To view audio clips in a project/folder.
3. Right-click the audio clip that should be deleted.
4. In the menu that appears, select **Delete**.
5. In the dialog box that opens, confirm the delete operation.

To view audio clips in a project/folder

1. Select a project/folder in the server tree.
2. Select the **A** tab of the Content Switcher.

Alternatively, perform the following steps:

1. Select a project/folder in the server tree.
2. Select **Audio** from the Server Menu.

Alternatively, perform the following steps:

1. Select a project/folder in the server tree.

2. Press the keyboard shortcut **CTRL+7** while holding the pointer over the Server area.

4.5 Working with Elements

This section contains information on the following procedures:

- To copy an element
- To create an independent copy of an element
- To move an element
- To search an element
- To create a new keyword for an element
- To assign the keyword to an element

To copy an element

To create an independent copy of an element in another project/folder, perform the following steps:

1. Select the project/folder where the source element is placed.
2. Drag the element to the target project/folder while pressing **CTRL**.

Alternatively, perform the following steps:

1. Select the project/folder where the source element is placed.
2. Select the source element.
3. Select **Copy** from the Server Menu.
4. Select the project/folder where the target element should be placed.
5. Select **Paste-Dup** from the Server Menu.

To create an independent copy of an element

1. Select the project/folder where the source element is placed.
2. Right-click the source element.
3. In the menu that appears, select **Duplicate**.
4. In the dialog box that opens, type a descriptive name for the new element.
5. Click the **Ok** button.
   The new copy will be in the same project/folder as the original element.

To move an element

- Drag the element from the Element List onto the target project/folder in the server tree.

Alternatively, perform the following steps:

1. Select the project/folder where the source element is placed.
2. Select the source element.
3. Select **Cut** from the Server Menu.
4. Select the project/folder where the target element should be placed.
5. Select **Paste-Dup** from the Server Menu.

To search an element

- Select **Search** from the Server Menu.

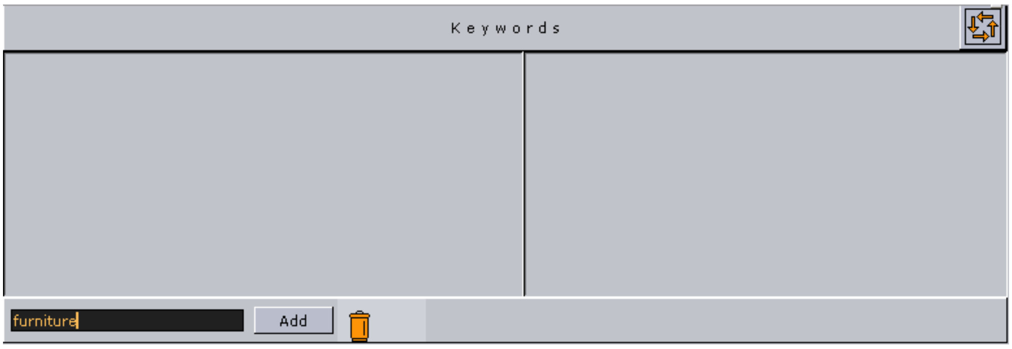
Alternatively, perform the following step:

- Press the keyboard shortcut **CTRL+F** while holding the pointer over the Server area.

To create a new keyword for an element

1. Select **Import** from the Main menu.

Figure 27: Keywords section in Import panel

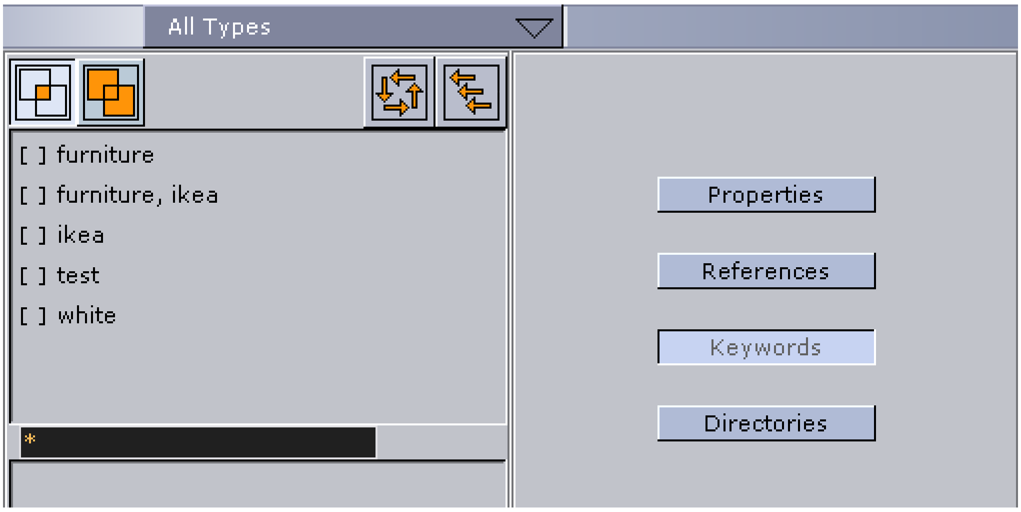


2. In the Import panel that appears, type a descriptive keyword in the text box.
3. Click the **Add** button.

To assign the keyword to an element

1. Select **Server** from the Main menu.
2. Press the keyboard shortcut **CTRL+F** while holding the pointer over the Server area.
3. In the Search window that opens, enable the Keywords search.
4. Type *** in the text box, and then press **ENTER**.

Figure 28: Search window showing all keywords in the database



5. Drag one or more keywords from the Search window to an element in the Element List.

Tip: Keywords can be assigned to multiple elements at once. To do so, first select the elements in the Element List, and then select the keywords in the Search window. Next, drag the keywords onto the elements. Finally, in the dialog box that opens, confirm the assign operation.

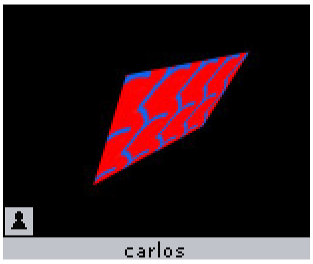
See Also

- Assigning Keywords
- Searching
- To link an element

4.5.1 Active Elements

Active elements that a user currently work on can be locked or checked out.

Figure 29: Element with locked icon



An active element has a small icon attached to the lower left corner of the thumbnail in the Element List.

- **Locked**: For a locked element, the icon represents a keyhole. If a user connected to the database opens an element, the element will be locked for all other users, marked with the keyhole icon. A locked element means that the element can be opened and viewed by all users, but only modified or saved by the user that has locked it. The element will be unlocked when the user closes the element or disconnects from the database. To lock an element.

- **Checked Out**: For a checked out element, the icon represents a stop sign. If a user connected to the database checks out an element, the element will be locked for all other users, marked with the stop icon. A checked out element means that the element can be opened and viewed by all users, but only modified and saved by the user that has checked it out. To cancel the check out, the element must be checked in. To check out an element and To check in an element.

Note: The element will be checked out even if the user closes the element or disconnects from the database.

- **Locked and Checked Out**: For a locked and checked out element, the icon represents a combined keyhole/stop sign. To lock and check out an element.

This section contains information on the following procedures:

- To lock an element
- To check out an element
- To check out one or more elements at once
- To check in an element
- To check in or more elements at once
- To lock and check out an element

To lock an element

- An element is automatically locked when a user opens the element. It is not possible to perform this operation manually.

To check out an element

1. Select the project/folder where the element is placed.
2. Right-click the element.
3. In the menu that appears, select **Check Out**.

Alternatively, perform the following steps:

1. Select the project/folder where the element is placed.
2. Select the element.
3. Press the keyboard shortcut **CTRL+O** while holding the pointer over the Server area.

To check out one or more elements at once

1. Select the project/folder where the elements are placed.
2. Select one or more elements.
3. Select **Check Out Files** from the Server Menu.

Alternatively, perform the following steps:

1. Select the project/folder where the elements are placed.
2. Select the elements.
3. Press the keyboard shortcut **CTRL+O** while holding the pointer over the Server area.

To check in an element

1. Select the project/folder where the element is placed.
2. Right-click the element.
3. In the menu that appears, select **Check In**.

Alternatively, perform the following steps:

1. Select the project/folder where the element is placed.
2. Select the element.
3. Press the keyboard shortcut **CTRL+I** while holding the pointer over the Server area.

To check in or more elements at once

1. Select the project/folder where the elements are placed.
2. Select one or more elements.
3. Select **Check In Files** from the Server Menu.

Alternatively, perform the following steps:

1. Select the project/folder where the elements are placed.
2. Select the elements.
3. Press the keyboard shortcut **CTRL+I** while holding the pointer over the Server area.

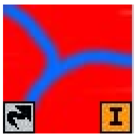
To lock and check out an element

- An element is automatically locked when a user opens the element. It is not possible to perform this operation manually. See To check out an element.

4.5.2 Linked Elements

Every element in the database is displayed in one or more projects/folders.

Figure 30: Element with linked icon



A linked element has a small link icon attached to the lower left corner of the thumbnail in the Element List.

Through the use of linking, an element can appear logically in multiple projects/folders, although the element is physically saved in the database only once. By opening the Properties Pane and selecting the File Info tab, the UUID of the element can be located. This identifier is equal for all the linked elements, while information such as path, name, and creation date may vary. If modifying the element in one of the projects/folders, for example cropping an image, all the linked items will reflect these changes.

To link an element

1. Select the project/folder where the source element is placed.
2. Drag the element to the target project/folder while pressing **ALT**.

Alternatively, perform the following steps:

1. Select the project/folder where the source element is placed.
2. Select the source element.
3. Select **Copy** from the Server Menu.
4. Select the project/folder where the target element should be placed.
5. Select **Paste-Link** from the Server Menu.

4.6 Editing Elements in the Database

Editors make it possible to modify elements in the database. Individual editors are available for materials, images, and fonts.

To open an element in one of the individual editors, double-click the element in the Element List.

Note: An element that is opened in an editor, is locked in the database. For more information, see Active Elements.

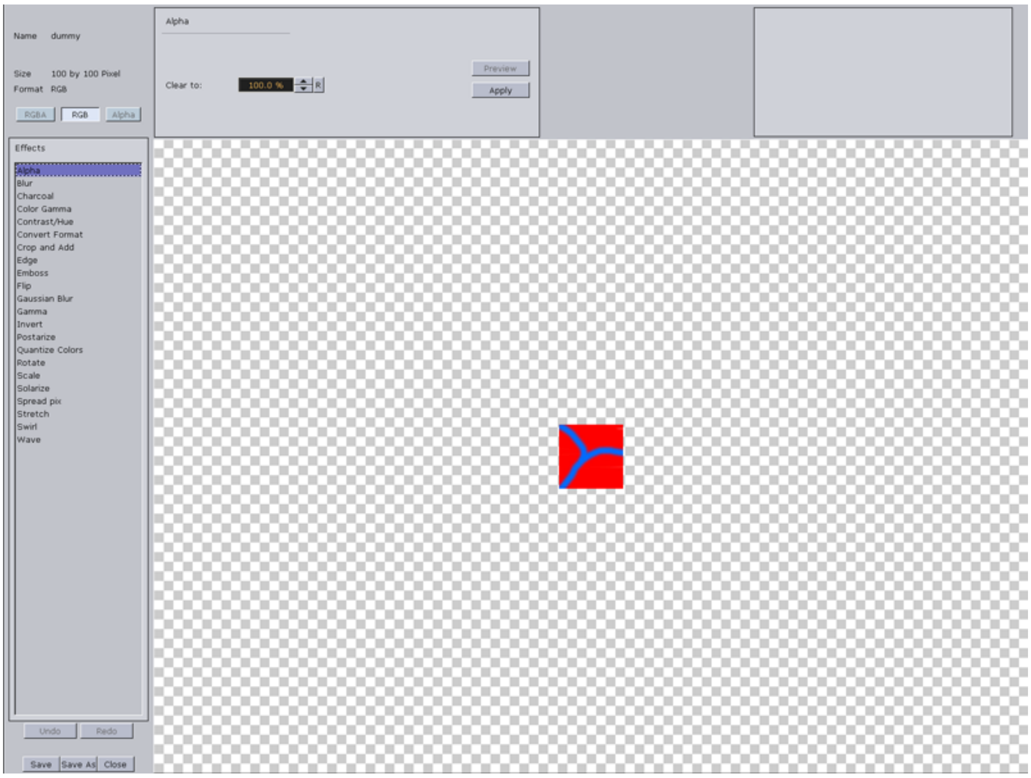
This section contains information on the following topics:

- Image Editor
- Fontstyle Editor

4.6.1 Image Editor

The Image editor is used to modify Images.

Figure 31: Image editor



Name dummy

Size 100 by 100 Pixel

Format RGB

Alpha

Clear to: 100.0 %

The modifications that can be performed in the Image editor are quite similar to the ones in other image editing software.

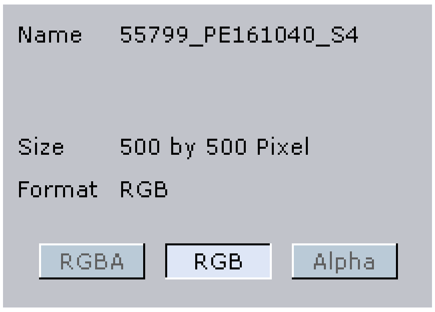
To open an element in the Image editor, double-click the element in the Element List. For more information, see To edit an image.

This section contains information on the following topics:

- General Image Information
- Effects
- Previewing and Applying Effects

General Image Information

Figure 32: General image information in Image editor



Name 55799_PE161040_S4

Size 500 by 500 Pixel

Format RGB

The upper left corner of the Image editor shows general information about the image:

- **Name**: Shows the image name.
- **Size**: Shows the image size in pixels.
- **Format**: Shows the image format (RGBA, RGB, or Alpha).

By clicking the format buttons, the image can be viewed in various formats, if available.

Note: All images are treated as RGB or alpha format. An imported image that has alpha can be viewed as either RGB or alpha. Modifications to an image are automatically applied to the RGB or alpha information.

Effects

The following effects can be applied to an image:

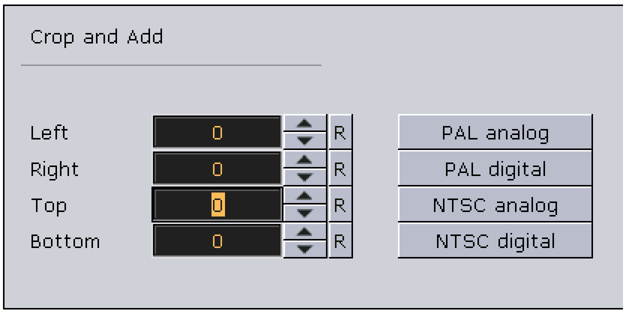
- **Alpha**: Makes the image transparent.

Note: This option only applies for Alpha format images.

- **Blur**: Makes the image blurry.
- **Charcoal**: Creates a posterized, smudged effect by redrawing the image. Major edges are boldly drawn, while mid tones are sketched using a diagonal stroke.
- **Color Gamma**: Adjusts the color tones of the image.
- **Contrast/Hue**: Adjusts the brightness, saturation, and hue settings for the image.
- **Convert Format**: Changes the format of the image to one of the following:
  - **RGB**
  - **RGBA**
  - **Alpha**
  - **Luminance**
  - **Luminance Alpha**

Note: The current format is grayed out.

Figure 33: Crop and add effects in Image editor



Crop and Add

Left 0

Right 0

Top 0

Bottom 0

- **Crop and Add**: The image can be cropped according to the values in the input boxes. The values can be entered manually, or the standard TV format buttons can be used.
  - **Left**: Crops the image the specified number of pixels from the left side of the image border.

    - **Right**: Crops the image the specified number of pixels from the right side of the image border.
    - **Top**: Crops the image the specified number of pixels from the top of the image border.
    - **Bottom**: Crops the image the specified number of pixels from the bottom of the image border.
    - **PAL Analog**: Crops the image to a resolution of 768x576 pixels.
    - **PAL Digital**: Crops or adds the image to a resolution of 720x576 pixels.
    - **NTSC Analog**: Crops or adds the image to a resolution of 640x486 pixels.
    - **NTSC Digital**: Crops or adds the image to a resolution of 720x486 pixels.
- **Emboss**: Makes the color at a given location of the filtered image correspond to the rate of color change at that location in the original image.
- **Flip**:
    - **Flip Vertical**: Flips the image upside down. To flip it back to its original settings, click the button again.
    - **Flip Horizontal**: Flips the image horizontally as if seen in reverse/through a mirror. To flip it back to its original settings, click the button again.
- **Gaussian Blur**: The visual effect of this blurring technique is a smooth blur resembling that of viewing the image through a translucent screen. It adds low frequency detail and can produce a hazy effect to the image.
- **Gamma**: Gamma measures the brightness of mid tone values produced by a device (often a monitor). A higher gamma value yields an overall darker image.
- **Invert**: Inverts the colors of an image. When inverting an image, the brightness value of each pixel in the channels is converted to the inverse value on the 256-step color values scale.

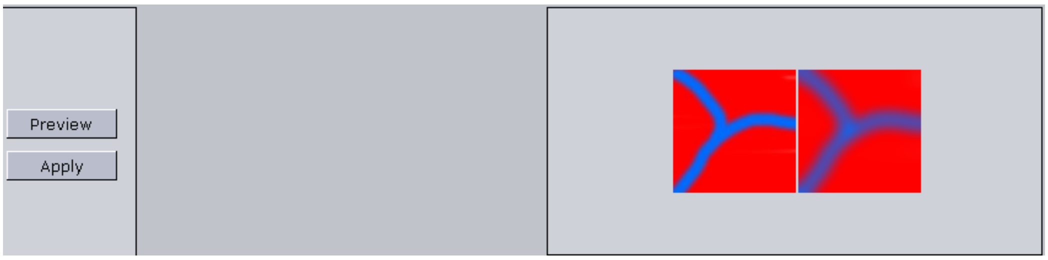
Example: A pixel in a positive image with a value of 255 is changed to 0, and a pixel with a value of 5 to 250.

- **Quantize Colors**: Reduces the number of colors in the image.
- **Rotate**: Rotates the image according to the specified number of degrees.
- **Scale**: Scales the image based on either size (percent) or width/height (pixels).
- **Solarize**: Blends a negative and positive image, similar to exposing a photographic print briefly to light during production.
- **Stretch**: Stretches the image based on either width/height in pixels or percent.
- **Swirl**: Swirls the image more sharply in the center than at the edges. Specifying the degree produces a swirl pattern.
- **Wave**: Creates an undulating pattern on the image, like ripples on the surface of a pond.
    - **Amplitude**: Defines the height of the waves.
    - **Wavelength**: Defines the width of the waves, the distance from one wave crest to the next.

Previewing and Applying Effects

Various effects can be selected from the Effects list at the left side of the Image editor. Selecting an effect in the list, makes it possible to define various settings in the section at the top of the Image editor.

Figure 34: Preview and Apply buttons in Image editor



Two buttons apply for all effects:

- **Preview**: Shows a preview of the changes in the upper right corner of the Image editor.
- **Apply**: Applies the changes to the image.

4.6.2 Fontstyle Editor

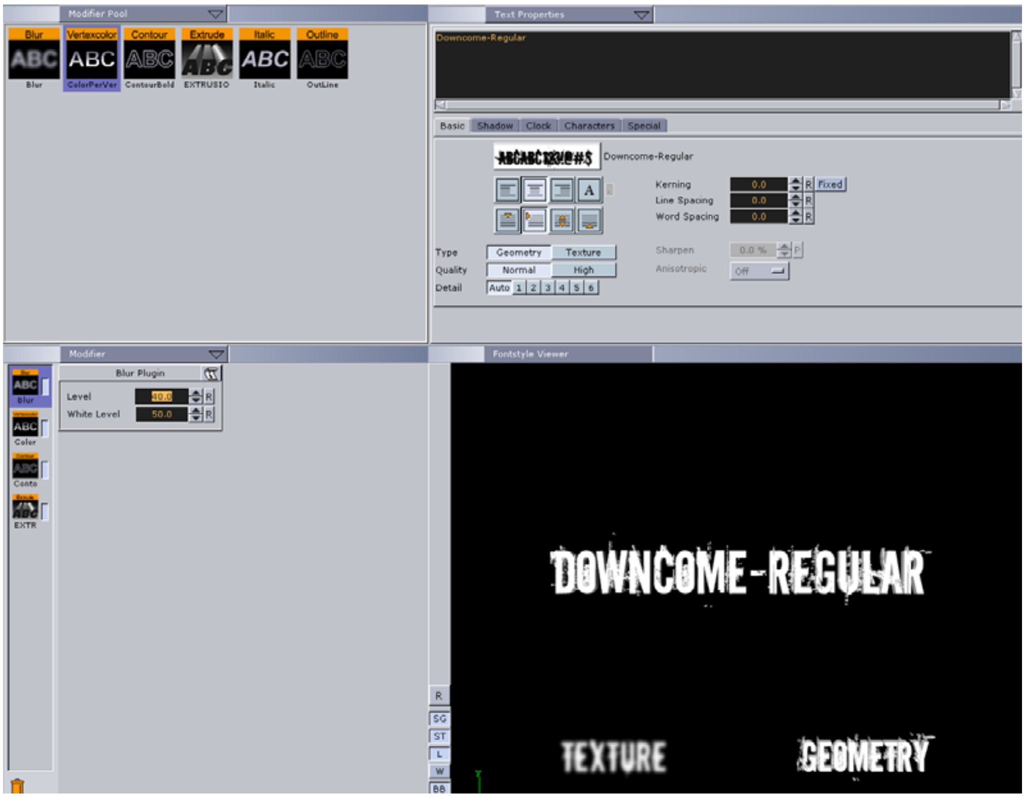
The Fontstyle editor is used to modify Fonts.

Viz Artist uses fonts to display text in scenes. When a font is imported into the database, Viz Artist imports the base font and creates up to eight different fontstyles; one fontstyle looking like the base font, up to four fontstyles in different blur levels, and up to three fontstyles using only the outline of the base font.

Tip: The number of blur levels and outlines for each font can be defined from the Font Options section of the Configuration.

The Fontstyle editor makes it possible to change the imported fontstyles through the use of Modifiers. For example, a fontstyle can be altered to reflect a corporate identity. As Viz Artist uses an open interface for the plug-ins used as modifiers, almost any changes can be applied to a fontstyle. New fontstyles can be created based on existing fontstyles through the Save As operation.

Figure 35: Fontstyle editor



Downcome-Regular

Kerning 0.0

Line Spacing 0.0

Word Spacing 0.0

Sharpen 0.0 %

Anisotropic Off

Level 40.0

White Level 50.0

The Fontstyle editor consists of two areas. The left side shows the available modifiers (top) and the values for the modifiers (bottom). The right side shows a preview of the modifications (bottom) and the properties of previewed text (top).

Note: The properties help to illustrate how the modifications will look like in a scene. Changes made on the right side of the editor, for example kerning and material, will not be saved with the font. However, there is one exception from this rule. If the kerning between two fixed characters are modified, this is saved with the font. To create a kerning between two fixed characters, use the keyboard shortcuts **ALT+LEFT/RIGHT ARROW**.

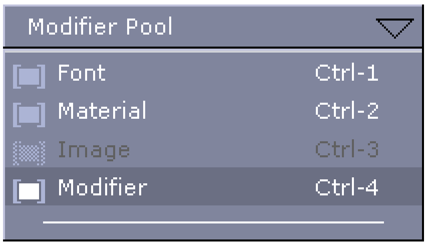
This section contains information on the following topics:

- Database Menu
- Modifier Menu
- Modifiers
- Preview Menu
- Fontstyle Viewer
- Fontstyle Viewer Menu
- Fontstyle Viewer Bar
- To open a font for editing

Database Menu

The Database menu is positioned at the top left of the Fontstyle editor.

Figure 36: Database menu in Fontstyle editor

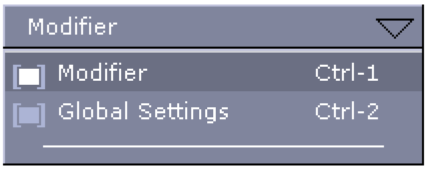


- **Font**: Shows all the fonts in the database.
- **Material**: Shows all the materials in the database.
- **Image**: Shows all the images in the database.
- **Modifier**: Shows all available modifiers that can be applied to the font.

Modifier Menu

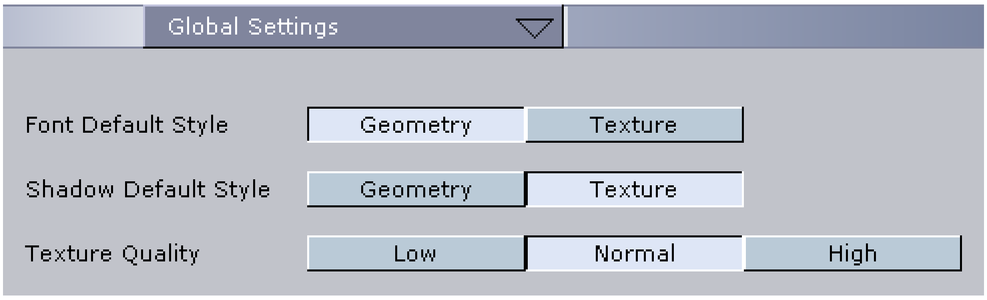
The Modifier menu is positioned in the middle left part of the Fontstyle editor.

Figure 37: Modifier menu in Fontstyle editor



- **Modifier**: Shows the properties for the modifier that is selected.

Figure 38: Global settings in Fontstyle editor



- **Global Settings**: Shows various properties for the font.
  - **Font Default Style**: Defines the font default style, can be set to Geometry or Texture.
  - **Shadow Default Style**: Defines the shadow default style, can be set to Geometry or Texture.
  - **Texture Quality**: Defines the quality of the texture, can be set to Low, Normal, or High.

Note: The texture quality cannot be changed in a text editor.

Modifiers

Modifiers are used to change either the texture or the geometry of a fontstyle, or both. A fixed set of modifiers come with a standard Viz Artist installation, but the system provides an open connection, so that additional modifiers can be obtained or created and added as plug-ins.

A modifier that is applied to a fontstyle is automatically enabled. An enabled modifier implies that modifications will be reflected in the preview area. When adding multiple modifiers, the overview in the preview area can get confusing, and it may be necessary to disable one or more modifiers.

To open a font for editing

- Simply double-click the font from the server view to open the font in the Fontstyle editor.

4.6.3 Working with Modifiers

This section contains information on the following topics:

- To add a modifier to a font
- To edit a modifier
- To enable/disable a modifier
- To remove a modifier from a font
- Blur
- Color Per Vertex
- Contour Bold
- Extrude
- Italic
- Outline
- Preview Menu
- Fontstyle Viewer
- Fontstyle Viewer Menu
- Fontstyle Viewer Bar

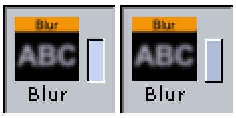
To add a modifier to a font

1. Select **Modifier** from the Database Menu.
2. Drag the modifier from the list of available modifiers (top left) to the Modifier bar (bottom left, above the trash can).
3. Click the **Save** button.

To edit a modifier

1. Select the modifier in the Modifier bar (bottom left, above the trash can).
2. Define the values for the modifier.
3. Click the **Save** button.

To enable/disable a modifier



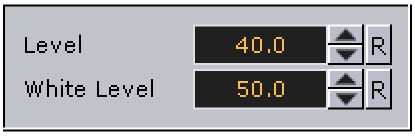
- Click the property switch next to the modifier icon in the Modifier bar.

To remove a modifier from a font

1. Drag the modifier from the Modifier bar to the trash can.
2. Click the **Save** button.

Blur

Use the Blur modifier to blur a font in texture mode.



Level 40.0

White Level 50.0

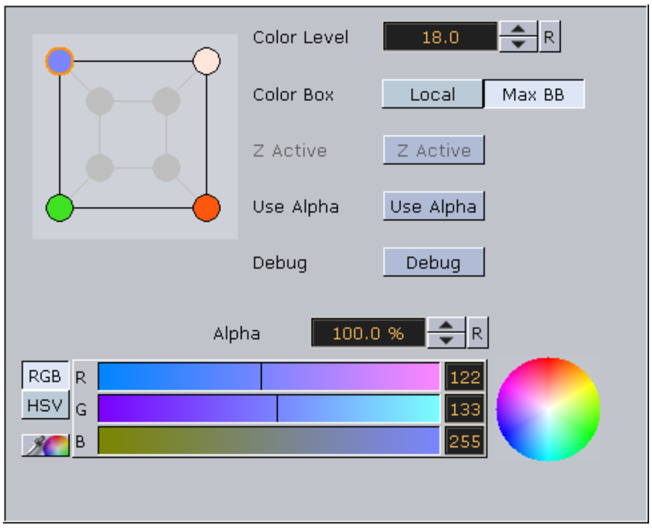
- **Level**: Sets the blur level.
- **White Level**: Sets the white level in the blur effect.

Color Per Vertex

Use the Color Per Vertex modifier to change the color of fonts by applying it to the vertices.

By clicking a vertex, it is possible to edit its color values. If clicking one of the lines that connect the vertices, the vertices on both ends of the line will be selected. Also, multiple vertices can be selected by clicking them while holding CTRL. Colors can be dragged from one vertex to another.

Figure 39: Color Per Vertex editor



Color Level 18.0 R

Color Box Local Max BB

Z Active Z Active

Use Alpha Use Alpha

Debug Debug

Alpha 100.0 % R

RGB R 122

HSV G 133

B 255

- **Color Level**: Defines how much color should be applied.
- **Color Box**:
  - **Local**: Applies the color to the vertices of every single character.
  - **Max BB**: Applies the color to the vertices of the bounding box.
- **Z Active**: If the Extrude modifier is used in addition to Color Per Vertex, the vertices can be changed both in the front and back. If no such modifier is applied, only the vertices in the front can be edited, as there are no back vertices.

Contour Bold

Use the Contour Bold modifier to change the bold level of fonts, which makes it appear thicker or thinner.

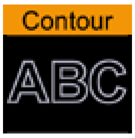
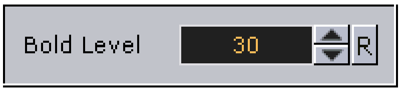


Figure 40: Contour Bold editor



Bold Level 30 R

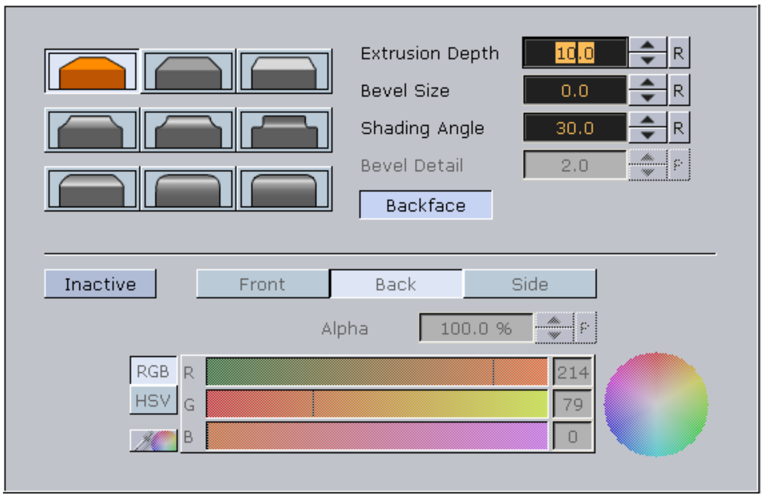
- **Bold Level**: Defines the bold level of a font. The higher the value, the bolder the font.

Extrude

Use the Extrude modifier to extrude a 2D font in the Z-axis, resulting in a 3D font with front, back, and sides.

An extrusion can be explained simply as sweeping the 2D font through space along its Z-axis. The sweeping path the outline of the font follows during this process is used to create a surface.

Figure 41: Extrude editor



Extrusion Depth 10.0
Bevel Size 0.0
Shading Angle 30.0
Bevel Detail 2.0
Alpha 100.0 %
R 214
G 79
B 0

The nine icons marked with different bevel types are used to set the type of bevel on the fontstyle.

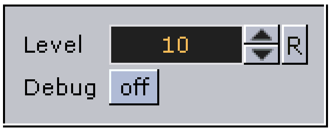
- **Extrusion Depth**: Defines the depth of the extrusion along the Z-axis.
- **Bevel Size**: Defines the size of the bevel.
- **Shading Angle**: Defines the angle of the shading function. To remove unevenness, increase the shading angle.
- **Bevel Detail**: Defines the degree of bevel detail. The lower the value, the more detailed the bevel is constructed and the smaller tilings it is constructed from.

Note: The Bevel Detail option is only available for certain bevel types.

- **Backface**: Enables or disables visualization of the backface.
- **Active/Inactive**: If colors are set to Active, the front, back, and side colors of the font can be applied.

Italic

Use the Italic modifier to create a cursive appearance of a font.

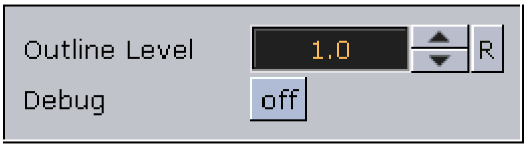

Level 10
Debug off

- **Level**: Defines the cursive level of a font. The higher the value, the more tilted is the text.

Outline

Use the Outline modifier to create an outline of a font.

Figure 42: Outline editor

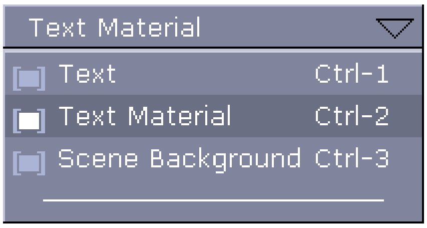

Outline Level 1.0
Debug off

- **Outline Level**: Defines the width of the outline level. The higher the value, the bolder the outline of a font.

Preview Menu

The Preview menu is positioned at the top right of the Fontstyle editor.

Figure 43: Preview menu in Fontstyle editor



Note: Only the applied modifiers will be saved with the font, not the settings described here. These settings are only used to give an overview of how the font will look like when being used.

- **Text Properties**: To preview how the font will look like with different settings, for example kerning, orientation, and content, select Text Properties from the Preview menu and modify the properties. For more information, see Text Editor.

- **Text Material**: To preview how a material would look like if used on the font, select Text Material from the Preview menu. In the Material editor that appears, select a color. For more information, see Working with Materials.

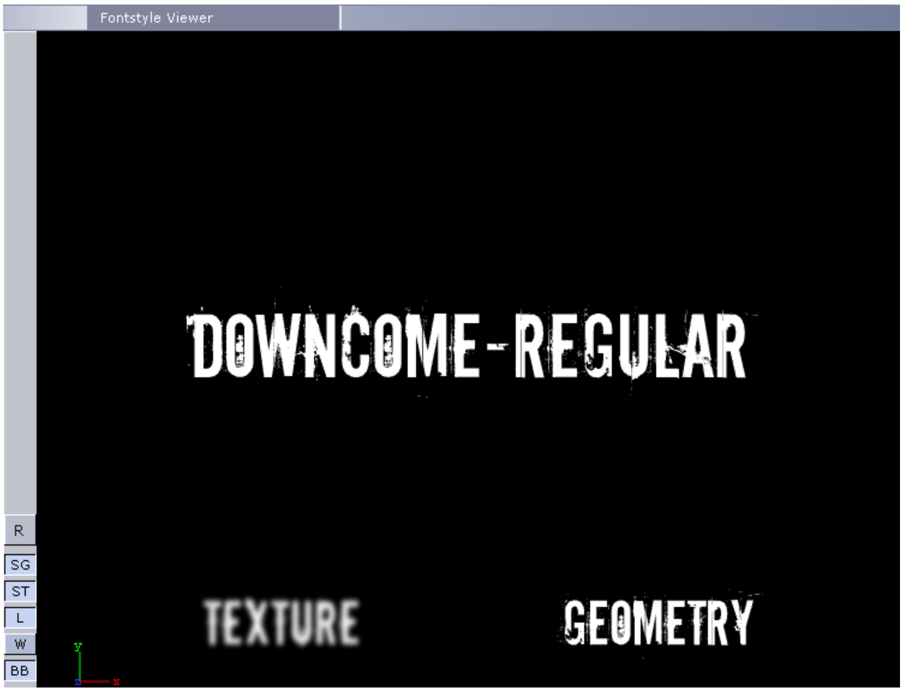
Tip: To use a material already saved in the database, select Material from the Database Menu, and then simply drag the preferred material to the color icon in the material editor of the font.

- **Scene Background**: To preview how a background color would look like if used together with the font, select Scene Background from the Preview menu, and then select a color. The scene background feature provides a better contrast, especially for drop shadows.

Fontstyle Viewer

To visualize how the fontstyle will look like when being used in scenes, the Fontstyle viewer in the lower right part of the Fontstyle editor shows a preview of the font.

Figure 44: Fontstyle viewer in Fontstyle editor



Additionally to the customizable text in the Fontstyle viewer, the two words Texture and Geometry will be displayed. These words show the modifications performed to Texture and/or Geometry applied by the modifiers, without reflecting the changes in the properties. For example, if kerning is applied to the customizable text, this will not be reflected in the Texture or Geometry texts.

The customizable text reflects either the Texture or Geometry text type, which is selected in the Global Settings, available from the Modifier Menu.

Fontstyle Viewer Menu

Right-clicking the Fontstyle Viewer opens a menu, similar to the menu in the render output, see Working with the Scene Editor.

Fontstyle Viewer Bar

At the left side of the Fontstyle Viewer is the Fontstyle Viewer bar. The bar contains the following icons:

- **R**: Resets the position, rotation, and scaling for the three elements in the Fontstyle viewer.
- **SG**: Shows/hides font with geometry style.
- **ST**: Shows/hides font with texture style.
- **L**: Enables/disables additional light.
- **W**: Switches on/off wire frame mode.
- **BB**: Switches on/off the bounding box visualization.

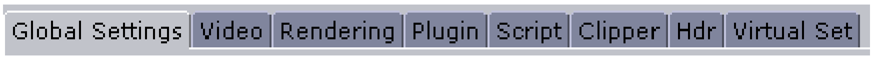
5 Global Tasks

This section contains information on the following topics:

- Scene Settings
- Implementing Video
- Searching
- Importing
- Archiving
- Post Rendering

5.1 Scene Settings

To open the Scene Settings editor, select Scene Settings from the Scene Setup menu. The Scene Settings editor makes it possible to modify the global parameters for the scene.



There are various types of scene settings. Click the tabs at the top of the Scene Settings editor to open the respective panels.

This section also contains information on the following topics:

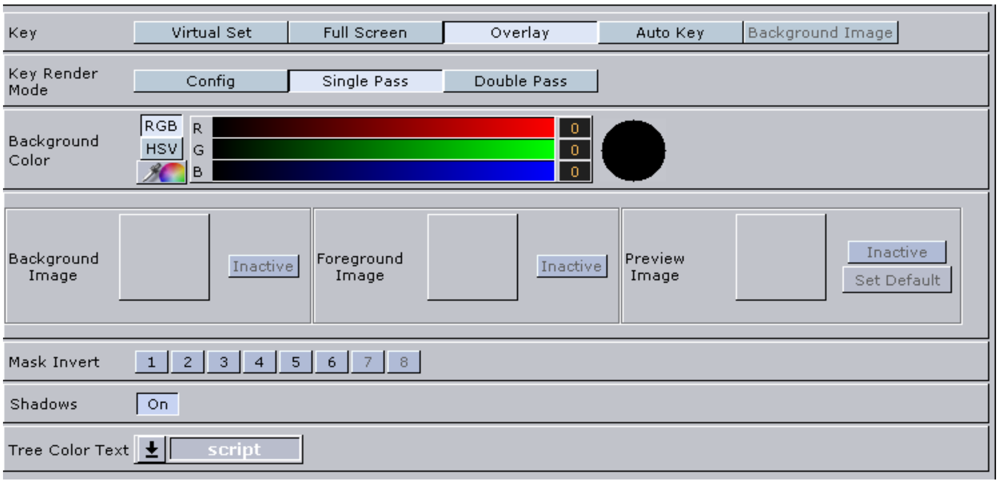
- Global Settings
- Rendering
- Plug-in
- Script
- Clipper
- Virtual Set

See Also

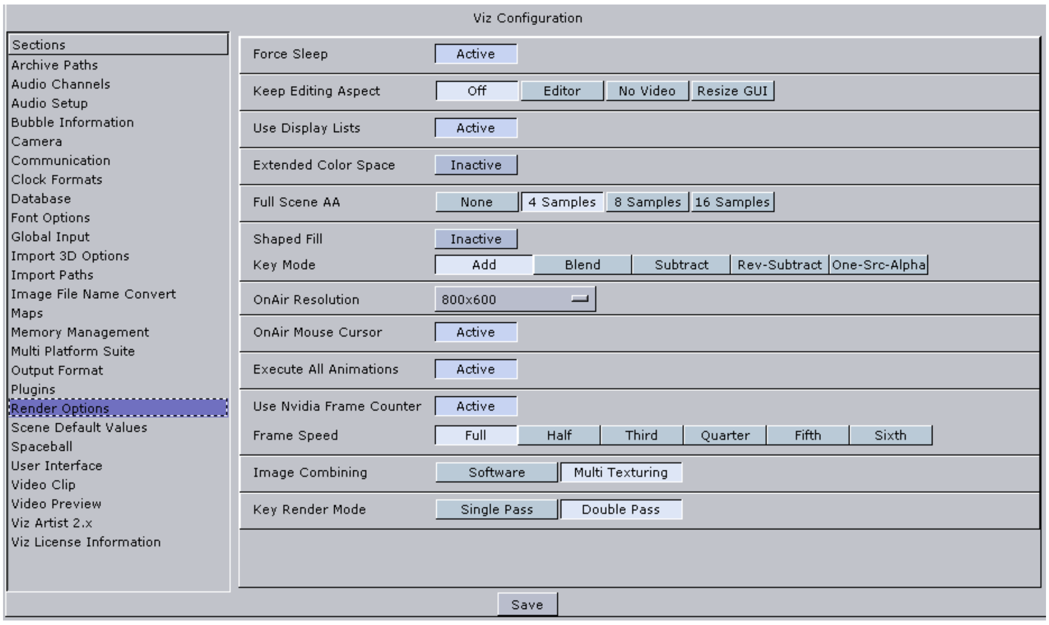
- Implementing Video

5.1.1 Global Settings

In the Global Settings panel, basic parameters for the scene can be modified.



- **Key**: Defines how the alpha key signal should be generated. The default is Overlay.
  - **Virtual Set**: This option is relevant only in combination with the Virtual Set solution. This mode switches from Overlay to Full Screen when the control mode of the current camera switches from Remote to Editor (see Camera Editor). On a Virtual Set, to perform a camera flight with the virtual camera when the real camera is stationary, the graphics must be switched to Full Screen to avoid a situation whereby the virtual and the real camera have different position values. When switching the camera from Remote to Editor to perform the flight, the keyed graphics automatically switches from Overlay to Full Screen.
  - **Full Screen**: Ouputs non-transparent key signal.
  - **Overlay**: Sets as key signal only those containers that have a Key function attached. Only containers in the scene tree with a Key function attached are rendered into the key signal.
  - **Auto Key**: This is the preferred mode for overlay graphics. The key signal is automatically generated from the containers in the scene tree and it does not require the application of Key functions to the container. This function provides you with a worry-free overlay.
  - **Background Image**: This option is only applicable if a background image with an alpha layer is added. The alpha values of the background image are used as a key signal and containers with a Key function attached will be rendered on top.
  - **Config**: Renders the key according to the settings in the Config.



Force Sleep: Active

Keep Editing Aspect: Off | Editor | No Video | Resize GUI

Use Display Lists: Active

Extended Color Space: Inactive

Full Scene AA: None | 4 Samples | 8 Samples | 16 Samples

Shaped Fill: Inactive

Key Mode: Add | Blend | Subtract | Rev-Subtract | One-Src-Alpha

OnAir Resolution: 800x600

OnAir Mouse Cursor: Active

Execute All Animations: Active

Use Nvidia Frame Counter: Active

Frame Speed: Full | Half | Third | Quarter | Fifth | Sixth

Image Combining: Software | Multi Texturing

Key Render Mode: Single Pass | Double Pass

    - **Single Pass** Uses one rendering step. Use this for all new scenes and for scenes imported from version 2.8, as it is much faster.
    - **Double Pass** Uses two rendering steps. Use for backward compatibility in scenes created in Viz Artist versions 3.0 to 3.3.
- **Color** Sets the color of the background. For information about how to select a color, see Changing Color.
- **Background Image** Makes it possible to use an image as the scene background.
- **Foreground Image** Makes it possible to use an image in the scene foreground, which means right behind the camera pane. If the image in the foreground has no alpha, then only the image will be displayed. If the image has alpha, the scene will be displayed through the image.
- **Preview Image** Makes it possible to add a fixed layout to the scene. This visualizes how the total preview (camera picture and/or graphics) will look like. To set the image as global default, click the Set Default button.

Tip: It is possible to drag results from a database search onto the background, foreground, and preview image drop zones.

- **Mask Invert** Makes it possible to set one or more of the mask layers in the scene to be inverted. With no mask inverted, the mask cuts out a hole where it covers the mask target object. With mask invert enabled, the mask target is only visible where the mask covers it.

Figure 45: Mask Not Inverted

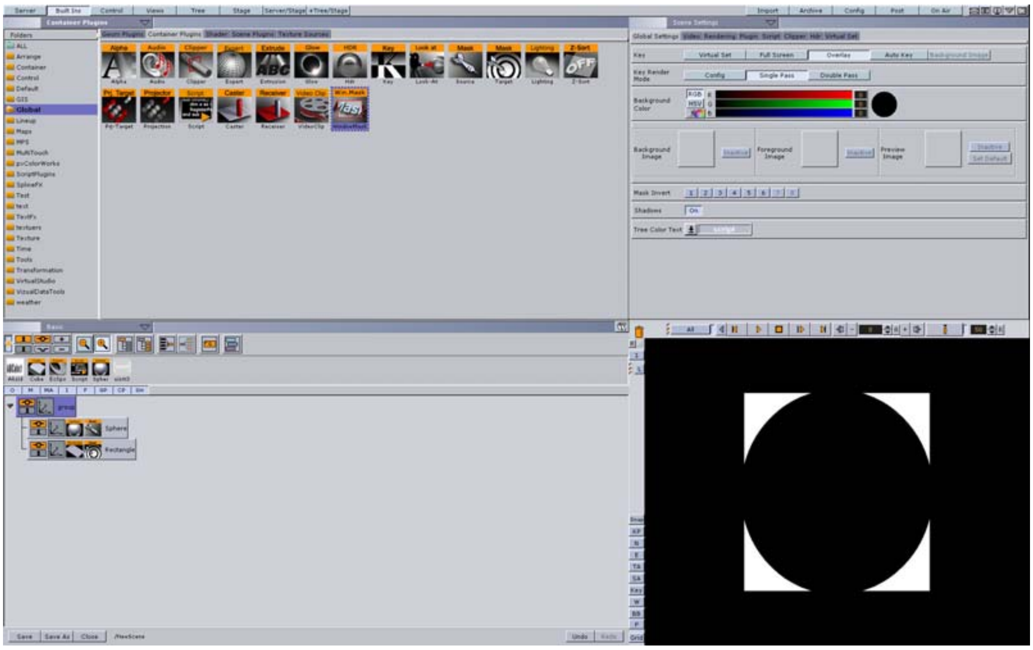
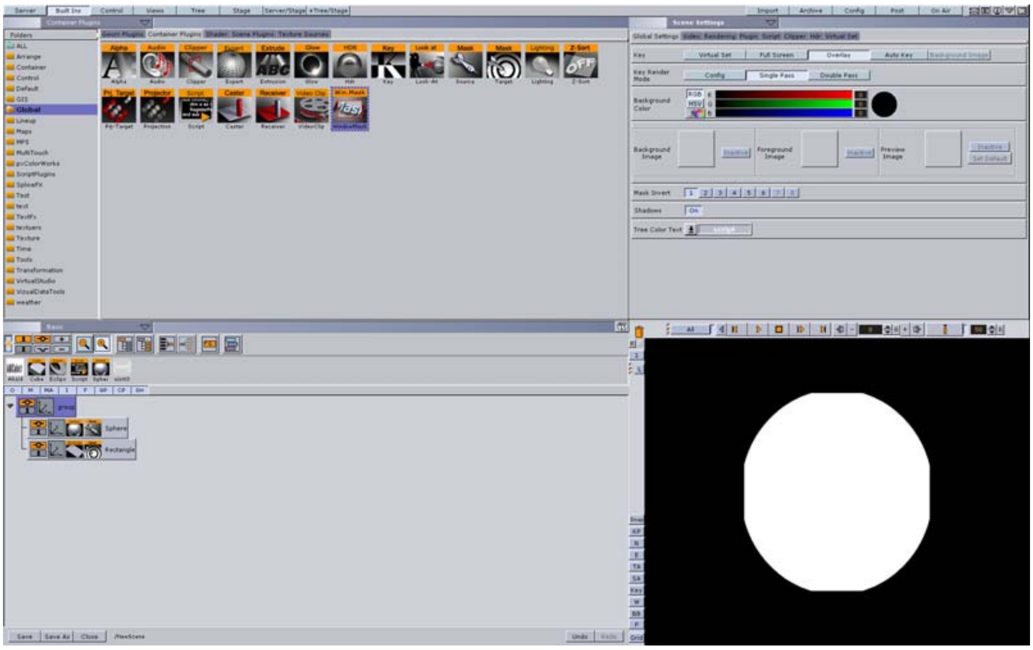


Figure 46: Mask Inverted



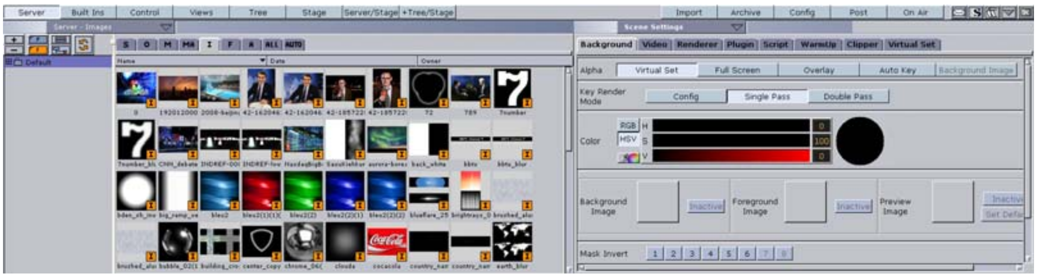
- **Shadows** Enables or disables the shadow function in the scene. The shadow function uses two layers from the mask function. With shadows enabled, there are only six layers available for the mask function (see Mask Source and Mask Target). In a scene where more than six mask layers are needed, shadows can be disabled. Note that only very complex mask configurations need more than six layers.
- **Tree Color Text** Makes it possible to add text to the scene tree colors that are available in the Favorites area. Note that this text is scene specific. Adding text to the global scene tree Colors can be done in the Configuration. To create customized colors for the scene tree, see Display Area.

Working with Images

The following procedures demonstrate what you can do with background, foreground and preview images.

To add an image

1. Click **Server**.
2. Click the **I** tab for Images.
3. Drag the desired image onto the drop zone.



A thumbnail of the image will then be shown in the drop zone button. Dragging a new image onto the drop zone will automatically replace the old image.

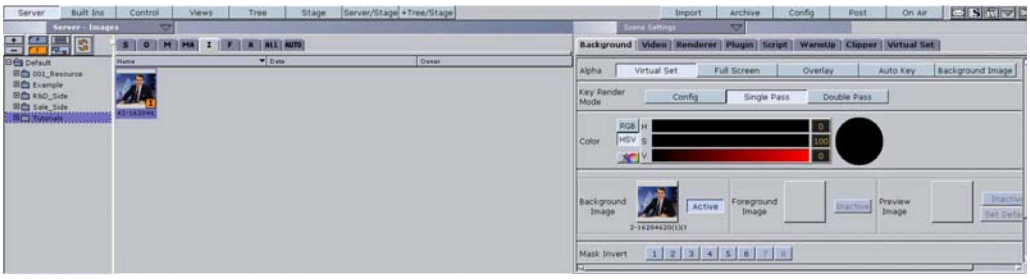
To remove an image

- Drag the image from the drop zone onto the trash can.

To view an image's source project or folder

- Click the image on the drop zone.

  The project/folder it is placed in is visible. If the image is linked, the first project/folder will be selected.



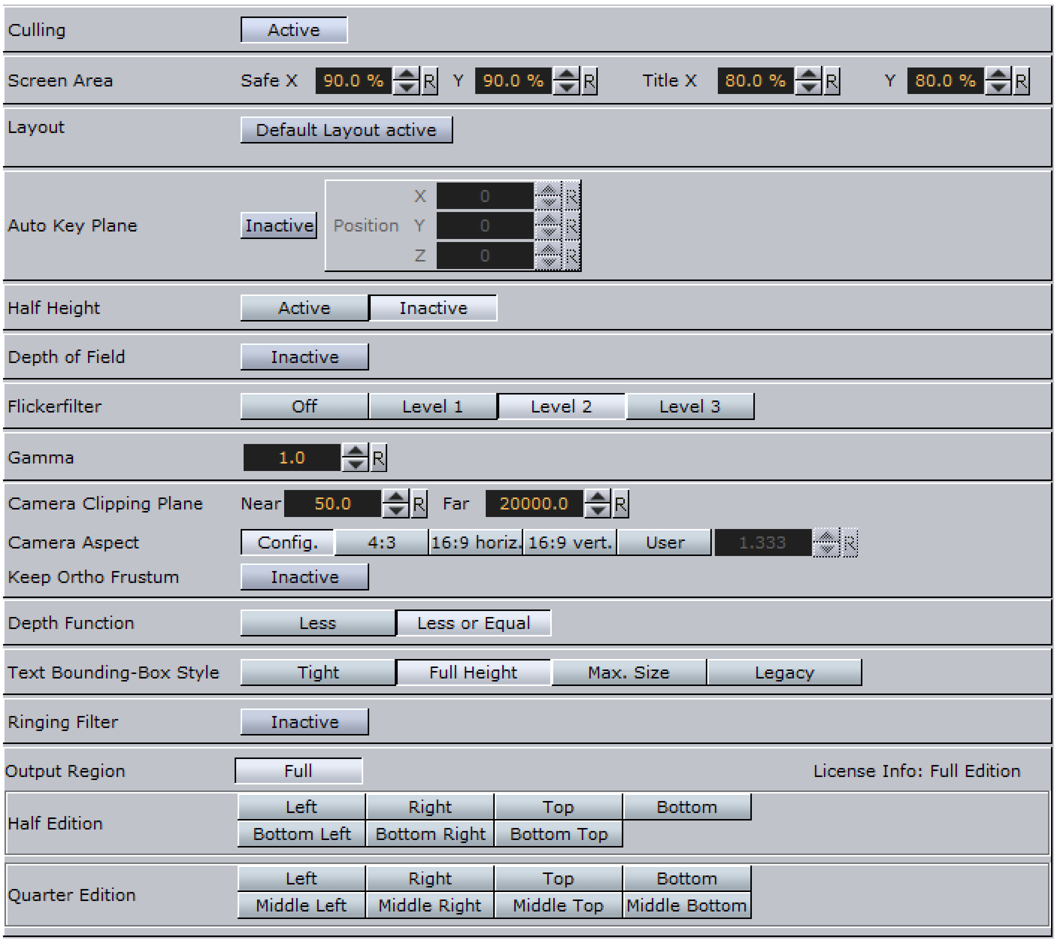
To show/hide an image in the Output Area

- Click the Active/Inactive button next to the drop zone.

  If the image has a different aspect ratio than the screen resolution, the image will be stretched to fit.

5.1.2 Rendering

In the Rendering panel, basic rendering parameters can be modified.



Culling: Active

Screen Area: Safe X 90.0 % R Y 90.0 % R Title X 80.0 % R Y 80.0 % R

Layout: Default Layout active

Auto Key Plane: Inactive Position X 0 R Y 0 R Z 0 R

Half Height: Active Inactive

Depth of Field: Inactive

Flickerfilter: Off Level 1 Level 2 Level 3

Gamma: 1.0 R

Camera Clipping Plane: Near 50.0 R Far 20000.0 R

Camera Aspect: Config. 4:3 16:9 horiz. 16:9 vert. User 1.333 R

Keep Ortho Frustum: Inactive

Depth Function: Less Less or Equal

Text Bounding-Box Style: Tight Full Height Max. Size Legacy

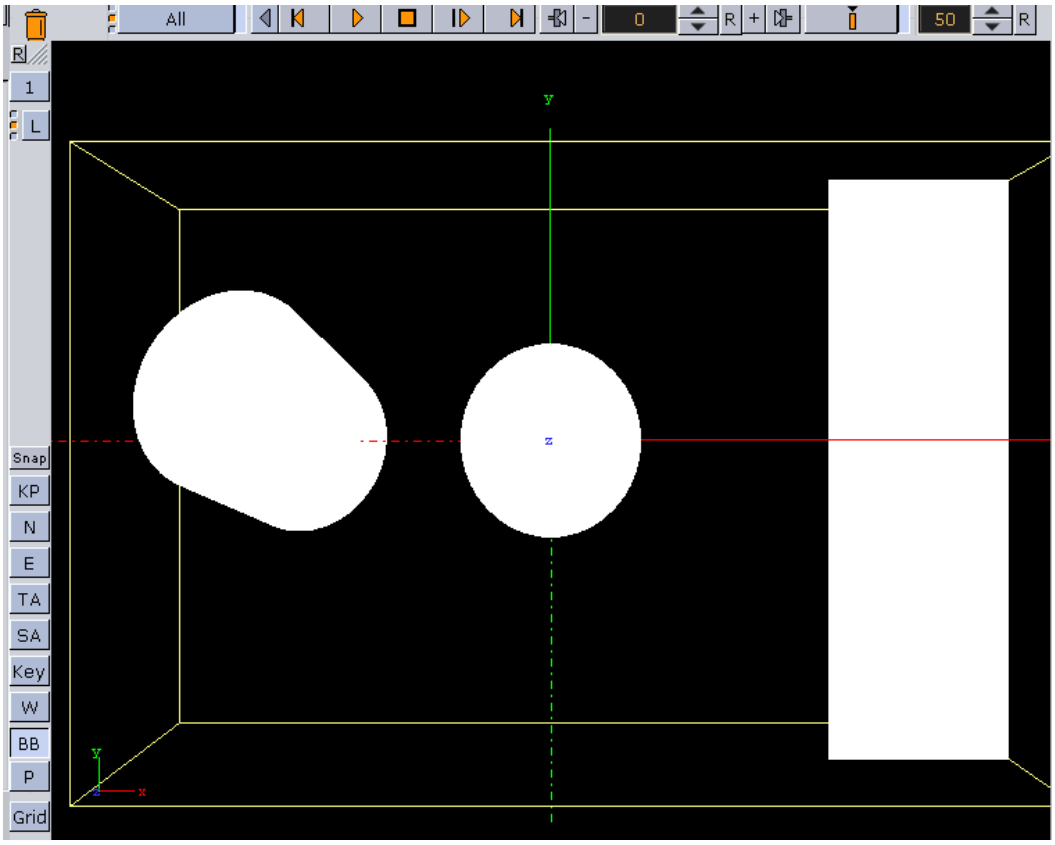
Ringing Filter: Inactive

Output Region: Full License Info: Full Edition

Half Edition: Left Right Top Bottom Bottom Left Bottom Right Bottom Top

Quarter Edition: Left Right Top Bottom Middle Left Middle Right Middle Top Middle Bottom

- **Culling** Enables/disables a culling function for the scene. The culling function is a performance saving function. With culling enabled, objects whose bounding box is outside the camera view are not rendered. This is of great importance to performance in large scenes and virtual sets. Normally the culling function should be Active.
- **Screen Area**
  - **Safe Area** Sets the size of the Safe area’s X and Y values in percent of the screen resolution. Double-clicking the R button will reset the values to default values.
  - **Title Area** Sets the size of the Title area’s X and Y values in percent of the screen resolution. Double-clicking the R button will reset the values to default values.

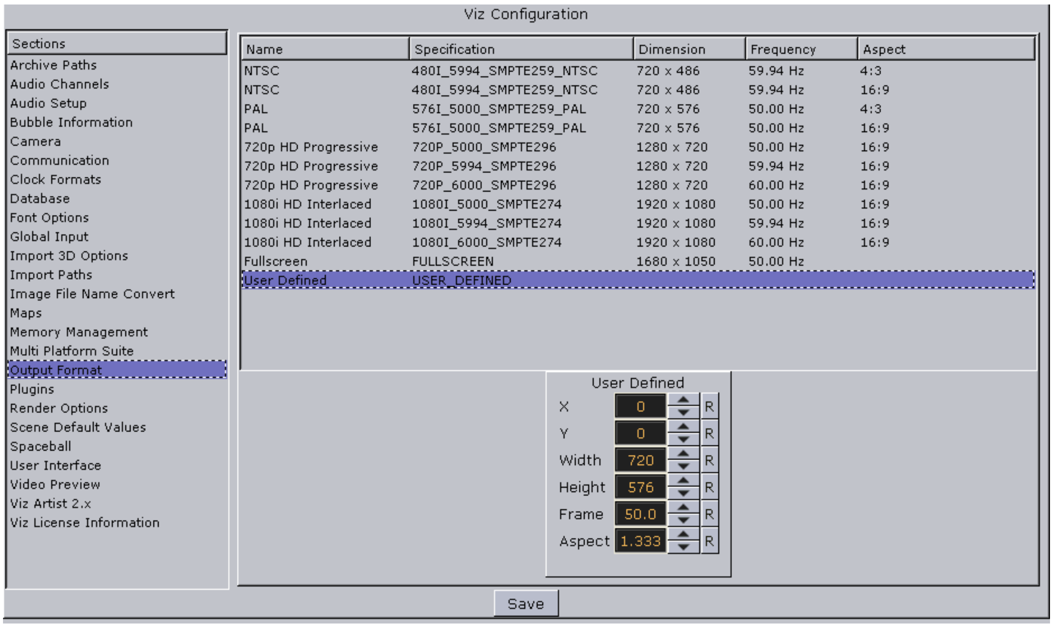


- **Auto Key Plane** Makes it possible to define the X, Y and Z Position that serves as a threshold between objects with and without key. Click the Active/Inactive button to enable the settings. The position can be set according to a tracking device carried by a person on the set, or it can be controlled by an operator. The auto key position decides which objects or parts of objects in the scene that are to be part of the key signal. Only objects positioned between the auto key position and the camera will be rendered with the key signal. A typical setup for a Virtual Set would be that the scene background is put into a separate channel of the external keyer. The objects of the background do not have a key function attached. The objects that are meant to be in front of the person on the set are placed in another channel/layer of the external keyer and have a key function attached. If the person on the set moves against the camera, the auto key function disables the key function of those objects that now are placed behind the person, thereby moving them into the background layer of the external keyer. To learn more about this, see the separate Virtual Set documentation.
- **Half Height** This option can be set to Active or Inactive. Active means that Viz Artist will render fields instead of full frames. Normally the system renders 50 or 60 full height frames and produce twice the amount of pixels needed. This is done to achieve better quality. If picking every second line of a full height frame and send it to the output, it will look better. Anti aliasing, texture filtering works “better” if rendering full height. So when Viz Artist renders half height, the quality will be reduced a little bit, especially on some textures, but it nearly renders twice as fast, because the pixel fill is half the size.
- **Depth of Field** emulates the behavior of a real camera where objects further away from the focal plane appear blurred and objects close to the focal plane appear sharp. The respective parameters can be set in the camera settings under the Camera’s Focus settings (seeFocus).

- **Flicker filter** Makes it possible to change the level of the flicker filter from Off and up to Level 3. The flicker filter is used to remove/reduce flickering from a scene. If a flicker filter is applied and the scene still flickers, try using the next level.
- **Gamma** Makes it possible to set a gamma value for the renderer.
- **Camera Clipping Plane** Sets the range of the virtual camera. Near sets the close range while Far defines the far range. Only containers within this range will be rendered.

Note: The camera range is where the Zbuffer is within. So if Zbuffer problems arise, they may be solved by editing the camera clipping plane settings.

- **Camera Aspect**: Sets the camera aspect.
  - **Config** makes the camera aspect equal to the settings in the Output Format panel of the configuration.
  - **4:3, 16:9 Horizontal** and **16:9 Vertical** makes the camera aspect independent of the aspect for the rest of the scene.
  - **User** enables the user to set a user defined camera aspect ratio. When building a scene with a customized aspect ratio for a **Viz Video Wall** set the user-defined camera aspect to the same value as the Aspect in the Config's Output Format's User Defined value for Aspect (which is set dynamically based on the other values). When building a scene with a customized aspect ratio for other e.g. **handheld/mobile devices** or **online publishing** set the Render Options' Keep Editing Aspect parameter to Editor as this will not have an affect on the On-Air or Viz Engine output. To design graphics with a user defined aspect ratio.



Name	Specification	Dimension	Frequency	Aspect
NTSC	480I_5994_SMPTE259_NTSC	720 x 486	59.94 Hz	4:3
NTSC	480I_5994_SMPTE259_NTSC	720 x 486	59.94 Hz	16:9
PAL	576I_5000_SMPTE259_PAL	720 x 576	50.00 Hz	4:3
PAL	576I_5000_SMPTE259_PAL	720 x 576	50.00 Hz	16:9
720p HD Progressive	720P_5000_SMPTE296	1280 x 720	50.00 Hz	16:9
720p HD Progressive	720P_5994_SMPTE296	1280 x 720	59.94 Hz	16:9
720p HD Progressive	720P_6000_SMPTE296	1280 x 720	60.00 Hz	16:9
1080i HD Interlaced	1080I_5000_SMPTE274	1920 x 1080	50.00 Hz	16:9
1080i HD Interlaced	1080I_5994_SMPTE274	1920 x 1080	59.94 Hz	16:9
1080i HD Interlaced	1080I_6000_SMPTE274	1920 x 1080	60.00 Hz	16:9
Fullscreen	FULLSCREEN	1680 x 1050	50.00 Hz	
User Defined	USER_DEFINED			

User Defined

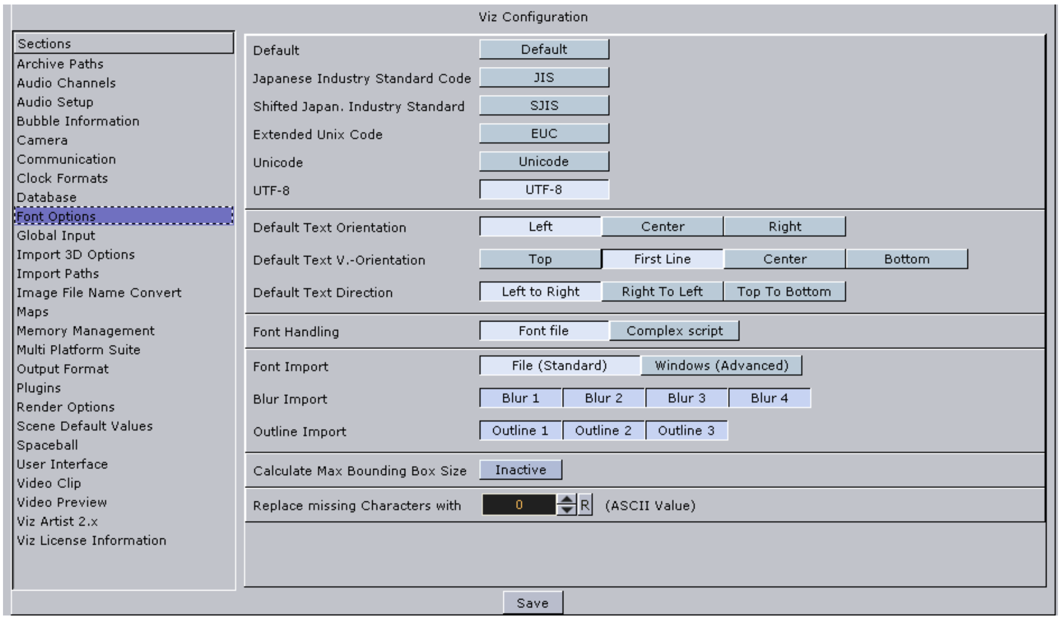
X	0	R
Y	0	R
Width	720	R
Height	576	R
Frame	50.0	R
Aspect	1.333	R

- **Keep Orthog. Frustum**: When active, takes the original frustum stored and does not allow its recomputation after changes have been made to the camera parameters in the orthogonal mode.
- **Depth Function** Specifies the function used to compare each incoming pixel depth value with the depth value present in the depth buffer. Equal: passes if the incoming depth value is equal to the stored depth value. Less or Equal: passes if the incoming depth value is less than or equal to the stored depth value.

Note: The Depth Function setting is OpenGL-specific.

- **Text Bounding Box Style** Tight means that the bounding box is drawn tight around all text used in this scene. This means just the height of the actual text used. When set to Full Height, the bounding box is higher: Viz Artist calculates the bounding box for characters that need some space underneath (e.g. the letter 'g') and for characters that need more space above (e.g. a parenthesis) even if these characters are not used in the text of the scene. Max. Size is for scenes imported from version 2.8 so that the various plug-ins which depend on font bounding boxes are compatible in version 3.5, such as Max Size and Autofollow; it also includes a small trailing space between the letter and the bounding box. Legacy produces the exact same bounding box as version 2.8.

Note: The Full Height setting does not work for special characters like the 'Ä' which is even higher in some fonts. There is a special config setting for this:



Viz Configuration

Sections
Archive Paths
Audio Channels
Audio Setup
Bubble Information
Camera
Communication
Clock Formats
Database
Font Options
Global Input
Import 3D Options
Import Paths
Image File Name Convert
Maps
Memory Management
Multi Platform Suite
Output Format
Plugins
Render Options
Scene Default Values
Spaceball
User Interface
Video Clip
Video Preview
Viz Artist 2.x
Viz License Information

Default: Default
Japanese Industry Standard Code: JIS
Shifted Japan. Industry Standard: SJIS
Extended Unix Code: EUC
Unicode: Unicode
UTF-8: UTF-8

Default Text Orientation: Left | Center | Right
Default Text V.-Orientation: Top | First Line | Center | Bottom
Default Text Direction: Left to Right | Right To Left | Top To Bottom

Font Handling: Font file | Complex script

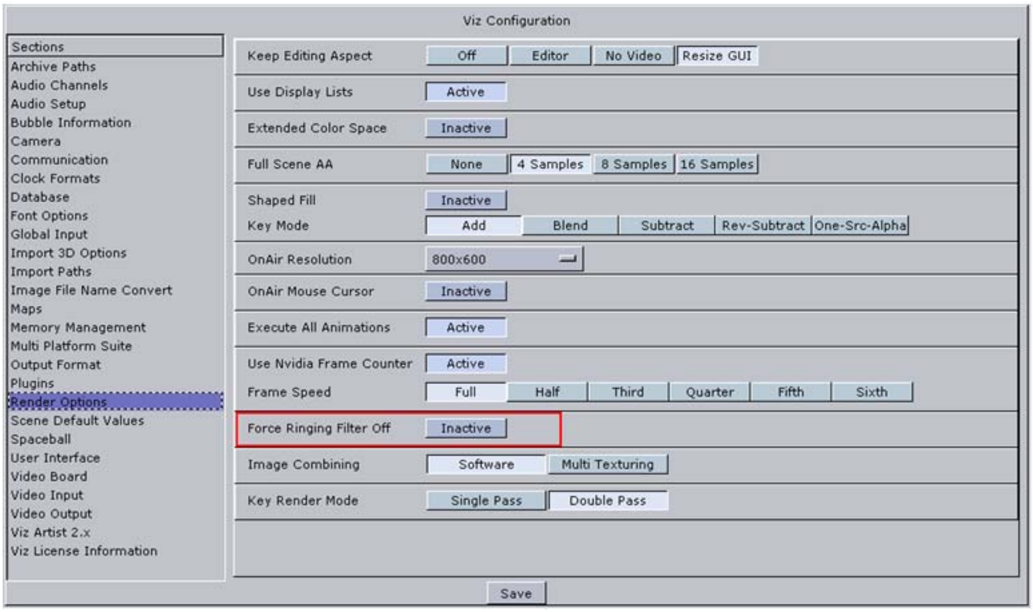
Font Import: File (Standard) | Windows (Advanced)
Blur Import: Blur 1 | Blur 2 | Blur 3 | Blur 4
Outline Import: Outline 1 | Outline 2 | Outline 3

Calculate Max Bounding Box Size: Inactive

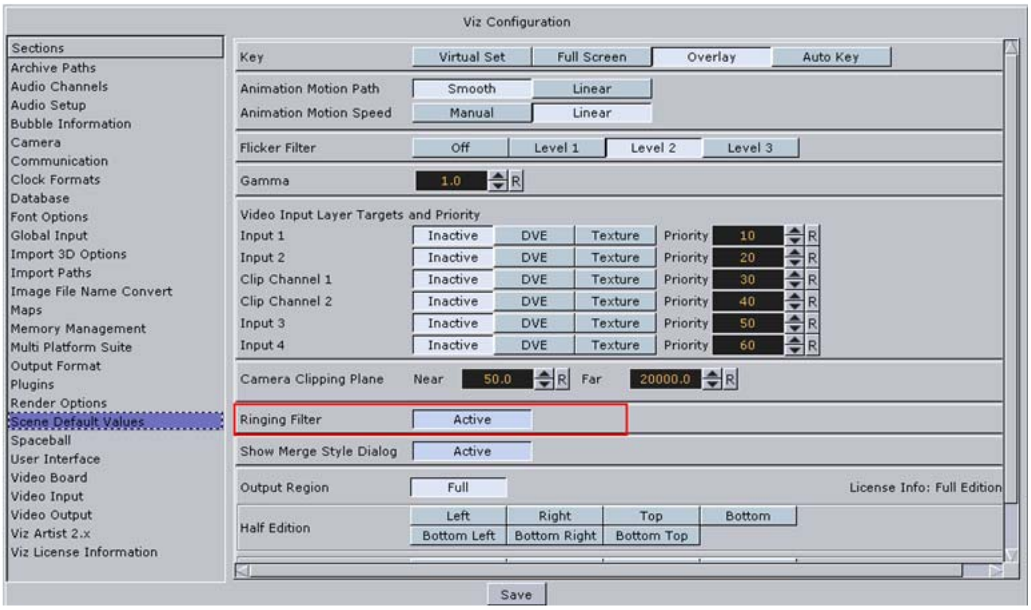
Replace missing Characters with: 0 R (ASCII Value)

Save

- **Ringing Filter** Reduces the ringing artifacts on the SDI output and applies only to SD formats. The ringing filter can be switched off with the config setting under Render Options; set Force Ringing Filter Off to Active.


Viz Configuration
Sections
Archive Paths
Audio Channels
Audio Setup
Bubble Information
Camera
Communication
Clock Formats
Database
Font Options
Global Input
Import 3D Options
Import Paths
Image File Name Convert
Maps
Memory Management
Multi Platform Suite
Output Format
Plugins
Render Options
Scene Default Values
Spaceball
User Interface
Video Board
Video Input
Video Output
Viz Artist 2.x
Viz License Information
Keep Editing Aspect: Off | Editor | No Video | Resize GUI
Use Display Lists: Active
Extended Color Space: Inactive
Full Scene AA: None | 4 Samples | 8 Samples | 16 Samples
Shaped Fill: Inactive
Key Mode: Add | Blend | Subtract | Rev-Subtract | One-Src-Alpha
OnAir Resolution: 800x600
OnAir Mouse Cursor: Inactive
Execute All Animations: Active
Use Nvidia Frame Counter: Active
Frame Speed: Full | Half | Third | Quarter | Fifth | Sixth
Force Ringing Filter Off: Inactive
Image Combining: Software | Multi Texturing
Key Render Mode: Single Pass | Double Pass
Save

Note: The default ringing filter is located in the config Scene Default Values.


Viz Configuration
Sections
Archive Paths
Audio Channels
Audio Setup
Bubble Information
Camera
Communication
Clock Formats
Database
Font Options
Global Input
Import 3D Options
Import Paths
Image File Name Convert
Maps
Memory Management
Multi Platform Suite
Output Format
Plugins
Render Options
Scene Default Values
Spaceball
User Interface
Video Board
Video Input
Video Output
Viz Artist 2.x
Viz License Information
Key: Virtual Set | Full Screen | Overlay | Auto Key
Animation Motion Path: Smooth | Linear
Animation Motion Speed: Manual | Linear
Flicker Filter: Off | Level 1 | Level 2 | Level 3
Gamma: 1.0 R
Video Input Layer Targets and Priority
Input 1: Inactive | DVE | Texture | Priority 10 R
Input 2: Inactive | DVE | Texture | Priority 20 R
Clip Channel 1: Inactive | DVE | Texture | Priority 30 R
Clip Channel 2: Inactive | DVE | Texture | Priority 40 R
Input 3: Inactive | DVE | Texture | Priority 50 R
Input 4: Inactive | DVE | Texture | Priority 60 R
Camera Clipping Plane: Near 50.0 R Far 20000.0 R
Ringing Filter: Active
Show Merge Style Dialog: Active
Output Region: Full
License Info: Full Edition
Half Edition: Left | Right | Top | Bottom | Bottom Left | Bottom Right | Bottom Top
Save

- **Output Region** Becomes inactive if you design a Half or Quarter Edition scene. To revert to a full screen scene, click Full.
- **Half Edition** If you have a full Viz Artist license, but want to design a scene for another license that has only a Half Edition, select which portion of the scene you want to design your scene in. Your options are:
  - Left
  - Right
  - Top
  - Bottom
  - Bottom Left
  - Bottom Right

  - Bottom Top
- **Quarter Edition** If you have a full Viz Artist license, but want to design a scene for another license that has only a Quarter Edition, select which portion of the scene you want to design your scene in. Your options are:
  - Left
  - Right
  - Top
  - Bottom
  - Middle Left
  - Middle Right
  - Middle Top
  - Middle Bottom

To design graphics with a user defined aspect ratio

1. **Start** Viz Artist
2. Click the **Config** button to open the configuration (see Configuring Viz)
3. If you plan to render graphics in many different aspects (e.g. still images for online publishing using Escenic Content Engine) you should select the Render Options section and set *Keep Editing Aspect* to **Editor**
   - This will ensure your graphics are rendered according to the scene's configured *camera aspect* (and not be affected by the Output Format aspect)
4. In the Output Format section set the **format** that best fits your screen's resolution and aspect (e.g. 16:9) and **close Viz Artist**
   - Viz Artist is resolution independent and its workspace is optimized for designing in 16:9 and 4:3 aspects.
   - When designing graphics with a user defined aspect, only set the *Output Format* to size the aspect of your Scene Editor and adjust the *camera aspect* further to define it according to your scene's required and final aspect.
5. **Start** Viz Artist
6. Click the **Server** button
7. From the Scene Setup menu select Scene Settings
8. Under the Rendering tab set **Camera Aspect** to **User**
9. Enter your aspect ratio in the field beside the User field
   - To calculate your aspect ratio simply divide width with height (e.g. 1600:900=1.778)

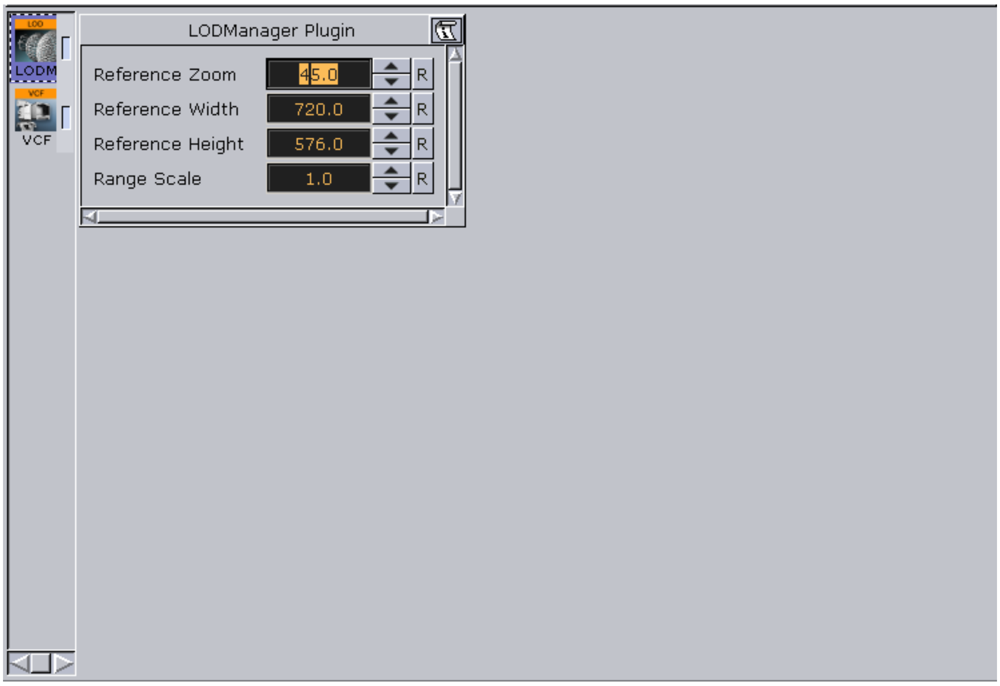
See Also

- Scene Editor Camera Controls and Shortcuts

5.1.3 Plug-in

In the Plugin panel, plugins can be applied to the scene. Scene plugins are "global" functions that have influence on the whole scene. They are mainly used when external programs are interfaced with Viz Artist. Viz Artist uses an open interface for plugins. New plugins can either be obtained or written by yourself and then installed into Viz Artist.

Viz Artist provides a set of already installed plugins. The installed scene plugins can be found if you click on **Built Ins** in the main menu and select **Function Scene** in the pull-down menu. For a detailed list and description of the plugins delivered with Viz Artist see Scene Plugins.

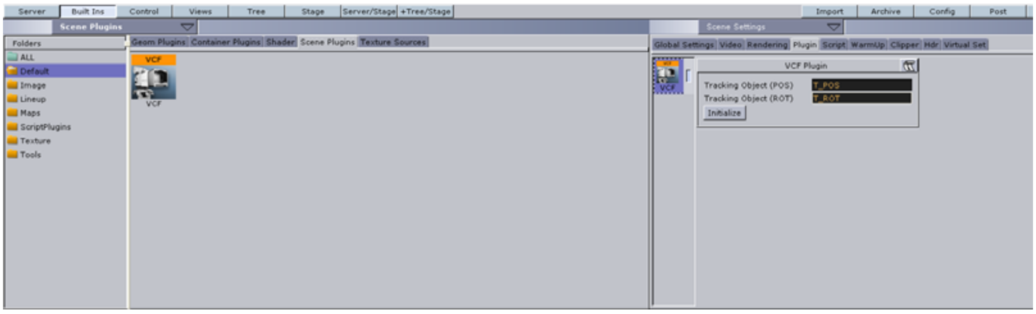


Reference Zoom 45.0
Reference Width 720.0
Reference Height 576.0
Range Scale 1.0

The area for the plugins is divided in two sections. On the left side you will find all plugins applied to your scene as icons. On the right side an editor for the properties of a plugin will open when the plugin is highlighted.

To add a plugin

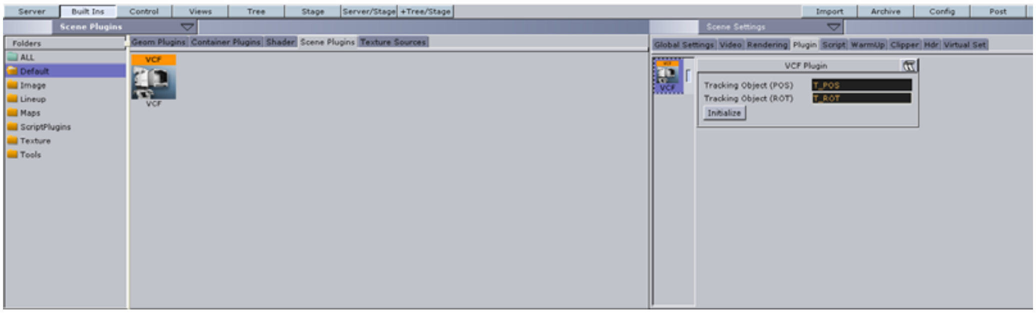
1. Click **Built Ins** and select **Scene Plugins**.
2. Browse for the plugin you want to apply to the scene.
3. Drag the selected plugin to the left side of the Scene Setting’s Plugin area.



Tracking Object (POS) T_POS
Tracking Object (ROT) T_ROT

To edit plugin properties

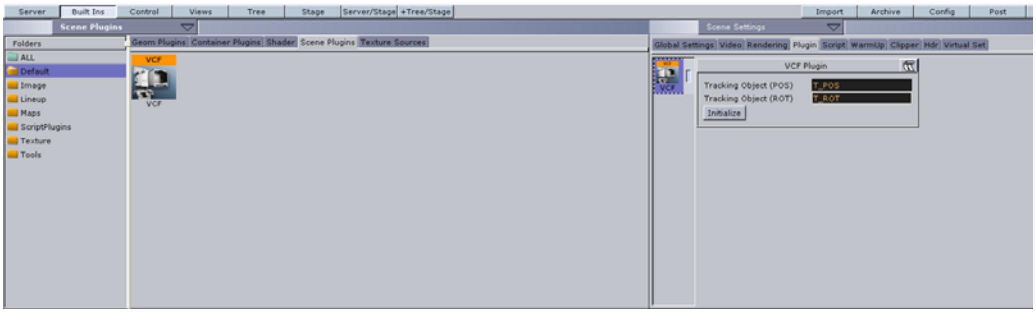
1. In the Scene Setting’s Plugin area, select the plugin.
2. Edit the plugin’s properties in the editor to the right.



Tracking Object (POS) T_POS
Tracking Object (ROT) T_ROT

To show plugin information

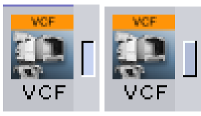
1. In the Scene Setting's Plugin area, select the plugin.
2. In the plugin editor, click the info icon.



Tracking Object (POS) T_POS
Tracking Object (ROT) T_ROT

To enable/disable a plugin

- In the Scene Setting's Plugin area, click on the enable/disable switch in the icon.



For greater convenience Viz Artist allows you to enable or disable the plugins already applied to the scene. This will help you not to loose the overview when adjusting the plugin properties to the values you need.

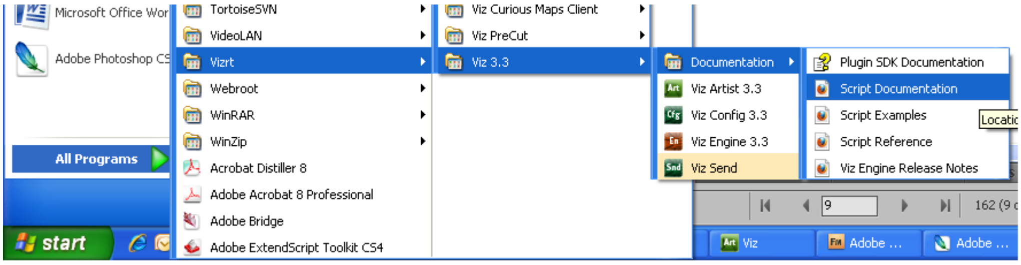
Note: After you add a plugin to the scene, it will automatically be enabled.

To Remove A Plugin

- Drag the plugin onto the trash can.

5.1.4 Script

In the Script panel, scripts can be added to the scene. Scene scripts are mostly used for holding global variables and for procedures used globally in the entire scene. For more information about scripting in Viz Artist, from the Windows desktop, you can also see scripting information by clicking **Start** -> **All Programs** -> **Vizrt** -> **Viz 3.x** -> **Documentation** -> **Script Documentation.** Alternatively, from within the Script panel, press <F1>.

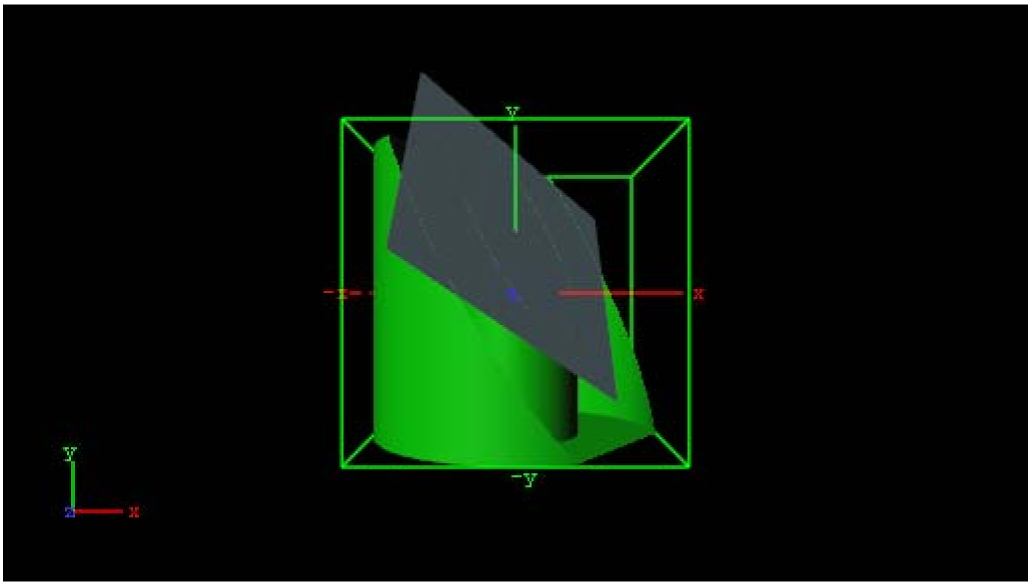


See Also

- Script plug-in
- Script Editor
- Script Editor Shortcuts

5.1.5 Clipper

In the Clipper panel, settings for scenes that have the Clipper plug-in attached can be modified.

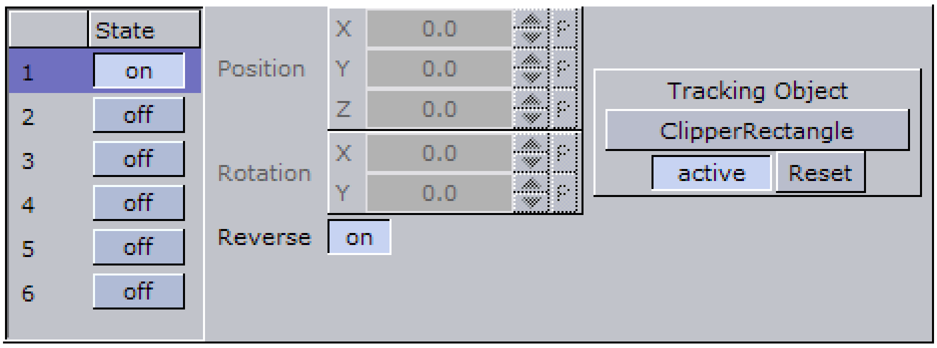


The Clipper plug-in is an alternative to the mask function; the clipper works in 3D and the mask in 2D. Objects using the clipper plug-in, when placed behind the clipper plane, will be masked/clipped. If an object is only partially behind a clipper plane, only the part of the object that is behind the plane will be clipped out since the clipping is done in true 3D space. The plug-in uses the OpenGL clipping planes that supports up to six planes.

This section contains the following topics:

- Properties and Parameters

Properties and Parameters



	State
1	on
2	off
3	off
4	off
5	off
6	off

Position X 0.0
Y 0.0
Z 0.0

Rotation X 0.0
Y 0.0

Reverse on

Tracking Object
ClipperRectangle

- **Number**: Shows the clipper plane number, from 1 to 6.
- **State**: Makes it possible to switch the selected clipper plane On or Off.
- **Position**: Makes it possible to modify the position of the clipper plane along the X, Y, and Z axis.
- **Rotation**: Makes it possible to modify the rotation along the X and Y axis.
- **Reverse**: If this option is set to Off, everything in front of the clipper plane is shown. If it is set to On, everything behind the clipper plane is shown. Tracking Objects with Clipper Plane
- **Tracking Object**: A clipper plane can be set up to track containers regarding position and rotation. Tracking position means that the position of the clipper plane will be in the center of the container, and tracking rotation means that the clipper plane will rotate if the object rotates. This is especially useful when animating a scene.

See Also

- Creating Animations
- Clipper

5.1.6 Virtual Set

The Virtual Set panel is only available if the Virtual Set option is installed. This panel makes it possible to set up and save configuration for virtual sets. For information about the parameters, see the separate Virtual Set documentation.

5.2 Implementing Video

How you implement video in Viz Artist depends on your machine and video board configuration. There are also two general approaches: from an external feed or from a video clip. Additionally, you implement it in Viz Artist in one of two ways. One is at the texture level, meaning the video is part of the graphics. The other approach is as DVE (Digital Video Effect). This is a feature of specific video boards (e.g. Targa, Matrox, but *not* DVS).

Before you can implement video, you need to configure Video Output and Video Input configuration settings. These settings determine the frame rate (PAL/NTSC) of the entire video system. See the Viz Engine documentation for details.

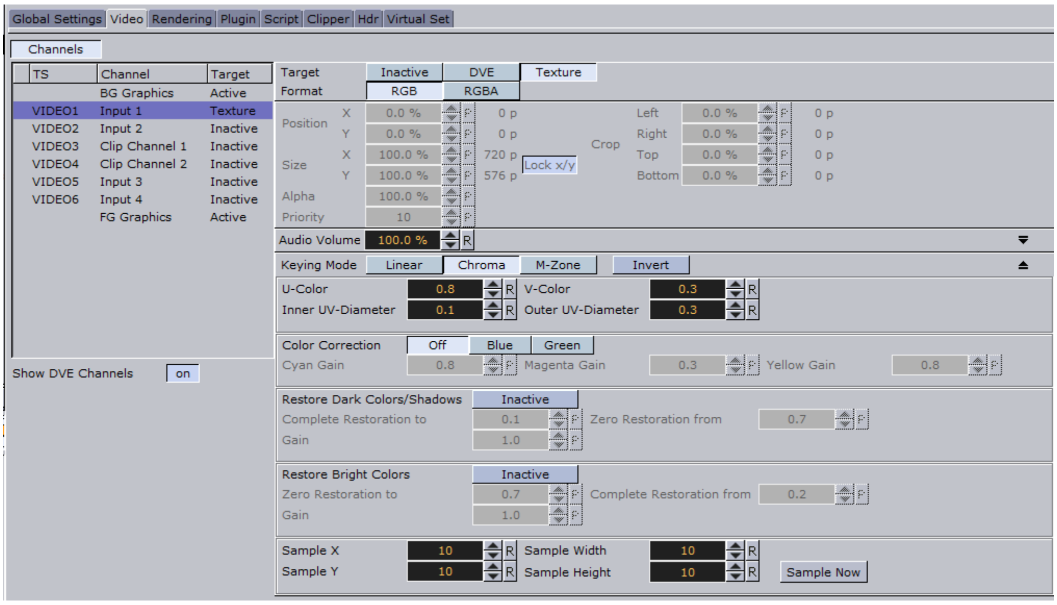
The following video cards supported:

- Matrox
- DVS
- CGP Panasonic
- Targa

There are a few varieties of Matrox cards: X.mio and X.mio2 mainly differ in the number of possible video inputs and outputs (X.mio2/44). DSX/LE2 in does not have any input, codec support or compositor. Furthermore, for X.mio and X.mio2 Matrox cards video inputs and clip channels can be disabled in the Matrox configuration.

IMPORTANT! The most basic rule for working with the video version of Viz Engine is to always have a valid genlock attached to the video board, no matter which board is installed. Without a proper genlock setup there is no guarantee that the Engine will function properly. See Video Output.

Video parameters are modified in Viz Artist in the Scene Setting’s Video Tab.



TS	Channel	Target
	BG Graphics	Active
VIDEO1	Input 1	Texture
VIDEO2	Input 2	Inactive
VIDEO3	Clip Channel 1	Inactive
VIDEO4	Clip Channel 2	Inactive
VIDEO5	Input 3	Inactive
VIDEO6	Input 4	Inactive
	FG Graphics	Active

Show DVE Channels on

Target: Inactive DVE Texture
Format: RGB RGBA
Position X 0.0 % 0 p
Position Y 0.0 % 0 p
Size X 100.0 % 720 p
Size Y 100.0 % 576 p
Lock x/y
Crop Left 0.0 % 0 p
Crop Right 0.0 % 0 p
Crop Top 0.0 % 0 p
Crop Bottom 0.0 % 0 p
Alpha 100.0 %
Priority 10
Audio Volume 100.0 %
Keying Mode: Linear Chroma M-Zone Invert
U-Color 0.8
V-Color 0.3
Inner UV-Diameter 0.1
Outer UV-Diameter 0.3
Color Correction: Off Blue Green
Cyan Gain 0.8
Magenta Gain 0.3
Yellow Gain 0.8
Restore Dark Colors/Shadows Inactive
Complete Restoration to 0.1
Zero Restoration from 0.7
Gain 1.0
Restore Bright Colors Inactive
Zero Restoration to 0.7
Complete Restoration from 0.2
Gain 1.0
Sample X 10
Sample Width 10
Sample Y 10
Sample Height 10
Sample Now

In the left pane, there are three columns, showing all available input channels.

- **TS** Channel’s internal name.

Note: This is the same name as in the command interface.

- **Channel** Corresponding to the Texture icon names.
- **Target** The channel’s state: Active or Inactive for foreground and background graphics; Inactive, DVE and Texture for input channels.

Clicking on a channel in the left pane opens its editor in the right pane.

This section also contains information on the topics:

- Methodological Considerations

- Clip Playback Use Cases
- Implementing Video Through an External Feed
- Implementing Video Through a Video Clip
- Configuring the Keying Mode for Video

5.2.1 Methodological Considerations

When using DVE, the graphics scene and the channel are composited on the video card (Matrox X.mio or Targa). An SDI monitor is needed to show the channel when using DVE mode. When using DVE, there is a limited set of 2D control options available in the design and animation of the Viz Artist scene (position, size, crop and alpha). Please take into consideration that chroma keying is not supported for channels set to DVE.

If more control of the video is needed, e.g. being able to map the video onto a 3D object in a scene, then Texture mode can be used. The video would then have the same control parameters as an image in Viz Artist. However, the video may now affect rendering performance, since it needs to composite the video into the render window.

5.2.2 Clip Playback Use Cases

This section contains information on the following common clip playback use cases:

- Playing Overlay Clips - Not Full Screen
- Clip with Audio - Not Full Screen
- Clip with Audio - Full Screen
- Clip with Audio - Full Screen - 3D Space Transformation

Playing Overlay Clips - Not Full Screen

Playing a clip which is not full screen in size, like a spinning logo, and insert snipe or a special effect without audio. These are known as overlay clips.

To play overlay clips it is recommended that you use the Soft Clip clip player. Even though Softclip can play clips up to HD 1080, it should be mainly used for overlay clip playback. For full screen playback there is a more efficient clip player available in Viz. The advantage of the Softclip clip player is that it can play clips in any pixel size, aspect ratio or frame rate. Softclip is based on the VFW (Video for Windows) AVI clip playback workflow on Windows.

Soft Clip is able to play all installed VFW (Video For Windows/AVI) codecs on your computer. There are dozens of codecs available, each with it's own advantages and drawbacks. There is no general rule which codec to use (see Supported Codecs), but it is recommended that you use the different types of Matrox codecs (depending on whether the alpha channel is necessary or not), avi uncompressed or the freely available HuffYuv codec. Softclip supports RGBA playback if the codec is able to store the alpha channel; avi uncompressed supports alpha, also the Matrox Mpg2iFrame&alpha codec and the HuffYuv codec, just to mention a few.

A typical use case for the Softclip player would be:

- The playback of a spinning looping logo with alpha channel. The size of this clip could be around 350x350 pixels.
- The playback of an explosion overlay clip with alpha. The size is around 500x500 pixels.
- The playback of a snipe/insert with alpha. The size is around 600x150 pixels.

Note: The Softclip player is not intended for full screen playback!

Clip with Audio - Not Full Screen

Playing a clip with audio which is not full screen in size, like a strap of an advertiser, a 3D object transition or an effect or a reveal of an element only.

To play clips which need audio but don't need to be played back in full screen mode, you should use MoViz. Moviz is very similar to Soft Clip, with the main difference being that it supports audio. Moviz is based on the directX clip playback workflow on Windows, meaning it uses a different library and codec filters than the Softclip plugin. If you install a VFW codec for avi playback like for Softclip, it is not guaranteed that your clip will also play in Moviz, since Moviz is directX-based. In general it is very helpful to install the free package of the main concept codecs, since with this package you get dozens of ffdshow filters for directX for free and this will raise the compatibility with which you will be able to play regular avi files via Moviz. To use the full potential of Moviz you should encode your clips whit ffd instead of avi. If you want to render clips into ffd as an example from Adobe After Effects, there will be a new section showing up in the render section named ffd, besides Quicktime and Video for Windows, TGA single frames and others. In the ffd settings you can select the necessary format, like dv, mpg and many others.

A typical use case for the Moviz player would be:

- The playback of a insert advertiser with audio on the bottom of your screen. The size of this clip would be something around 720x150 pixels.
- The playback of a “clip universe”: the animated Viz camera is flying through space and passes clips.

Note: The Moviz player is not intended for full screen playback!

Clip with Audio - Full Screen

Playing a clip with audio in full screen mode, but only using it as a background in my Viz scene. The clip does not need to be rotated in 3D space.

To play full screen background clips it is recommended that you use the Matrox clip channels. The dimensions and frame rate of these clips must be either HD 1080, HD 720, PAL or NTSC. Any free pixel size, aspect ratio or frame rate is not supported on the Matrox channels.

If the clip is 2D, meaning there is no transformation/rotation in 3D space needed, then you should set your Matrox clip channel to DVE. In DVE mode you get the maximal clip quality achievable.

When using DVE mode, don't forget to apply a key function to your Viz elements or switch the renderer to autokey, otherwise they will not appear in front of the clip. See Configuring the Keying Mode for Video.

If you want to use an alpha channel on this clip, use the Matrox Mpg2iFrame&alpha codec and configure your clip channels in the Viz Config correctly (activate alpha). Also here it is recommended that you use the Matrox AVI codecs, but the Clip channels support many more formats than just the Matrox codecs. See Supported Codecs.

A typical use case for the DVE setting would be:

- The playback of a clip at it's best quality, with or without any overlaying graphics.

Note: Using the Matrox clip channels for playback is the most efficient clip playback for full screen clips!

Clip with Audio - Full Screen - 3D Space Transformation

Playing a clip with audio in full screen mode and transforming it in 3D space.

To play full screen clips it is recommended that you use the Matrox clip channels. The dimensions and frame rate of these clips must be either HD 1080, HD 720, PAL or NTSC. Any free pixel size, aspect ratio or frame rate is not supported on the Matrox channels.

To be able to transform the clip in 3D space you must set the clip channel to Texture. Compared to DVE mode, the clip is slightly more blurred, but this is essential to avoid noise on the clip texture when moved or rotated in 3D space.

If you want to use an alpha channel on this clip, use the Matrox Mpg2iFrame&alpha codec and configure your clip channels in the Viz Config correctly (activate alpha). Also here it is recommended that you use the Matrox AVI codecs, but the Clip channels support many more formats than just the Matrox codecs. See Supported Codecs.

A typical use case for the Texture setting would be:

- The playback of a clip in 3D space with transformations on rotation, scaling and position.
- The playback of a clip on a 3D object.

Note: Using the Matrox clip channels for playback is the most efficient clip playback for full screen clips!

5.2.3 Implementing Video Through an External Feed

External video is implemented as follows:

- External Video Through DVE
- External Video Through Texture

External Video Through DVE

This section covers the following procedures:

- To implement external video (DVE/Matrox)
- To implement external video (DVE/DVS)
- To implement external video (DVE/Targa)

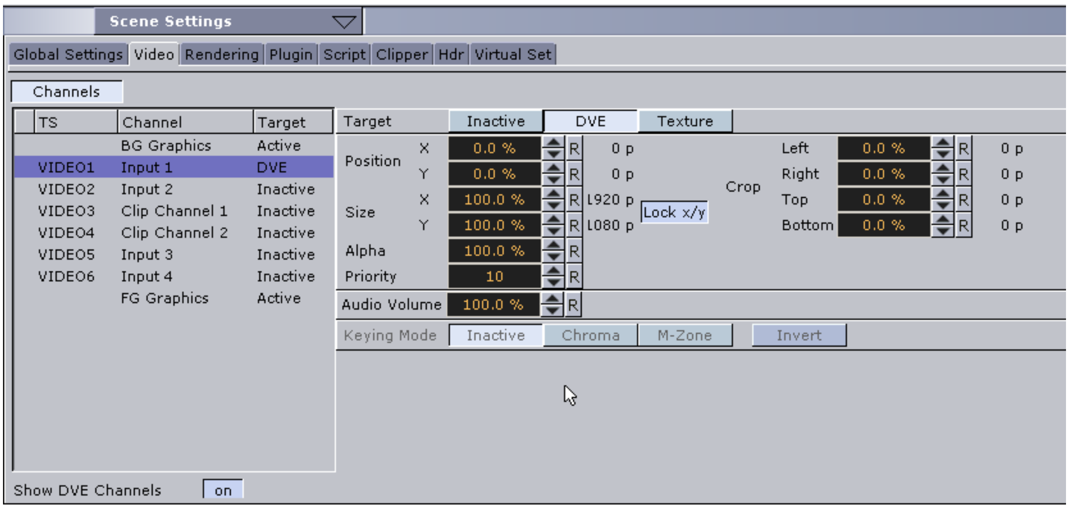
Note: Implementing external video through DVE does not have any impact on rendering performance, since it is handled solely on the board.

To implement external video (DVE/Matrox)

1. Click **Config** and configure the alpha in Video Input.

Tip: **w/Alpha** Enable this setting when you need use external video with alpha. Two of the video inputs to the Matrox board are combined to be the fill and key. Input A and B will be combined and input C and D will be combined, respectively.

2. Save and restart Viz Artist.
3. Click **Server**.
4. In the right pane, from the pull-down menu, select **Scene Settings** and click the **Video** tab.
5. In the left pane of the Video parameters, click **Input 1**.
6. In the right pane of the Video parameters, in the Target row, click the **DVE** button. You can now adjust the parameters in the DVE control panel.
   - **Position**
   - **Size**
   - Cropping
   - **Alpha** For each layer, you can adjust the level of opacity, where 100% is fully opaque and 0% is fully transparent.
   - **Priority** Assigns the position of each layer in the layer stack, where 0 is the bottom-most layer, and 100 is the top-most layer.



TS	Channel	Target
	BG Graphics	Active
VIDEO1	Input 1	DVE
VIDEO2	Input 2	Inactive
VIDEO3	Clip Channel 1	Inactive
VIDEO4	Clip Channel 2	Inactive
VIDEO5	Input 3	Inactive
VIDEO6	Input 4	Inactive
	FG Graphics	Active

Target: Inactive / DVE / Texture

Position X 0.0 % 0 p
Position Y 0.0 % 0 p
Size X 100.0 % 1920 p
Size Y 100.0 % 1080 p
Alpha 100.0 %
Priority 10
Audio Volume 100.0 %

Crop Left 0.0 % 0 p
Crop Right 0.0 % 0 p
Crop Top 0.0 % 0 p
Crop Bottom 0.0 % 0 p

Show DVE Channels on

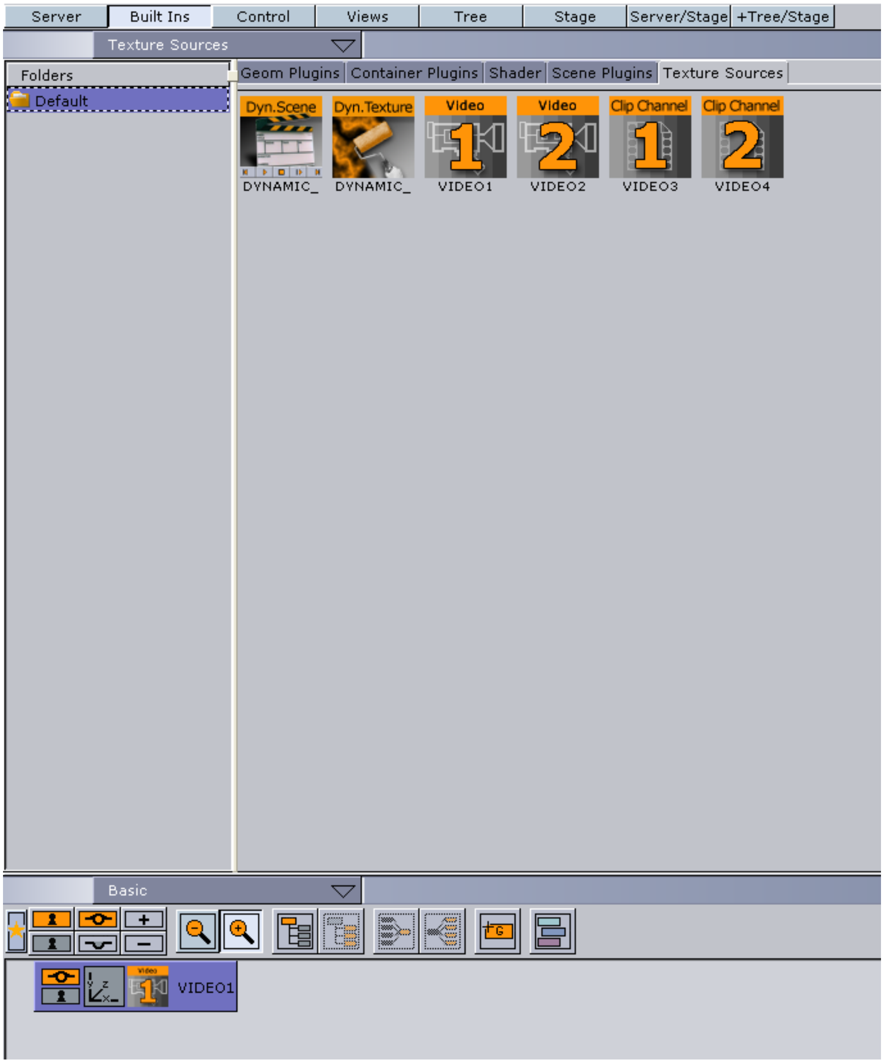
Note: BG Graphics are always the lowest in the stack, even below zero, and FG graphics are assigned a default of 100. As such, you cannot assign a priority to BG Graphics.

The Scene Editor displays a blue rectangle, showing that the input channel is properly configured.

To implement external video (DVE/DVS)

Note: In DVS, DVE is simulated through the software.

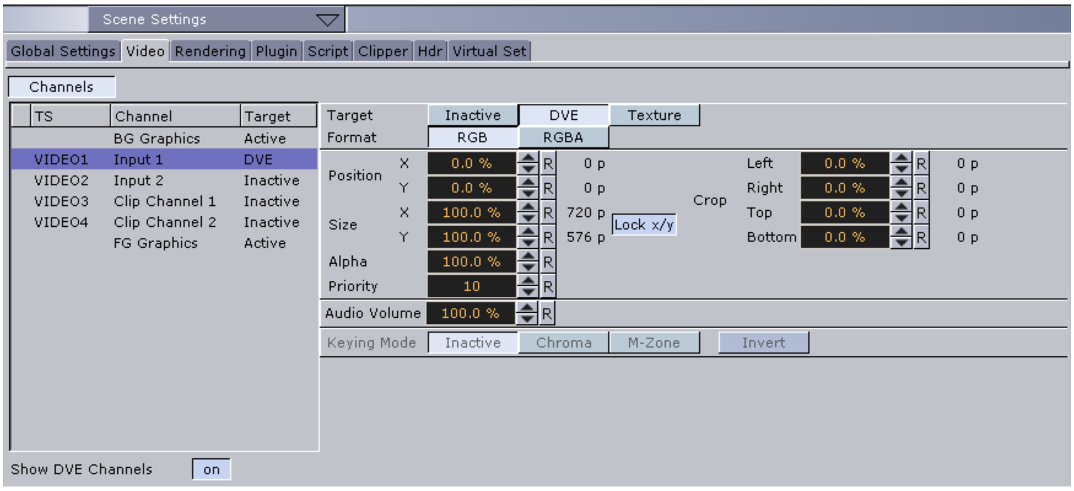
1. Click **Built Ins.**
2. In the drop-down menu underneath, select **Texture Sources.**
3. Drag and drop the **Video1** plugin into the scene tree.



- You can now do to the video anything you would do with a texture.

To implement external video (DVE/Targa)

1. Click **Server**.
2. In the right pane, from the pull-down menu, select **Scene Settings** and click the **Video** tab.
3. In the left pane of the Video parameters, click **Input 1**.
4. In the right pane of the Video parameters, in the Target row, click the **DVE** button. You can now adjust the parameters in the DVE control panel.
   - **Position**
   - **Size**
   - Cropping
   - **Alpha** For each layer, you can adjust the level of opacity, where 100% is fully opaque and 0% is fully transparent.
   - **Priority** Assigns the position of each layer in the layer stack, where 0 is the bottom-most layer, and 100 is the top-most layer.



TS	Channel	Target
	BG Graphics	Active
VIDEO1	Input 1	DVE
VIDEO2	Input 2	Inactive
VIDEO3	Clip Channel 1	Inactive
VIDEO4	Clip Channel 2	Inactive
	FG Graphics	Active

Target: Inactive | DVE | Texture
Format: RGB | RGBA
Position X 0.0 % 0 p
Position Y 0.0 % 0 p
Size X 100.0 % 720 p
Size Y 100.0 % 576 p
Lock x/y
Alpha 100.0 %
Priority 10
Crop Left 0.0 % 0 p
Crop Right 0.0 % 0 p
Crop Top 0.0 % 0 p
Crop Bottom 0.0 % 0 p
Audio Volume 100.0 %
Keying Mode: Inactive | Chroma | M-Zone | Invert
Show DVE Channels on

Note: BG Graphics are always the lowest in the stack, even below zero, and FG graphics are assigned a default of 100. As such, you cannot assign a priority to BG Graphics.

The Scene Editor displays a blue rectangle, showing that the input channel is properly configured.

5. In the right pane of the Video settings, in the Alpha row, select either **RGB** or **RGBA**.
   - **RGB** The input is taken as-is, meaning, it is used as fill.
   - **RGBA** Both inputs of the Targa are combined to one signal having fill and key.

   Now the layer is part of the compositing on the video hardware.

External Video Through Texture

This section covers the following procedures:

- To implement external video (Texture/Matrox)
- To implement external video (Texture/DVS)
- To implement external video (Texture/CGP Panasonic)
- To implement external video (Texture/Targa)

Note: Other than implementing external video through DVE, implementing through texture does have an impact on rendering performance, since the video stream needs to be transferred to the graphics board.

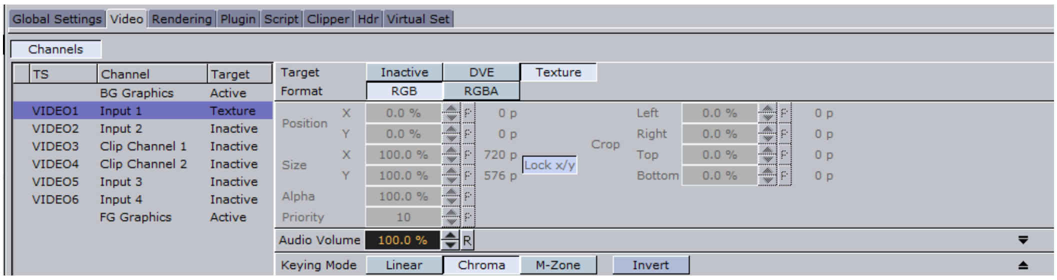
To implement external video (Texture/Matrox)

1. Click **Config** and configure the alpha in Video Input.

Tip: **w/Alpha** Enable this setting when you need use external video with alpha. Two of the video inputs to the Matrox board are combined to be the fill and key. Input A and B will be combined and input C and D will be combined, respectively.

2. Save and restart Viz Artist.
3. Click **Server**.

4. In the right pane, from the pull-down menu, select **Scene Settings** and click the **Video** tab.
5. In the left pane of the Video parameters, click **Input 1**.
6. In the right pane of the Video parameters, in the Target row, click the **Texture** button.
7. Select the keying mode.
   - **Linear** No chroma keying.
   - **Chroma** Uses the internal chroma keyer.
   - **M-Zone** Uses the multi-zone keyer.
   - **Invert** Only works with Chroma.



TS	Channel	Target
	BG Graphics	Active
VIDEO1	Input 1	Texture
VIDEO2	Input 2	Inactive
VIDEO3	Clip Channel 1	Inactive
VIDEO4	Clip Channel 2	Inactive
VIDEO5	Input 3	Inactive
VIDEO6	Input 4	Inactive
	FG Graphics	Active

Target: Inactive | DVE | Texture
Format: RGB | RGBA

Position X: 0.0 % — 0 p
Position Y: 0.0 % — 0 p
Size X: 100.0 % — 720 p
Size Y: 100.0 % — 576 p
Lock x/y
Alpha: 100.0 %
Priority: 10

Crop Left: 0.0 % — 0 p
Crop Right: 0.0 % — 0 p
Crop Top: 0.0 % — 0 p
Crop Bottom: 0.0 % — 0 p

Audio Volume: 100.0 %

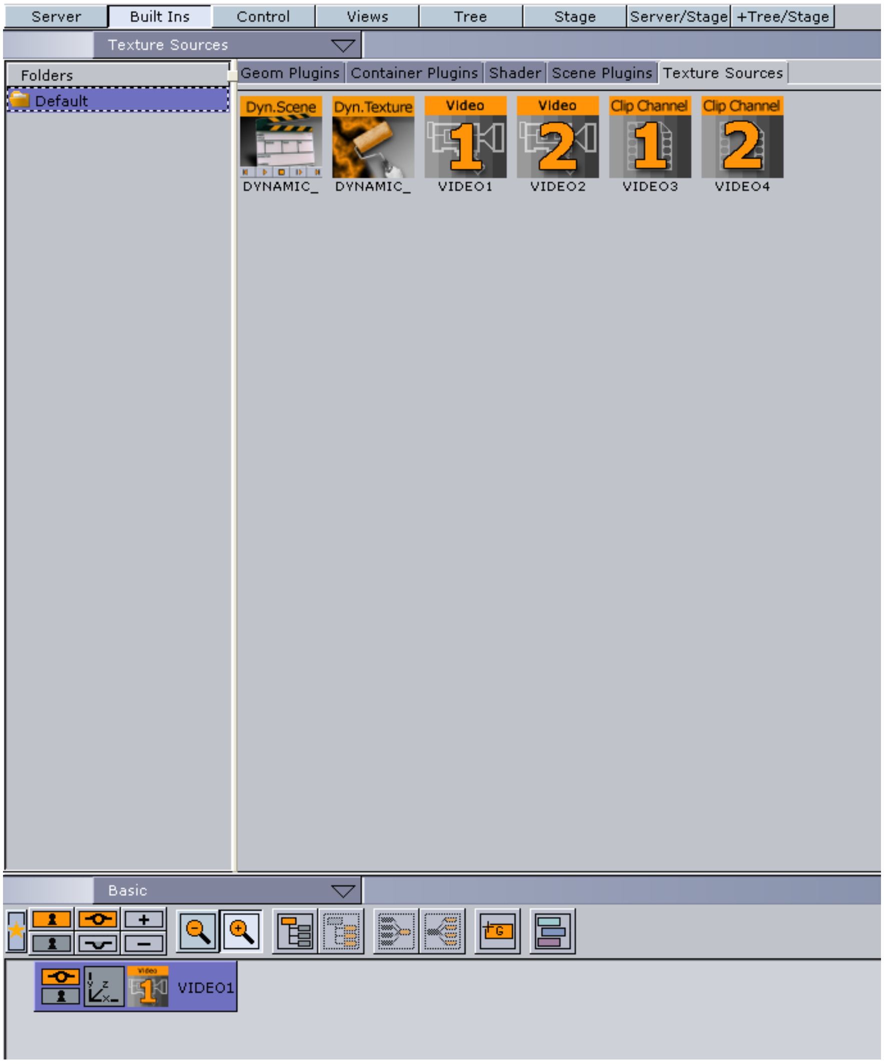
Keying Mode: Linear | Chroma | M-Zone | Invert

   - You can now do to the video anything you would do with a texture.

Tip: If you drag the Video1 plugin straight into the scene tree, the Texture button will be active by default.

To implement external video (Texture/DVS)

1. Click **Built Ins**.
2. In the drop-down menu underneath, select **Texture Sources**.
3. Drag and drop the **Video1** plugin into the scene tree.

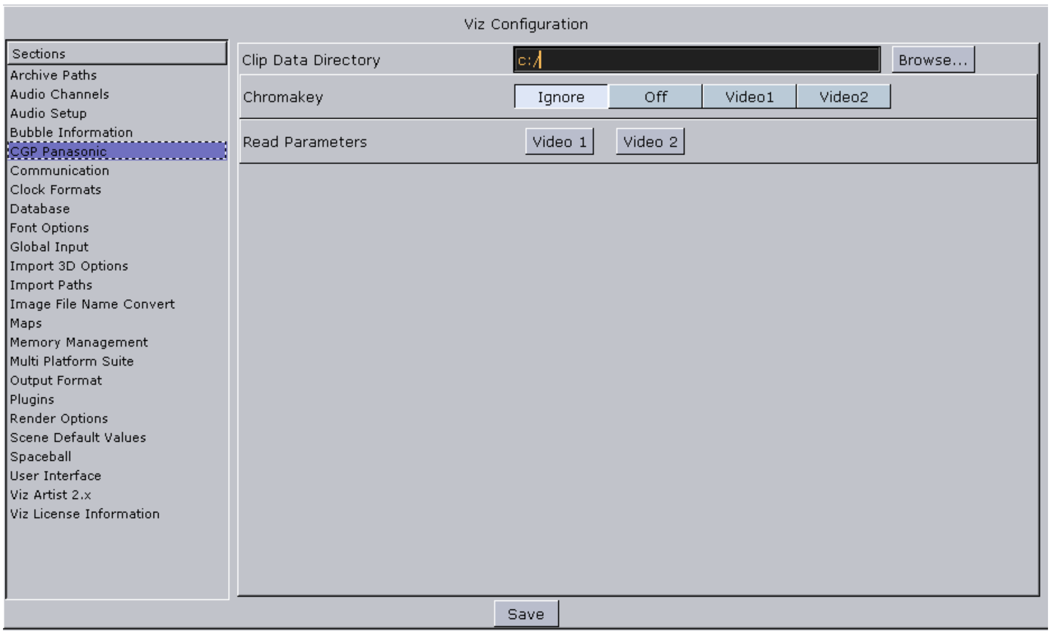


- You can now do to the video anything you would do with a texture.

Tip: If you drag the Video1 plugin straight into the scene tree, the Texture button will be active by default.

To implement external video (Texture/CGP Panasonic)

1. Click **Config** and in the left pane, select **CGP Panasonic.**
2. In the Chroma Key row, select one of the following to set the mode:
   - **Ignore** Does nothing.
   - **Off** Disables chroma key for both Video-In 1 and Video-In 2
   - **Video 1** Enables chroma key on Video-In 1, disables chroma key on Video-In 2.
   - **Video 2** Enables chroma key on Video-In 2, disables chroma key on Video-In 1.
3. In the Read Parameters row, select which video source you want to read the chroma key parameters from.
   - **Video 1** Reads chroma keyer settings from Video-In 1.
   - **Video 2** Reads chroma keyer settings from Video-In 2.



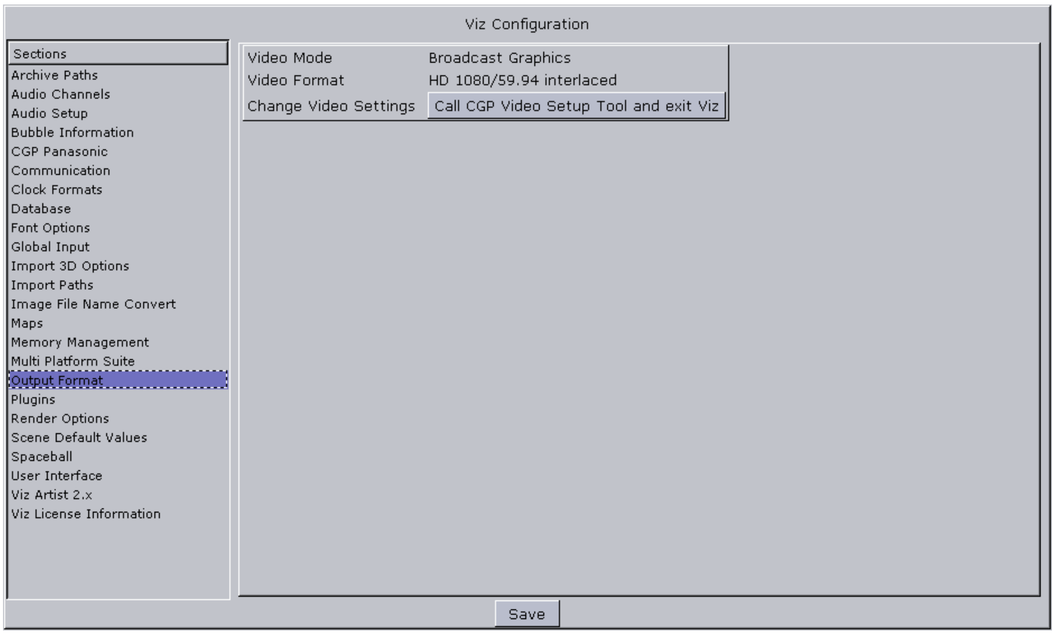
Clip Data Directory c:/

Chromakey Ignore

Read Parameters

4. In the left pane, click **Output Format**.
5. Click **Save**.
6. Click **Call CGP Video Setup Tool and Exit Viz**.

   The CGP setup tool is launched. To configure it, consult the relevant Panasonic documentation.

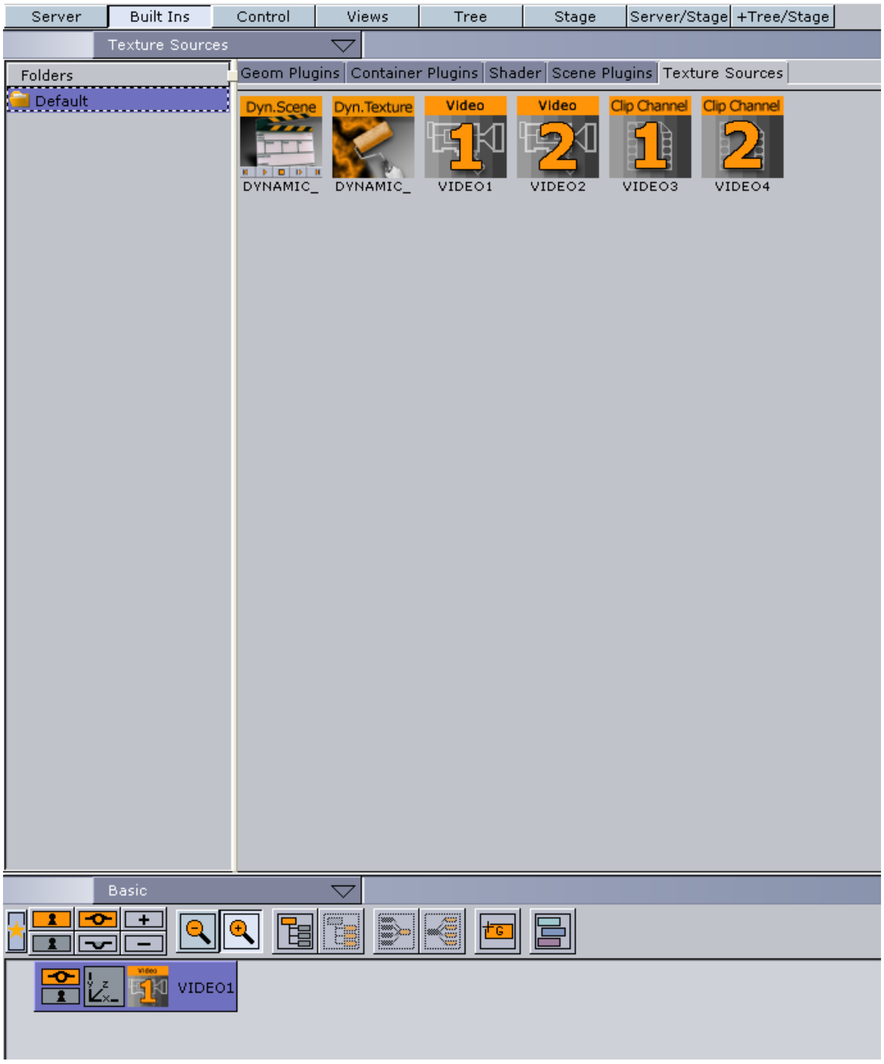


Video Mode Broadcast Graphics

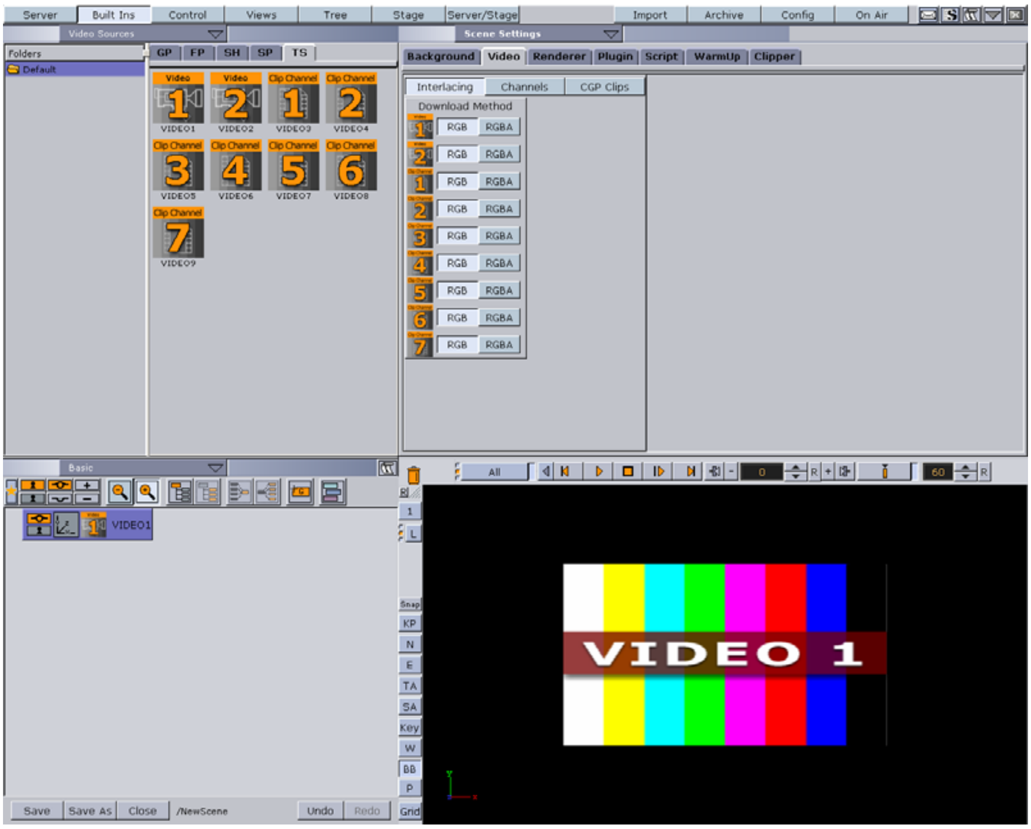
Video Format HD 1080/59.94 interlaced

Change Video Settings

7. Restart Viz Artist.
8. Click **Built Ins**.
9. In the drop-down menu underneath, select **Texture Sources**.
10. Drag and drop the **Video1** plugin into the scene tree.



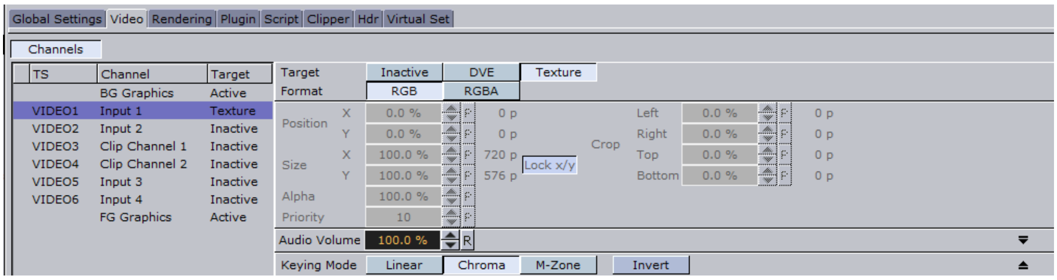
11. Click **Server**.
12. In the right pane, from the pull-down menu, select **Scene Settings** and click the **Video** tab.
13. In the left pane of the Video parameters, click the **Interlacing** button, and select either **RGB** or **RGBA**.



- You can now do to the video anything you would do with a texture.

To implement external video (Texture/Targa)

1. Click **Server**.
2. In the right pane, from the pull-down menu, select **Scene Settings** and click the **Video** tab.
3. In the left pane of the Video parameters, click **Input 1**.
4. In the right pane of the Video parameters, in the Target row, click the **Texture** button.
5. Select the keying mode.
   - **Linear** No chroma keying.
   - **Chroma** Uses the internal chroma keyer.
   - **M-Zone** Uses the multi-zone keyer.
   - **Invert** Only works with Chroma.



Global Settings | Video | Rendering | Plugin | Script | Clipper | Hdr | Virtual Set

Channels

TS	Channel	Target
	BG Graphics	Active
VIDEO1	Input 1	Texture
VIDEO2	Input 2	Inactive
VIDEO3	Clip Channel 1	Inactive
VIDEO4	Clip Channel 2	Inactive
VIDEO5	Input 3	Inactive
VIDEO6	Input 4	Inactive
	FG Graphics	Active

Target: Inactive | DVE | Texture
Format: RGB | RGBA

Position X 0.0 % 0 p
Position Y 0.0 % 0 p
Size X 100.0 % 720 p
Size Y 100.0 % 576 p
Lock x/y
Alpha 100.0 %
Priority 10

Crop Left 0.0 % 0 p
Crop Right 0.0 % 0 p
Crop Top 0.0 % 0 p
Crop Bottom 0.0 % 0 p

Audio Volume 100.0 %

Keying Mode: Linear | Chroma | M-Zone | Invert

- You can now do to the video anything you would do with a texture.

Tip: If you drag the Video1 plugin straight into the scene tree, the Texture button will be active by default.

5.2.4 Implementing Video Through a Video Clip

Depending on sizes and formats, using clips will affect Viz Engine rendering performance. Hence, clip playback should be used sparingly.

When building the scene in Viz Artist that will support clip playback, there are three possible plugins you can optionally use:

- Image Clip
- MoViz
- Soft Clip

Consider the following general guidelines when deciding on a specific method to use for playing back video clips:

- If no 3D effects or animation control – and only limited 2D effects – on the clips are needed, use the Clip Channel through DVE.
- Full screen (e.g. background) clips should be done by the Clip Channel through DVE.
- If video must be mapped onto a texture in the scene or requires minor adjustments, use Clip Channel through Texture.
- Non-standard image format/resolution/codec, use SoftClip or MoViz plugins.
- If more than two small clips are required, use SoftClip or MoViz plugins.
- For low resolution and short animations use the ImageClip plugin.

See Also

- Clip Playback Use Cases

You should always test the performance of the different solutions and work with the solution that gives the best quality versus performance ratio. Performance testing should always take place on a machine with the same specifications as those used for production playout.

Note: Reading and decompressing a clip impacts system performance both with DVE and texture.

All Viz Artist and Viz Engine machines, as well as Vizrt clients (with preview – e.g. Viz Trio or Viz Weather), should come prepared with a pre-mapped video (V:\) drive. This drive is configured for faster access and speeds. This is where all video clips should be stored. It is not recommended to store video clips on the C:\ drive, as this is where the operating system is installed.

Viz Video Hub integrates with Viz Artist for the purposes of implementing clips for playout. To enable this, consult the *Viz Link Administrator Guide*. For actual use, consult the *Viz Video Hub User Guide*.

Video clips are implemented as follows:

- Video Clip Through DVE
- Video Clip Through Texture

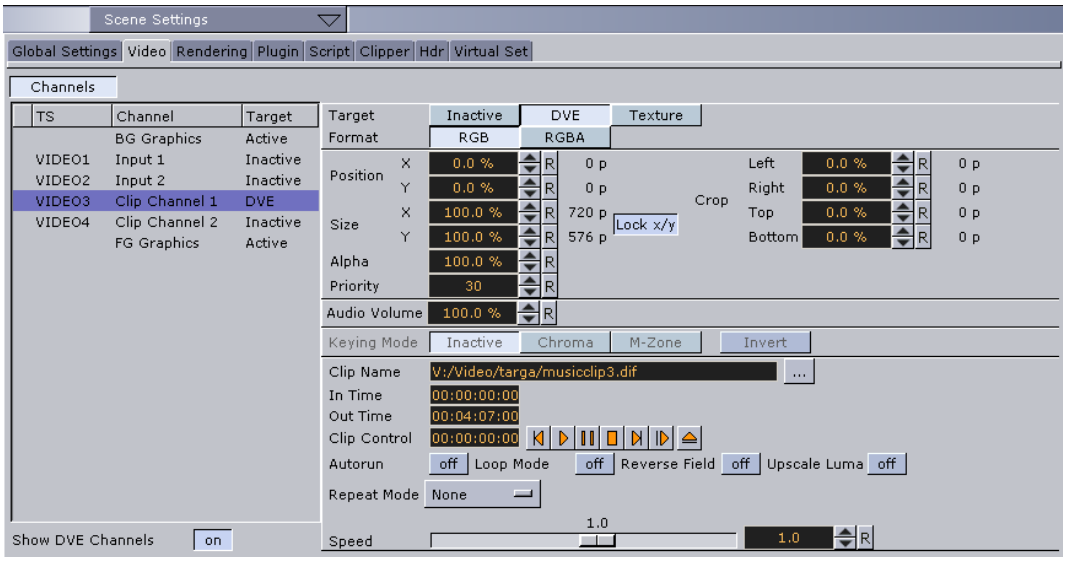
Video Clip Through DVE

This section covers the following procedures:

- To implement a video clip (DVE/Targa)
- To implement a video clip (DVE/Matrox)
- To implement a video clip (DVE/DVS)

To implement a video clip (DVE/Targa)

1. In the right pane, select **Scene Settings**, and click the **Video** tab.
2. In the left pane of the Video settings, select **Clip Channel 1**.
3. In the right pane of the Video settings, in the Target row, click the **DVE** button. You can now adjust the parameters in the DVE control panel.
   - **Position**
   - **Size**
   - Cropping
   - **Alpha** For each layer, you can adjust the level of opacity, where 100% is fully opaque and 0% is fully transparent.
   - **Priority** Assigns the position of each layer in the layer stack, where 0 is the bottom-most layer, and 100 is the top-most layer.



TS	Channel	Target
	BG Graphics	Active
VIDEO1	Input 1	Inactive
VIDEO2	Input 2	Inactive
VIDEO3	Clip Channel 1	DVE
VIDEO4	Clip Channel 2	Inactive
	FG Graphics	Active

Show DVE Channels: on

Target: Inactive | DVE | Texture
Format: RGB | RGBA

Position X: 0.0 % R 0 p
Position Y: 0.0 % R 0 p
Size X: 100.0 % R 720 p
Size Y: 100.0 % R 576 p
Lock x/y
Alpha: 100.0 % R
Priority: 30 R

Crop
Left: 0.0 % R 0 p
Right: 0.0 % R 0 p
Top: 0.0 % R 0 p
Bottom: 0.0 % R 0 p

Audio Volume: 100.0 % R
Keying Mode: Inactive | Chroma | M-Zone | Invert

Clip Name: V:/Video/targa/musicclip3.dif ...
In Time: 00:00:00:00
Out Time: 00:04:07:00
Clip Control: 00:00:00:00
Autorun: off
Loop Mode: off
Reverse Field: off
Upscale Luma: off
Repeat Mode: None
Speed: 1.0 1.0 R

Note: BG Graphics are always the lowest in the stack, even below zero, and FG graphics are assigned a default of 100. As such, you cannot assign a priority to BG Graphics.

4. In the right pane of the Video settings, click the ellipsis (...) to select a video clip for import.

IMPORTANT! Ensure that the clip you select has the same resolution as the channel in question.

IMPORTANT! You must select a codec that the Targa board supports. See Targa Codecs.

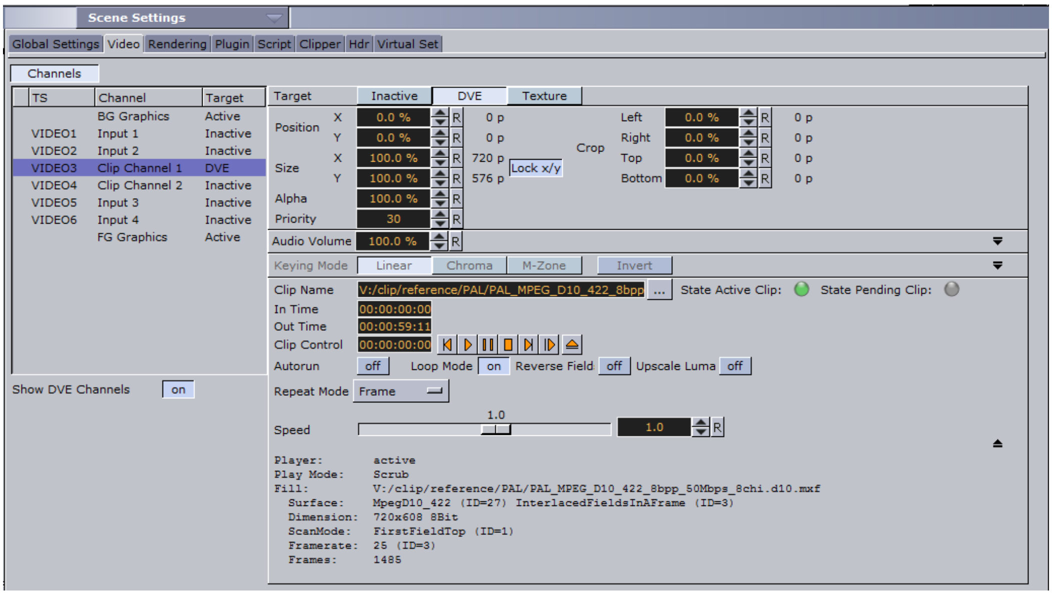
5. In the right pane of the Video settings, in the Alpha row, select either **RGB** or **RGBA**.
   - **RGB** The clip is taken as-is, meaning, it is used as fill.
   - **RGBA** You need to implement two clips; the first is the fill, and second is the key clip. If you only implement one clip, the key will be created dynamically and be fully opaque. The naming convention is that the alpha/key clip must have the same name with an attached `_key` before the file extension, this means if the main/fill clip is named `example.dif`, the alpha/key clip must be named as follows: `example_key.dif`.
6. In the Clip Control row, click the Play button to test your video clip.

To implement a video clip (DVE/Matrox)

1. Click **Config** and configure the alpha in Video Input.

Tip: w/Alpha Enable this setting when you need play clips with alpha. You will therefore either need two clips, or one clip that contains both the fill and key. If you only implement one clip containing the fill (without a key), the key will be created dynamically and be fully opaque. For the naming convention for two clips, the alpha/key clip must have the same name with an attached `_key` before the file extension, this means if the main/fill clip is named `example.dif`, the alpha/key clip must be named as follows: `example_key.dif`. Check also the Supported Codecs.

2. Save and restart Viz Artist.
3. Click **Server**.
4. In the right pane, select **Scene Settings**, and click the **Video** tab.
5. In the left pane of the Video settings, select **Clip Channel 1**.
6. In the right pane of the Video settings, in the Target row, click the **DVE** button. You can now adjust the parameters in the DVE control panel.
   - **Position**
   - **Size**
   - Cropping
   - **Alpha** For each layer, you can adjust the level of opacity, where 100% is fully opaque and 0% is fully transparent.
   - **Priority** Assigns the position of each layer in the layer stack, where 0 is the bottom-most layer, and 100 is the top-most layer.



TS	Channel	Target
	BG Graphics	Active
VIDEO1	Input 1	Inactive
VIDEO2	Input 2	Inactive
VIDEO3	Clip Channel 1	DVE
VIDEO4	Clip Channel 2	Inactive
VIDEO5	Input 3	Inactive
VIDEO6	Input 4	Inactive
	FG Graphics	Active

Show DVE Channels: on

Target: Inactive | DVE | Texture

Position	X	0.0 %	0 p	Crop	Left	0.0 %	0 p
	Y	0.0 %	0 p		Right	0.0 %	0 p
Size	X	100.0 %	720 p	Lock x/y	Top	0.0 %	0 p
	Y	100.0 %	576 p		Bottom	0.0 %	0 p
Alpha		100.0 %					
Priority		30					

Audio Volume: 100.0 %

Keying Mode: Linear | Chroma | M-Zone | Invert

Clip Name: V:/clip/reference/PAL/PAL_MPEG_D10_422_8bpp ... State Active Clip: ● State Pending Clip: ●

In Time: 00:00:00:00

Out Time: 00:00:59:11

Clip Control: 00:00:00:00

Autorun: off Loop Mode: on Reverse Field: off Upscale Luma: off

Repeat Mode: Frame

Speed: 1.0 1.0

```
Player:       active
Play Mode:    Scrub
Fill:         V:/clip/reference/PAL/PAL_MPEG_D10_422_8bpp_50Mbps_8chi.d10.mxf
  Surface:    MpegD10_422 (ID=27) InterlacedFieldsInAFrame (ID=3)
  Dimension:  720x608 8Bit
  ScanMode:   FirstFieldTop (ID=1)
  Framerate:  25 (ID=3)
  Frames:     1485
```

Note: BG Graphics are always the lowest in the stack, even below zero, and FG graphics are assigned a default of 100. As such, you cannot assign a priority to BG Graphics.

7. In the right pane of the Video settings, click the ellipsis (...) to select a video clip for import.

IMPORTANT! Ensure that the clip you select has the same resolution as the channel in question.

IMPORTANT! You must select a proper container and codec. Check the Supported Codecs.

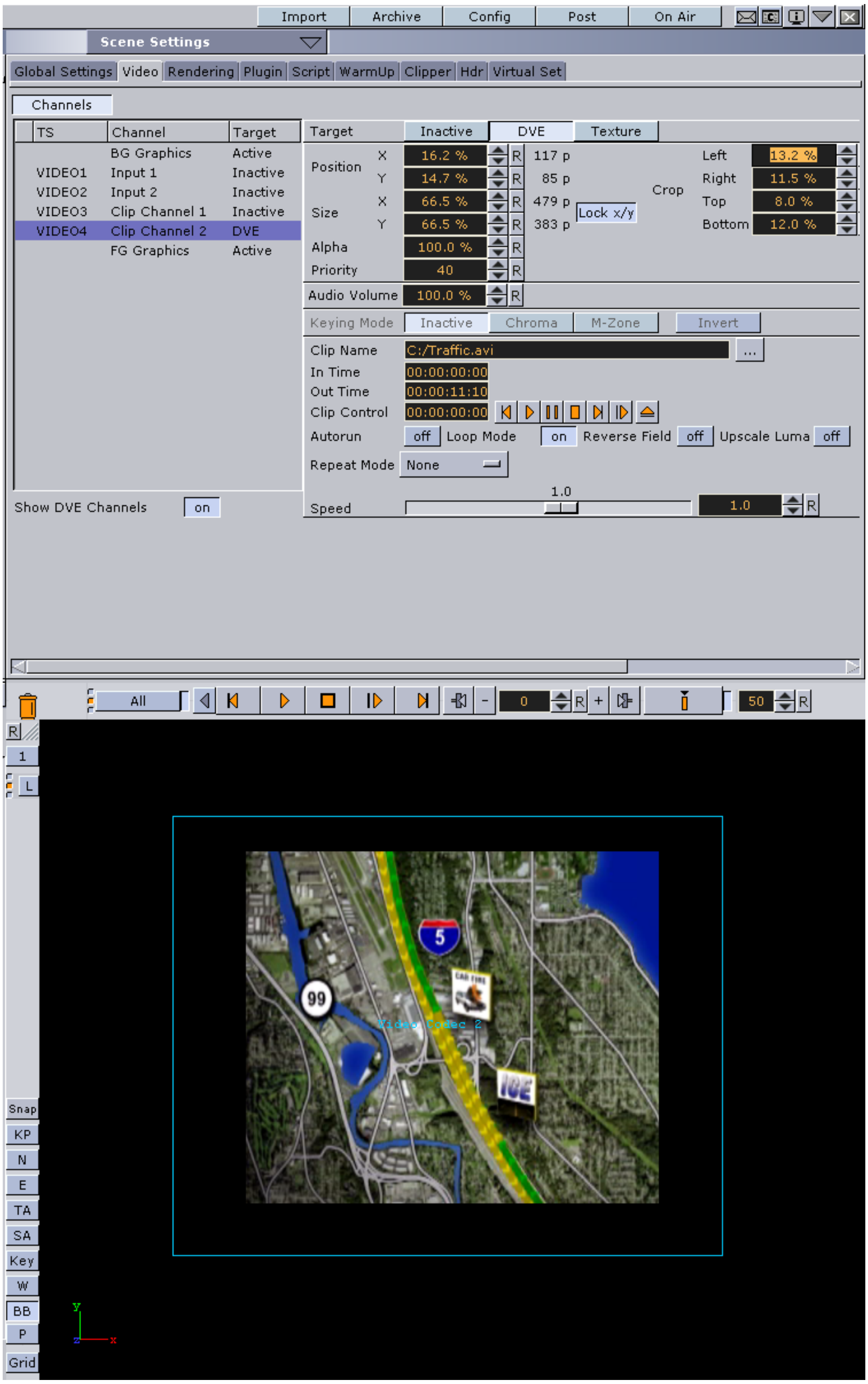
8. Configure the playback behavior as follows:
   - **Autorun** - **Off** means that as soon as the clip is loaded, the first frame is presented. **On** means that as soon as the clip is loaded, it begins playing.
   - **Loop Mode** - **Off** means that the clip will play only once. **On** means that the clip will play in a loop.
   - **Reverse Field** - **Off** means that when using backward playback (i.e. speed is negative), the fields in interlaced mode are played out as-is. **On** means the first as second fields in interlaced mode are swapped.
   - **Upscale Luma** - **Off** means that the key will be used as-is. **On** means that the key information will be upscaled from 16-235 to 0-255.
   - **Repeat Mode** - **None** means the playout of black frames when the clip is in an erroneous state. **Field** means that the first field of the last frame will be played out when the clip is paused. **Frame** means that the last frame will be played out when the clip is paused.
   - **Speed** - Controls playback speed. 1.0 is the normal speed. Values between 0 and 1.0 are forward playback in slow motion. Values above 1.0 are forward playback in fast motion. Likewise, values between 0 and -1.0 are reverse playback in slow motion. Values less than -1.0 are reverse playback in fast motion.
9. In the Clip Control row, click the Play button to test your video clip.

A green ‘traffic light’ indicates that the clip has been loaded successfully. A grey ‘traffic light’ indicates that no clip has been loaded. A red ‘traffic light’ indicates that there was an error during the clip loading process.

Clicking the expand button displays addition information. If the clip loaded successfully, clip details are shown. If the clip did not load successfully, additional error indications are shown.

To implement a video clip (DVE/DVS)

1. If you intend to have audio, you must first enable it by modifying the Viz configuration file. Close Viz Artist and in Windows Explorer, browse to `C:\Program Files\Vizrt\Viz3` and open the **VIZ-*<computer name>*.cfg** file in the text editor of your choice.
2. Ensure that the `EnableDirectShowAudio` flag = **1**.
3. Ensure that the `clip_load_wouldnt_block` flag = **0**.
4. Save and close the config file and restart Viz Artist.
5. In the left pane of the Video settings, select **Clip Channel 1**.
6. In the right pane of the Video settings, in the Target row, click the **DVE** button. You can now adjust the parameters in the DVE control panel.
   - **Position**
   - **Size**
   - Cropping
   - **Alpha** For each layer, you can adjust the level of opacity, where 100% is fully opaque and 0% is fully transparent.
   - **Priority** Assigns the position of each layer in the layer stack, where 0 is the bottom-most layer, and 100 is the top-most layer.
7. In the right pane, select **Scene Settings**, and click the **Video** tab.



TS	Channel	Target
	BG Graphics	Active
VIDEO1	Input 1	Inactive
VIDEO2	Input 2	Inactive
VIDEO3	Clip Channel 1	Inactive
VIDEO4	Clip Channel 2	DVE
	FG Graphics	Active

Show DVE Channels: on

Target: Inactive | DVE | Texture

Position X: 16.2 % (117 p)
Position Y: 14.7 % (85 p)
Size X: 66.5 % (479 p)
Size Y: 66.5 % (383 p)
Lock x/y
Alpha: 100.0 %
Priority: 40

Crop Left: 13.2 %
Crop Right: 11.5 %
Crop Top: 8.0 %
Crop Bottom: 12.0 %

Audio Volume: 100.0 %

Keying Mode: Inactive | Chroma | M-Zone | Invert

Clip Name: C:/Traffic.avi ...
In Time: 00:00:00:00
Out Time: 00:00:11:10
Clip Control: 00:00:00:00
Autorun: off
Loop Mode: on
Reverse Field: off
Upscale Luma: off
Repeat Mode: None
Speed: 1.0

Note: BG Graphics are always the lowest in the stack, even below zero, and FG graphics are assigned a default of 100. As such, you cannot assign a priority to BG Graphics.

8. In the right pane of the Video settings, click the ellipsis (...) to select a video clip for import.

IMPORTANT! You must select a video format that works in Windows Media Player. If you want to use Matrox codecs, see Advanced Issues with Video Codecs.

9. In the Clip Control row, click the Play button to test your video clip.

Video Clip Through Texture

This section covers the following procedures:

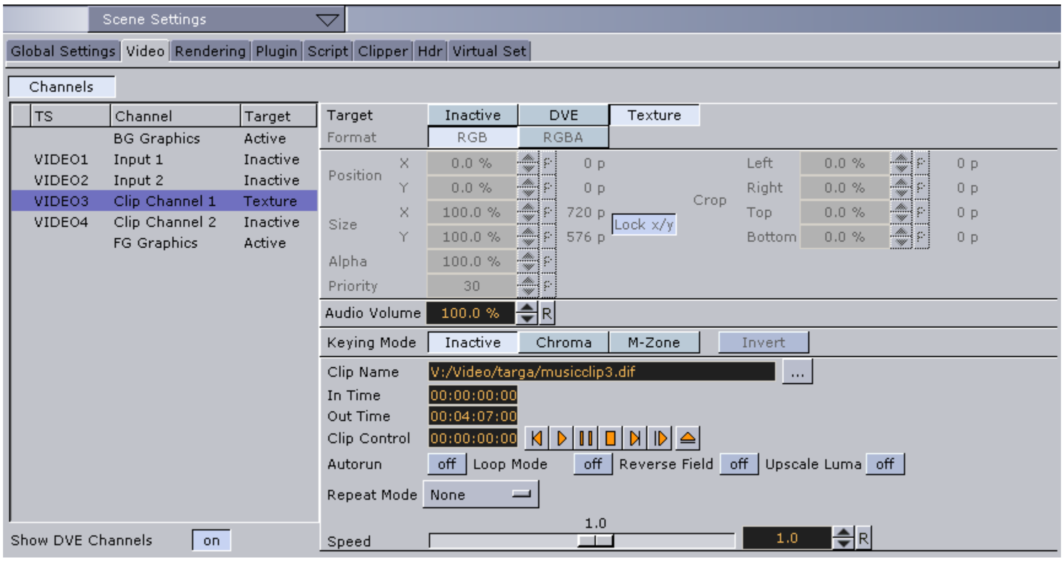
- To implement a video clip (Texture/Targa)
- To implement a video clip (Texture/Matrox)
- To implement a video clip (Texture/DVS)
- To implement a video clip (Texture/CGP Panasonic)

To implement a video clip (Texture/Targa)

1. In the right pane, select **Scene Settings**, and click the **Video** tab.
2. In the left pane of the Video settings, select **Clip Channel 1**.
3. In the right pane of the Video settings, in the Target row, click the **Texture** button.
4. In the right pane of the Video settings, click the ellipsis (...) to select a video clip for import.

IMPORTANT! Ensure that the clip you select has the same resolution as the channel in question.

IMPORTANT! You must select a codec that the Targa board supports. See Targa Codecs.



TS	Channel	Target
	BG Graphics	Active
VIDEO1	Input 1	Inactive
VIDEO2	Input 2	Inactive
VIDEO3	Clip Channel 1	Texture
VIDEO4	Clip Channel 2	Inactive
	FG Graphics	Active

Audio Volume 100.0 %

Clip Name V:/Video/targa/musicclip3.dif

In Time 00:00:00:00

Out Time 00:04:07:00

Clip Control 00:00:00:00

Autorun off Loop Mode off Reverse Field off Upscale Luma off

Repeat Mode None

Speed 1.0

Note: BG Graphics are always the lowest in the stack, even below zero, and FG graphics are assigned a default of 100. As such, you cannot assign a priority to BG Graphics.

5. In the Clip Control row, click the Play button to test your video clip. You can now do to the video anything you would do with a texture.

To implement a video clip (Texture/Matrox)

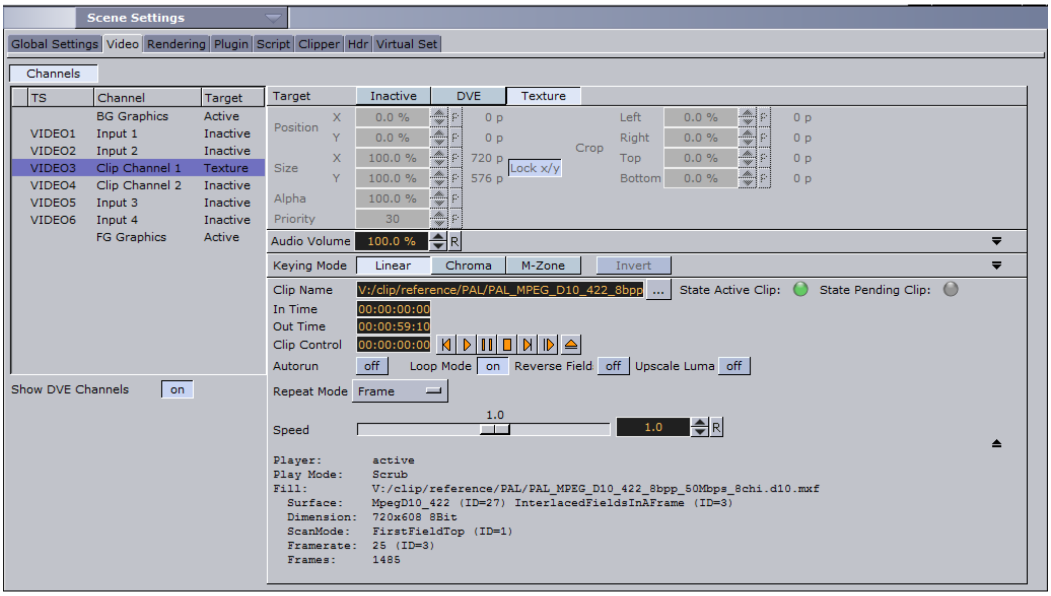
1. Click **Config** and configure the alpha in Video Input.

Tip: **w/Alpha** Enable this setting when you need play clips with alpha. You will therefore either need two clips, or one clip that contains both the fill and key. If you only implement one clip containing the fill (without a key), the key will be created dynamically and be fully opaque. For the naming convention for two clips, the alpha/key clip must have the same name with an attached `_key` before the file extension, this means if the main/fill clip is named `example.dif`, the alpha/key clip must be named as follows: `example_key.dif`. Also check the Supported Codecs.

2. Save and restart Viz Artist.
3. Click **Server**.
4. In the right pane, select **Scene Settings**, and click the **Video** tab.
5. In the left pane of the Video settings, select **Clip Channel 1**.
6. In the right pane of the Video settings, in the Target row, click the **Texture** button.
7. In the right pane of the Video settings, click the ellipsis (...) to select a video clip for import.

IMPORTANT! Ensure that the clip you select has the same resolution as the channel in question.

IMPORTANT! You must select a proper container and codec. Check the Supported Codecs.



TS	Channel	Target
	BG Graphics	Active
VIDEO1	Input 1	Inactive
VIDEO2	Input 2	Inactive
VIDEO3	Clip Channel 1	Texture
VIDEO4	Clip Channel 2	Inactive
VIDEO5	Input 3	Inactive
VIDEO6	Input 4	Inactive
	FG Graphics	Active

Show DVE Channels: on

Target: Inactive | DVE | Texture

Position X: 0.0 % 0 p
Position Y: 0.0 % 0 p
Size X: 100.0 % 720 p
Size Y: 100.0 % 576 p
Lock x/y
Alpha: 100.0 %
Priority: 30

Crop Left: 0.0 % 0 p
Crop Right: 0.0 % 0 p
Crop Top: 0.0 % 0 p
Crop Bottom: 0.0 % 0 p

Audio Volume: 100.0 %

Keying Mode: Linear | Chroma | M-Zone | Invert

Clip Name: V:/clip/reference/PAL/PAL_MPEG_D10_422_8bpp ... State Active Clip: ● State Pending Clip: ●
In Time: 00:00:00:00
Out Time: 00:00:59:10
Clip Control: 00:00:00:00
Autorun: off Loop Mode: on Reverse Field: off Upscale Luma: off
Repeat Mode: Frame
Speed: 1.0

```
Player:       active
Play Mode:    Scrub
Fill:         V:/clip/reference/PAL/PAL_MPEG_D10_422_8bpp_50Mbps_8chi.d10.mxf
  Surface:    MpegD10_422 (ID=27) InterlacedFieldsInAFrame (ID=3)
  Dimension:  720x608 8Bit
  ScanMode:   FirstFieldTop (ID=1)
  Framerate:  25 (ID=3)
  Frames:     1485
```

Note: BG Graphics are always the lowest in the stack, even below zero, and FG graphics are assigned a default of 100. As such, you cannot assign a priority to BG Graphics.

8. In the Clip Control row, click the Play button to test your video clip.

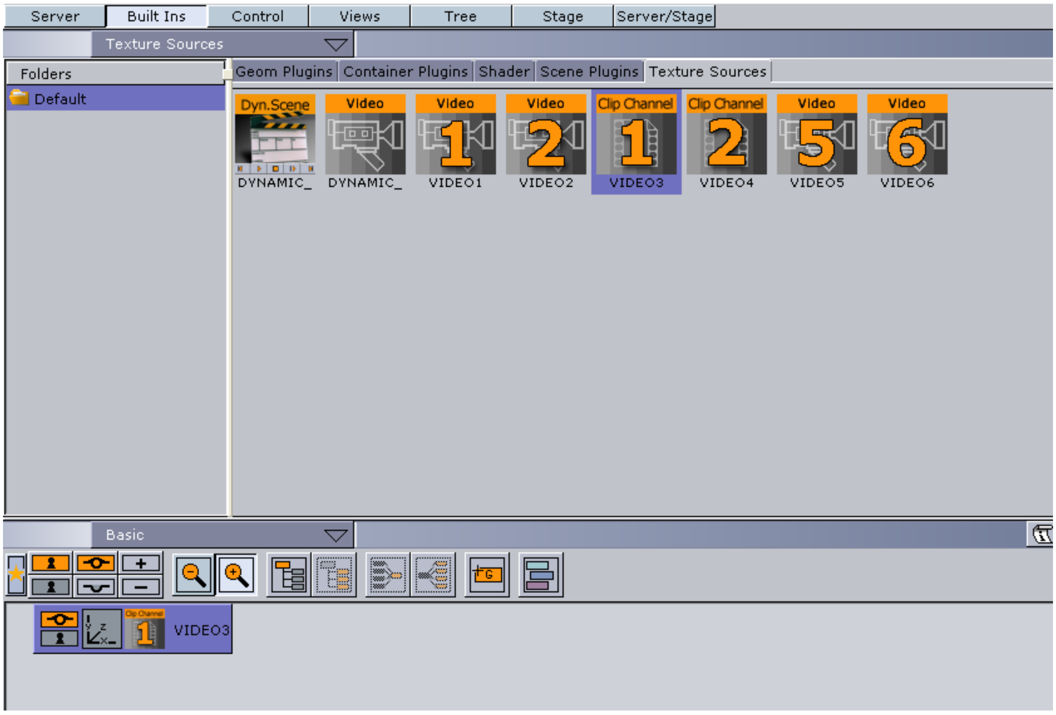
A green ‘traffic light’ indicates that the clip has been loaded successfully. A grey ‘traffic light’ indicates that no clip has been loaded. A red ‘traffic light’ indicates that there was an error during the clip loading process.

Clicking the expand button displays addition information. If the clip loaded successfully, clip details are shown. If the clip did not load successfully, additional error indications are shown.

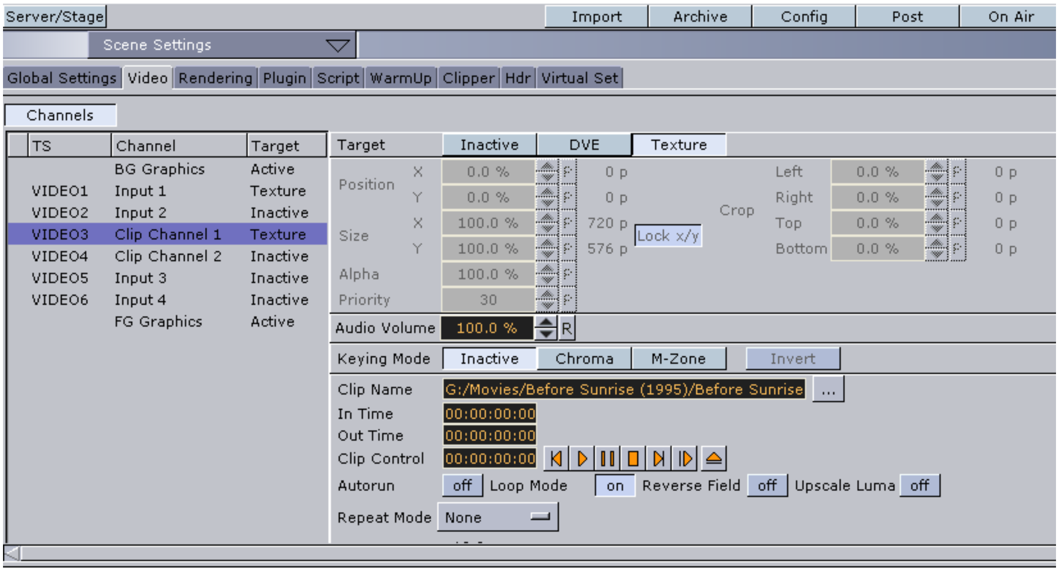
Once loaded you can do to the video anything you would do with a texture.

To implement a video clip (Texture/DVS)

1. If you intend to have audio, you must first enable it by modifying the Viz configuration file. Close Viz Artist and in Windows Explorer, browse to `C:\Program Files\Vizrt\Viz3` and open the **VIZ-***<computer name>***.cfg** file in the text editor of your choice.
2. Ensure that the `EnableDirectShowAudio` flag = 1.
3. Ensure that the `clip_load_wouldnt_block` flag = **0.**
4. Save and close the config file and restart Viz Artist.
5. Click **Built Ins.**
6. In the drop-down menu underneath, select **Texture Sources.**
7. Drag and drop the **Clip Channel 1** plugin into the scene tree.



8. In the right pane, select **Scene Settings**, and click the **Video** tab.



TS	Channel	Target
	BG Graphics	Active
VIDEO1	Input 1	Texture
VIDEO2	Input 2	Inactive
VIDEO3	Clip Channel 1	Texture
VIDEO4	Clip Channel 2	Inactive
VIDEO5	Input 3	Inactive
VIDEO6	Input 4	Inactive
	FG Graphics	Active

Position X 0.0 % 0 p
Position Y 0.0 % 0 p
Size X 100.0 % 720 p
Size Y 100.0 % 576 p
Alpha 100.0 %
Priority 30
Crop Left 0.0 % 0 p
Crop Right 0.0 % 0 p
Crop Top 0.0 % 0 p
Crop Bottom 0.0 % 0 p
Audio Volume 100.0 % R
Clip Name G:/Movies/Before Sunrise (1995)/Before Sunrise
In Time 00:00:00:00
Out Time 00:00:00:00
Clip Control 00:00:00:00
Autorun off
Loop Mode on
Reverse Field off
Upscale Luma off
Repeat Mode None

9. In the left pane of the Video settings, select **Clip Channel 1**.
10. In the right pane of the Video settings, click the ellipsis (...) to select a video clip for import.

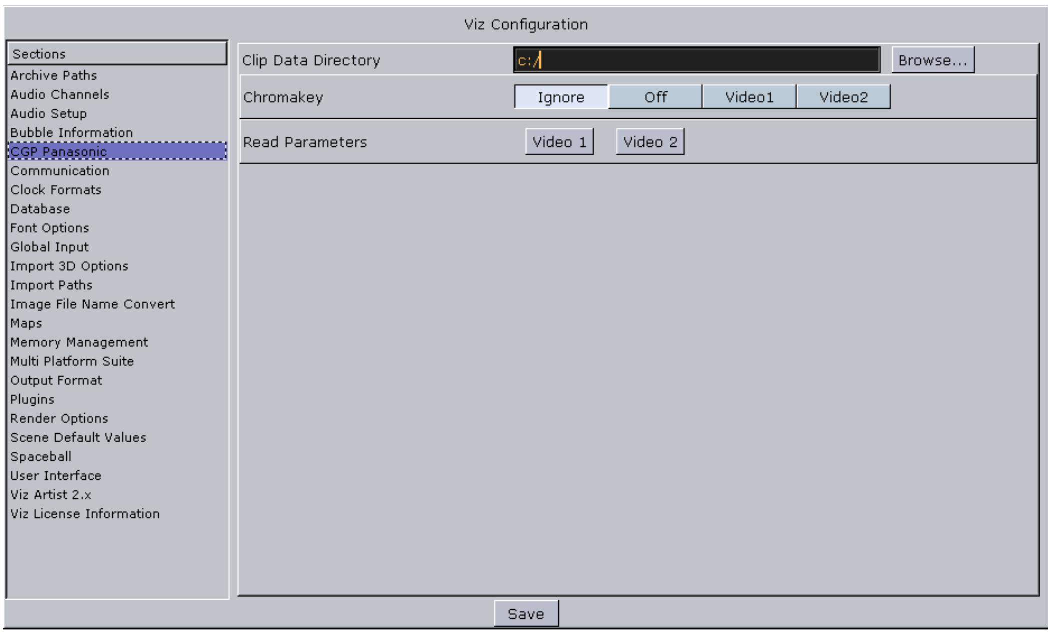
IMPORTANT! You must select a video format that works in Windows Media Player.

11. In the Clip Control row, click the Play button to test your video clip.

    You can now do to the video anything you would do with a texture.

To implement a video clip (Texture/CGP Panasonic)

1. Click **Config** and in the left pane, select **CGP Panasonic**.
2. In the Clip Data Directory field, enter the folder path where your CGP clips are stored.
3. In the Chroma Key row, select one of the following to set the mode:
    - **Ignore** Does nothing.
    - **Off** Disables chroma key for both Video-In 1 and Video-In 2
    - **Video 1** Enables chroma key on Video-In 1, disables chroma key on Video-In 2.
    - **Video 2** Enables chroma key on Video-In 2, disables chroma key on Video-In 1.
4. In the Read Parameters row, select which video source you want to read the chroma key parameters from.
    - **Video 1** Reads chroma keyer settings from Video-In 1.
    - **Video 2** Reads chroma keyer settings from Video-In 2.



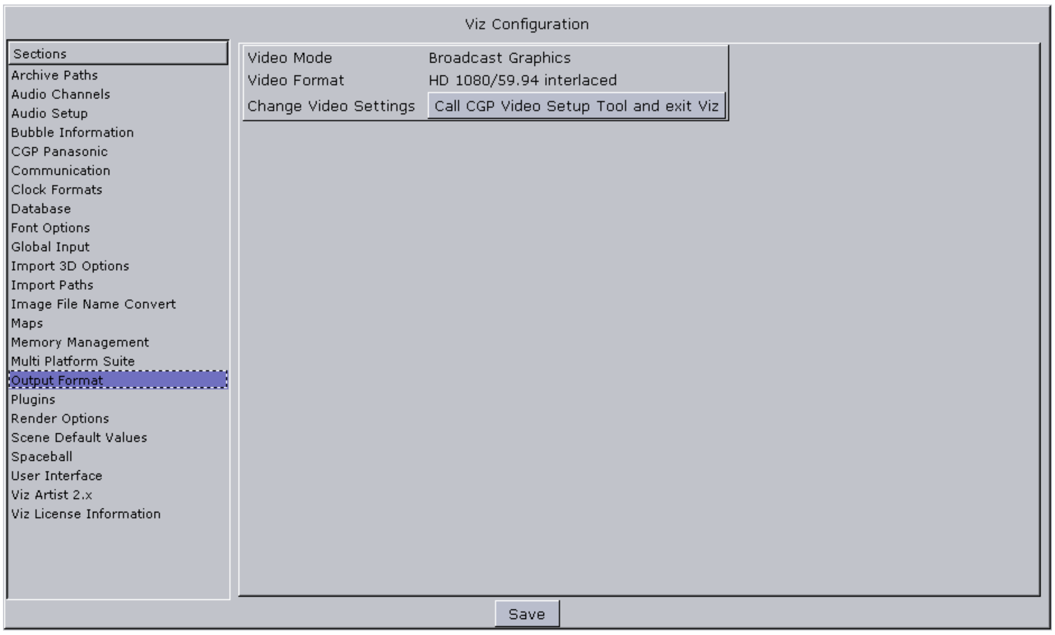
Clip Data Directory c:/

Chromakey [✓ Ignore] Off Video1 Video2

Read Parameters Video 1 Video 2

5. In the left pane, click **Output Format**.
6. Click **Save**.
7. Click **Call CGP Video Setup Tool and Exit Viz**.

   The CGP setup tool is launched. To configure it, consult the relevant Panasonic documentation.

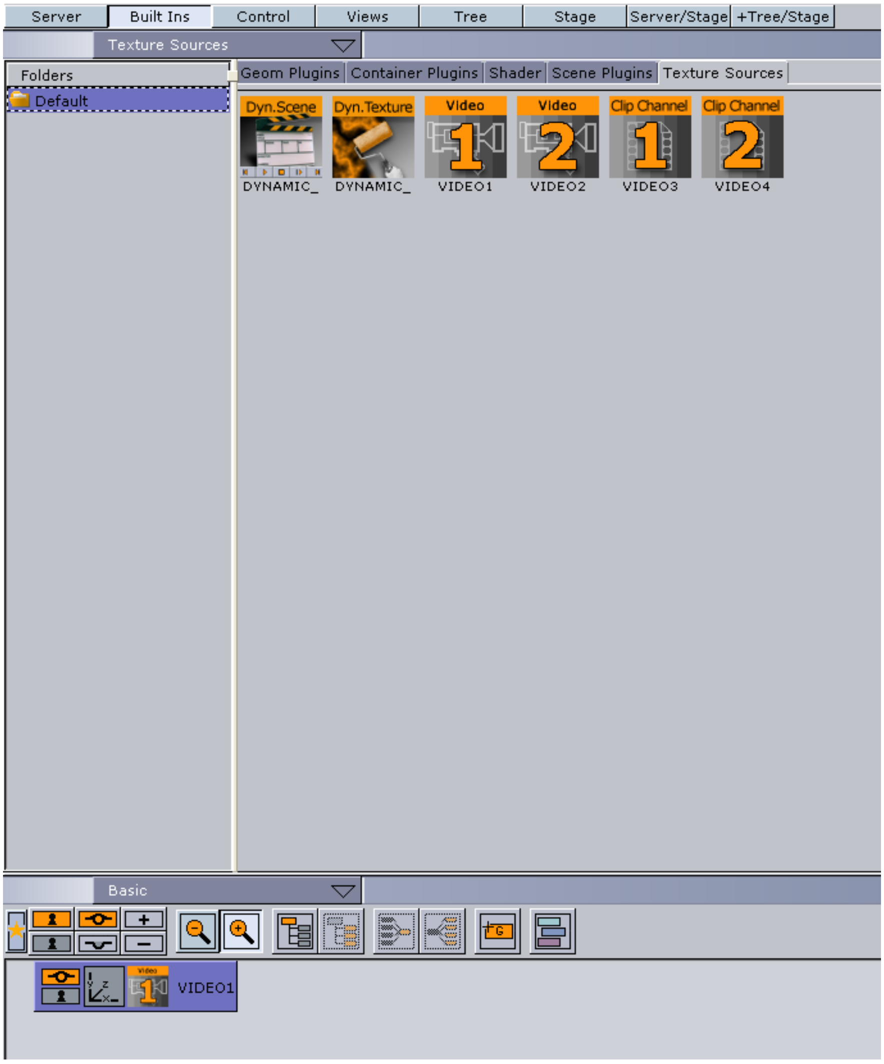


Video Mode Broadcast Graphics

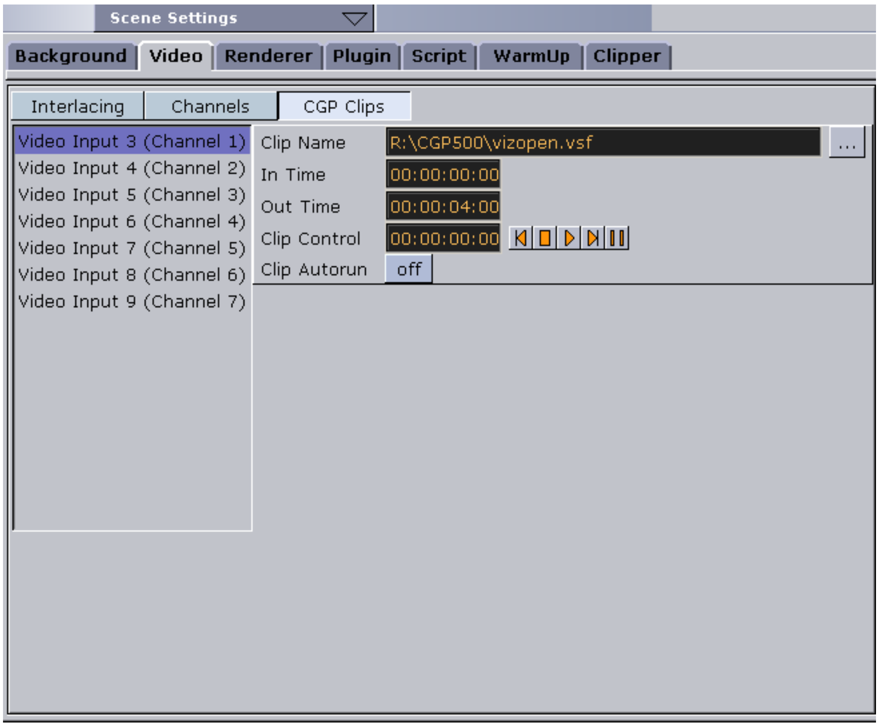
Video Format HD 1080/59.94 interlaced

Change Video Settings Call CGP Video Setup Tool and exit Viz

8. Restart Viz Artist.
9. Click **Built Ins**.
10. In the drop-down menu underneath, select **Texture Sources**.
11. Drag and drop the **Clip Channel 1** plugin into the scene tree.



12. Click **Server**.
13. In the right pane, from the pull-down menu, select **Scene Settings** and click the **Video** tab.
14. In the left pane of the Video parameters, click the **Interlacing** button, and select either **RGB** or **RGBA**.
15. Click the **CGP Clips** button.
16. In the right pane of the Video settings, click the ellipsis (...) to select a video clip for import.
17. In the Clip Control row, click the Play button to test your video clip.


Scene Settings
Background
Video
Renderer
Plugin
Script
WarmUp
Clipper
Interlacing
Channels
CGP Clips
Video Input 3 (Channel 1)
Video Input 4 (Channel 2)
Video Input 5 (Channel 3)
Video Input 6 (Channel 4)
Video Input 7 (Channel 5)
Video Input 8 (Channel 6)
Video Input 9 (Channel 7)
...

Clip Name R:\CGP500\vizopen.vsf
In Time 00:00:00:00
Out Time 00:00:04:00
Clip Control 00:00:00:00
Clip Autorun off

- You can now do to the video anything you would do with a texture.

Clip Animation in the Stage

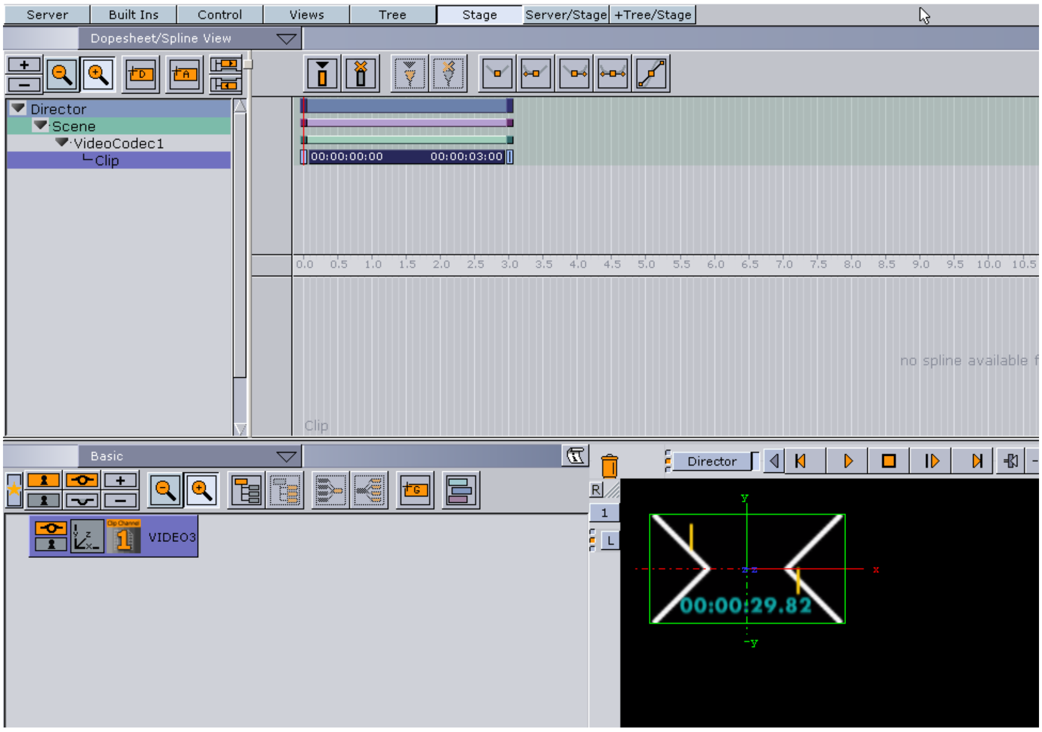
You can also implement video as part of an animation. This is useful when clips are required to interact with animation.

Before you can animate a clip, it must first be implemented in one of the following ways:

- Video Clip Through DVE
- Video Clip Through Texture

To animate a clip

1. In the main menu, click **Stage**.
2. In the far left upper panel, click the new director button, marked by a **D**.
3. Right-click the **Director** that appears below and select **Add Scene**.
4. Right-click the Scene that appears below, and from the context menus, select **VideoCodec1** (or **VideoCodec2**, depending on the desired channel) and then select **Clip**.



5. In the keyframe that appears to the right, click the dark blue area that is in the same row as the **Clip** in the director.

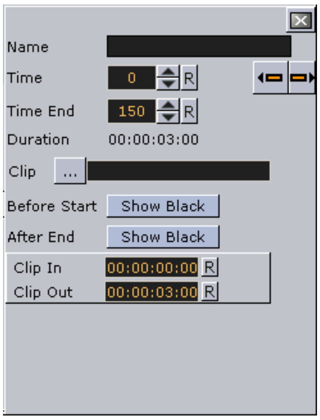
   The director's editor appears at the far right.

6. In the director's editor, click the ellipsis (...) and browse and select the desired video clip for the animation.

Note: This does not necessarily have to be the clip enabled in the previous procedures. It can be any video clip that Viz Artist can play.

IMPORTANT! If there is a conflict, the last clip selected, whether in the Stage or the Scene Settings, will be the one to play, with overriding effect.

7. Configure the director's settings as follows:
   - **Name** of keyframe
   - **Time** is the start time
   - **End time**
   - **Duration** is the interval between the start and end times
   - **Clip** is the clip path and name
   - **Before Start** defines what should happen before the clip starts to play within the animation. If the button is disabled, it shows the in-frame. If the button is enabled, it shows black until the animation gets to the clip.
   - **After End** defines what should happen after the animation leaves the clip key frame. If the button is disabled, it shows the out-frame. If the button is enabled, it shows black.
   - **Clip In** defines at what time within the clip it should start playing.
   - **Clip Out** defines at what time within the clip it should begin to loop (until the animation ends).



Name

Time 0 R

Time End 150 R

Duration 00:00:03:00

Clip ...

Before Start Show Black

After End Show Black

Clip In 00:00:00:00 R

Clip Out 00:00:03:00 R

5.2.5 Configuring the Keying Mode for Video

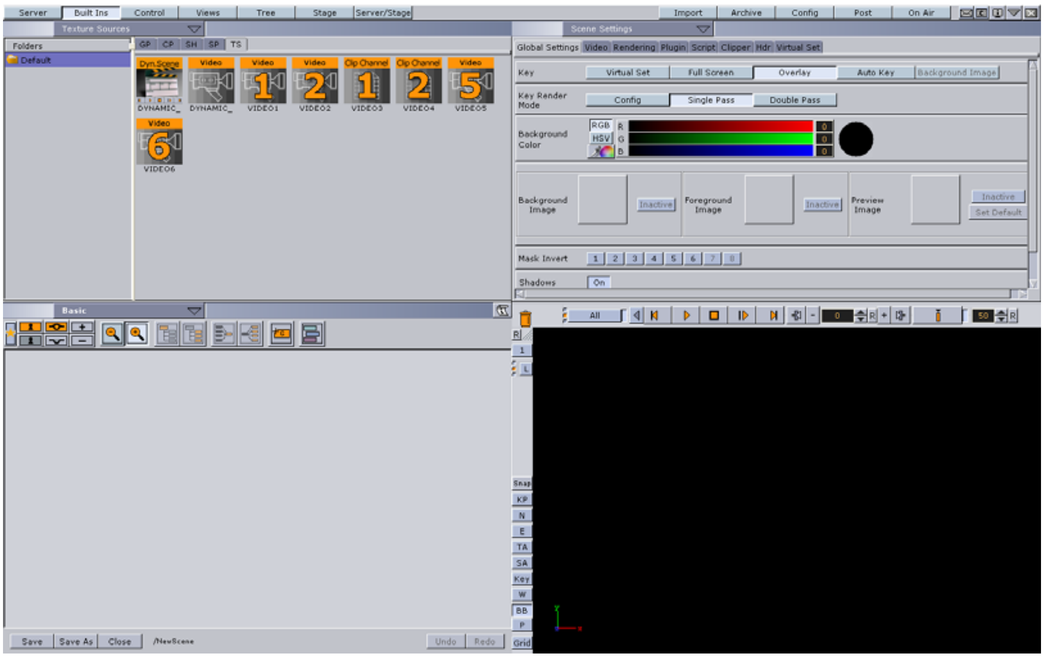
There are three different types of keying built into Viz Artist.

- **Chroma key** is mostly used with a virtual studio.
  - In the chroma keying mode the key/alpha channel for the video source is generated for each field or frame. When used in a virtual studio, the video source should be dragged into the foreground image. This way elements in the scenes without a key will be drawn first followed by the video source and last elements with key information. This drawing order allows for semi transparent objects to be drawn in front of the talent.
  - The chroma keyer can also be used to cutout people or objects which where filmed in front of a blue or green screen for head shots or product placement. For sports applications where only one color needs to be keyed out the chroma keyer can also be used. Each input channel has a freely configurable chroma keyer.
- **M-zone key** is often used in outdoor sporting events broadcasts.
  - The multi zone keyer prepares the input video to be keyable by the chroma keyer plugins in the scene tree. It is typically used in sports productions where more than one color needs to be keyed out. The video source needs to be dragged into the background image in global settings. An unlimited number of plugins can be created in the scene, each keying out a different color.
- **Linear key** is standard keying technique in Viz, and can be used for most applications. This is a logical key (enabled by the Key plugin).
  - If a video or clip channel is applied to an object it is treated the same way as if it would be a static image.
  - If it is used as a background image it is the first element to be drawn and all objects in the scene are drawn on top of it regardless of the key settings in the global section of scene settings.
  - If it is set as the foreground image it will be the last element to be drawn and all other objects will only be visible in the parts of the screen where the key/alpha of the video is not 100% (opaque)

To configure the keying mode

1. Click **Built Ins.**
2. Click the **Texture Sources** tab.

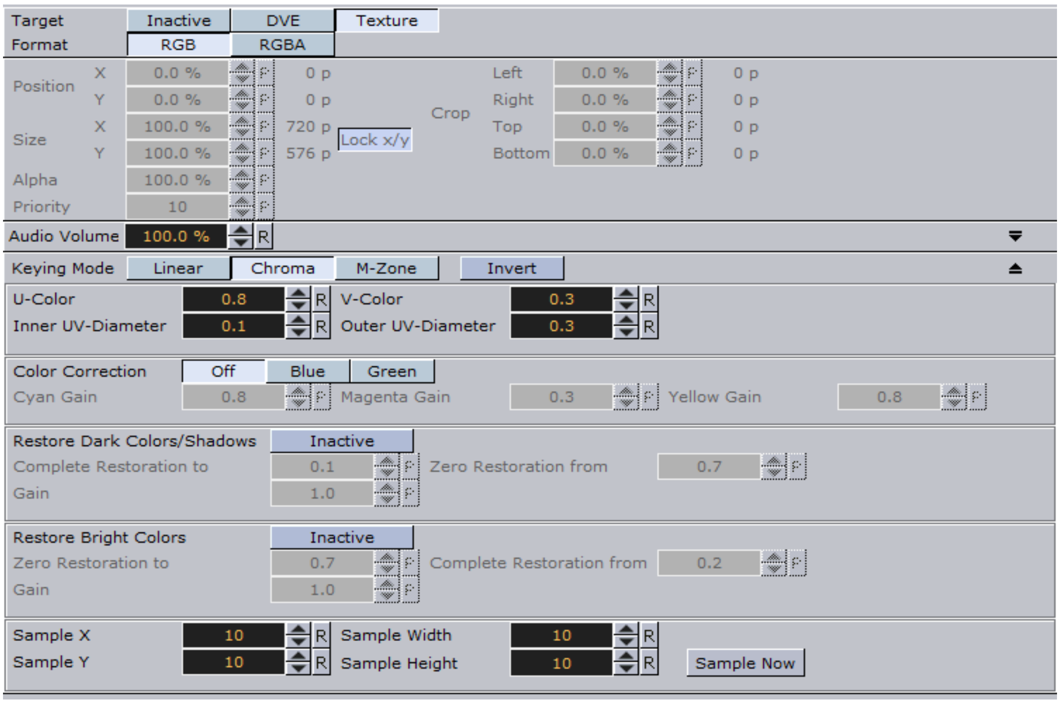
3. From the drop-down menu in the right pane, select **Scene Settings** and click the **Global Settings** tab.
4. Select a video source and drag it onto the **Background Image** or **Foreground Image** drop-zone.



5. Click the **Video** tab.
6. In the Keying Mode row, click the relevant button to enable the keying method. Your options are:
   - **Linear**
   - **Chroma key**
   - **M-zone key**

Note: For the M-zone key, you must drag and drop the Chroma Key plugin onto the container holding the video texture in the scene tree.

7. At the far right of the Keying Mode row, click the arrow to open up the keying parameters.



Target: Inactive | DVE | Texture
Format: RGB | RGBA

Position X: 0.0 % R 0 p
Position Y: 0.0 % R 0 p
Size X: 100.0 % R 720 p
Size Y: 100.0 % R 576 p
Lock x/y
Alpha: 100.0 % R
Priority: 10 R

Crop Left: 0.0 % R 0 p
Crop Right: 0.0 % R 0 p
Crop Top: 0.0 % R 0 p
Crop Bottom: 0.0 % R 0 p

Audio Volume: 100.0 % R

Keying Mode: Linear | Chroma | M-Zone | Invert

U-Color: 0.8 R
V-Color: 0.3 R
Inner UV-Diameter: 0.1 R
Outer UV-Diameter: 0.3 R

Color Correction: Off | Blue | Green
Cyan Gain: 0.8 R
Magenta Gain: 0.3 R
Yellow Gain: 0.8 R

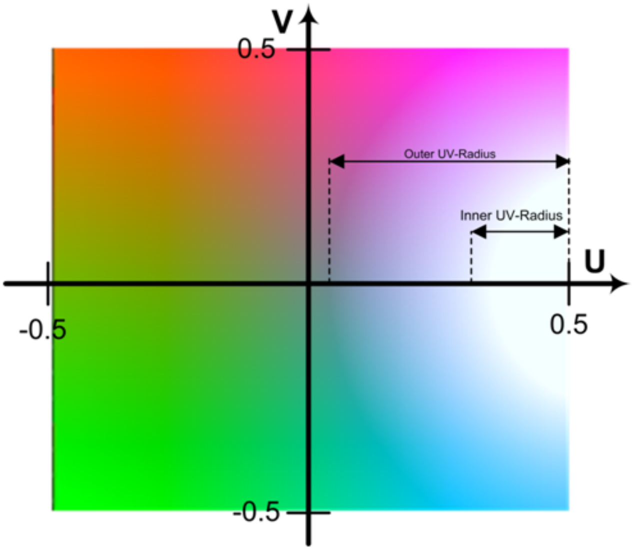
Restore Dark Colors/Shadows: Inactive
Complete Restoration to: 0.1 R
Zero Restoration from: 0.7 R
Gain: 1.0 R

Restore Bright Colors: Inactive
Zero Restoration to: 0.7 R
Complete Restoration from: 0.2 R
Gain: 1.0 R

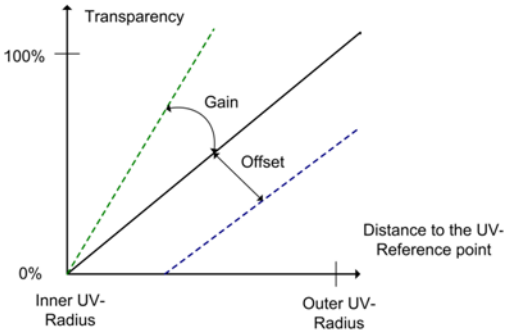
Sample X: 10 R
Sample Width: 10 R
Sample Y: 10 R
Sample Height: 10 R
Sample Now

Note: When in DVE mode, only the Linear keying mode is available.

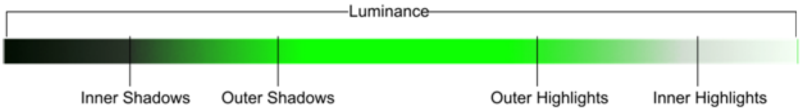
8. Configure the obligatory parameters, as follows.



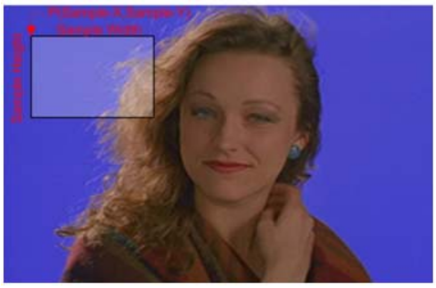
- With the first two parameters you set the background color in the YUV color space. The point set by the two parameters **U-Color** and **V-Color** is the reference for the next parameters, which determine the colors to remove. All colors between the point defined by the **U/V-Color** and the **Inner UV-Radius** will be removed completely and for all colors between the **Inner UV-Radius** and the **Outer UV-Radius** the transparency will be set to a value between 0% and 100%.



9. Configure these optional parameters, as follows.
   - **Color Correction** allows you to remove blue or green colors completely from the image. This is especially helpful to remove chroma bleed. If the Color Correction mode is set to **Blue**, the parameters for the two neighboring colors (**Cyan**, **Magenta**) are active. If the Color Correction mode is set to **Green** then the neighboring colors are **Yellow** and **Cyan**. High values for all these parameters will mean that nearly all tones of this color will remain unchanged. Low values signify that this tone will be removed completely from the image.
   - **Luminance Control** - With the **Restore Dark Colors** option all pixels with a luminance lower than complete restoration to the key/alpha will be making the pixel 100% opaque. A pixel with a luminance between **Complete Restoration to** and **Zero Restoration from** the key value is interpolated between 0 and 100%. This key is compared with the results of the chroma key and the higher value will be used. The key value of a pixel where the luminance is higher than Zero Restoration from will remain unchanged. **Gain** can be used to darken dark colors even more if it is set below 1.0. This is useful in a virtual studio where the set design uses dark colors in the shadow seen through the camera is lighter in color than the floor. Darkening the shadow will make the final composition more believable. **Gain** is only applied to pixels with luminance between 0 and **Zero Restoration To.** With the **Restore Bright Colors** option all pixels with a luminance higher than complete restoration to the key/alpha will be making the pixel 100% opaque. For a pixel with a luminance between **Complete Restoration to** and **Zero Restoration from** the key value is interpolated between 0 and 100%. This key is compared with the results of the chroma key and the higher value will be used. The key value of a pixel where the luminance is lower than Zero Restoration from will remain unchanged. **Gain** can be used to make bright colors appear even brighter. Gain is only applied to pixels with a luminance between 1 and **Zero Restoration To.**



10. **Self-configuration** functions provide a starting point, but you will still need to adopt some parameters to increase results. Set the **Sample-X** /**-Y** coordinates and the **Sample Width** / **Height.** These parameters define a section of the background in the video-input. Pressing the button **Sample Now** will start the self-configuration.



Best Practices for Configuring the Keyer

Take advantage of the self-configuration

Before you start to play with the different parameters, let the system configure by itself. Select a section of the image, which contains in best case all color tones, you want to key. If you have no foreground objects, you can set the whole input video as section for the self-configuration.

Modify the keying color

Put your real foreground objects into the scene. Mostly your objects will still have a blue edge after the self-configuration. To eliminate it, increase the Outer and Inner UV-Radius. When increasing the Inner UV-Radius have a look at opaque foreground objects in the video. It is perfectly configured if it cuts off also a little bit of the object. Have a look at tiny things, like hair or transparent objects and modify the Outer UV-Radius to get a smooth fade out at the edges of these objects.

Desaturate the blue/green edges

Up till now you have probably had tiny edges around people or objects which are close to the background color. If you enable color correction by setting it to blue or green, the color will be replaced.

Add highlights and shadows

By modifying the add highlights and add shadows, borders of objects will appear, that you've keyed before by setting the inner and outer radius.

Play around with it

You can either try to improve the keyer by increasing or decreasing the values, or you just play around with it.

5.2.6 Supported Codecs

Similar to the Pinnacle Targa 3200, the Matrox X.mio and X.mio2 supports a selection of different codecs for both SD and HD.

All codecs are implemented in software. Matrox uses the card itself as a kind of dongle for the software codecs, therefore certain codecs require certain card classes as follows:

- 6000 class = SD codecs only. No D10
- 8000 class = same as 6000 + HD codecs and D10, D12

The Matrox X.mio 24/6000 card, including all SD codecs, except D10/D12, is Vizrt's standard card. This model can be upgraded to any higher model-class by performing a license upgrade.

The following codec types are supported:

- Targa Codecs
- PAL Codecs
- NTSC Codecs
- 720P50 Codecs
- 720P60M Codecs
- 1080i25 Codecs
- 1080i30M Codecs

5.2.7 Targa Codecs

The Targa card supports these codecs in the following formats:

- DV: DV25, DV50
- MPEG2 I frame, 4:2:2
- IMX D10

5.2.8 PAL Codecs

Table 15:

Name	Container	Codec / Resolution		Audio
Generic Interleaved	.avi	MPEG2-IFrame422 720x576	alpha: no bitrate: 10-50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG2-IFrame4224 720x576	alpha: yes bitrate: 10-50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP SD 4:2:0, Elementary 720x576	alpha: no bitrate: 1-15 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz

Table 15:

Name	Cont ainer	Codec / Resolution		Audio
Generic Interleaved	.avi	MPEG-2 IBP SD 4:2:0, Programm 720x576	alpha: no bitrate: 1-15 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP SD 4:2:0, Transport 720x576	alpha: no bitrate: 1-15 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP SD 4:2:2, Elemetary 720x576	alpha: no bitrate: 5-50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP SD 4:2:2, Programm 720x576	alpha: no bitrate: 5-50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP SD 4:2:2, Transport 720x576	alpha: no bitrate: 5-50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	DV/DVCAM 4:2:0 720x576	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	DVCPRO 4:1:1 720x576	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz

Table 15:

Name	Container	Codec / Resolution		Audio
Generic Interleaved	.avi	DVCPRO 50 4:2:2 720x576	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	Uncompressed YUVU 4:2:2 720x576	alpha: no bitrate: 160 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	Uncompressed YUAVUA 4:2:2:4 720x576	alpha: no bitrate: 240 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MJpeg Lossless 720x576	alpha: no bitrate: VBR bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
XDCAM (OP1a)	.mxf	DVCAM 4:2:0 720x576	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 4 ch: 16 in 16bit; 48kHz
XDCAM (OP1a)	.mxf	D10 (IMX) 4:2:2 720x608	alpha: no bitrate: 30, 40, 50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 4 ch: 24 in 32bit; 48kHz 8 ch: 16 in 16bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2 IBP SD 4:2:0, Elemetary 720x576	alpha: no bitrate: 1-15 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2 IBP SD 4:2:0, Programm 720x576	alpha: no bitrate: 1-15 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz

Table 15:

Name	Container	Codec / Resolution		Audio
Generic MPEG-2	.mxf	MPEG-2 IBP SD 4:2:0, Transport 720x576	alpha: no bitrate: 1-15 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2 IBP SD 4:2:2, Elemetary 720x576	alpha: no bitrate: 5-50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2 IBP SD 4:2:2, Programm 720x576	alpha: no bitrate: 5-50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2 IBP SD 4:2:2, Transport 720x576	alpha: no bitrate: 5-50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
DVCPRO	.mxf	DVCPRO 4:1:1 720x576	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
DVCPRO	.mxf	DVCPRO 50 4:2:2 720x576	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Panasonic P2	.mxf	DVCAM 4:2:0 720x576	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Panasonic P2	.mxf	DVCPRO 4:1:1 720x576	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz

Table 15:

Name	Container	Codec / Resolution		Audio
Panasonic P2	.mxf	DVCPRO 50 4:2:2 720x576	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
XDCAM	.mov	DVCAM 4:2:0 720x576	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
DVCPRO	.mov	DVCPRO 4:1:1 720x576	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
DVCPRO	.mov	DVCPRO 50 4:2:2 720x576	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
H.246	.mov	H.264/AVC 720x576	alpha: no bitrate: VBR bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
H.246	.mp4	H.264/AVC 720x576	alpha: no bitrate: VBR bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
DV	.dv	DVCAM 4:2:0 720x576	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz
DV	.dv	DVCPRO 4:1:1 720x576	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz

Table 15:

Name	Container	Codec / Resolution		Audio
DV	.dv	DVCPRO 50 4:2:2 720x576	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz
Generic Separate	.avi	MPEG2-IFrame422 720x576	alpha: no bitrate: 10-50 bitdepth: 8 VBI: N/A timecode: N/A	
Generic Separate	.avi	MPEG2-IFrame4224 720x576	alpha: yes bitrate: 10-50 bitdepth: 8 VBI: N/A timecode: N/A	
Generic Separate	.avi	MPEG-2 IBP SD 4:2:0, Elemetary 720x576	alpha: no bitrate: 1-15 bitdepth: 8 VBI: N/A timecode: N/A	
Generic Separate	.avi	MPEG-2 IBP SD 4:2:0, Programm 720x576	alpha: no bitrate: 1-15 bitdepth: 8 VBI: N/A timecode: N/A	
Generic Separate	.avi	MPEG-2 IBP SD 4:2:0, Transport 720x576	alpha: no bitrate: 1-15 bitdepth: 8 VBI: N/A timecode: N/A	
Generic Separate	.avi	MPEG-2 IBP SD 4:2:2, Elemetary 720x576	alpha: no bitrate: 5-50 bitdepth: 8 VBI: N/A timecode: N/A	
Generic Separate	.avi	MPEG-2 IBP SD 4:2:2, Programm 720x576	alpha: no bitrate: 5-50 bitdepth: 8 VBI: N/A timecode: N/A	

Table 15:

Name	Cont ainer	Codec / Resolution	Audio
Generic Separate	.avi	MPEG-2 IBP SD 4:2:2, Transport 720x576	alpha: no bitrate: 5-50 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	DV/DVCAM 4:2:0 720x576	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	DVCPRO 4:1:1 720x576	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	DVCPRO 50 4:2:2 720x576	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	Uncompressed YUVU 4:2:2 720x576	alpha: no bitrate: 160 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	Uncompressed YUAVUA 4:2:2:4 720x576	alpha: no bitrate: 240 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	MJpeg Lossless 720x576	alpha: no bitrate: VBR bitdepth: 8 VBI: N/A timecode: N/A
XDCAM	.mov	DVCAM 4:2:0 720x576	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A

Table 15:

Name	Cont ainer	Codec / Resolution		Audio
DVCPRO	.mov	DVCPRO 4:1:1 720x576	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	
DVCPRO	.mov	DVCPRO 50 4:2:2 720x576	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A	
H.246	.mov	H.264/AVC 720x576	alpha: no bitrate: VBR bitdepth: 8 VBI: N/A timecode: N/A	
Generic Separate	.wav			PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Separate	_vbi. avi	vbi 720x1-17 Start Line: 6	bitdepth: 10 VBI: YUYV422	

5.2.9 NTSC Codecs

Table 16:

Name	Cont ainer	Codec / Resolution		Audio
Generic Interleaved	.avi	MPEG2-IFrame422 720x480	alpha: no bitrate: 10-50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG2-IFrame4224 720x480	alpha: yes bitrate: 10-50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz

Table 16:

Name	Container	Codec / Resolution		Audio
Generic Interleaved	.avi	MPEG-2 IBP SD 4:2:0, Elemetary 720x480	alpha: no bitrate: 1-15 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP SD 4:2:0, Programm 720x480	alpha: no bitrate: 1-15 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP SD 4:2:0, Transport 720x480	alpha: no bitrate: 1-15 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP SD 4:2:2, Elemetary 720x480	alpha: no bitrate: 5-50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP SD 4:2:2, Programm 720x480	alpha: no bitrate: 5-50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP SD 4:2:2, Transport 720x480	alpha: no bitrate: 5-50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	DV/DVCAM 4:1:1 720x480	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz

Table 16:

Name	Container	Codec / Resolution		Audio
Generic Interleaved	.avi	DVCPRO 4:1:1 720x480	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	DVCPRO 50 4:2:2 720x480	alpha: no bitrate: 1-15 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	Uncompressed YUVU 4:2:2 720x480	alpha: no bitrate: 160 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	Uncompressed YUAVUA 4:2:2:4 720x480	alpha: no bitrate: 240 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MJpeg Lossless 720x480	alpha: no bitrate: VBR bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
XDCAM (OP1a)	.mxf	DVCAM 4:1:1 720x480	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 4 ch: 16 in 16bit; 48kHz
XDCAM (OP1a)	.mxf	D10 (IMX) 4:2:2 720x512	alpha: no bitrate: 30, 40, 50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 4 ch: 24 in 32bit; 48kHz 8 ch: 16 in 16bit; 48kHz

Table 16:

Name	Container	Codec / Resolution		Audio
Generic MPEG-2	.mxf	MPEG-2 IBP SD 4:2:0, Elemetary 720x480	alpha: no bitrate: 1-15 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2 IBP SD 4:2:0, Programm 720x480	alpha: no bitrate: 1-15 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2 IBP SD 4:2:0, Transport 720x480	alpha: no bitrate: 1-15 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2 IBP SD 4:2:2, Elemetary 720x480	alpha: no bitrate: 5-50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2 IBP SD 4:2:2, Programm 720x480	alpha: no bitrate: 5-50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2 IBP SD 4:2:2, Transport 720x480	alpha: no bitrate: 5-50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
DVCPRO	.mxf	DVCPRO 4:1:1 720x480	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
DVCPRO	.mxf	DVCPRO 50 4:2:2 720x480	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz

Table 16:

Name	Cont ainer	Codec / Resolution		Audio
Panasonic P2	.mxf	DV/DVCAM 4:1:1 720x480	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Panasonic P2	.mxf	DVCPRO 4:1:1 720x480	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Panasonic P2	.mxf	DVCPRO 50 4:2:2 720x480	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
XDCAM (OP1a)	.mov	DVCAM 4:1:1 720x480	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
DVCPRO	.mov	DVCPRO 4:1:1 720x480	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
DVCPRO	.mov	DVCPRO 50 4:2:2 720x480	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
H.246	.mov	H.264/AVC 720x480	alpha: no bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
H.246	.mp4	H.264/AVC 720x480	alpha: no bitrate: VBR bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz

Table 16:

Name	Container	Codec / Resolution		Audio
DV	.dv	DV/DVCAM 4:1:1 720x480	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz
DV	.dv	DVCPRO 4:1:1 720x480	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz
DV	.dv	DVCPRO 50 4:2:2 720x480	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz
Generic Separate	.avi	MPEG2-IFrame422 720x480	alpha: no bitrate: 10-50 bitdepth: 8 VBI: N/A timecode: N/A	
Generic Separate	.avi	MPEG2-IFrame4224 720x480	alpha: yes bitrate: 10-50 bitdepth: 8 VBI: N/A timecode: N/A	
Generic Separate	.avi	MPEG-2 IBP SD 4:2:0, Elemetary 720x480	alpha: no bitrate: 1-15 bitdepth: 8 VBI: N/A timecode: N/A	
Generic Separate	.avi	MPEG-2 IBP SD 4:2:0, Programm 720x480	alpha: no bitrate: 1-15 bitdepth: 8 VBI: N/A timecode: N/A	
Generic Separate	.avi	MPEG-2 IBP SD 4:2:0, Transport 720x480	alpha: no bitrate: 1-15 bitdepth: 8 VBI: N/A timecode: N/A	

Table 16:

Name	Cont ainer	Codec / Resolution	Audio
Generic Separate	.avi	MPEG-2 IBP SD 4:2:2, Elemetary 720x480	alpha: no bitrate: 5-50 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	MPEG-2 IBP SD 4:2:2, Programm 720x480	alpha: no bitrate: 5-50 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	MPEG-2 IBP SD 4:2:2, Transport 720x480	alpha: no bitrate: 5-50 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	DV/DVCAM 4:1:1 720x480	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	DVCPRO 4:1:1 720x480	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	DVCPRO 50 4:2:2 720x480	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	Uncompressed YUVU 4:2:2 720x480	alpha: no bitrate: 160 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	Uncompressed YUAVUA 4:2:2:4 720x480	alpha: no bitrate: 240 bitdepth: 8 VBI: N/A timecode: N/A

Table 16:

Name	Container	Codec / Resolution		Audio
Generic Separate	.avi	MJpeg Lossless 720x480	alpha: no bitrate: VBR bitdepth: 8 VBI: N/A timecode: N/A	
XDCAM (OP1a)	.mov	DVCAM 4:1:1 720x480	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	
DVCPRO	.mov	DVCPRO 4:1:1 720x480	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	
DVCPRO	.mov	DVCPRO 50 4:2:2 720x480	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A	
H.246	.mov	H.264/AVC 720x486	alpha: no bitrate: VBR bitdepth: 8 VBI: N/A timecode: N/A	
Generic Separate	.wav			PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Separate	_vbi.avi	vbi 720x1-16 Start Line: 7	bitdepth: 10 VBI: YUYV422	

5.2.10 720P50 Codecs

Table 17:

Name	Container	Codec / Resolution		Audio
Generic Interleaved	.avi	MPEG2-IFrame422 1280x720	alpha: no bitrate: 50-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG2-IFrame4224 1280x720	alpha: yes bitrate: 50-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP HD 4:2:0, Elemetary 1280x720	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP HD 4:2:0, Program 1280x720	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP HD 4:2:0, Transport 1280x720	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP HD 4:2:2, Elemetary 1280x720	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP HD 4:2:2, Program 1280x720	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz

Table 17:

Name	Container	Codec / Resolution		Audio
Generic Interleaved	.avi	MPEG-2 IBP HD 4:2:2, Transport 1280x720	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
DVCPRO	.avi	DVCPRO HD 1280x720	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	Uncompressed YUVU 4:2:2 1280x720	alpha: no bitrate: 700 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	Uncompressed YUAVUA 4:2:2:4 1280x720	alpha: no bitrate: 1050 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
XDCAM (OP1a)	.mxf	XDCAM HD, IBP HD 4:2:0,Elementary 1280x720	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
XDCAM (OP1a)	.mxf	XDCAM HD, IBP HD 4:2:0, Program 1280x720	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
XDCAM (OP1a)	.mxf	XDCAM HD, IBP HD 4:2:0, Transport 1280x720	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
XDCAM (OP1a)	.mxf	XDCAM HD422, IBP HD 4:2:2, Elementary 1280x720	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz

Table 17:

Name	Container	Codec / Resolution		Audio
XDCAM (OP1a)	.mxf	XDCAM HD422, IBP HD 4:2:2, Program 1280x720	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz
XDCAM (OP1a)	.mxf	XDCAM HD422, IBP HD 4:2:2, Transport 1280x720	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2, IBP HD 4:2:0, Elementary 1280x720	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2, IBP HD 4:2:0, Program 1280x720	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2, IBP HD 4:2:0, Transport 1280x720	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2, IBP HD 4:2:2, Elementary 1280x720	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2, IBP HD 4:2:2, Program 1280x720	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2, IBP HD 4:2:2, Transport 1280x720	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz

Table 17:

Name	Container	Codec / Resolution		Audio
DVCPRO	.mxf	DVCPRO HD 1280x720	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Panasonic P2	.mxf	AVCIntra Class 50 1280x720	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz
Panasonic P2	.mxf	AVCIntra Class 100 1280x720	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz
Panasonic P2	.mxf	DVCPRO HD 1280x720	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz
DVCPRO Quicktime	.mov	DVCPRO HD 1280x720	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz
HDV	.mov	HDV 1280x720	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
XDCAM (OP1a)	.mov	XDCam EX 1280x720	alpha: no bitrate: 25CBR, 35VBR bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
XDCAM (OP1a)	.mov	XDCam HD 422 1280x720	alpha: no bitrate: 18-35 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz

Table 17:

Name	Container	Codec / Resolution		Audio
DNxHD	.mov	DNxHD 4:2:2 1280x720	alpha: no bitrate: 36, 145, 220 bitdepth: 8; 10 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
H.264	.mov	H.264/AVC 1280x720	alpha: no bitrate: VBR bitdepth: 8; 10 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
H.264	.mp4	H.264/AVC 1280x720	alpha: no bitrate: VBR bitdepth: 8; 10 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
DV	.dv	DVCPRO HD 1280x720	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz
ProRes 422	.mov	I-Frame 4:2:2 1280x720	alpha: no bitrate: 147, 220 bitdepth: 10 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
Generic Separate	.avi	MPEG2-IFrame422 1280x720	alpha: no bitrate: 50-300 bitdepth: 8 VBI: N/A timecode: N/A	
Generic Separate	.avi	MPEG2-IFrame4224 1280x720	alpha: yes bitrate: 50-300 bitdepth: 8 VBI: N/A timecode: N/A	
Generic Separate	.avi	MPEG-2 IBP HD 4:2:0, Elemetary 1280x720	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	

Table 17:

Name	Container	Codec / Resolution	Audio
Generic Separate	.avi	MPEG-2 IBP HD 4:2:0, Program 1280x720	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	MPEG-2 IBP HD 4:2:0, Transport 1280x720	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	MPEG-2 IBP HD 4:2:2, Elemetary 1280x720	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	MPEG-2 IBP HD 4:2:2, Program 1280x720	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	MPEG-2 IBP HD 4:2:2, Transport 1280x720	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A
DVCPRO	.avi	DVCPRO HD 1280x720	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	Uncompressed YUVU 4:2:2 1280x720	alpha: no bitrate: 700 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	Uncompressed YUAVUA 4:2:2:4 1280x720	alpha: no bitrate: 1050 bitdepth: 8 VBI: N/A timecode: N/A

Table 17:

Name	Container	Codec / Resolution		Audio
DVCPRO Quicktime	.mov	DVCPRO HD 1280x720	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A	
HDV	.mov	HDV 1280x720	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	
XDCAM (OP1a)	.mov	XDCam EX 1280x720	alpha: no bitrate: 25CBR, 35VBR bitdepth: 8 VBI: N/A timecode: N/A	
XDCAM (OP1a)	.mov	XDCam HD 422 1280x720	alpha: no bitrate: 18-35 bitdepth: 8 VBI: N/A timecode: N/A	
DNxHD	.mov	DNxHD YUVU 4:2:2 1280x720	alpha: no bitrate: 36, 145, 220 bitdepth: 8; 10 VBI: N/A timecode: N/A	
H.264	.mov	H.264 YUVU 4:2:2 1280x720	alpha: no bitrate: VBR bitdepth: 8; 10 VBI: N/A timecode: N/A	
Generic Separate	.wav			PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Separate	_vbi.avi	vbi 1280x1-19 Start Line: 7	bitdepth: 10 VBI: YUYV422	

5.2.11 720P60M Codecs

Table 18:

Name	Container	Codec / Resolution		Audio
Name	Container	Codec / Resolution		Audio
Generic Interleaved	.avi	MPEG2-IFrame422 1280x720	alpha: no bitrate: 50-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG2-IFrame4224 1280x720	alpha: yes bitrate: 50-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP HD 4:2:0, Elemetary 1280x720	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP HD 4:2:0, Program 1280x720	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP HD 4:2:0, Transport 1280x720	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP HD 4:2:2, Elemetary 1280x720	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz

Table 18:

Name	Container	Codec / Resolution		Audio
Generic Interleaved	.avi	MPEG-2 IBP HD 4:2:2, Program 1280x720	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP HD 4:2:2, Transport 1280x720	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
DVCPRO	.avi	DVCPRO HD 1280x720	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	Uncompressed YUVU 4:2:2 1280x720	alpha: no bitrate: 850 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	Uncompressed YUAVUA 4:2:2:4 1280x720	alpha: no bitrate: 1275 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
XDCAM (OP1a)	.mxf	XDCAM HD, IBP HD 4:2:0,Elementary 1280x720	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
XDCAM (OP1a)	.mxf	XDCAM HD, IBP HD 4:2:0, Program 1280x720	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
XDCAM (OP1a)	.mxf	XDCAM HD, IBP HD 4:2:0, Transport 1280x720	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz

Table 18:

Name	Container	Codec / Resolution		Audio
XDCAM (OP1a)	.mxf	XDCAM HD422, IBP HD 4:2:2, Elementary 1280x720	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz
XDCAM (OP1a)	.mxf	XDCAM HD422, IBP HD 4:2:2, Program 1280x720	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz
XDCAM (OP1a)	.mxf	XDCAM HD422, IBP HD 4:2:2, Transport 1280x720	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2, IBP HD 4:2:0, Elementary 1280x720	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2, IBP HD 4:2:0, Program 1280x720	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2, IBP HD 4:2:0, Transport 1280x720	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2, IBP HD 4:2:2, Elementary 1280x720	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2, IBP HD 4:2:2, Program 1280x720	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz

Table 18:

Name	Container	Codec / Resolution		Audio
Generic MPEG-2	.mxf	MPEG-2, IBP HD 4:2:2, Transport 1280x720	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
DVCPRO	.mxf	DVCPRO HD 1280x720	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Panasonic P2	.mxf	AVCIntra Class 50 1280x720	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz
Panasonic P2	.mxf	AVCIntra Class 100 1280x720	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz
Panasonic P2	.mxf	DVCPRO HD 1280x720	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz
DVCPRO Quicktime	.mov	DVCPRO HD 1280x720	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz
HDV	.mov	HDV 1280x720	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
XDCAM (OP1a)	.mov	XDCam EX 1280x720	alpha: no bitrate: 25CBR, 35VBR bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz

Table 18:

Name	Container	Codec / Resolution		Audio
XDCAM (OP1a)	.mov	XDCam HD 422 1280x720	alpha: no bitrate: 18-35 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
DNxHD	.mov	DNxHD 4:2:2 1280x720	alpha: no bitrate: 36, 145, 220 bitdepth: 8; 10 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
H.264	.mov	H.264/AVC 1280x720	alpha: no bitrate: VBR bitdepth: 8; 10 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
H.264	.mp4	H.264/AVC 1280x720	alpha: no bitrate: VBR bitdepth: 8; 10 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
DV	.dv	DVCPRO HD 1280x720	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz
ProRes 422	.mov	I-Frame 4:2:2 1280x720	alpha: no bitrate: 147, 220 bitdepth: 10 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
Generic Separate	.avi	MPEG2-IFrame422 1280x720	alpha: no bitrate: 50-300 bitdepth: 8 VBI: N/A timecode: N/A	
Generic Separate	.avi	MPEG2-IFrame4224 1280x720	alpha: yes bitrate: 50-300 bitdepth: 8 VBI: N/A timecode: N/A	

Table 18:

Name	Cont ainer	Codec / Resolution	Audio
Generic Separate	.avi	MPEG-2 IBP HD 4:2:0, Elemetary 1280x720	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	MPEG-2 IBP HD 4:2:0, Program 1280x720	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	MPEG-2 IBP HD 4:2:0, Transport 1280x720	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	MPEG-2 IBP HD 4:2:2, Elemetary 1280x720	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	MPEG-2 IBP HD 4:2:2, Program 1280x720	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	MPEG-2 IBP HD 4:2:2, Transport 1280x720	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A
DVCPRO	.avi	DVCPRO HD 1280x720	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	Uncompres sed YUVU 4:2:2 1280x720	alpha: no bitrate: 850 bitdepth: 8 VBI: N/A timecode: N/A

Table 18:

Name	Container	Codec / Resolution		Audio
Generic Separate	.avi	Uncompressed YUAVUA 4:2:2:4 1280x720	alpha: no bitrate: 1275 bitdepth: 8 VBI: N/A timecode: N/A	
DVCPRO Quicktime	.mov	DVCPRO HD 1280x720	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A	
HDV	.mov	HDV 1280x720	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	
XDCAM (OP1a)	.mov	XDCam EX 1280x720	alpha: no bitrate: 25CBR, 35VBR bitdepth: 8 VBI: N/A timecode: N/A	
XDCAM (OP1a)	.mov	XDCam HD 422 1280x720	alpha: no bitrate: 18-35 bitdepth: 8 VBI: N/A timecode: N/A	
DNxHD	.mov	DNxHD YUVU 4:2:2 1280x720	alpha: no bitrate: 36, 145, 220 bitdepth: 8; 10 VBI: N/A timecode: N/A	
H.264	.mov	H.264 YUVU 4:2:2 1280x720	alpha: no bitrate: VBR bitdepth: 8; 10 VBI: N/A timecode: N/A	
Generic Separate	.wav			PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz

Table 18:

Name	Container	Codec / Resolution		Audio
Generic Separate	_vbi.avi	vbi 1280x1-19 Start Line: 7	bitdepth: 10 VBI: YUYV422	

5.2.12 1080i25 Codecs

Table 19:

Name	Container	Codec / Resolution		Audio
Generic Interleaved	.avi	MPEG2-IFrame422 1920x1080	alpha: no bitrate: 50-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG2-IFrame4224 1920x1080	alpha: yes bitrate: 50-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP HD 4:2:0, Elemetary 1920x1080	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP HD 4:2:0, Program 1920x1080	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP HD 4:2:0, Transport 1920x1080	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz

Table 19:

Name	Container	Codec / Resolution		Audio
Generic Interleaved	.avi	MPEG-2 IBP HD 4:2:2, Elemetary 1920x1080	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP HD 4:2:2, Program 1920x1080	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP HD 4:2:2, Transport 1920x1080	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
DVCPRO	.avi	DVCPRO HD 1920x1080	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	Uncompressed YUVU 4:2:2 1920x1080	alpha: no bitrate: 800 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	Uncompressed YUAVUA 4:2:2:4 1920x1080	alpha: no bitrate: 1200 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
XDCAM (OP1a)	.mxf	XDCAM HD, IBP HD 4:2:0,Elementary 1440x1080	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz

Table 19:

Name	Container	Codec / Resolution		Audio
XDCAM (OP1a)	.mxf	XDCAM HD, IBP HD 4:2:0, Program 1440x1080	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
XDCAM (OP1a)	.mxf	XDCAM HD, IBP HD 4:2:0, Transport 1440x1080	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
XDCAM (OP1a)	.mxf	XDCAM HD422, IBP HD 4:2:2, Elementary 1920x1080	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz
XDCAM (OP1a)	.mxf	XDCAM HD422, IBP HD 4:2:2, Program 1920x1080	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz
XDCAM (OP1a)	.mxf	XDCAM HD422, IBP HD 4:2:2, Transport 1920x1080	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2, IBP HD 4:2:0, Elementary 1920x1080	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2, IBP HD 4:2:0, Program 1920x1080	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2, IBP HD 4:2:0, Transport 1920x1080	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz

Table 19:

Name	Container	Codec / Resolution		Audio
Generic MPEG-2	.mxf	MPEG-2, IBP HD 4:2:2, Elementary 1920x1080	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2, IBP HD 4:2:2, Program 1920x1080	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2, IBP HD 4:2:2, Transport 1920x1080	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
DVCPRO	.mxf	DVCPRO HD 1920x1080	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Panasonic P2	.mxf	AVCIntra Class 50 1920x1080	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz
Panasonic P2	.mxf	AVCIntra Class 100 1920x1080	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz
Panasonic P2	.mxf	DVCPRO HD 1920x1080	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz
DVCPRO Quicktime	.mov	DVCPRO HD 1920x1080	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz

Table 19:

Name	Container	Codec / Resolution		Audio
HDV	.mov	HDV 1440x1080	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
XDCAM (OP1a)	.mov	XDCam EX 1920x1080	alpha: no bitrate: 25CBR, 35VBR bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
XDCAM (OP1a)	.mov	XDCam HD 1440x1080	alpha: no bitrate: 35 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
XDCAM (OP1a)	.mov	XDCam HD 422 1920x1080	alpha: no bitrate: 18–35 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
DNxHD	.mov	DNxHD 4:2:2 1920x1080	alpha: no bitrate: 36, 145, 220 bitdepth: 8; 10 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
H.264	.mov	H.264/AVC 1920x1080	alpha: no bitrate: VBR bitdepth: 8; 10 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
H.264	.mp4	H.264/AVC 1920x1080	alpha: no bitrate: VBR bitdepth: 8; 10 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
DV	.dv	DVCPRO HD 1920x1080	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz

Table 19:

Name	Container	Codec / Resolution		Audio
ProRes 422	.mov	I-Frame 4:2:2 1920x1080	alpha: no bitrate: 42, 63 bitdepth: 10 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
Generic Separate	.avi	MPEG2-IFrame422 1920x1080	alpha: no bitrate: 50-300 bitdepth: 8 VBI: N/A timecode: N/A	
Generic Separate	.avi	MPEG2-IFrame4224 1920x1080	alpha: yes bitrate: 50-300 bitdepth: 8 VBI: N/A timecode: N/A	
Generic Separate	.avi	MPEG-2 IBP HD 4:2:0, Elemetary 1920x1080	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	
Generic Separate	.avi	MPEG-2 IBP HD 4:2:0, Program 1920x1080	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	
Generic Separate	.avi	MPEG-2 IBP HD 4:2:0, Transport 1920x1080	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	
Generic Separate	.avi	MPEG-2 IBP HD 4:2:2, Elemetary 1920x1080	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A	
Generic Separate	.avi	MPEG-2 IBP HD 4:2:2, Program 1920x1080	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A	

Table 19:

Name	Container	Codec / Resolution	Audio
Generic Separate	.avi	MPEG-2 IBP HD 4:2:2, Transport 1920x1080	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A
DVCPRO	.avi	DVCPRO HD 1920x1080	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	Uncompressed YUVU 4:2:2 1920x1080	alpha: no bitrate: 800 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	Uncompressed YUAVUA 4:2:2:4 1920x1080	alpha: no bitrate: 1200 bitdepth: 8 VBI: N/A timecode: N/A
DVCPRO Quicktime	.mov	DVCPRO HD 1920x1080	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A
HDV	.mov	HDV 1440x1080	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A
XDCAM (OP1a)	.mov	XDCam EX 1920x1080	alpha: no bitrate: 25CBR, 35VBR bitdepth: 8 VBI: N/A timecode: N/A
XDCAM (OP1a)	.mov	XDCam HD 1440x1080	alpha: no bitrate: 35 bitdepth: 8 VBI: N/A timecode: N/A

Table 19:

Name	Cont ainer	Codec / Resolution		Audio
XDCAM (OP1a)	.mov	XDCam HD 422 1920x1080	alpha: no bitrate: 18-35 bitdepth: 8 VBI: N/A timecode: N/A	
DNxHD	.mov	DNxHD YUVU 4:2:2 1920x1080	alpha: no bitrate: 36, 145, 220 bitdepth: 8; 10 VBI: N/A timecode: N/A	
H.264	.mov	H.264 YUVU 4:2:2 1920x1080	alpha: no bitrate: VBR bitdepth: 8; 10 VBI: N/A timecode: N/A	
Generic Separate	.wav			PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Separate	_vbi. avi	vbi 1920x1-15 Start Line: 6	bitdepth: 10 VBI: YUYV422	

5.2.13 1080i30M Codecs

Table 20:

Name	Cont ainer	Codec / Resolution		Audio
Generic Interleaved	.avi	MPEG2-IFrame422 1920x1080	alpha: no bitrate: 50-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG2-IFrame4224 1920x1080	alpha: yes bitrate: 50-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz

Table 20:

Name	Container	Codec / Resolution		Audio
Generic Interleaved	.avi	MPEG-2 IBP HD 4:2:0, Elemetary 1920x1080	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP HD 4:2:0, Program 1920x1080	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP HD 4:2:0, Transport 1920x1080	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP HD 4:2:2, Elemetary 1920x1080	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP HD 4:2:2, Program 1920x1080	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	MPEG-2 IBP HD 4:2:2, Transport 1920x1080	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
DVCPRO	.avi	DVCPRO HD 1920x1080	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz

Table 20:

Name	Container	Codec / Resolution		Audio
Generic Interleaved	.avi	Uncompressed YUVU 4:2:2 1920x1080	alpha: no bitrate: 950 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Interleaved	.avi	Uncompressed YUAVUA 4:2:2:4 1920x1080	alpha: no bitrate: 1425 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
XDCAM (OP1a)	.mxf	XDCAM HD, IBP HD 4:2:0,Elementary 1440x1080	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
XDCAM (OP1a)	.mxf	XDCAM HD, IBP HD 4:2:0, Program 1440x1080	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
XDCAM (OP1a)	.mxf	XDCAM HD, IBP HD 4:2:0, Transport 1440x1080	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
XDCAM (OP1a)	.mxf	XDCAM HD422, IBP HD 4:2:2, Elementary 1920x1080	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz
XDCAM (OP1a)	.mxf	XDCAM HD422, IBP HD 4:2:2, Program 1920x1080	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz
XDCAM (OP1a)	.mxf	XDCAM HD422, IBP HD 4:2:2, Transport 1920x1080	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz

Table 20:

Name	Container	Codec / Resolution		Audio
Generic MPEG-2	.mxf	MPEG-2, IBP HD 4:2:0, Elementary 1920x1080	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2, IBP HD 4:2:0, Program 1920x1080	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2, IBP HD 4:2:0, Transport 1920x1080	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2, IBP HD 4:2:2, Elementary 1920x1080	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2, IBP HD 4:2:2, Program 1920x1080	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Generic MPEG-2	.mxf	MPEG-2, IBP HD 4:2:2, Transport 1920x1080	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
DVCPRO	.mxf	DVCPRO HD 1920x1080	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz
Panasonic P2	.mxf	AVCIntra Class 50 1920x1080	alpha: no bitrate: 50 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz

Table 20:

Name	Container	Codec / Resolution		Audio
Panasonic P2	.mxf	AVCIntra Class 100 1920x1080	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz
Panasonic P2	.mxf	DVCPRO HD 1920x1080	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz
DVCPRO Quicktime	.mov	DVCPRO HD 1920x1080	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz
HDV	.mov	HDV 1440x1080	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
XDCAM (OP1a)	.mov	XDCam EX 1920x1080	alpha: no bitrate: 25CBR, 35VBR bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
XDCAM (OP1a)	.mov	XDCam HD 1440x1080	alpha: no bitrate: 35 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
XDCAM (OP1a)	.mov	XDCam HD 422 1920x1080	alpha: no bitrate: 18–35 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
DNxHD	.mov	DNxHD 4:2:2 1920x1080	alpha: no bitrate: 36, 145, 220 bitdepth: 8; 10 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz

Table 20:

Name	Container	Codec / Resolution		Audio
H.264	.mov	H.264/AVC 1920x1080	alpha: no bitrate: VBR bitdepth: 8; 10 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
H.264	.mp4	H.264/AVC 1920x1080	alpha: no bitrate: VBR bitdepth: 8; 10 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
DV	.dv	DVCPRO HD 1920x1080	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz
ProRes 422	.mov	I-Frame 4:2:2 1920x1080	alpha: no bitrate: 42, 63 bitdepth: 10 VBI: N/A timecode: N/A	PCM 2 ch: 16 in 16bit; 48kHz 4 ch: 16 in 16bit; 48kHz 8 ch: 16 in 16bit; 48kHz
Generic Separate	.avi	MPEG2-IFrame422 1920x1080	alpha: no bitrate: 50-300 bitdepth: 8 VBI: N/A timecode: N/A	
Generic Separate	.avi	MPEG2-IFrame4224 1920x1080	alpha: yes bitrate: 50-300 bitdepth: 8 VBI: N/A timecode: N/A	
Generic Separate	.avi	MPEG-2 IBP HD 4:2:0, Elemetary 1920x1080	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	
Generic Separate	.avi	MPEG-2 IBP HD 4:2:0, Program 1920x1080	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A	

Table 20:

Name	Cont ainer	Codec / Resolution	Audio
Generic Separate	.avi	MPEG-2 IBP HD 4:2:0, Transport 1920x1080	alpha: no bitrate: 5-80 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	MPEG-2 IBP HD 4:2:2, Elemetary 1920x1080	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	MPEG-2 IBP HD 4:2:2, Program 1920x1080	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	MPEG-2 IBP HD 4:2:2, Transport 1920x1080	alpha: no bitrate: 5-300 bitdepth: 8 VBI: N/A timecode: N/A
DVCPRO	.avi	DVCPRO HD 1920x1080	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	Uncompressed YUVU 4:2:2 1920x1080	alpha: no bitrate: 950 bitdepth: 8 VBI: N/A timecode: N/A
Generic Separate	.avi	Uncompressed YUAVUA 4:2:2:4 1920x1080	alpha: no bitrate: 1425 bitdepth: 8 VBI: N/A timecode: N/A
DVCPRO Quicktime	.mov	DVCPRO HD 1920x1080	alpha: no bitrate: 100 bitdepth: 8 VBI: N/A timecode: N/A

Table 20:

Name	Container	Codec / Resolution		Audio
HDV	.mov	HDV 1440x1080	alpha: no bitrate: 25 bitdepth: 8 VBI: N/A timecode: N/A	
XDCAM (OP1a)	.mov	XDCam EX 1920x1080	alpha: no bitrate: 25CBR, 35VBR bitdepth: 8 VBI: N/A timecode: N/A	
XDCAM (OP1a)	.mov	XDCam HD 1440x1080	alpha: no bitrate: 35 bitdepth: 8 VBI: N/A timecode: N/A	
XDCAM (OP1a)	.mov	XDCam HD 422 1920x1080	alpha: no bitrate: 18-35 bitdepth: 8 VBI: N/A timecode: N/A	
DNxHD	.mov	DNxHD YUVU 4:2:2 1920x1080	alpha: no bitrate: 36, 145, 220 bitdepth: 8; 10 VBI: N/A timecode: N/A	
H.264	.mov	H.264 YUVU 4:2:2 1920x1080	alpha: no bitrate: VBR bitdepth: 8; 10 VBI: N/A timecode: N/A	
Generic Separate	.wav			PCM 2 ch: 24 in 32bit; 48kHz 4 ch: 24 in 32bit; 48kHz 8 ch: 24 in 32bit; 48kHz 16 ch: 24 in 32bit; 48kHz
Generic Separate	_vbi.avi	vbi 1920x1-15 Start Line: 6	bitdepth: 10 VBI: YUYV422	

5.3 Searching

This section contains the information on the following topics related to searching:

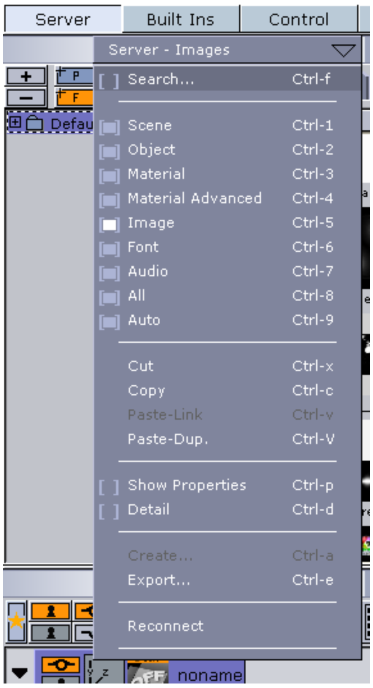
- Initiating a Search
- Search Area
- Search Editors
- Combination Searches
- SmartView Search
- Search Result Area
- Assigning Keywords

5.3.1 Initiating a Search

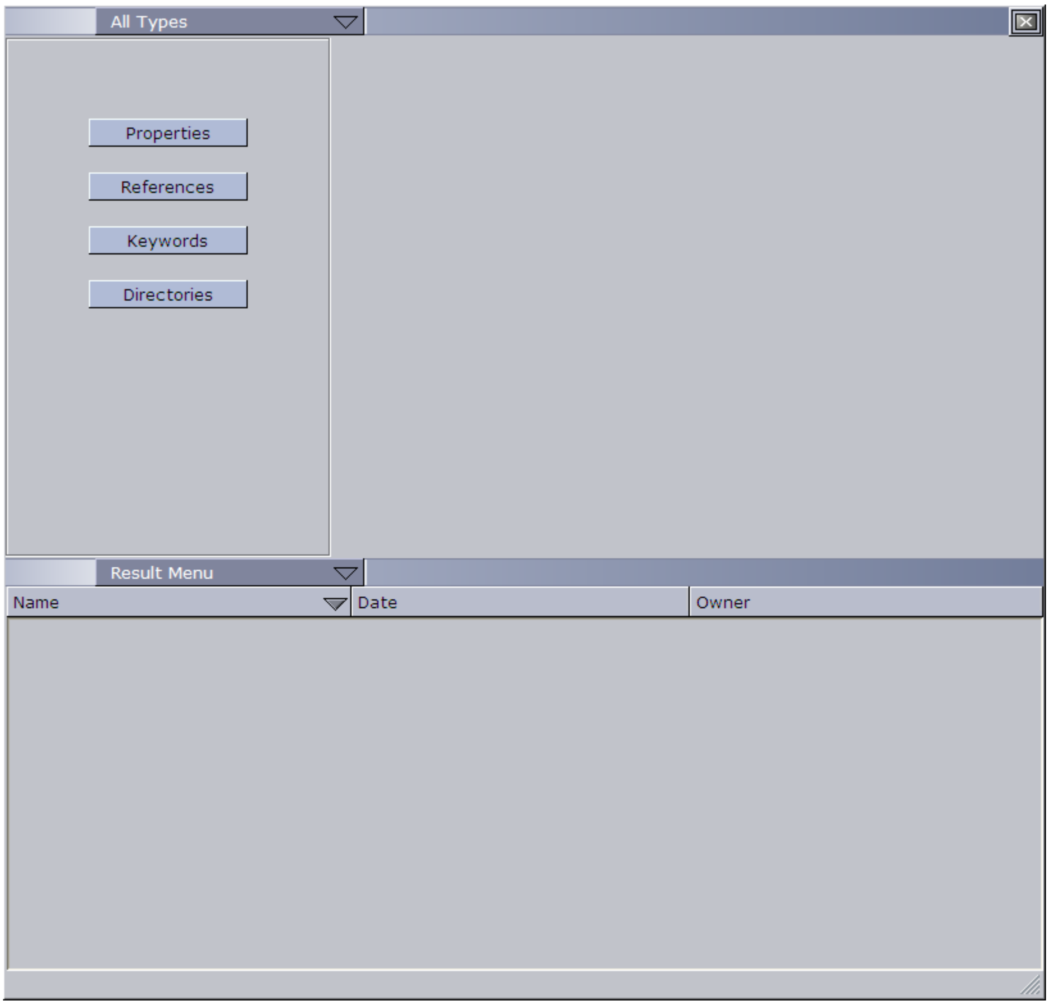
Note: Make sure to hold the mouse pointer over the Server area.

To open the search window

1. Click **Server**.
2. In the Server menu, click **Search**.



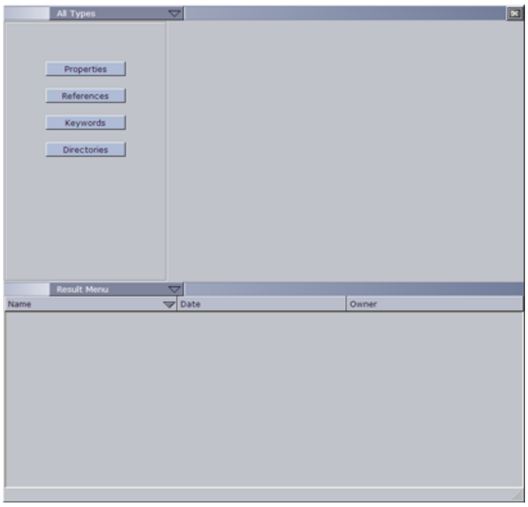
The search window is displayed.



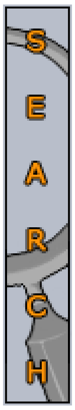
The Search window is divided in two areas; Search Area and Result Area.

5.3.2 Search Area

The Search area is positioned at the top of the Search window.



At the right side of the Search area are the following three buttons:

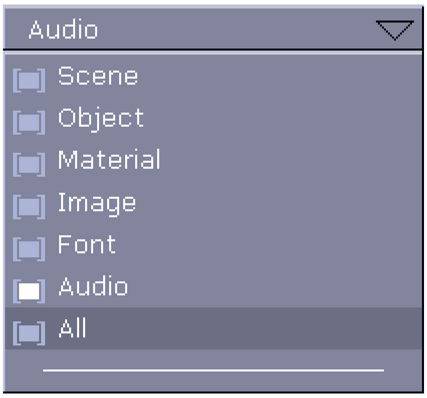


- **Search** Performs the actual search, and lists the hits in the Result area.

- **Close All Editors** Closes all search editors.

- **New Search** Clears all settings, so that a new search can be performed.

5.3.3 Search Menu



The Search menu is positioned at the top of the Search area. Selecting an element type from the menu narrows down the search. If for example Image is selected, only search hits that are images will be listed. Clicking All on the menu will not narrow down the search, but show hits from all element types. All is selected by default.

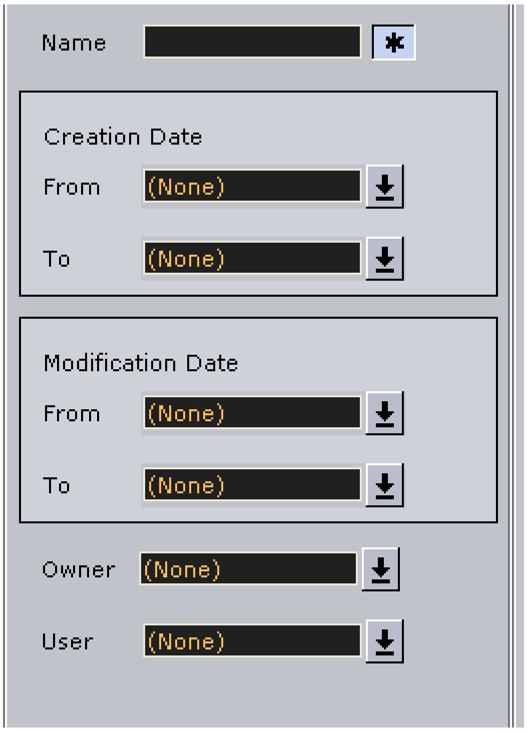
5.3.4 Search Editors

Elements can be searched based on various criteria. Four Search editors are located in the upper part of the Search window. Searches can be based upon four different criteria; Properties, References, Keywords, and Directories.

Tip: Up to three search criteria can be combined in one search request. The search will be performed using a logical AND of the selected criteria.

For information on keywords, see Assigning Keywords.

Properties



This editor searches the database for elements based upon properties.

- **Name** Searches for elements with the specified name. Clicking the asterisk allows you to perform wildcard searches, whereby you will get results that include your search criteria. With the asterisk disabled, only the exact search string is used.
- **Creation Date** Searches for elements that were created in the specified time period.
- **Modification Date** Searches for elements that were created in the specified time period.

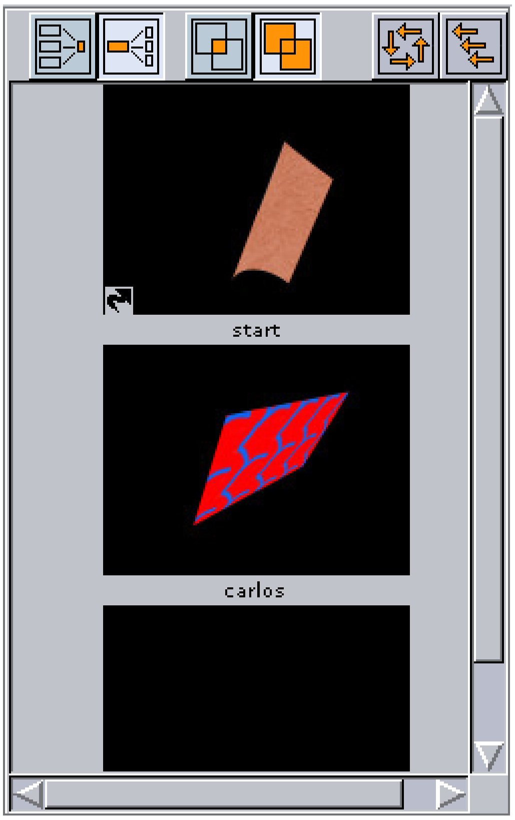
Note: The creation and modification dates can be set to; Today, 1 day, 1 week, 2 weeks, 1 month, or Older.

- **Owner** Searches for elements that were created by the specified user.
- **User** Searches for elements that were last modified by the specified user.

- Clears the settings in the Properties editor, so that a new search can be performed.

- Closes the Properties editor.

References

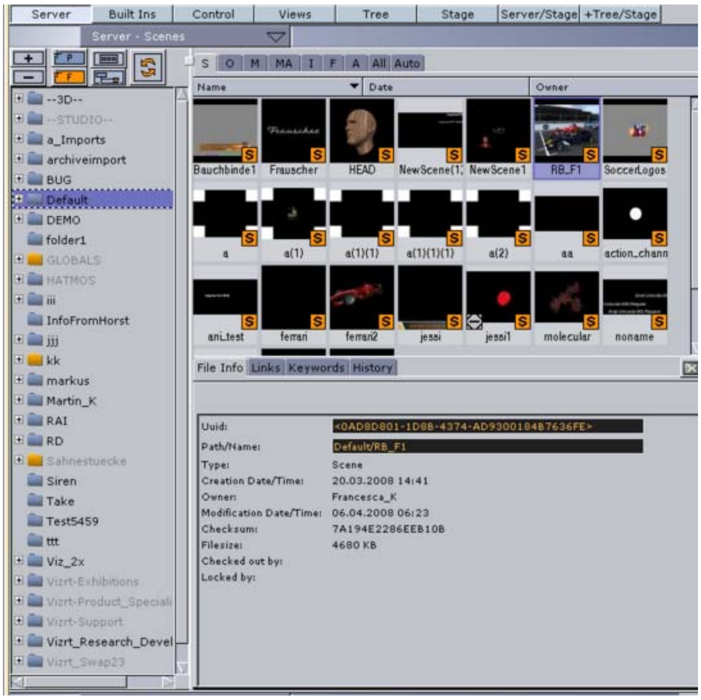


All graphics elements stored in Viz Graphics Hub have a unique ID. When elements from Viz Graphics Hub are used in a scene, they are imported by reference into that scene (and any others where they are used). Any object can reference to any other object in Viz Graphics Hub. To find out where any given Viz Graphics Hub element is used, you can search by reference.

To view an element’s unique ID

1. Click **Server**.
2. Locate the desired element, right-click it and select **Show Properties**.

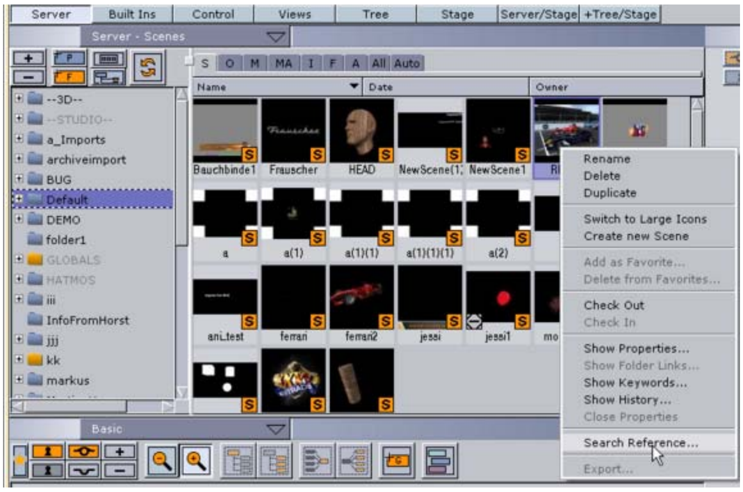
   The properties are displayed. Unique ID is shown in the red box.



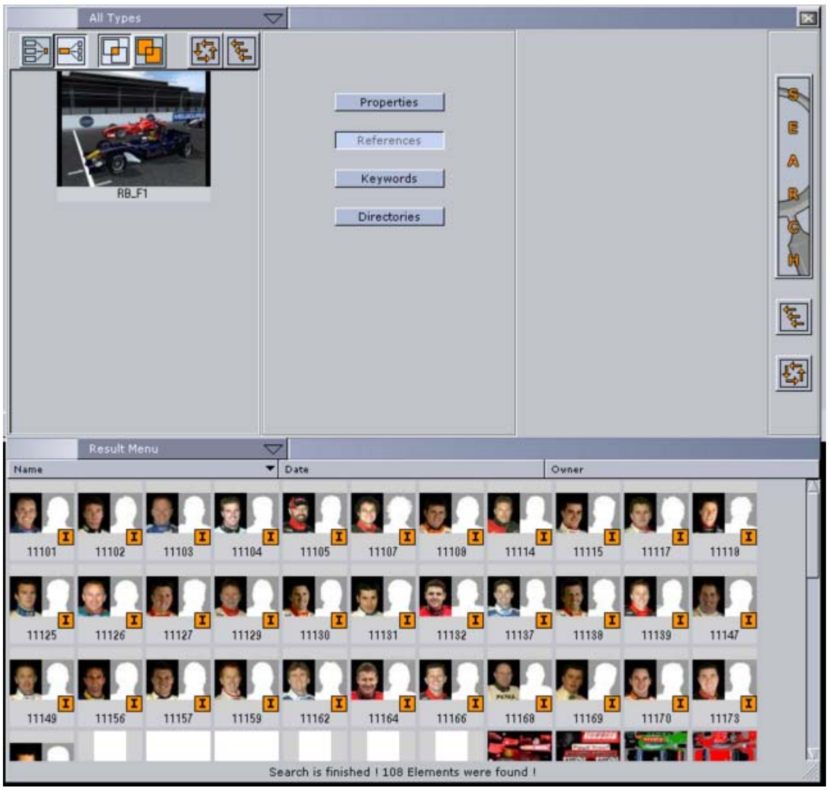
Uuid: <0AD8D801-1D8B-4374-AD9300184B7636FE>
Path/Name: Default/RB_F1
Type: Scene
Creation Date/Time: 20.03.2008 14:41
Owner: Francesca_K
Modification Date/Time: 06.04.2008 06:23
Checksum: 7A194E2286EEB10B
Filesize: 4680 KB
Checked out by:
Locked by:

To search by reference

1. Click **Server**.
2. Locate the desired element, right-click it and select **Search Reference**.



The results are displayed in the search editor.



Search is finished ! 108 Elements were found !

Note: It is possible to add multiple elements to the References editor.

- Searches for which objects use this object, e.g. images, geometries, fonts.

- Searches for which objects are used in this object, e.g. images, geometries, fonts.

- Performs a logical AND search of the elements that are added. The result must be in all searched elements.

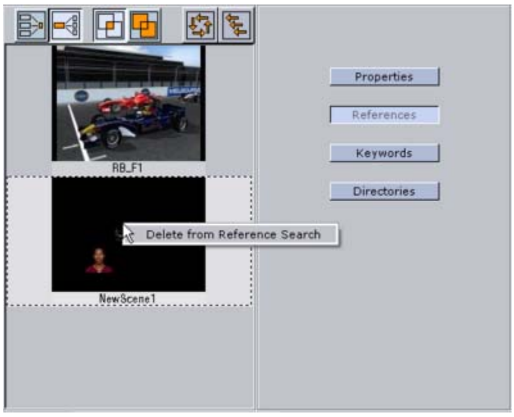
- Performs a logical OR search of the elements that are added. The result must be at least one of the searched elements.

- Clears the settings in the References editor, so that a new search can be performed.

- Closes the References editor.

To delete a element from a search by reference

- Right-click the element you wish to remove and select **Delete from Reference Search**.

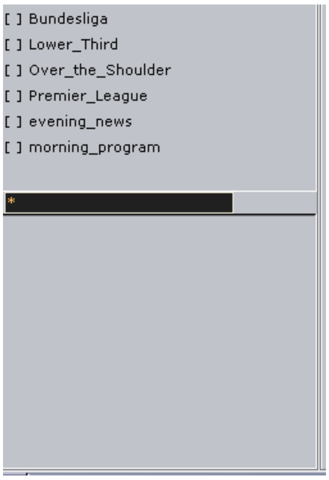


Keywords

This editor searches the database for elements based upon keywords.

To search for a keyword

1. Type the keyword in the input box, and press <Enter>.



[ ] Bundesliga
[ ] Lower_Third
[ ] Over_the_Shoulder
[ ] Premier_League
[ ] evening_news
[ ] morning_program

*

2. Do *one* of the following:
   - Click to select the desired keyword and drag it to the bottom pane.
   - Click inside the parentheses [] to the left of the desired keyword.

   The keyword search is performed on the keywords that have been moved into the bottom pane.

- Performs a logical AND search of the keywords that are added. The result must be in all searched elements.

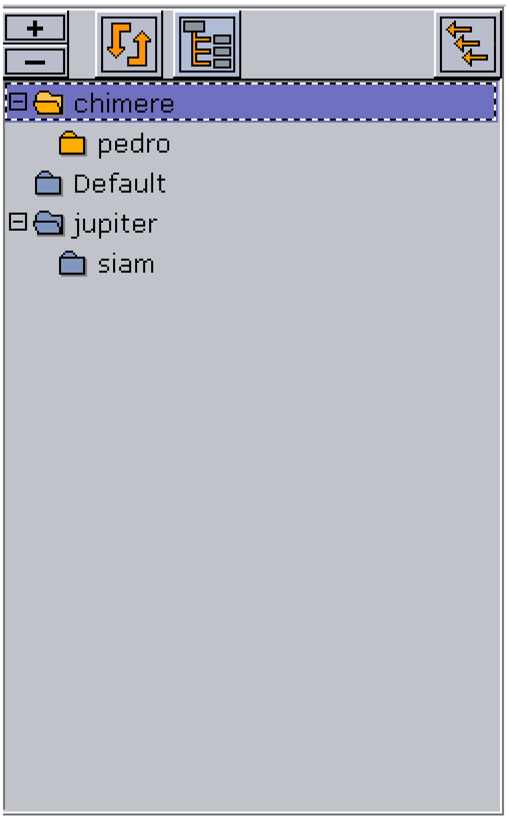
- Performs a logical OR search of the keywords that are added. The result must be in at least one of the searched elements.

- Clears the settings in the Keywords editor, so that a new search can be performed.

- Closes the Keywords editor.

Tip: Keywords can be assigned to elements from the Search window. To do so, select the keywords by highlighting them, and then drag them onto an element in the Server area. For more information, see Assigning Keywords.

Directories



This editor narrows down searches to the selected directory.

To search for elements, select a folder/project from the tree, and then click the Search button.

+

- Expands the currently selected branch and all its sub-branches.

Note: A branch that can be expanded has a Plus icon to the left of the folder/project name.

⊞

Tip: A branch can also be expanded by clicking the Plus icon to the left of the folder/ project name, but this will not expand the subbranches.

- Collapses the currently selected branch and all its subbranches.

Note: A branch that can be collapsed has a Minus icon to the left of the folder/ project name.

⊟

Tip: A branch can also be collapsed by clicking the Minus icon to the left of the folder/project name.

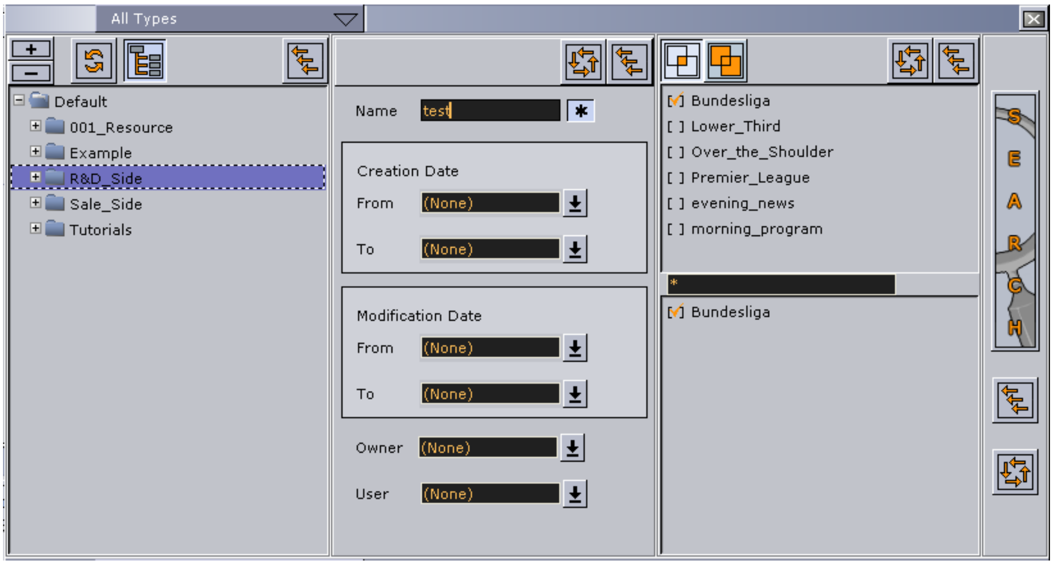
- Refreshes the tree.

- Searches for elements in the selected folder/project, and recursively in all sub folders/subprojects. When disabled, the search is performed in the selected folder only.

- Closes the Directories editor.

5.3.5 Combination Searches

You can further refine searches to file names in specific projects/folders with specified keywords. You can also have references, all within a single search.



Name: test

Creation Date
From: (None)
To: (None)

Modification Date
From: (None)
To: (None)

Owner: (None)

User: (None)

[✓] Bundesliga
[ ] Lower_Third
[ ] Over_the_Shoulder
[ ] Premier_League
[ ] evening_news
[ ] morning_program

*

[✓] Bundesliga

In the example above, the search will be conducted for files that contain the word **test**, in the **R&D_Side** folder and sub folders, having the keyword **Bundesliga** assigned.

5.3.6 SmartView Search

Note: This feature is only available with Viz Graphics Hub version 2.2 and up.

A more comprehensive search can be performed with the SmartView feature. A SmartView search includes any or all of the following:

- File names
- Folder names
- File owners
- File UUID (if enabled in Viz Graphics Hub)
- File type (geometry, font, images, etc.)
- Assigned keywords

Search String Syntax

SmartView search string syntax is very similar to Google search string syntax.

Example: jessi scene -project -folder

In the above example, the search will be conducted on any file names, file owners, UUIDs file types, and keywords that include the string `jessi` and `scene`. This search *excludes* projects and folders, plus any of the other searched items that include the words `project` and `folder`.

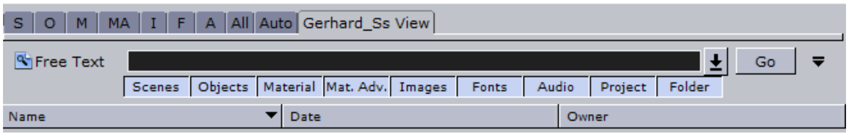
Example: francesca -image -geom

In the above example, the search will be conducted on any file names, file owners, UUIDs, file types and keywords that include the string `francesca`. This search *excludes* images and geometries, plus any of the other searched items that include the words `image` and `geom`.

Search String Builder Buttons

The SmartView search user interface includes a number of buttons that help you build search strings.

Figure 47: SmartView search string builder buttons



By clicking the Expand button (to the right of the Go button, highlighted above), the following search builder buttons are displayed:

- Scenes
- Objects
- Materials
- Mat. Adv. (Materials Advanced)
- Images
- Fonts
- Audio
- Project
- Folder

Note the following mouse and key shortcuts:

- Click on one or more buttons to select or deselect.
- **<Alt>** + **mouse click** activates only the button over which your mouse is placed.
- **<Ctrl>** + **mouse click** activates all buttons.
- **<Shift>** + **mouse click** deactivates all buttons.

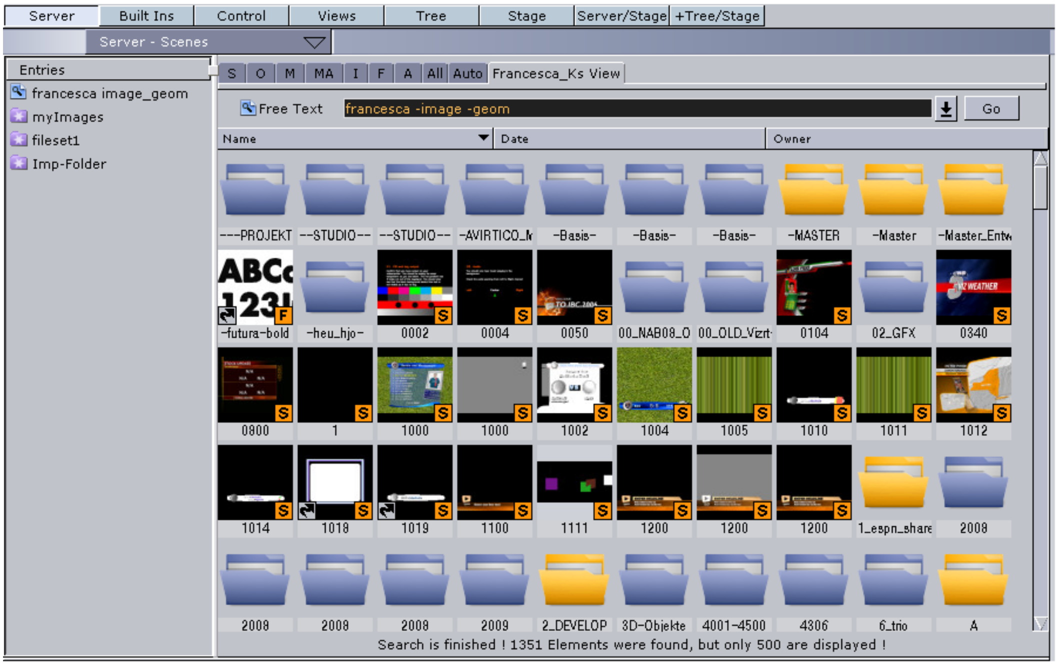
The buttons selected will refine the SmartView search within those particular items.

File Sets

File sets are a collection of objects saved under an assigned name. File sets are unique to each user on each instance of Viz Graphics Hub. Placing objects and folders in a file set does not move them from their location in Viz Graphics Hub hierarchy; rather it just makes them all accessible from the SmartView tab.

To initiate a SmartView search

1. Click **Server**.
2. Click the <**Username**> **View** tab.
3. In the Free Text field, enter a search string.
   See Search String Syntax and Search String Builder Buttons
4. Click the **Go** button (or press <Enter>).



francesca -image -geom

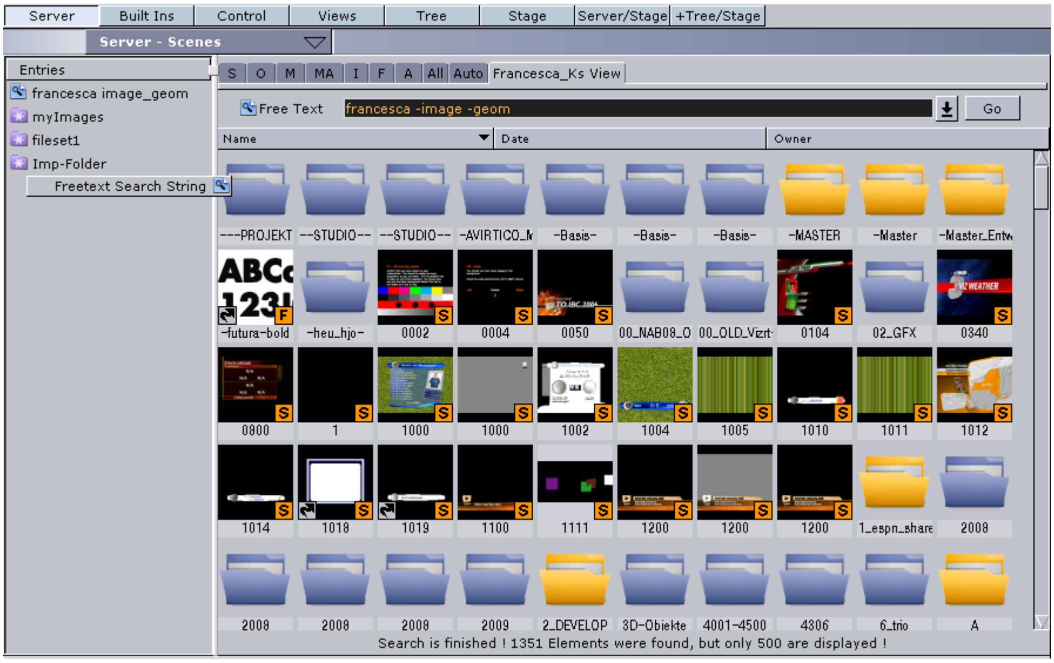
Search is finished ! 1351 Elements were found, but only 500 are displayed !

The results are displayed in the Server area.

- You can drag objects directly to where they are needed.
- You can double-click on folders to view sub folders and contents.
- You can right-click an object and select **Jump to Server View** to change the Entries column into the Viz Graphics Hub hierarchy. The object is then seen in its home directory.

To save a SmartView search

1. Drag the Free Text icon into the Entries column to the left of the SmartView area.



francesca -image -geom

Search is finished ! 1351 Elements were found, but only 500 are displayed !

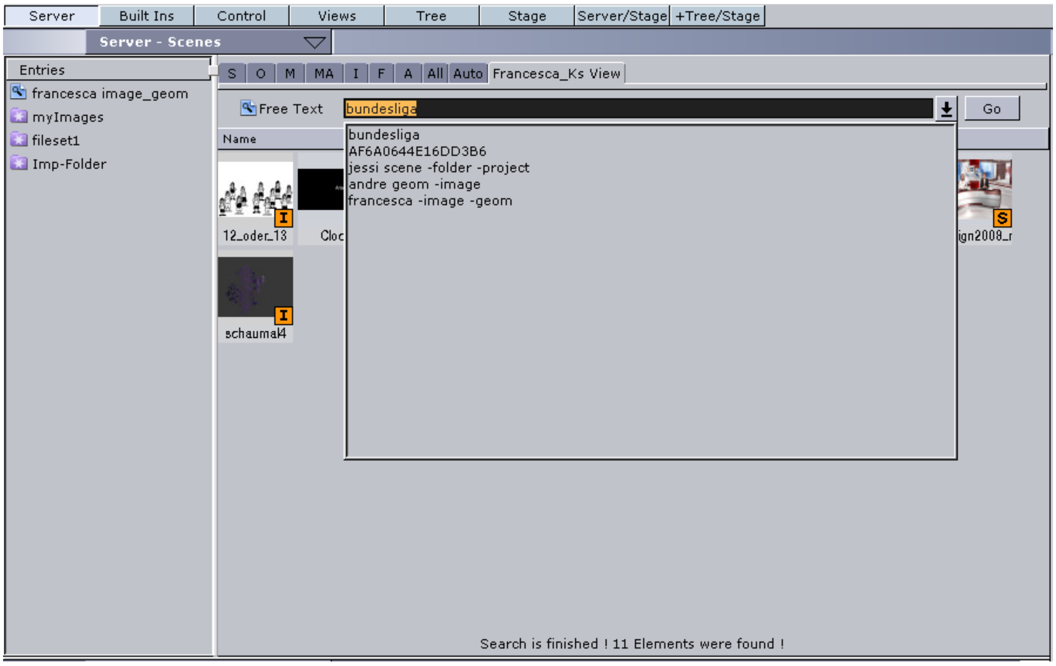
2. When prompted, give the search string a name and press <Enter>.

To use a saved SmartView search string

- In the Entries column to the left of the SmartView area, *click once* a saved search string to place it in the Free Text field without executing the search.
- In the Entries column to the left of the SmartView area, *double-click* a saved search string to place it in the Free Text field and execute the search.

To view previous SmartView searches

- Click the arrow to the right of the Free Text field to view all SmartView searches performed since Viz Artist was launched.



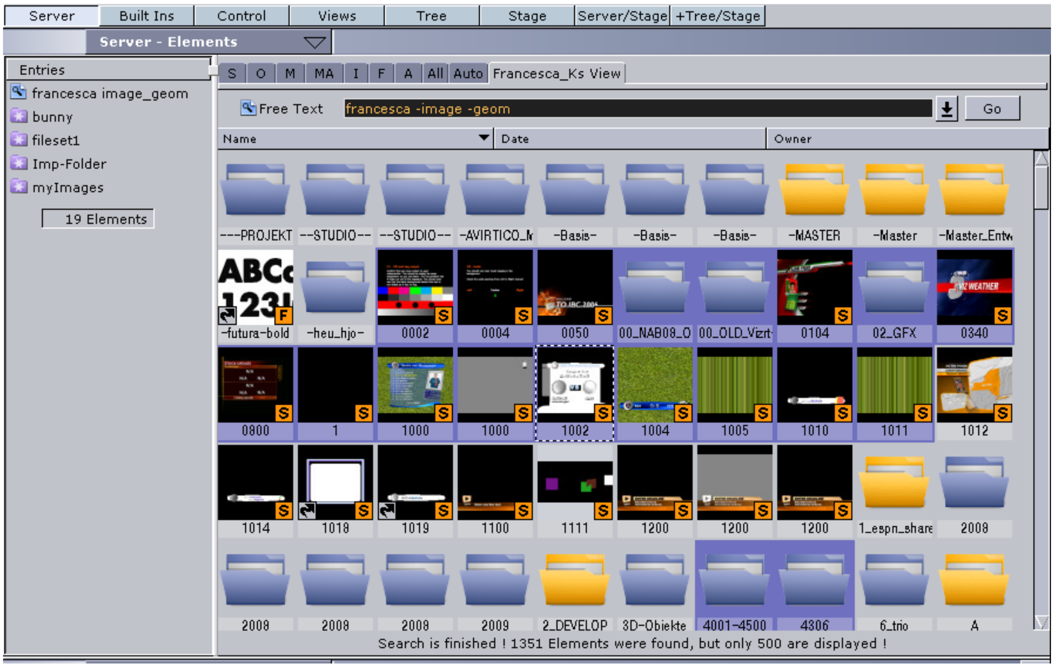
Free Text: bundesliga

bundesliga
AF6A0644E16DD3B6
jessi scene -folder -project
andre geom -image
francesca -image -geom

Search is finished ! 11 Elements were found !

To save a SmartView file set

1. Select all desired objects from a search result and drag them into the Entries column to the left of the SmartView area.



Search is finished ! 1351 Elements were found, but only 500 are displayed !

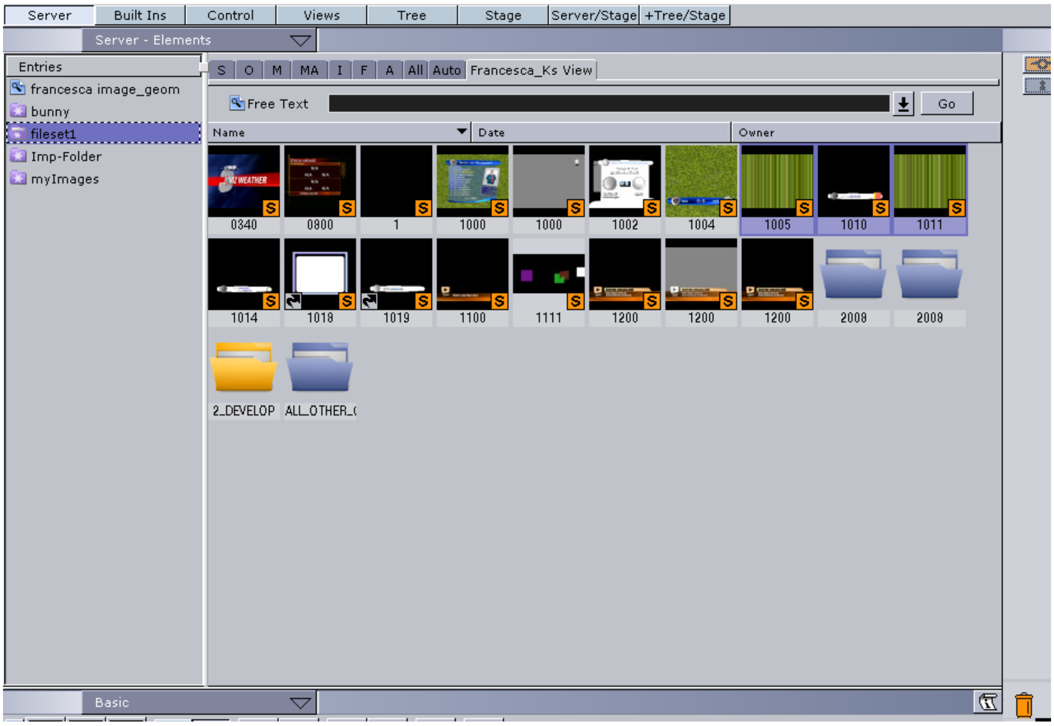
2. When prompted, enter a name for the file set and click **OK**.

To add files to a SmartView file set

- Select all desired objects from a search result and drag them into the target file set in the Entries column to the left of the SmartView area.

To remove files from a SmartView file set

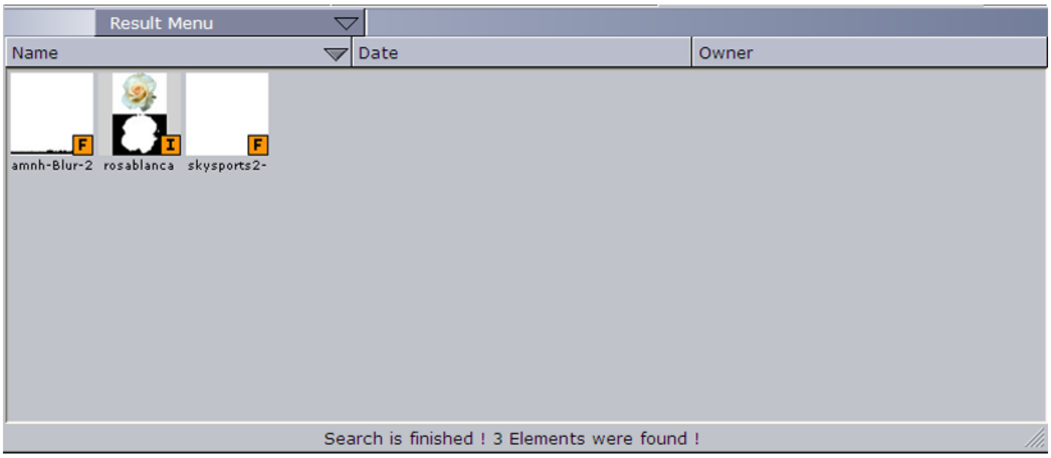
1. In the Entries column to the left of the SmartView area, click the relevant file set.



2. In the result area, select the desired objects, and do *one* of the following:
   - Drag the objects to the trash can.
   - Right-click the objects and select **Remove from file set.**

5.3.7 Search Result Area

The Result area is positioned at the bottom of the Search window.



Search is finished ! 3 Elements were found !

The Result area shows the search results.

Double-clicking an element in the Result area also selects the element in the Server area.

You can drag an element from the Result area directly into a scene tree for use. You can also drag them directly into the Favorites Bar.

Tip: Selecting multiple elements in the Search window and dragging them, manipulates all elements at once.

To move an element

- Drag the element from the Result list onto the target folder/project in the Server area.

IMPORTANT! Be careful when moving a linked element, as the element will be moved from the first folder/project found to the new destination.

To copy an element

- Press <Ctrl> and drag the element from the Result list onto the target folder/ project in the Server area.

To create a folder link

- Press <Alt> and drag the element from the Result list onto the target folder/ project in the Server area.

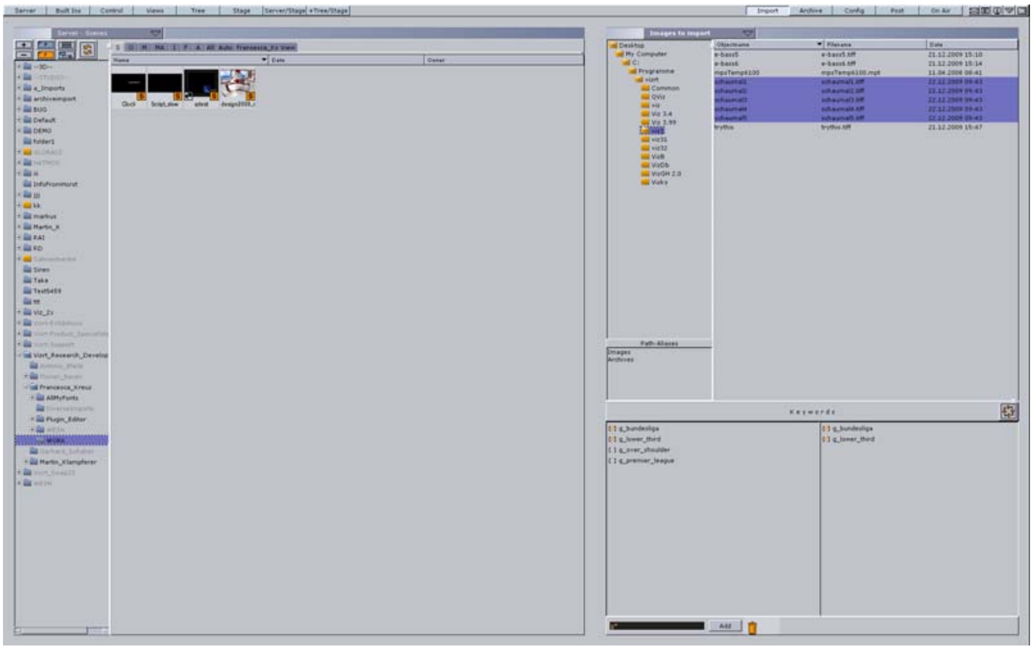
5.3.8 Assigning Keywords

When importing files into the database, you can add keywords to the file. When performing a database search, you can then search by keyword. Up to 20 keywords may be assigned to an element.

There are four ways to assign keywords. Each method starts from a different workflow point.

To assign a keyword while importing

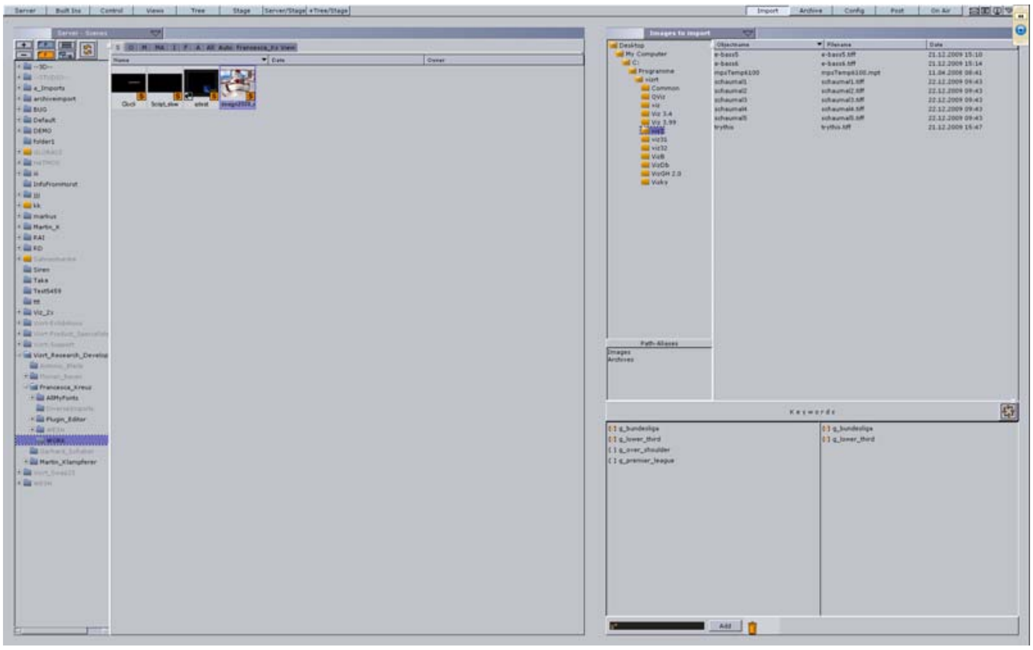
1. Click **Import**.
2. Click the source folder.
3. Search for keywords in the keyword search text field.
4. Select the keyword to search for.
5. Select the files to import.
6. Drag the selected files into the Server area.



To assign a keyword from the Keyword Editor

Note: This procedure assumes that you have already at least one object in the Server area.

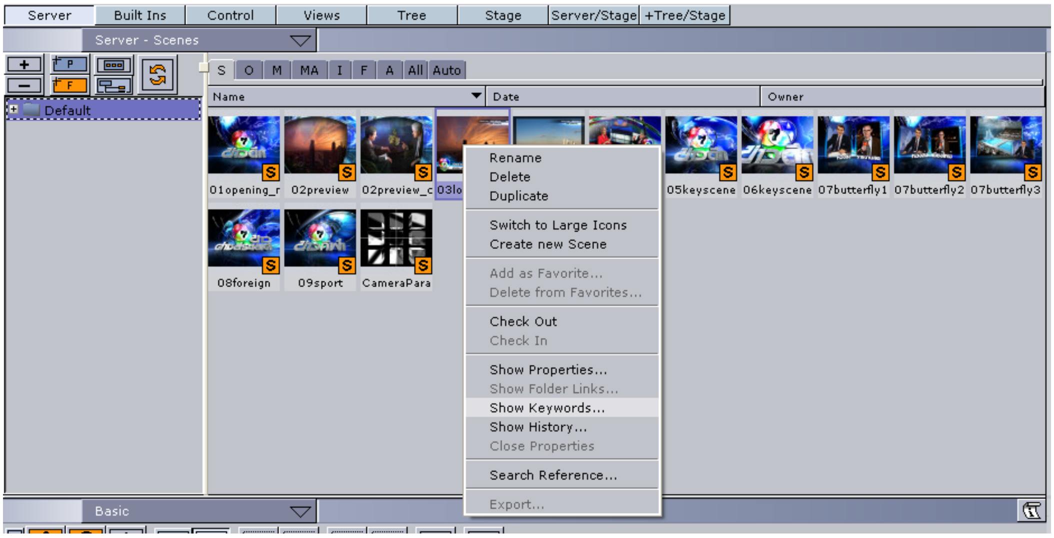
1. Search for keywords in the keyword search field.
2. Select the keyword to search for.
3. Drag the selected keywords into the Server area.
4. Click **Yes** to confirm that you want to assign the selected keyword(s) to the file(s).



To assign additional keywords from the Property Editor

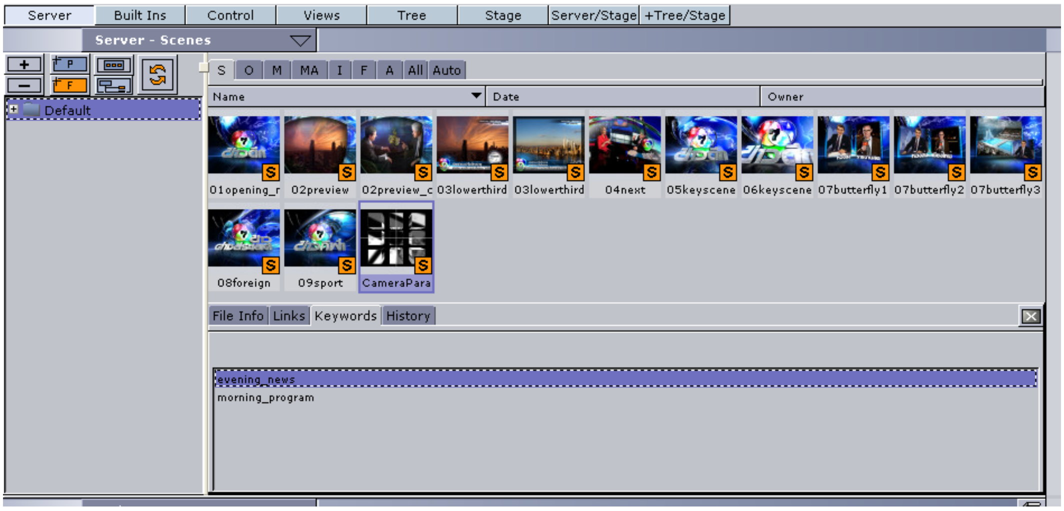
Note: This procedure assumes that you have files displayed in the Server area and that keywords are already assigned to them. You could use this method if, for example, you are creating a new scene and you want to replicate keywords from other objects.

1. In the Server area, right-click the source object and select **Show Keywords**.



Keywords assigned to the selected object are displayed in the pane below.

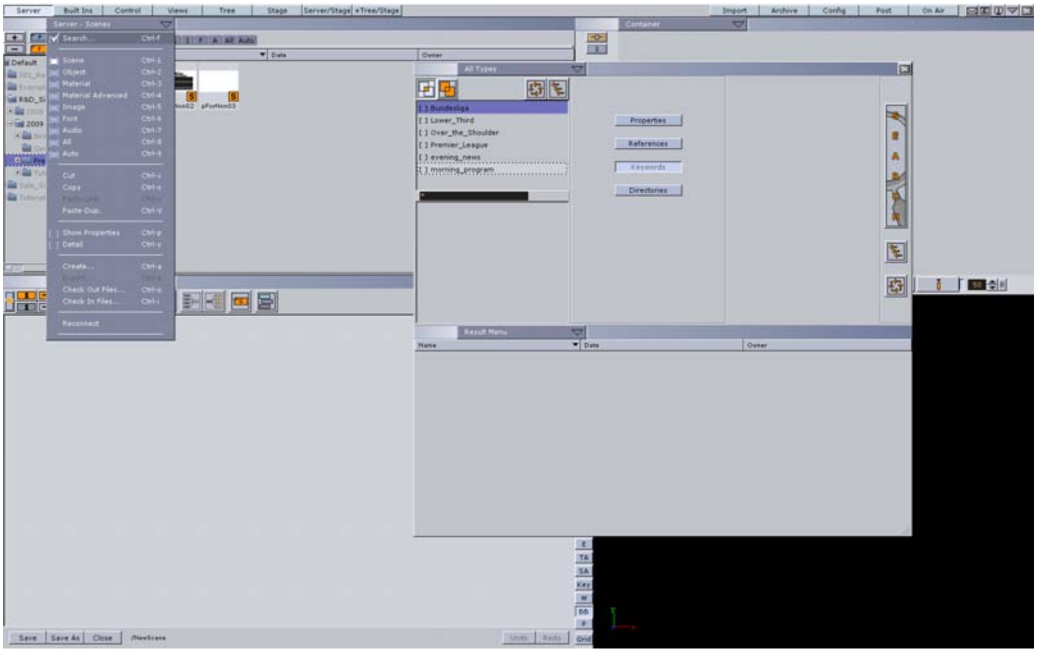
2. Click the desired keyword(s) and drag and drop onto the desired object in the Server Area.



Note: If you select more than one object in the Server area, and one keyword is common to all, but another is not, then the common keywords will be available, and the non-common keywords will be displayed, but greyed out.

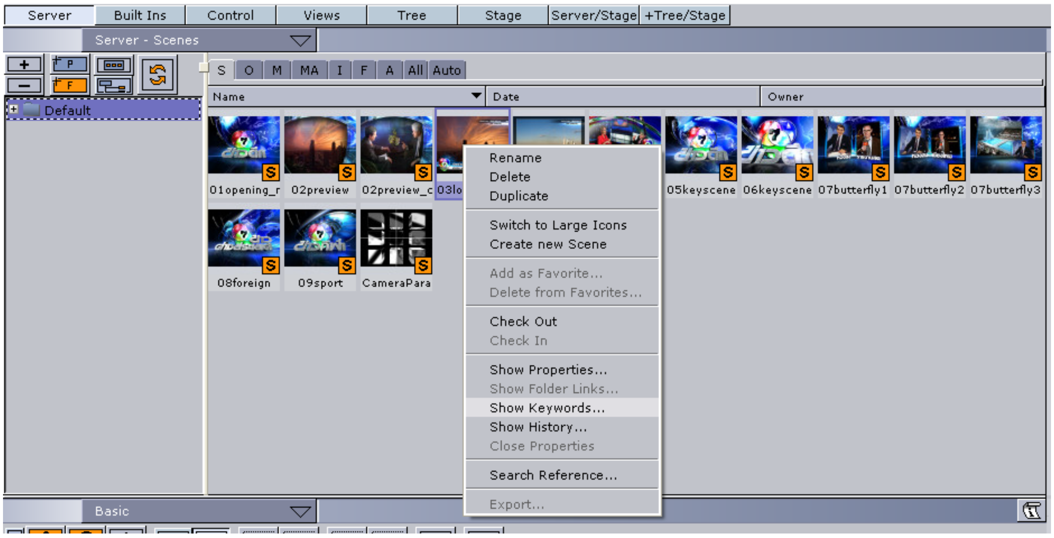
To assign a keyword from the Search Editor

1. Click **Server**.
2. From the drop-down menu, select **Search**.
3. Click the **Keyword** button.
4. Search for keywords in the keyword search field.
5. Select the keyword to search for.
6. Drag the selected keywords into the Server area.
7. Click **Yes** to confirm that you want to assign the selected keyword(s) to the file(s).



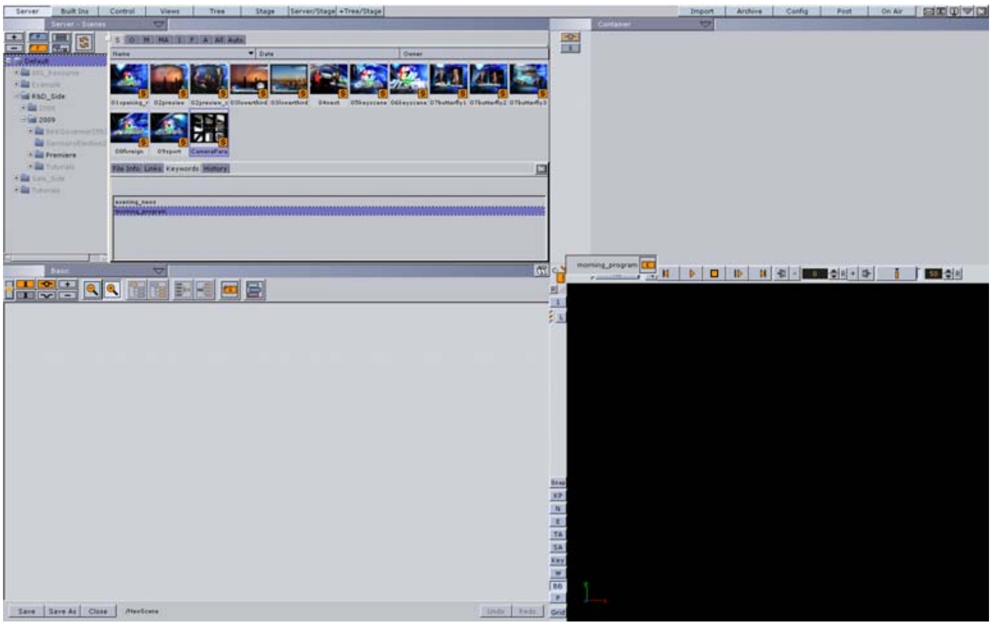
To remove a keyword from a file

1. In the Server area, right-click the source object and select **Show Keywords**.



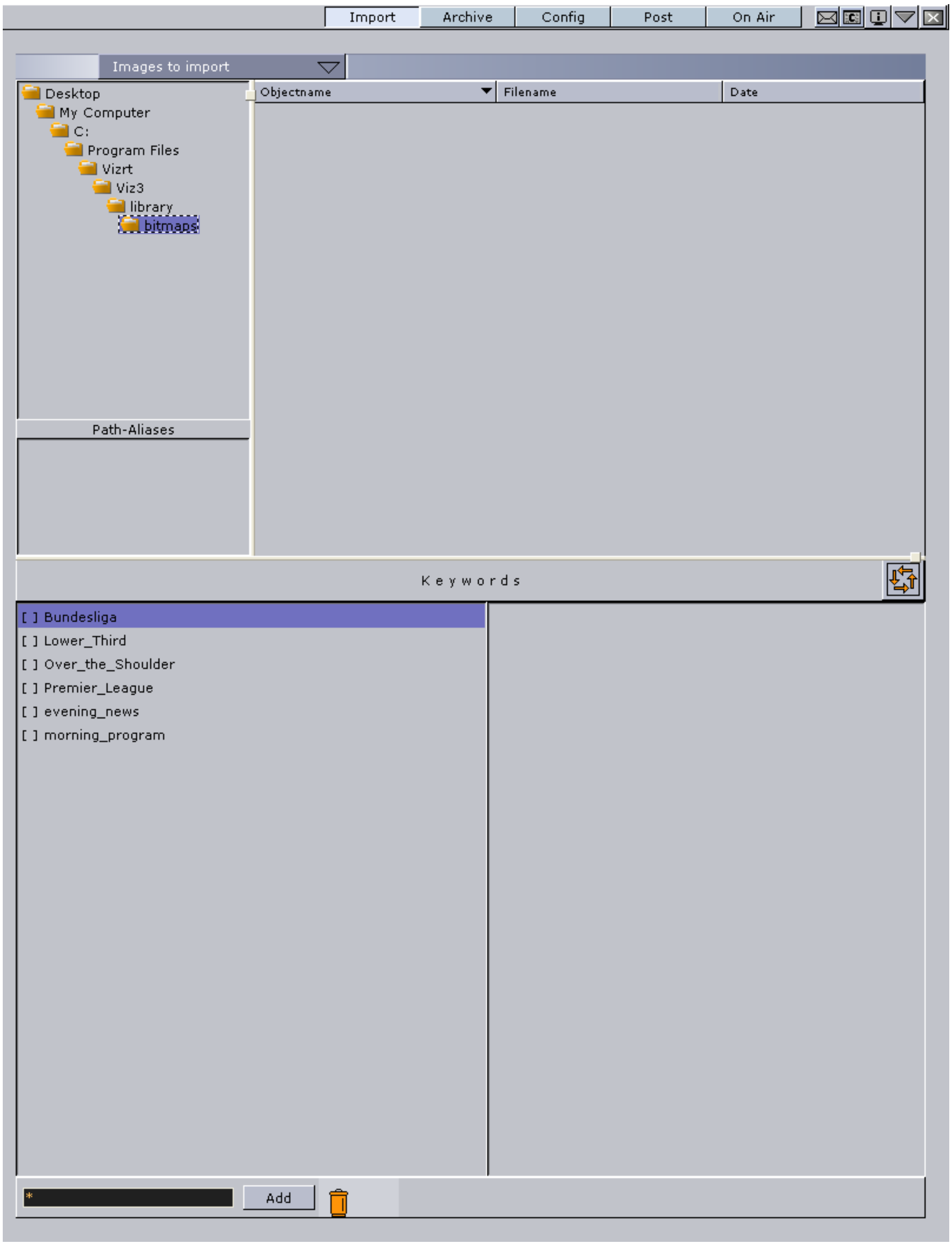
Keywords assigned to the selected object are displayed in the pane below.

2. Click the desired keyword(s) and drag and drop onto the trash can.



To remove a keyword from the Viz Graphics Hub

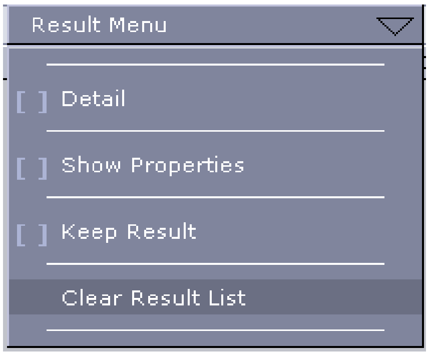
1. Click **Import**.
2. Search for keywords in the keyword search field.
3. Click the desired keyword and drag in onto the keyword editor’s trash can.



Note: Keywords can only be removed from Viz Graphics Hub if they are not assigned to any files.

5.3.9 Result Menu

The Result menu is positioned at the top of the Result area.

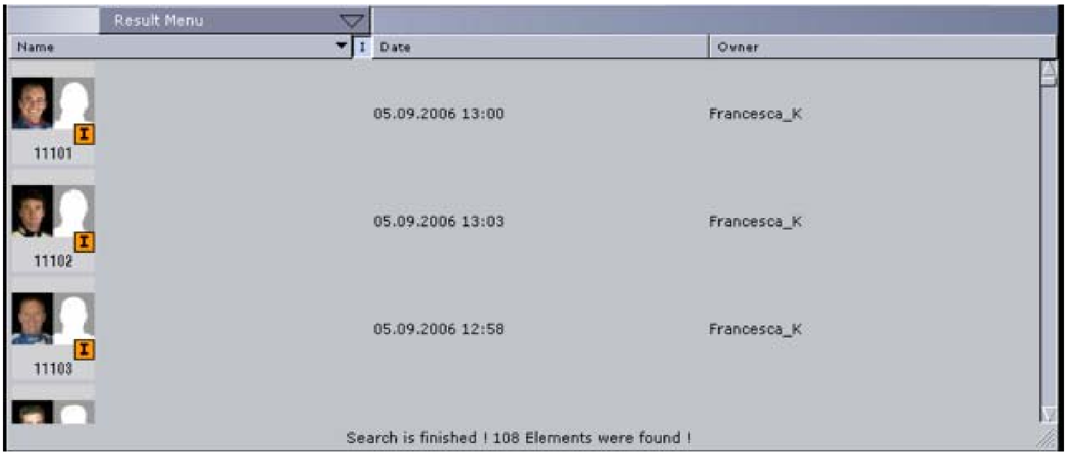


The Result menu defines the appearance of the results.

Detail By default, elements are presented in the Result list with thumbnail and element name. Clicking Detail shows a vertical list with the additional information of date of creation and username of element owner.

Clicking the I column heading enables/disables the thumbnails.

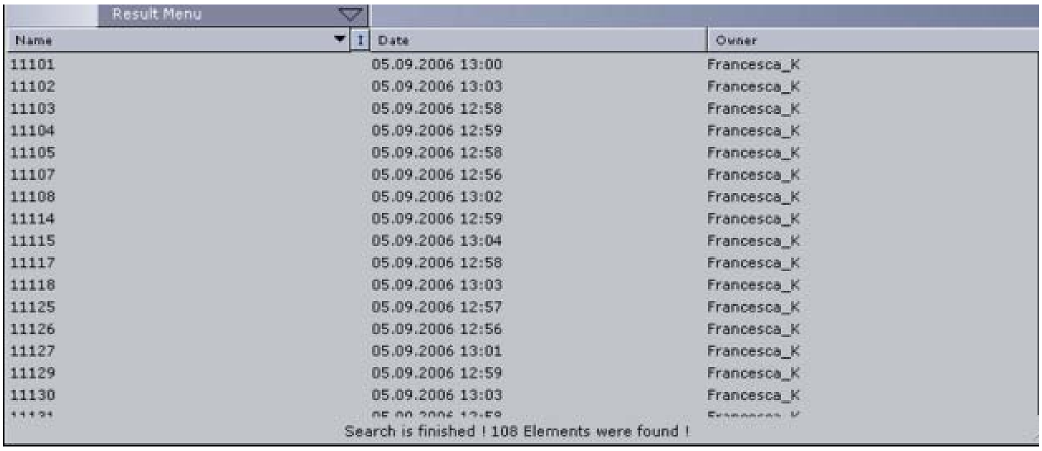
Figure 48: Thumbnails enabled



Name	I	Date	Owner
11101		05.09.2006 13:00	Francesca_K
11102		05.09.2006 13:03	Francesca_K
11103		05.09.2006 12:58	Francesca_K

Search is finished ! 108 Elements were found !

Figure 49: Thumbnails disabled



Name	I	Date	Owner
11101		05.09.2006 13:00	Francesca_K
11102		05.09.2006 13:03	Francesca_K
11103		05.09.2006 12:58	Francesca_K
11104		05.09.2006 12:59	Francesca_K
11105		05.09.2006 12:58	Francesca_K
11107		05.09.2006 12:56	Francesca_K
11108		05.09.2006 13:02	Francesca_K
11114		05.09.2006 12:59	Francesca_K
11115		05.09.2006 13:04	Francesca_K
11117		05.09.2006 12:58	Francesca_K
11118		05.09.2006 13:03	Francesca_K
11125		05.09.2006 12:57	Francesca_K
11126		05.09.2006 12:56	Francesca_K
11127		05.09.2006 13:01	Francesca_K
11129		05.09.2006 12:59	Francesca_K
11130		05.09.2006 13:03	Francesca_K

Search is finished ! 108 Elements were found !

Show Properties Displays the Properties Pane at the right side of the Result area, with the File Info tab selected.

Keep Result Keeps the current results in the Result list, while performing a new a search. The results from the new search will be added to the list.

Clear Result List Clears the Result list.

5.4 Importing

Viz Artist 3 offers the ability to import fonts, images, objects, scenes, audio clips, and archives from the source disk and network drives.

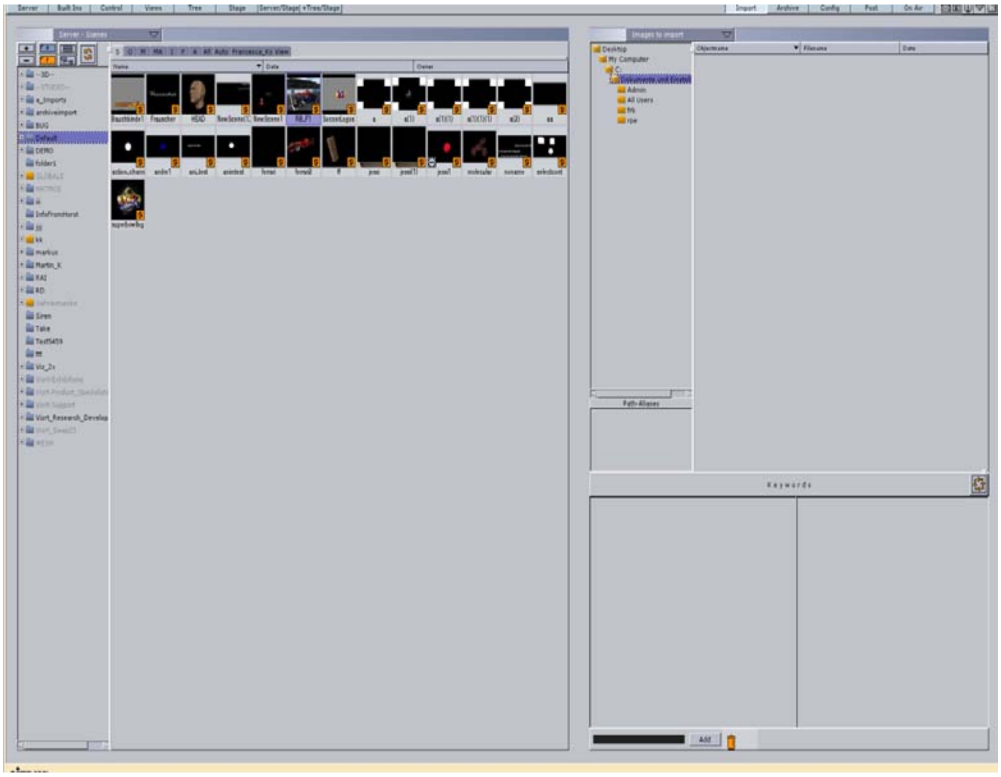
Note: If Viz Artist is running in Viz Engine mode, the import is limited to only import archives.

This section contains information on the following topics:

- Import Pane
- Import Menu
- Importing Files
- Importing Archives

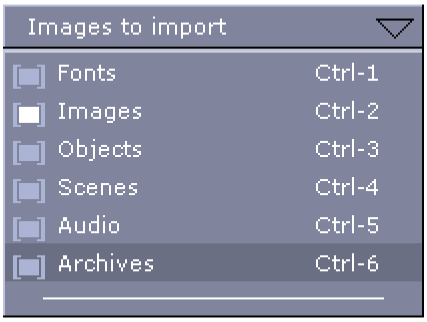
5.4.1 Import Pane

To open the Import pane, click Import from the Main menu.



The Server view is located at the left side of the Import pane, and gives access to all folders/projects in Viz Graphics Hub. To the right is the Import area that gives access to the source drives, where assets can be imported from.

5.4.2 Import Menu



The Import menu is positioned at the top of the Import pane. When selecting an element type from the menu, only the assets of that particular type will be listed in the directory tree.

5.4.3 Importing Files

Viz Artist allows you to import fonts, images, objects, scenes (from external applications), audio, and archives (entire Viz Artist scenes). The following file formats are supported.

Supported Font Types

- Adobe Type1 fonts (.pfb files)
- Adobe Type2 fonts
- True type fonts (recommended, most common format)

Supported Image Formats

- .bw
- .gif
- .hdr
- .png
- .ppm
- .psd (Photoshop, Layers defined in the Photoshop file will be imported as separate images into the database).
- .rgb (SGI’s)
- .rgba
- .tex
- .tga
- .tif
- .txc
- .xpm

Supported 3D Formats

- Wavefront .obj
- 3Dstudio .3Ds
- Autocad .dxf
- VRML 2.0
- VRML including animation

- Softimage .fbx (see Notes on Importing FBX Files)

IMPORTANT! All objects to be imported into Viz Artist must be “polygon objects”.

Supported Materials and Texture Vertices

- .obj
- .3Ds

Path Alias

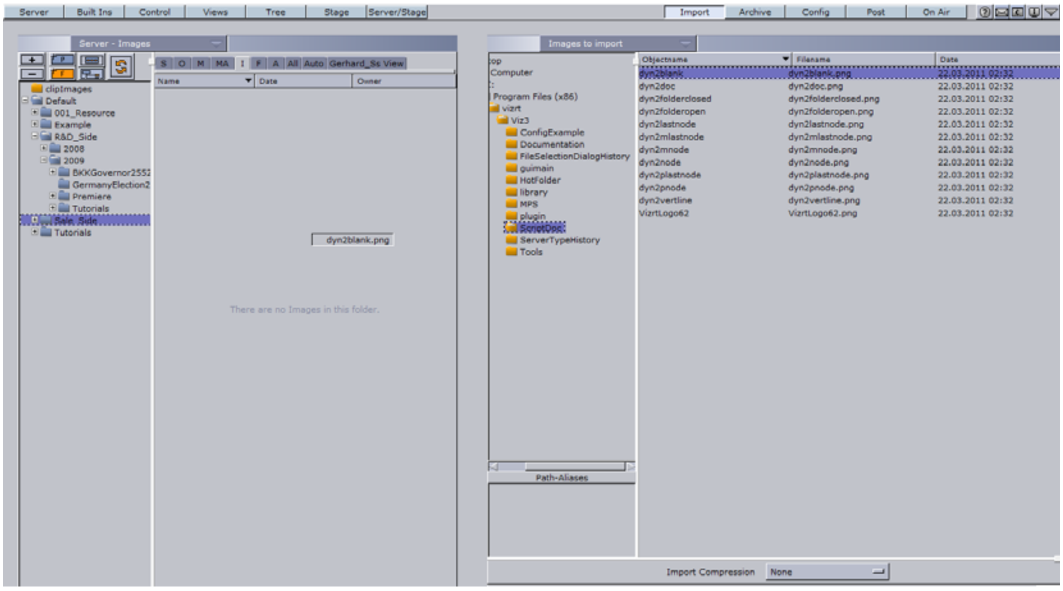
Using a path alias can help you streamline the import process. Defining Path Aliases pre-configures file paths that you often use for import operations.

Browsing for Files to Import

You can browse the directories on a disk using the directory tree on the right side of the import view. A double click on a directory will open it showing its sub directories. A single click on a directory displays the contents in the right window. Furthermore you can click on a path alias to quickly jump to this directory. For ease of use, multiple files may be selected by holding down the <Ctrl> key and clicking on all the desired files.

Importing the Selected Files

1. Click **Server**.
2. Open the target project/folder.
3. Click **Import**.
4. In the Import area, select the source folder and file(s).
5. For images only, select the image compression. Your options are:
   - **None**
   - **Select automatically**: If the source image has an alpha channel, DXT5 will be used. If the source image does not have an alpha channel, DXT1 will be used.
   - **DXT1**: Works best on images without an alpha channel.
   - **DXT5**: Works best on images with an alpha channel.
6. Drag the file(s) directly onto the project/folder in the Server area.



There are no Images in this folder.

Objectname	Filename	Date
dyn2blank	dyn2blank.png	22.03.2011 02:32
dyn2doc	dyn2doc.png	22.03.2011 02:32
dyn2folderclosed	dyn2folderclosed.png	22.03.2011 02:32
dyn2folderopen	dyn2folderopen.png	22.03.2011 02:32
dyn2lastnode	dyn2lastnode.png	22.03.2011 02:32
dyn2mlastnode	dyn2mlastnode.png	22.03.2011 02:32
dyn2mnode	dyn2mnode.png	22.03.2011 02:32
dyn2node	dyn2node.png	22.03.2011 02:32
dyn2plastnode	dyn2plastnode.png	22.03.2011 02:32
dyn2pnode	dyn2pnode.png	22.03.2011 02:32
dyn2vertline	dyn2vertline.png	22.03.2011 02:32
VizrtLogo62	VizrtLogo62.png	22.03.2011 02:32

Import Compression: None

Notes on Importing FBX Files

The FBX Importer reads Autodesk's exchange file format for 3D assets. FBX files (.fbx) are normally saved in a binary (or native) format, but they can also be saved in ASCII format. Binary FBX files and ASCII FBX files both use the .fbx filename extension.

Which of your scene feature you can expect to be carried over to a Viz scene depends on:

1. The FBX support of your 3D modeling application
2. The availability of the contained features in Viz Engine

- **Tips to avoid transformation being baked**

If unsupported features are detected the transformations of that nodes is baked. This means that individual transformations are not retained in Viz. You get the transformed vertices in Viz, but with all transformation parameters reset to their default values.

To avoid this you have to ensure that unsupported features are not employed in your scene and, ideally, that FBX properties that import into the same transformation in Viz are not used at the same time:

  - Don't use a separate scaling pivot

This might be tricky, depending on your 3D modeling application. In Maya the two pivots move together by default, which preserves the possibility to map them both to a single axis center in Viz Engine.

If your modeling app moves the rotation pivot separately by default, and you scale the object afterwards, you have two transformations based on different pivots and Viz Engine will have to bake the results to reflect them correctly.

Table 21: FBX Import: Avoid separate scaling pivot

Rotation and scaling pivot identical	Object rotated and/or scaled	OK, fully retained in Viz
Rotation and scaling pivot different	Object not scaled	OK
	Object scaled	**Baked**

- Don't use a scaling offset

An extra offset applied after rotation and before scaling. It depends on your modeling app if its available and under which name. To be on the safe side try to limit yourself to the regular translation and the rotation pivot.

You can built more complex transformation hierarchies and “simulate” multiple pivots by placing your objects in groups and apply additional transformations to the group(s). This is fully reproduced in the scene tree when importing to Viz Engine.

- **Tips to avoid transformations being combined**

If during import two FBX properties have to be combined into a single property in Viz they can't be animated separately anymore. Limitations are not fully specified yet, since animation import is still under development, but to retain 100% the same control over transformations as in your original scene it is a good idea to consider the following tips also:

- Don't use a rotation offset
- Don't use geometric translation/rotation/scaling in 3ds Max
- Don't use axis conversion

FBX uses an Y-up axis system, this is also the default setting for the FBX export. If your modeling application uses a different axis system it will apply pre-rotation of ± 90° to all objects to change the orientation of the scene to match the FBX axis system. This may interfere with animation.

Maya uses an Y-up axis system by default so all pre- and post-rotations will be set to zero, which is perfect. If your modeling app uses another axis system (e.g. 3ds Max uses Z-up) and you run into problems you may want to select a different option in the “Axis Conversion” section in the FBX export settings, so that no corrective rotation has to be applied.

Note: Import of lights are not currently supported.

5.4.4 Importing Archives

An Archive is a single file that holds various types of elements and remembers the project/folder they where stored in.

Note: Archives created with Viz Artist 2.x must not be imported to Viz Artist 3.x in this way. They need to be imported using the Viz GH Manager. See the separate Viz Graphics Hub documentation.

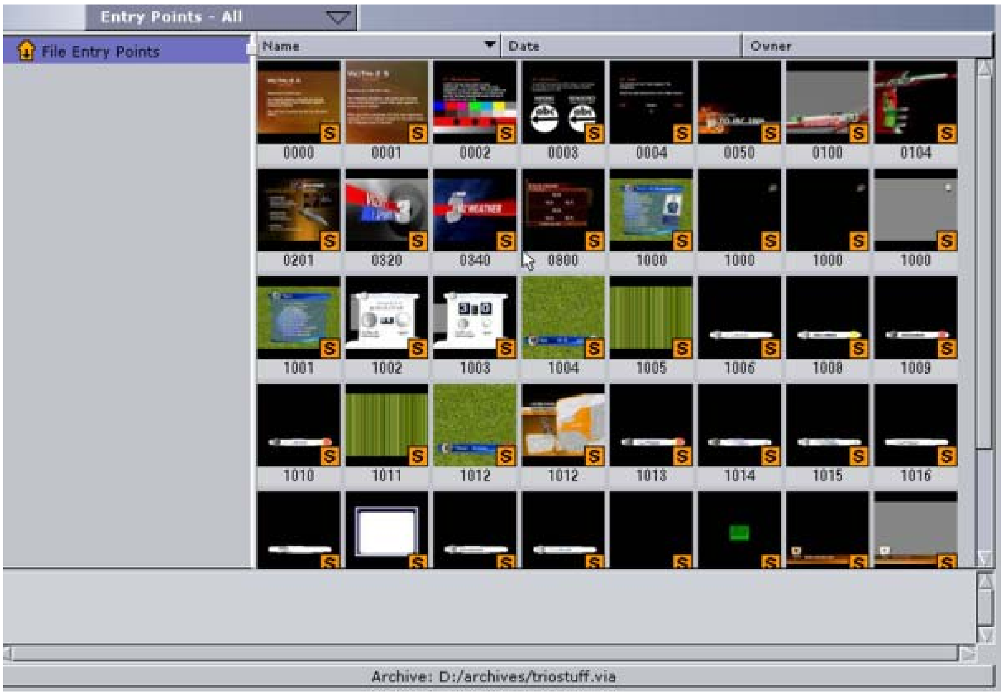
Path Alias

Using a path alias can help you streamline the import process. Defining Path Aliases pre-configures file paths that you often use for import operations.

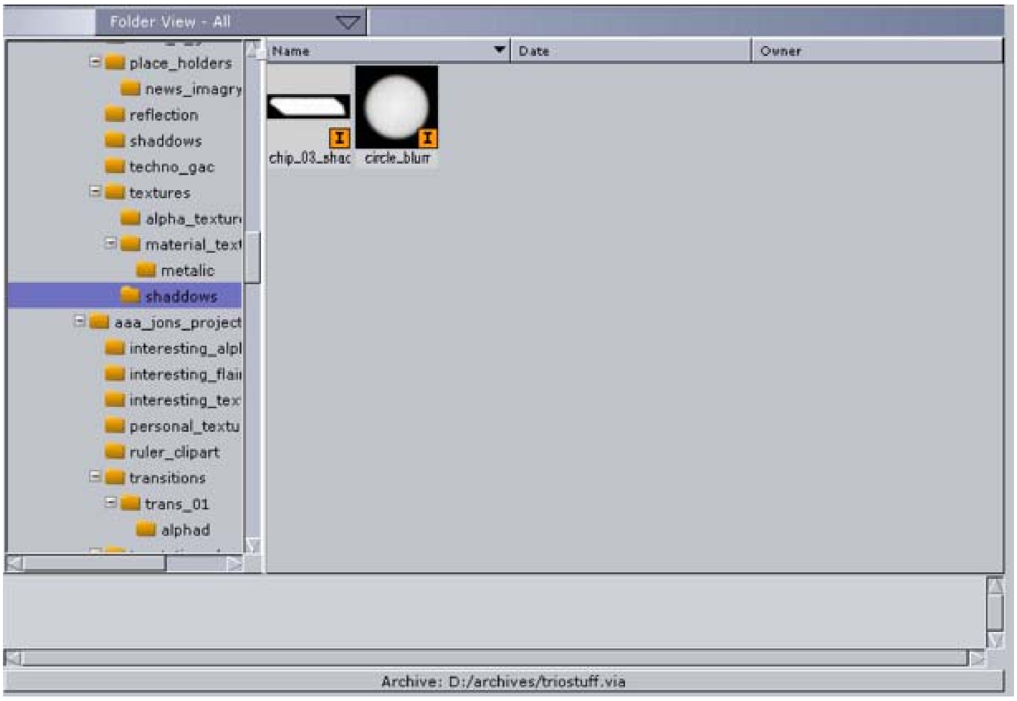
Archive Content Appearance

You can see differing aspects of an element’s composition by using the pull-down menu in the preview pane.

Entry Point View Will show you all Entry Points. An entry point is created for each element or project/folder you add to an archive via drag and drop.

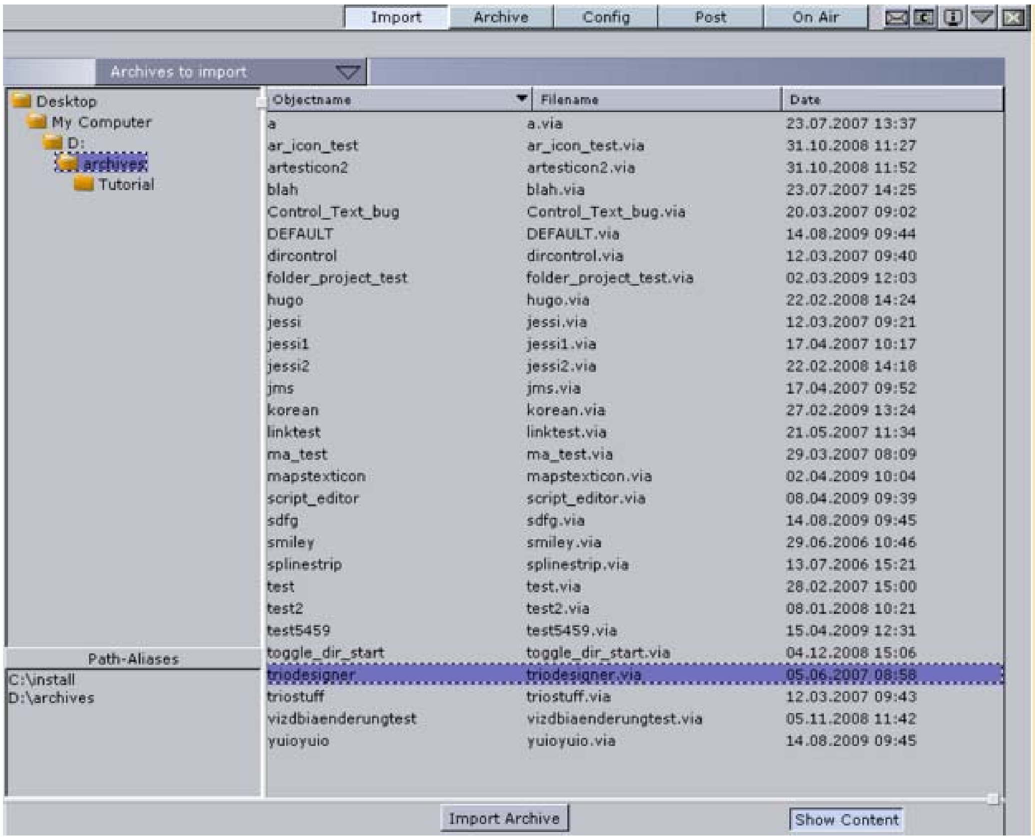


Folder View Will show you all the objects that are contained in the archive, including referenced objects as well.



To import an entire archive to the root of Viz Graphics Hub

1. Select the desired archive.
2. Click Import Archive.

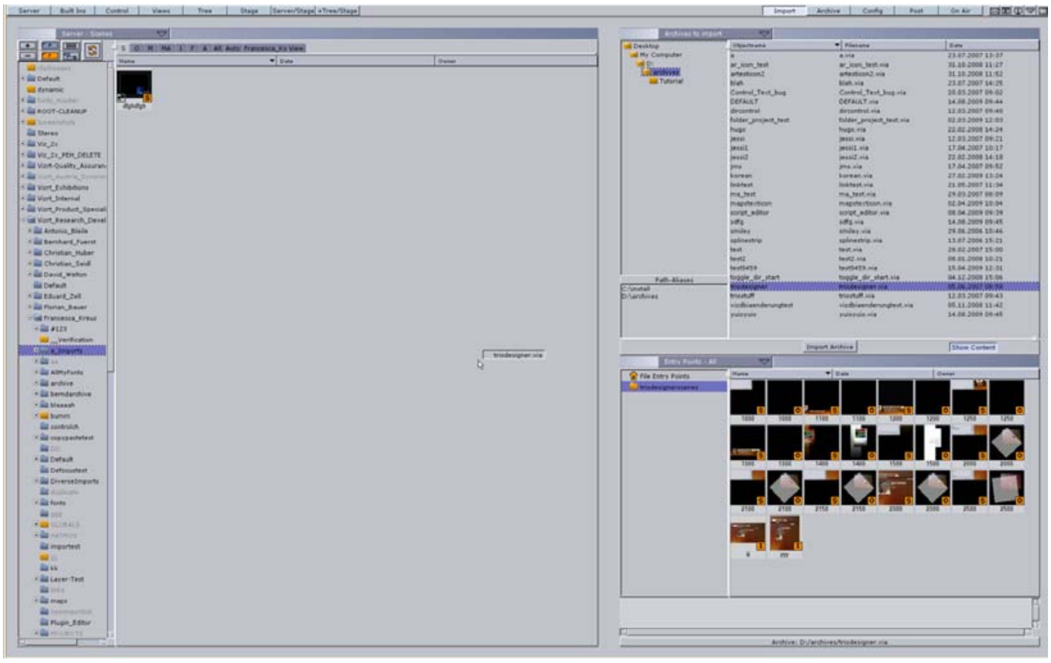


Objectname	Filename	Date
a	a.via	23.07.2007 13:37
ar_icon_test	ar_icon_test.via	31.10.2008 11:27
artesticon2	artesticon2.via	31.10.2008 11:52
blah	blah.via	23.07.2007 14:25
Control_Text_bug	Control_Text_bug.via	20.03.2007 09:02
DEFAULT	DEFAULT.via	14.08.2009 09:44
dircontrol	dircontrol.via	12.03.2007 09:40
folder_project_test	folder_project_test.via	02.03.2009 12:03
hugo	hugo.via	22.02.2008 14:24
jessi	jessi.via	12.03.2007 09:21
jessi1	jessi1.via	17.04.2007 10:17
jessi2	jessi2.via	22.02.2008 14:18
jms	jms.via	17.04.2007 09:52
korean	korean.via	27.02.2009 13:24
linktest	linktest.via	21.05.2007 11:34
ma_test	ma_test.via	29.03.2007 08:09
mapstexticon	mapstexticon.via	02.04.2009 10:04
script_editor	script_editor.via	08.04.2009 09:39
sdfg	sdfg.via	14.08.2009 09:45
smiley	smiley.via	29.06.2006 10:46
splinestrip	splinestrip.via	13.07.2006 15:21
test	test.via	28.02.2007 15:00
test2	test2.via	08.01.2008 10:21
test5459	test5459.via	15.04.2009 12:31
toggle_dir_start	toggle_dir_start.via	04.12.2008 15:06
triodesigner	triodesigner.via	05.06.2007 08:58
triostuff	triostuff.via	12.03.2007 09:43
vizdbiaenderungtest	vizdbiaenderungtest.via	05.11.2008 11:42
yuioyuio	yuioyuio.via	14.08.2009 09:45

Note: Pressing the Show Content button will display the archive's content in the pane below. Since sometimes an archive may take some time to display, you can disable the Show Content button for faster performance.

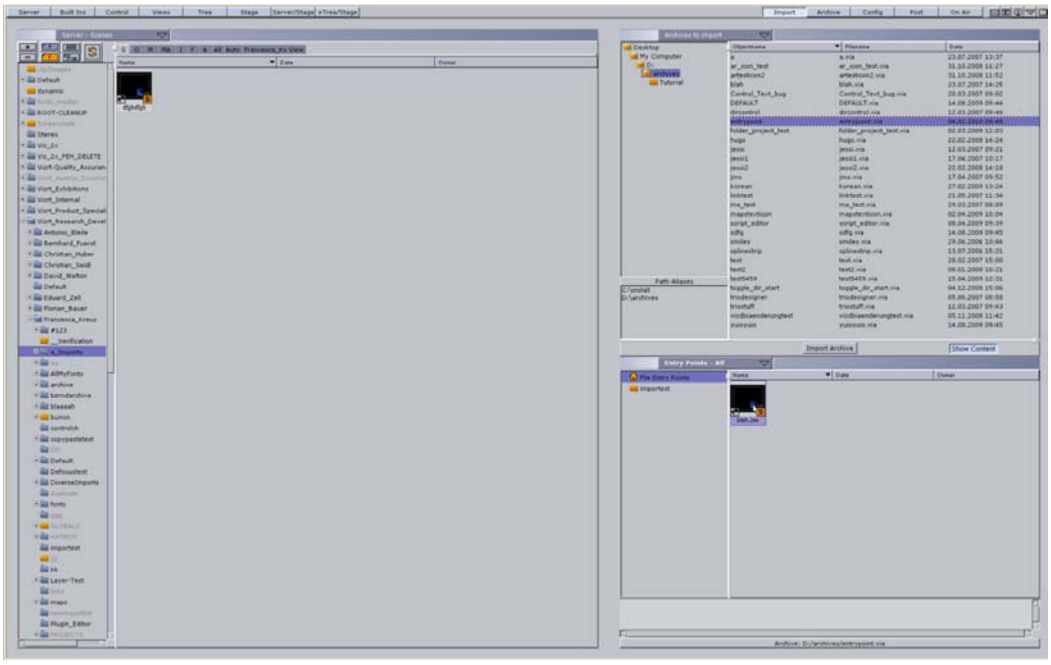
To import an entire archive to a specific location in Viz Graphics Hub

- Select the desired archive and drag it into the desired location in Viz Graphics Hub.



To import a partial archive

1. Select the desired archive.
2. From the pull-down menu, select **Entry Points View**.



3. Select one of the following options:
   - To import all entry point files from the archive, drag the **File Entry Points** file folder and drop it in the left pane.
   - To import a folder and its contents from the archive, drag the desired folder from the folder entry points and drop it in the left pane.
   - To import one or more files, drag and drop them into the left pane.

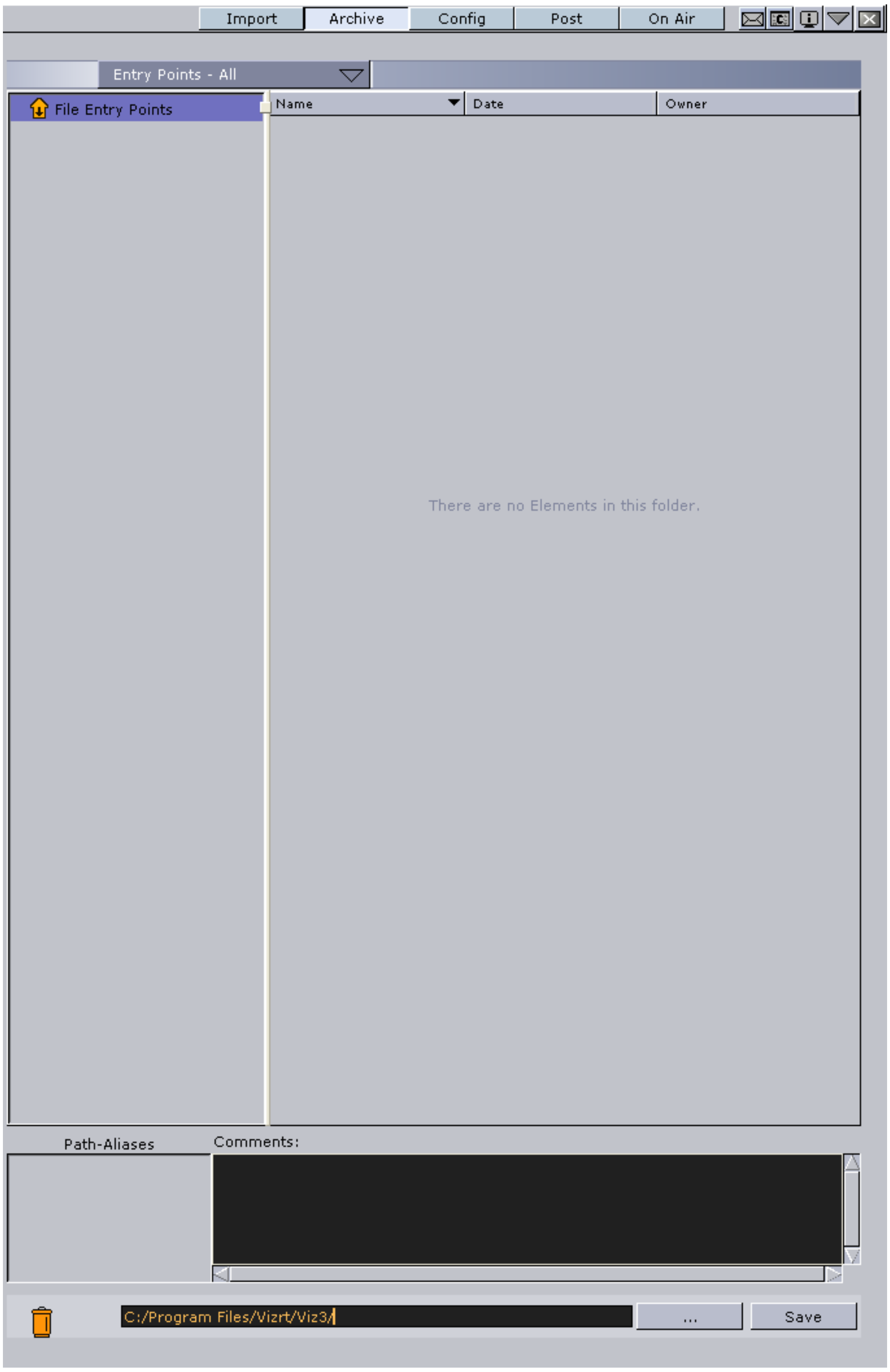
   The archive is imported into the folder selected in the Viz Graphics Hub tree. All reference files are included to guarantee full functionality.

Note: To import projects/folders, the same procedure applies, with the exception that you select **Folder View** from the pull-down menu, and drag and drop the desired folder.

5.5 Archiving

Archiving a scene allows you to export it. To open the archive view, in the main menu, click **Archive**. The archive view is displayed in the right pane.

Figure 50: Archive view

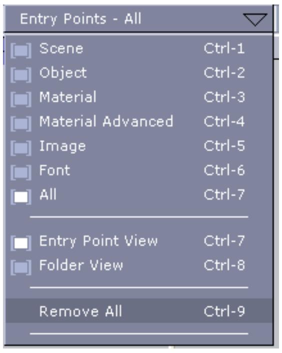


- The pull-down menu is located just below the main menu.
- The left pane is the target area for folders you will archive.
- The right pane is the target area of source files to archive.
- In the bottom pane, there is the Path Aliases, which gives you quick access to directories in your source disks. There is also the Comments area for assigning comments to the archive you want to save.
- At the very bottom is the trash can, for removing files from archives. To the right, there is a field to assign names to archives, with Browse and Save buttons.

Entry points represent the main objects in an archive. Other objects in the archive are referenced and attached automatically to the archive.

Pull-down Menu

At the top of the archive view there is a pull-down menu enabling you to switch the way the content of your archive is shown.

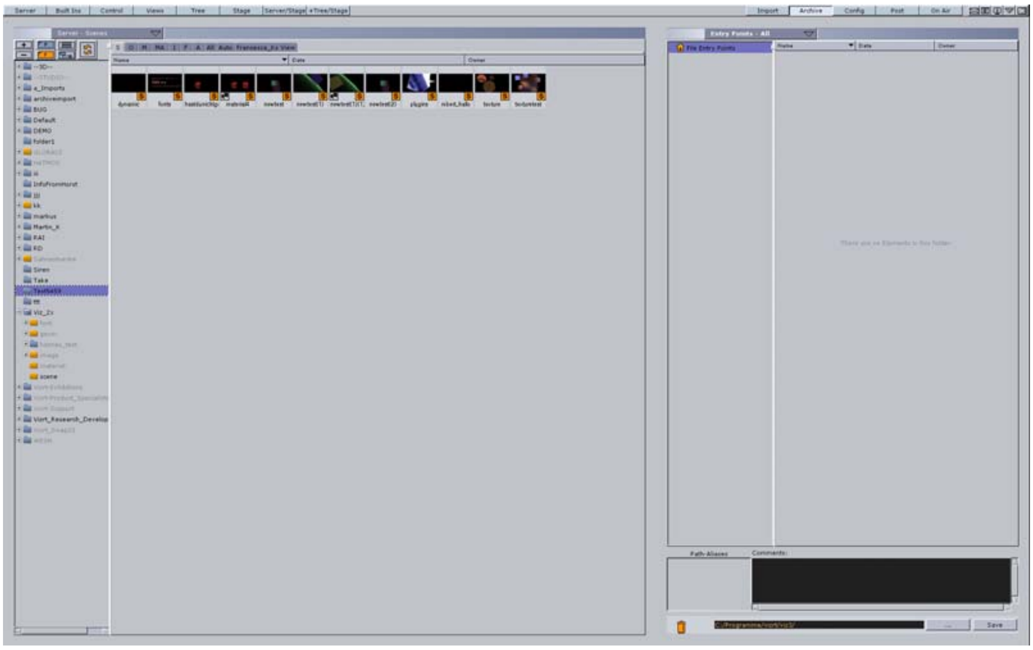


- **Scene to Font** Only shows the elements of the selected file type.
- **All** Shows elements of all file types.
- **Entry Point View** Shows only entry points.
- **Folder View** Shows the folder hierarchy.
- **Remove All** Removes all elements from the archive.

5.5.1 Creating an Archive

To create an archive

1. In the main menu, click **Archive.**
   The Archive view is displayed.



2. From the Server side, drag folders into the Archive view's left pane and files into the right pane.
3. If required, enter a comment in the comments area.

Note: Comments can be a maximum of 255 characters.

4. Either manually or by clicking the ellipsis button (...), enter a file path for the archive.
5. After the file path, enter a file name for the archive.
6. Click the **Save** button.

IMPORTANT! When archiving a set of transition logic scenes in Viz Artist, the archive will not include the geometry (GEOM) objects that are generated by Viz Template Wizard when creating a template for the control applications. It is therefore recommended to archive scenes intended for Viz Content Pilot by using Viz Content Pilot or Viz Template Wizard. If Viz Artist is used, the objects can be added manually.

5.6 Post Rendering

Viz Artist scenes can be exported as video or image files. This process is called post rendering.

Note: Post rendering is not possible with clips or live video feeds. However, you can post render clips through the Soft Clip plugin if it is animated through the stage.

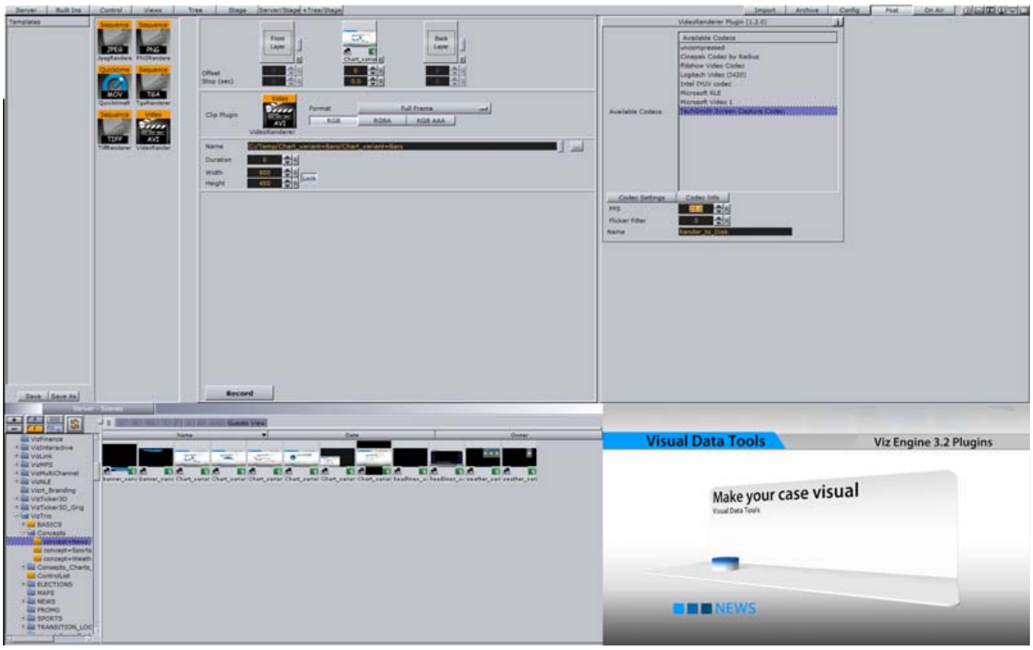
This section contains information on the following topics:

- Post Rendering Screen
- Post Rendering a Scene

- Post Rendering Templates
- Post Rendering Plugins
- Advanced Issues with Video Codecs
- Post Rendering to Targa playable Clips

5.6.1 Post Rendering Screen

Figure 51: Post rendering screen



In the post render video you will find:

- Template (top far left)
- Post rendering setup (top left)
- Server area (bottom l eft)
- Plugin property editor (top right)
- Output (bottom right)

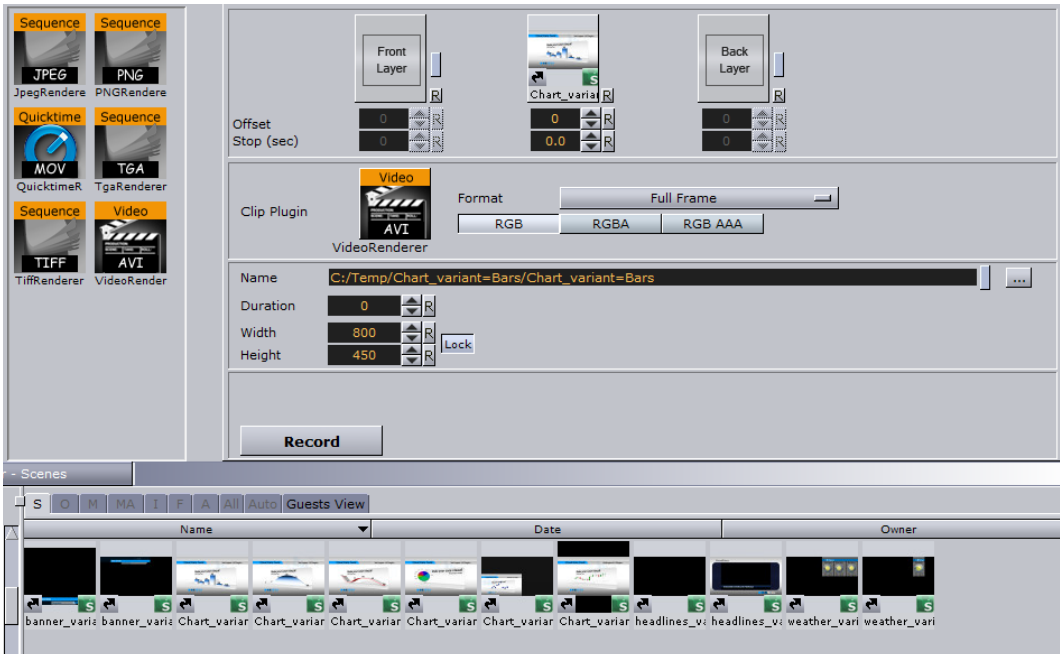
- On the far left is the bar where all installed plugins are listed. Additional plugins can be installed at any time.
- **Main Layer** shows the scene to be rendered. If you switch to post rendering, the scene you currently worked on will be set as the main layer.
- **Front Layer** enables you to set a scene running in front of the main layer.
- **Back Layer** enables you to set a scene running behind the main layer.
- **Offset** allows you to set an offset for the main, front and back layer, so they will start at different times.
- **Stop** is the number of seconds to pause during the scene’s animation stop points. If there are stop points in the scene, these are normally lost in the post rendering process, unless an interval is assigned here.

5.6.2 Post Rendering a Scene

To post render a scene

1. In the main menu, click **Post**.
2. If no scene is loaded by default, click one in the Server area and drag it onto **Main Layer** target area.
3. Select a post rendering plugin, and drag it onto **Clip Plugin** target area. Your default options are:
   - JPEG
   - PNG
   - MOV
   - TGA
   - TIFF
   - VideoRenderer
   - MatroxFileWriter

Note: Matrox File Writer requires a Matrox card to be installed.



Offset: 0 | 0 | 0
Stop (sec): 0 | 0.0 | 0

Clip Plugin: VideoRenderer
Format: Full Frame

Name: C:/Temp/Chart_variant=Bars/Chart_variant=Bars
Duration: 0
Width: 800
Height: 450

Note: See Post Rendering Plugins section for additional Matrox File Writer settings.

4. Optionally, set an Offset.

   You can see the how the offset affects the scene in the post rendering area.
5. Optionally, set a Stop (in seconds).

   This stops the post rendering after the assigned interval.
6. Select a frame format. Viz always renders 50 images per second for PAL or 60 for NTSC. Depending on the post rendering format, not all of them or not the full frames are stored. Your options are:

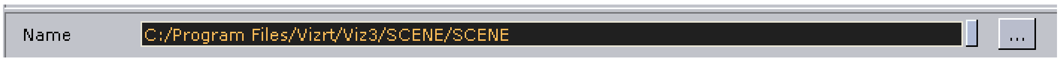
- **Full Frame** will store 50 or 60 full height frames for each second of video. The number depends on the video settings, PAL/50 or NTSC/60 frames, and the size also changes with the video format (720x576 for PAL digital and 720x486 for NTSC digital). Normally TV is 25 or 30 interleaved frames, but in this case the post rendering produces twice the amount of pixels needed. An editing system (or wherever you put the images you render) can throw away 50% of the pixels and still produce a “normal” video stream.
- **Fields Top/Fields Bottom** will store 50 fields per second of video. It is the same amount of data as interlaced top/bottom, but instead of shuffling the data the post rendering process writes different files for each field.
- **Interl. Top, Interl. Bottom** will store 25 interlaced frames for each second of video. Each frame consists of 2 fields (half size of PAL or NTSC for example 720x288). These two fields are combined line by line. Interlaced top means that the first line of the first field is the first line in the interlaced frame. In interlaced bottom the first line of the second field is the first line in the interlaced frame. This determines field dominance.
- **Full Frame Skip** will store every other full frame, resulting in 25 (or 30 for NTSC) frames per second.

7. Select the color settings. Your options are:
   - **RGB** creates RGB images. Here a pixel is represented by 3 bytes containing a red, green and blue sample.
   - **RGBA** is a red, green and blue sample, along with an additional color sample, all in a single image. When displayed on top of other images, RGBA pixels are blended with the background pixel according to the value of this alpha component.
   - **RGB AAA** For each image rendered, an RGB and an Alpha image are created.

Note: These settings are already fixed for the Matrox File Writer, see the Post Rendering Plugins section.

8. In the **Name** field, assign a destination file path and name.

Note: When you drag the scene into the drop zone, path and filename are automatically assigned. To keep another path that you have selected, click the Keep Path button, which is the small rectangle to the right of the Name field.


Name C:/Program Files/Vizrt/Viz3/SCENE/SCENE

9. In the **Duration** field, set the number of frames to be rendered.
10. In the **Width** field, set a scaling value for the width.
11. In the **Height** field, set a scaling value for the height.

Note: Beside the Width and Height fields, there is a Lock button. Enabling the lock allows you to scale height and width proportionately. Disabling the lock allows you to scale height and width separately. The default setting is locked.

12. Click the **Record** button.

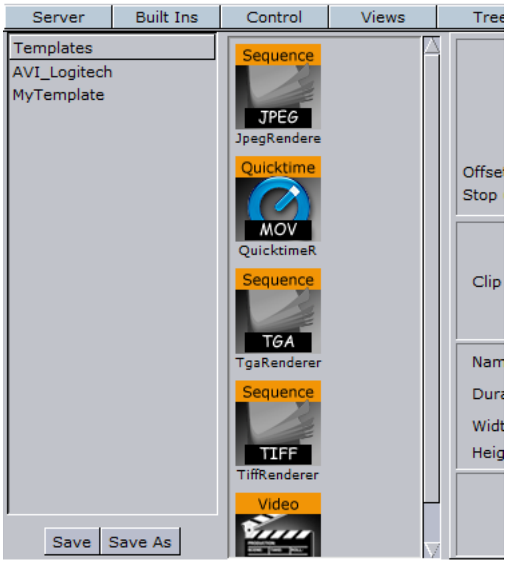
- During the rendering process the stop button will be shown as above, which would enable you to stop the rendering at any time. If you stop the rendering process the file(s) already written will be kept.

See Also

- Post Rendering Plugins
- Advanced Issues with Video Codecs

5.6.3 Post Rendering Templates

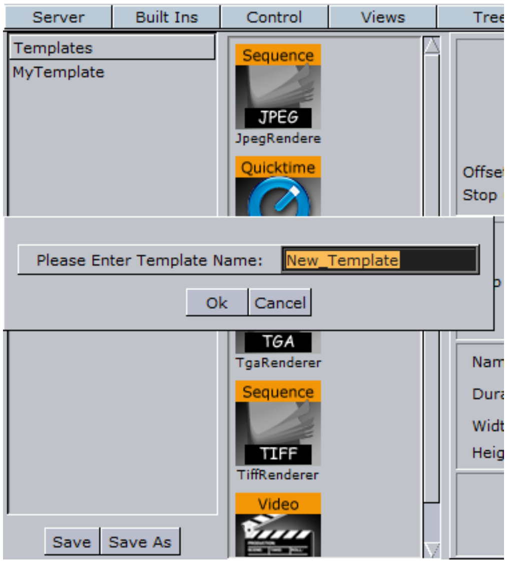
Figure 52: Post Rendering Templates



If you post render scenes often using the same settings, you can create templates that will allow you to use the same settings each time. In the upper left portion of the Post Rendering screen, there is a pane for saved templates.

Note: Post rendering templates are saved in the Viz Graphics Hub *and not* on the local machine.

To save a post rendering template

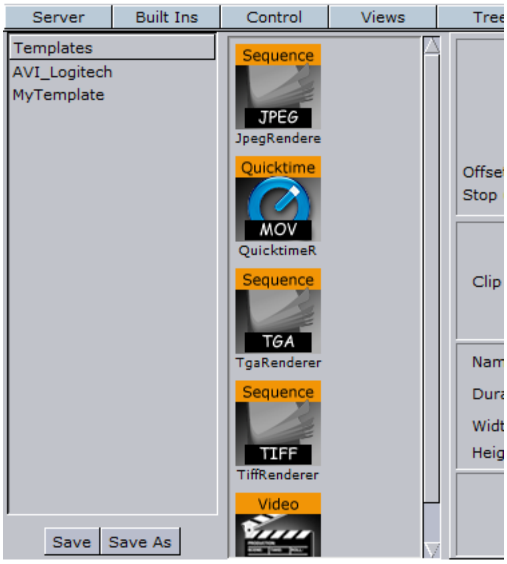


Please Enter Template Name: New_Template

Ok Cancel

1. Configure all the settings in the procedure To post render a scene.
2. In the Templates pane, click **Save**.
3. Enter a name for the template.
4. Click **OK**.
   Saved templates appear in the Templates pane.

To change a post rendering template

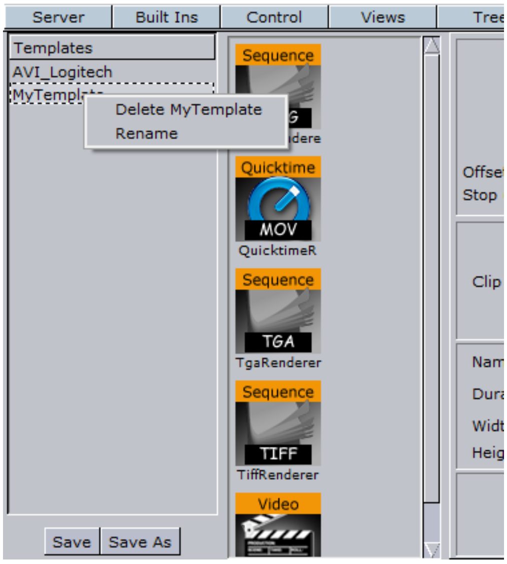


1. Double-click the post rendering template in the Templates pane.
2. Change the desired settings.
3. Click **Save**.

To load a post rendering template

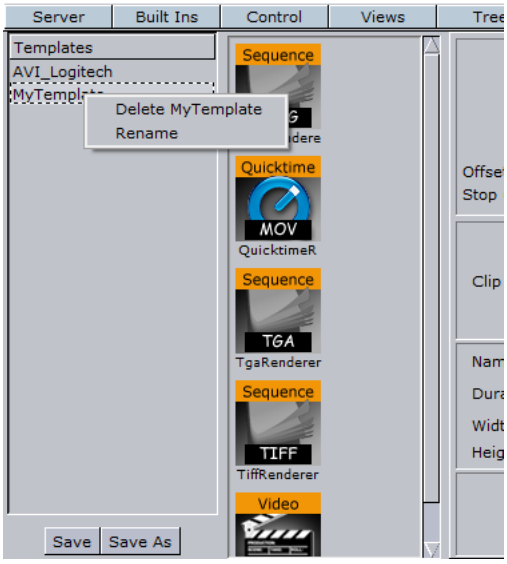
- Double-click the post rendering template in the Templates pane.

To rename a post rendering template



4. In the Templates pane, right-click the template and select **Rename**.
5. Enter a new name and click **OK**.

To delete a post rendering template



- In the Templates pane, right-click the template and select **Delete <name>**.

5.6.4 Post Rendering Plugins

Post rendering plugins define the format of the post rendered files. Out-of-the-box Viz Artist already offers you a set of image and video rendering plugins:

- Image rendering plugins
- Video clip rendering plugins

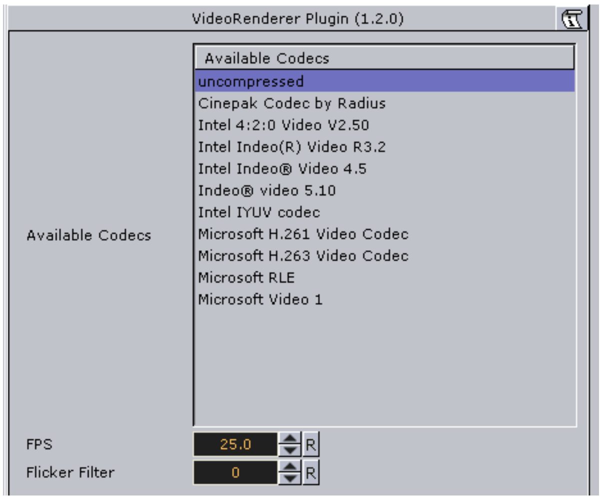
Image rendering plugins

Image rendering plugins, such as JPEG, PNG, TGA and TIFF, will all create a single image of every frame in your scene. The images will be named as you set in the Name field in the post render setup, with incremental numbers.

Some of the plugins offer a Quality setting that allows you to set the quality level of the jpeg images to be created.

Video clip rendering plugins

Figure 53: VideoRenderer plugin



VideoRenderer Plugin (1.2.0)

Available Codecs

Available Codecs:
- uncompressed
- Cinepak Codec by Radius
- Intel 4:2:0 Video V2.50
- Intel Indeo(R) Video R3.2
- Intel Indeo® Video 4.5
- Indeo® video 5.10
- Intel IYUV codec
- Microsoft H.261 Video Codec
- Microsoft H.263 Video Codec
- Microsoft RLE
- Microsoft Video 1

FPS: 25.0

Flicker Filter: 0

The video rendering plugins render a scene to a video clip. If you use the **Video Renderer** or **Quicktime Renderer** plugin you will find a menu showing all installed video codecs.

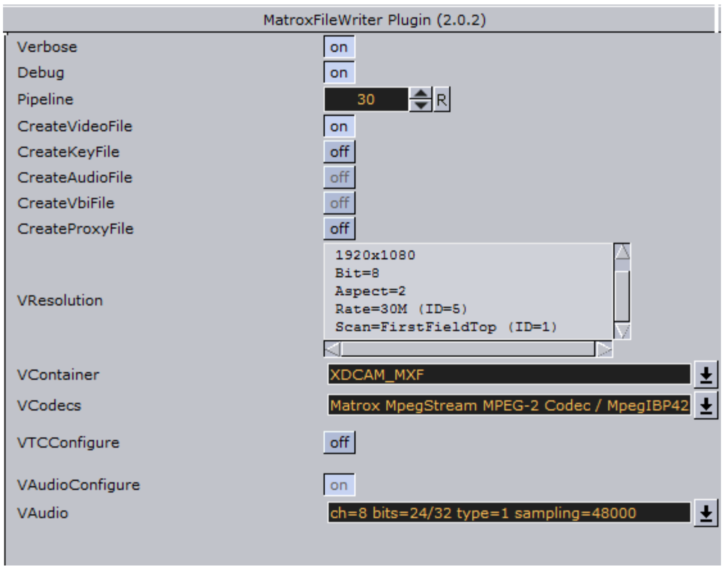
Select the codec you want to use and specify the parameters for Frame rate and Flicker Filter. If the video codec allows you to specify advanced parameters, clicking the Codec Settings and Codec Info buttons will open advanced menus related to the codec.

The **Matrox File Writer** plugin is also a video rendering plugin that is available to users with a Matrox board installed, and will render the scene into a Matrox encoded clip. It uses the Matrox video board to encode and write clips to file in real-time.

To use the Matrox File Writer plugin you must set the Output Format to match the desired clip format. You must also set the Video Output DVI out setting to **Inactive.**

When performing the procedure To post render a scene, note the following additional settings for this plugin:

Figure 54: MatroxFileWriter plugin



MatroxFileWriter Plugin (2.0.2)

Verbose	on
Debug	on
Pipeline	30
CreateVideoFile	on
CreateKeyFile	off
CreateAudioFile	off
CreateVbiFile	off
CreateProxyFile	off

VResolution	1920x1080
	Bit=8
	Aspect=2
	Rate=30M (ID=5)
	Scan=FirstFieldTop (ID=1)

VContainer	XDCAM_MXF
VCodecs	Matrox MpegStream MPEG-2 Codec / MpegIBP42
VTCConfigure	off
VAudioConfigure	on
VAudio	ch=8 bits=24/32 type=1 sampling=48000

- **Verbose**: produce verbose messages on the console and in the post render feedback window area. bwi can show you where that is in the main GUI.
- **Pipeline**: The number of internal buffers in frames for the matrox framework to operate. for SD+720p+1080i should be around one second (~25-30). It is recommended that you leave the default values.
- **CreateVideoFile**: Set to On to create a fill clip.
- **CreateKeyFile**: Set to On to create a key file.
- **CreateProxyFile**: Set to On to specify a custom resolution copy of the clip. These settings behave in a similar manner to the video file options and appear when this is set to On.
- **VResolution**: Provides clip information.
- **VContainer**: Select a container format for the clip. The options depend on the resolution. Matrox containers are only available with the standard HD/SD resolutions.
- **VCodecs**: Select a codec. The codecs available depend on the container selected. Supported Codecs.

5.6.5 Advanced Issues with Video Codecs

Many codecs have specific limitations and will only work on certain types of input. It is not possible to list all things to be aware of, but here are some:

- 24bit uncompressed video is valid but rare. Some programs may have difficulty in reading such files. Try creating an uncompressed video using RGBA to create a more common 32bit uncompressed video.
- Not all codecs support both 24bit and 32bit video. You may need to change between RGB and RGBA to get a specific codec to work.
- Some codecs will work with RGBA but will nevertheless store the video as 24bit.
- If you plan to use an RGBA video in a program that can handle alpha, you need to make sure that you have KEYFunctions on containers you want to have visible. Otherwise the alpha channel will be completely blank, thus making the whole video totally transparent.

- Some codecs require a handle to the top level application window. The Viz Artist plugin interface is unable to supply such a handle. As a result the configuration windows of some of the codecs will fail to appear. Also, some codecs will crash unless they get a handle.
- Files created in raw uncompressed mode looks weird in Windows Media Player. This is some oddity of the Media Player. The AVI video file contains pure RGB data, but the Media Player dithers it (often using only 16 colors) for some unknown reason. As said, the video file is correct and you should be able to import it into other video editing applications without any problems.
- In order to create Matrox clips, like DVCPROHD or MPEG I-Frame, you need to have VFW codecs installed. Since May 14, 2010 Matrox has provided a free codec pack. The installer is called `VFWCodecs-1.0.exe`. On machines with a Matrox board and driver version 7.5 and higher, you do not need to install VFW-Codecs since they are included in the driver. On machines with a Matrox board and drivers before version 7.5, there's always a matching VFW-Codecs pack available with the driver. However do not install VFWCodecs-1.0.exe or any non-matching VFW-Codecs pack on a machine with a Matrox board and drivers already installed. The following codecs are included in Matrox's free codec pack:
    - Matrox DV/DVCAM
    - Matrox DVCPRO
    - Matrox DVCPRO50
    - Matrox DVCPRO HD
    - Matrox MPEG-2 I-frame in SD and HD resolutions, with or without alpha
    - Matrox Uncompressed in SD and HD resolutions, with or without alpha
    - Matrox Offline HD
    - Matrox M-JPEG and HDV (playback only)

5.6.6 Post Rendering to Targa playable Clips

It is possible to render scenes to Targa clips. The following configuration must be done prior to the post rendering:

- The PinnacleDV25 or DVCPRO codec must be installed.
- The VideoRenderer plugin must be configured to use this codec.

After this is done, normal post rendering can be performed and the clips are directly playable by one of the two clip channels on the Targa card. This can typically be used to save performance in scenes by post rendering background elements to a clip file and thereafter playing the clip file instead of the original background.

Separate Fill and Key Clip

This post rendering also works when rendering separate fill and key clips, for example you can set up the Video Renderer plugin to create two separate clips, one for fill and one for key.

6 Lights and Materials

This section contains information on the following topics:

- Working with Light
- Working with Materials

6.1 Working with Light

The Light editor makes it possible to modify the light settings in the scene. A scene can have up to eight active simultaneous light sources.

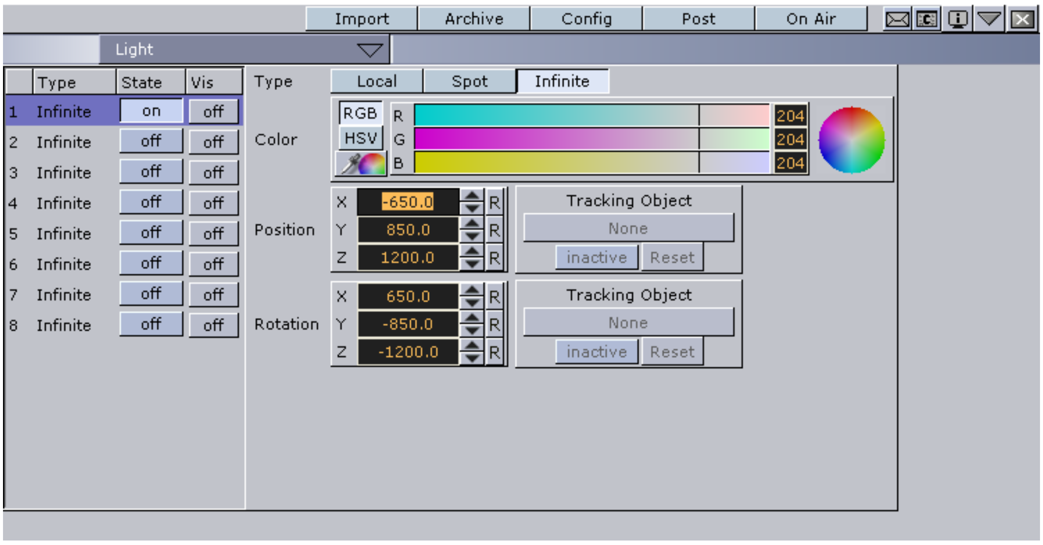
This section contains information on the following topics:

- Light Editor
- Adjusting Light Parameters
- Tracking Objects with Light
- Changing Light Source Parameters in Orthogonal View
- Copying Properties from one Light Source to another
- Light Source Animation

6.1.1 Light Editor

To open the Light editor

- In the right pane, from the drop down menu, select **Light**.



	Type	State	Vis
1	Infinite	on	off
2	Infinite	off	off
3	Infinite	off	off
4	Infinite	off	off
5	Infinite	off	off
6	Infinite	off	off
7	Infinite	off	off
8	Infinite	off	off

Type: Local | Spot | Infinite

Color: R 204, G 204, B 204

Position: X -650.0, Y 850.0, Z 1200.0 — Tracking Object: None, inactive, Reset

Rotation: X 650.0, Y -850.0, Z -1200.0 — Tracking Object: None, inactive, Reset

The left panel of the Light editor shows a list of all eight editable light sources. The right panel shows the parameters for the currently selected light source.

- **Number**: Shows the light source number, from 1 to 8.
- **Type**: Shows the light source type that is selected in the parameter panel.

- **State**: Turns the light On or Off.
- **Vis**: Shows the status of the light source:

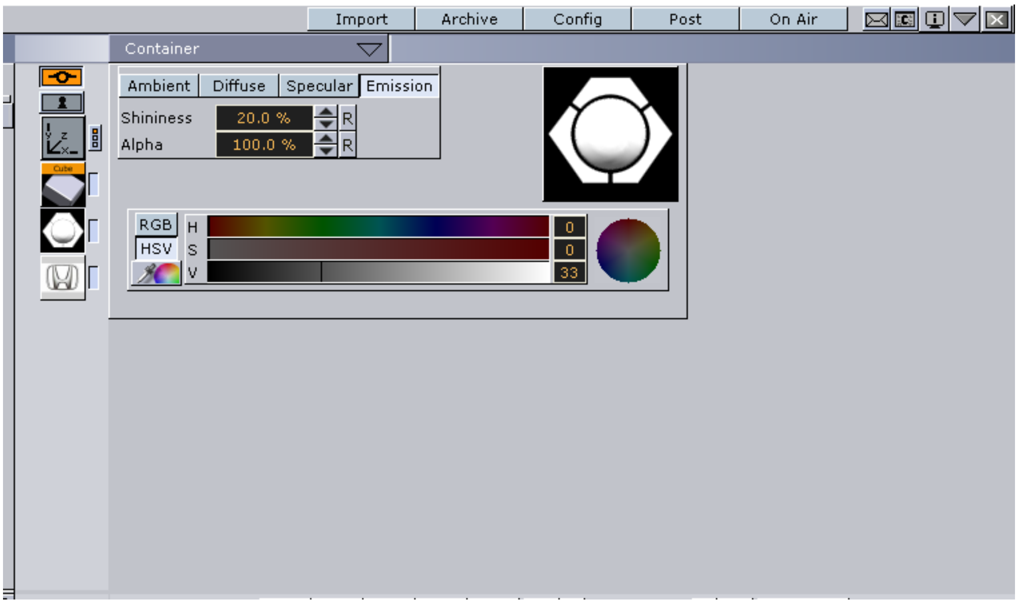
Note: This option is affected by the Bounding Box feature that is available in the various views. If the Bounding Box is activated, the light source is invisible whether Vis is set to On or Off.

  - **On**: You can see the light source in the output window. For example, if you have a spotlight, you will see the center of the light and the light cone.
  - **Off**: You cannot see the light source in the output window.

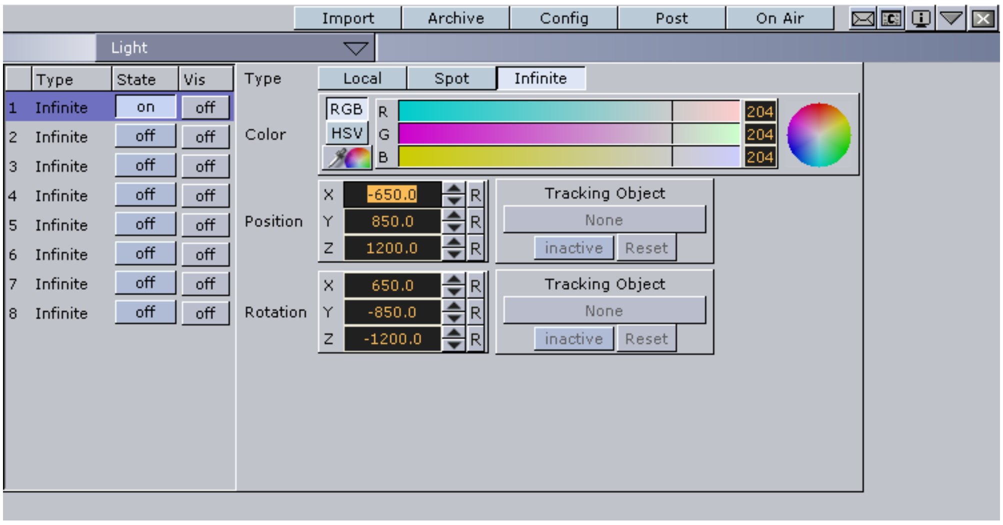
IMPORTANT! You must use a material to show light effects.

To apply light

1. Click **Built Ins.**
2. Click the **Geom Plugins** tab.
3. Drag an object into the scene tree.
4. Click **Server.**
5. Click the **M** tab, for Materials.
6. Drag the material onto the object in the scene tree.
7. Define a color for the material (see Changing Color).
8. Make sure the material is active by clicking the button down in the material editor.



9. In the right pane, from the drop down menu, select **Light**.
   The light editor is displayed. Light number 1 is turned on by default.



	Type	State	Vis
1	Infinite	on	off
2	Infinite	off	off
3	Infinite	off	off
4	Infinite	off	off
5	Infinite	off	off
6	Infinite	off	off
7	Infinite	off	off
8	Infinite	off	off

Color: R 204, G 204, B 204

Position: X -650.0, Y 850.0, Z 1200.0

Rotation: X 650.0, Y -850.0, Z -1200.0

10. In the left pane, click any or all of the light sources in the State column to show different light perspectives.

Figure 55: No light

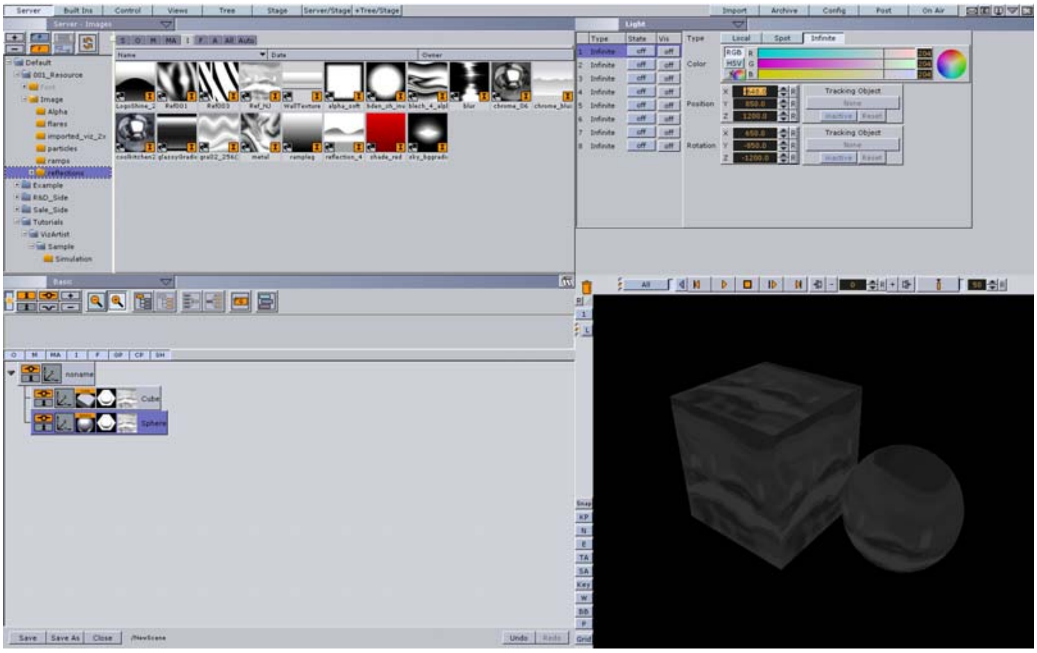


Figure 56: Light 1

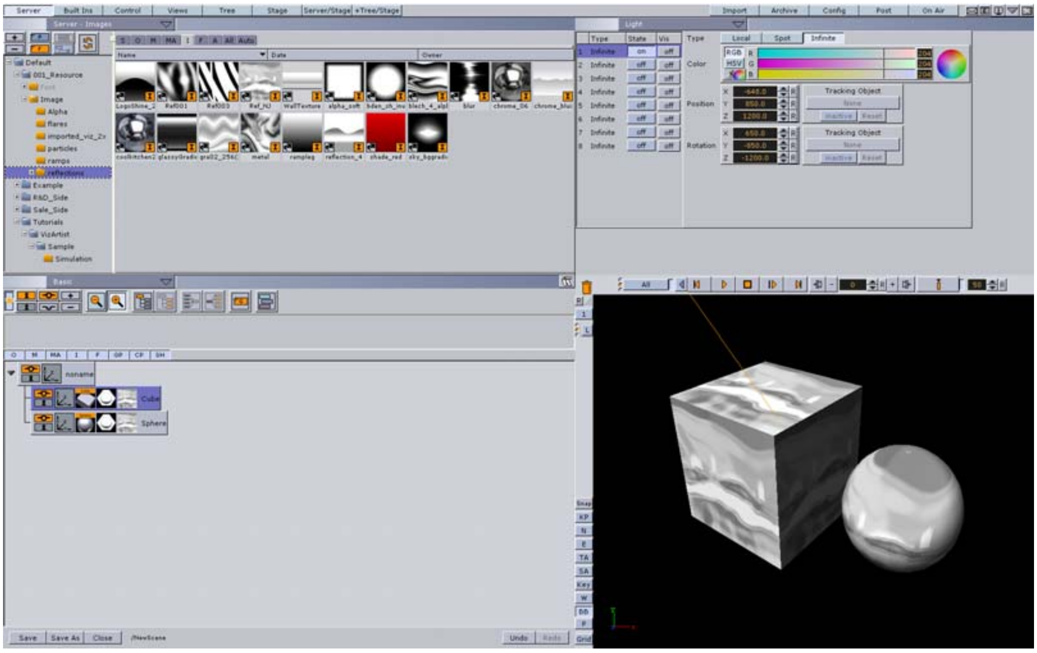
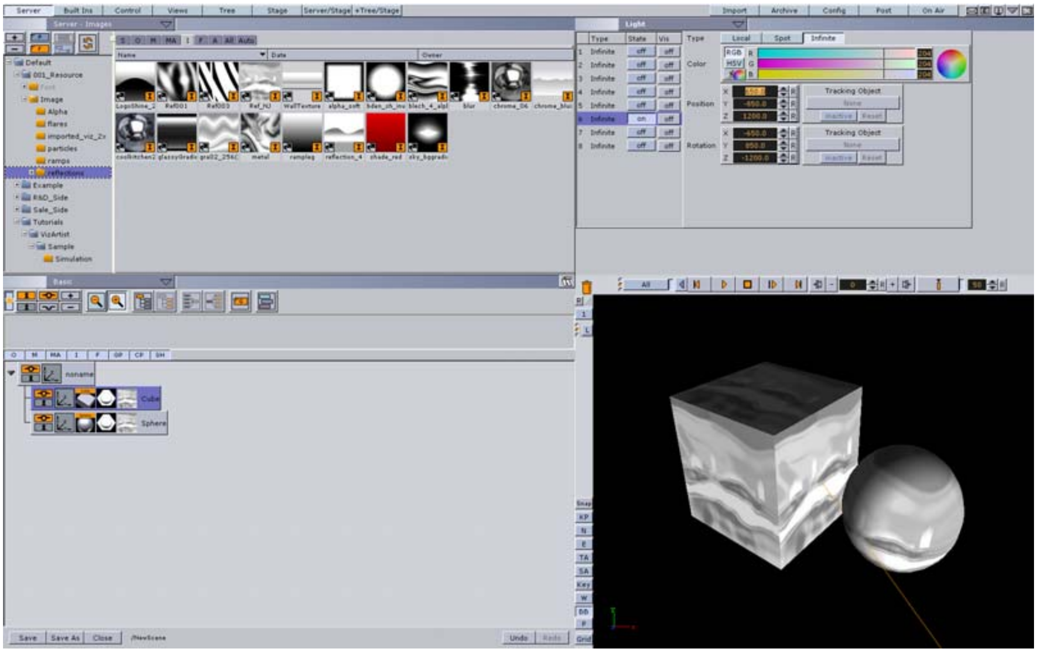


Figure 57: Light 6



See Also

- Lighting

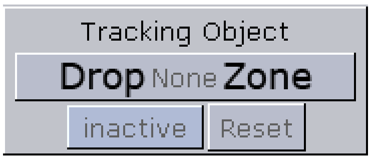
6.1.2 Adjusting Light Parameters

The following parameters of a light source can be modified:

- **Type**: There are three light source types:
  - **Local**: The Local light source is a positional light. It is near or within the scene, and the direction of its rays is taken into account in lighting calculations. Local lights have a greater performance cost than directional lights, due to the additional calculations. A real-life equivalent to a local light source is a light bulb. The Local light source has properties for color and position.
  - **Spot**: The Spot light source emits a cone of light. The only objects that are illuminated are those within the cone. The Spot light source has properties for color, position, rotation, and attenuation.
  - **Infinite**: The Infinite light source is a directional light. It is considered to be an infinite distance away from the objects in the scene. Because of the distance, the rays of light are considered parallel by the time they reach the object. A real life equivalent to an infinite light source is sunlight. The Infinite light source has properties for color, position, and rotation. Infinite is selected by default.
- **Color**: Sets the light source color. For information about how to select a color, see Changing Color.
- **Position**: Sets the position of the light source along the X, Y, and Z axis.
- **Rotation**: Sets the values for Pan, Tilt, and Twist for Spot or Infinite light sources.
- **Attenuation**: Sets the concentration of the light within the light cone of the Spot light source. When set to zero, the whole light cone has the same

intensity. If set greater than zero, the intensity decreases away from the centre.

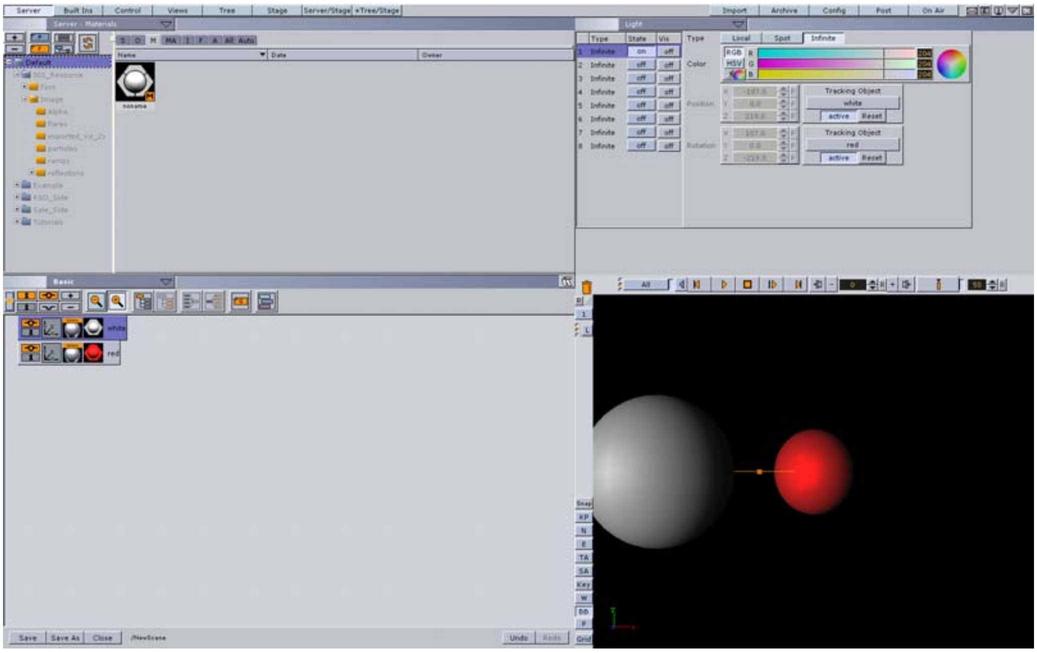
6.1.3 Tracking Objects with Light



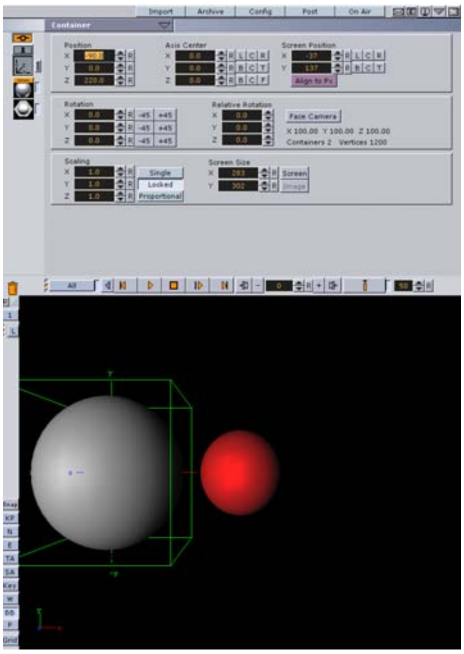
A light source can be set up to track containers regarding both position and direction. Tracking position means that the position of the source will be in the center of the container. Tracking direction means that the light source will look at the center of the container, for example throughout an animation.

To use a light source to track containers

1. In the scene tree create an object to track position and an object to track rotation.
2. In the right pane, from the drop down menu, select **Light**.
3. Drag the position object into the position drop zone and the rotation object into the rotation drop zone.



4. In the right pane, from the drop down menu, select **Container**.
5. Track the objects by adjusting the Position.



Position X -90.0 Y 0.0 Z 220.0

Axis Center X 0.0 Y 0.0 Z 0.0

Screen Position X -37 Y 137

Rotation X 0.0 Y 0.0 Z 0.0

Relative Rotation X 0.0 Y 0.0 Z 0.0

X 100.00 Y 100.00 Z 100.00

Containers 2 Vertices 1200

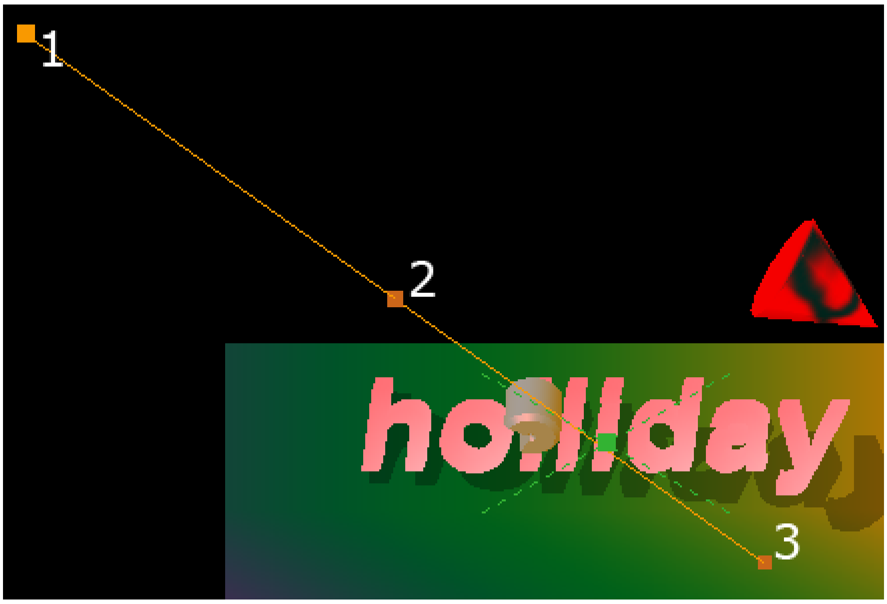
Scaling X 1.0 Y 1.0 Z 1.0

Screen Size X 283 Y 302

The following parameters can also be adjusted for tracking with light:

- **Active/Inactive**: By dragging a container to the drop zone, the light source tracking is automatically activated. To inactivate the tracking, click the Active/ Inactive button.
- **Reset**: Disables the light source tracking.

6.1.4 Changing Light Source Parameters in Orthogonal View



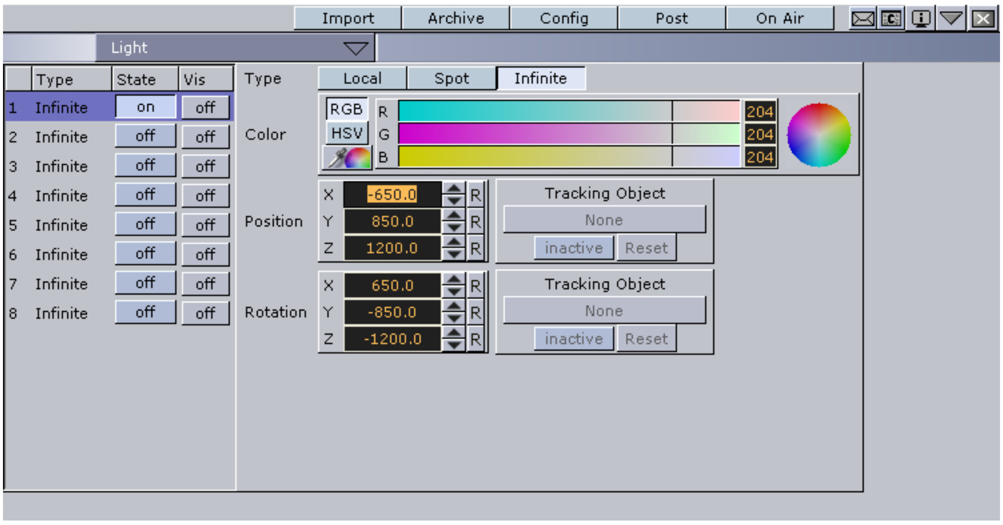
For a light source, position and direction can be set directly in the output area or the side views by repositioning the light handles. To do this, the light must be activated and selected in the Light editor, or alternatively the Vis option must be enabled for that light source.

In the render output, a Local light source is visualized as a single point. Infinite and Spot light sources are visualized as an orange line with three small squares, which are the handles. In addition, a Spot light source has a “wireframe box” that shows the attenuation or in other words, the scope of the light. The squares at the ends of the light source slightly differ in size and color (when the light source is not selected). Handle (1) shows the light source and handle (3) sets the direction and distance of the light. Handle (2) moves the whole light source without altering the proportionality of position and direction. Simply drag the handles around to change them.

6.1.5 Copying Properties from one Light Source to another

To copy light source properties

1. In the right pane, from the drop down menu, select **Light**.



	Type	State	Vis
1	Infinite	on	off
2	Infinite	off	off
3	Infinite	off	off
4	Infinite	off	off
5	Infinite	off	off
6	Infinite	off	off
7	Infinite	off	off
8	Infinite	off	off

Type: Infinite

Color: R 204, G 204, B 204

Position: X -650.0, Y 850.0, Z 1200.0

Rotation: X 650.0, Y -850.0, Z -1200.0

2. Adjust the source properties in the desired numbered light source.
3. Click the numbered source light and drag it onto the target light.
   The target light will inherit all properties belonging to the source.

6.1.6 Light Source Animation

All light source parameters can be animated in the same way as all other animations:

1. Set a value in the Light editor,
2. Add a keyframe,
3. Change the value
4. Add a keyframe again.

Animations can also easily be created in one of the orthogonal views. There, the light sources appear as symbols, and by repositioning them and adding keyframes, animation sequences are created. For more information, see Changing Light Source Parameters in Orthogonal View.

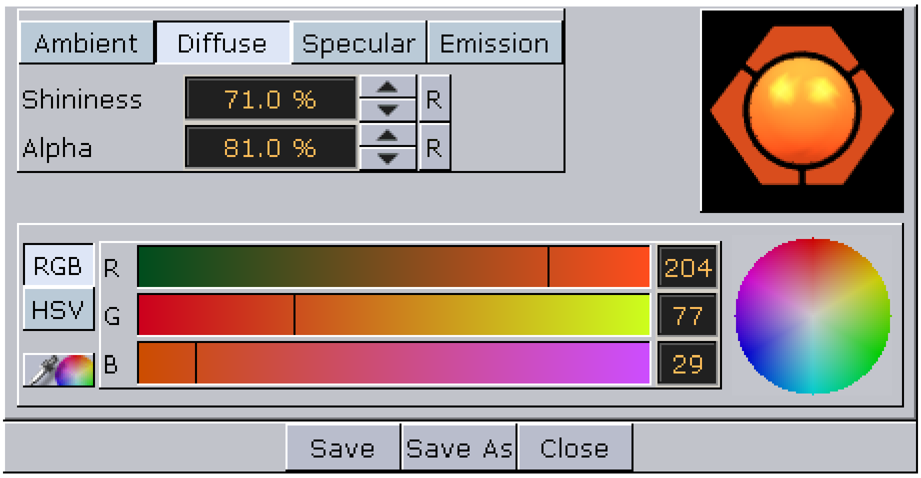
More advanced light source animations are normally created with the light source tracking feature, see Tracking Objects with Light.

For more general animation information, see Creating Animations.

6.2 Working with Materials

The Material editor is used to create and modify materials with customized colors and light effects.

Figure 58: Material editor



The icon in the upper right corner of the editor shows the current state of the material. The color for a lit object is shown on the sphere within the icon, while the three parts around the sphere show the plain color for an object that is not lit. For information about the various effects and types of lights that can be applied to a material.

Note: When a material has been added to a scene, the scene-specific material can be modified in a Material editor that opens in the Element pane. In this case, only the scene-specific material will be modified, the original material in the database will not be affected. To modify the original material, double-click it in the Server area.

This section contains information on the following topics:

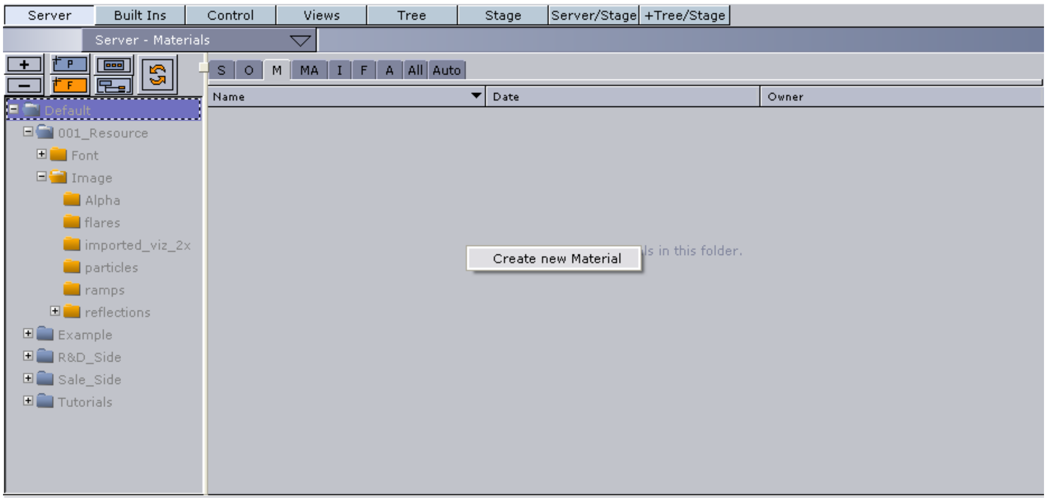
- Creating Materials
- Working with Advanced Materials
- Color Properties and Schemes
- Changing Color

6.2.1 Creating Materials

There are two options for creating materials. You can create one either in the Server area, or directly in a container.

To create a material in the Server area

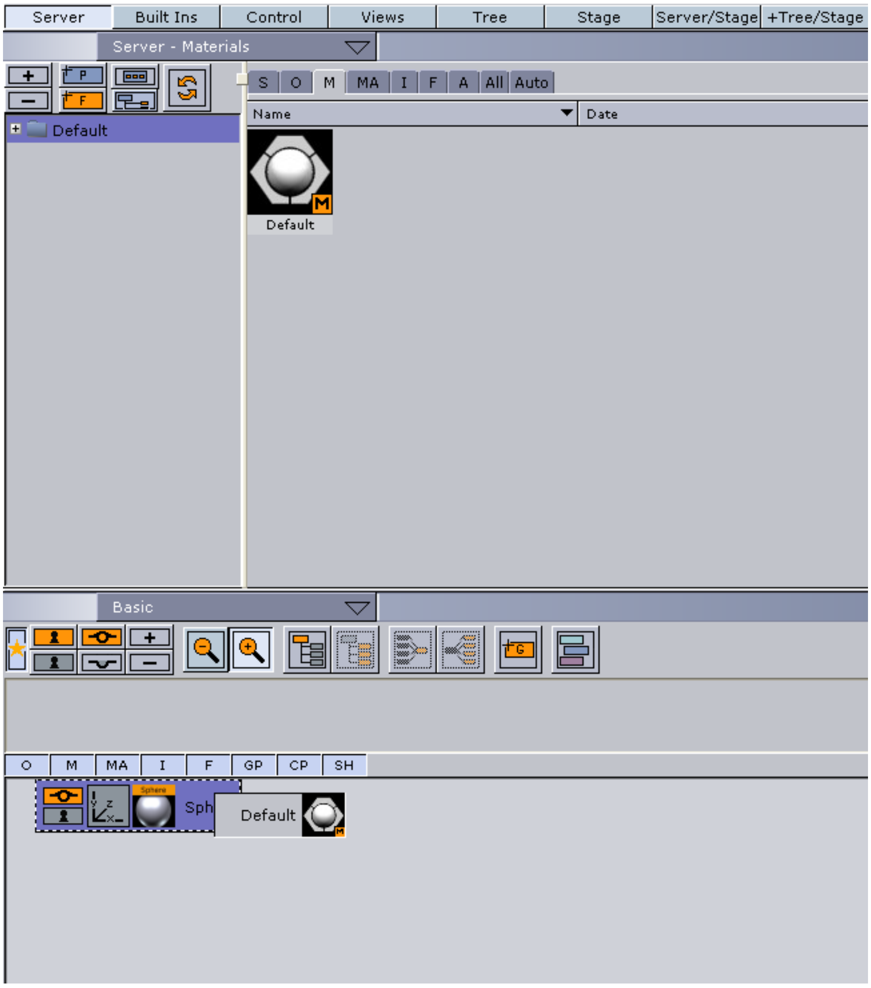
1. Click **Server**.
2. Click the **M** tab (for materials).
3. Right-click and select **Create new material**.



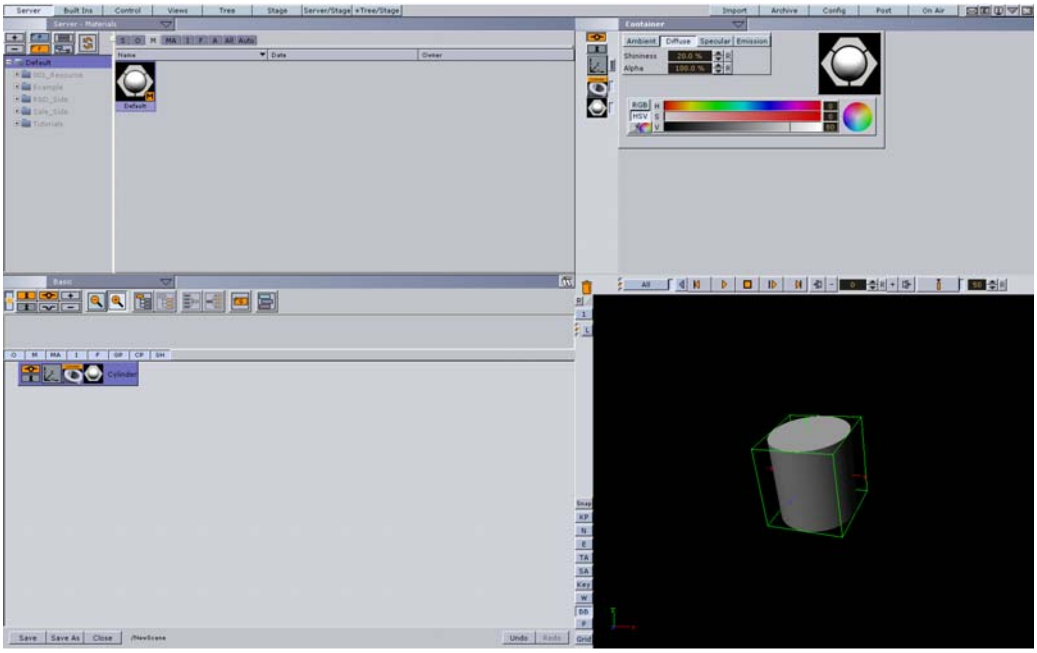
4. Enter a name for the material and click **OK**.
5. Double-click the material to open the Material editor.
6. Adjust the Color Properties and Schemes as necessary.
7. Adjust the color as necessary. See Changing Color.
8. Save any changes.
   - **Save**: Saves the material in Viz Graphics Hub.
   - **Save As**: Creates a new material in Viz Graphics Hub. In the Material Save As dialog box that appears, type the material name, and then select the folder/ project where the material should be placed.
   - **Close**: Closes the Material editor without saving the changes.

To create a material in a container

1. Click **Server**.
2. Click the **M** tab (for materials).
3. Click a material and drag it onto the desired container in the scene tree.



4. Click the material icon in the container.
   A material editor is displayed in the upper right quadrant.



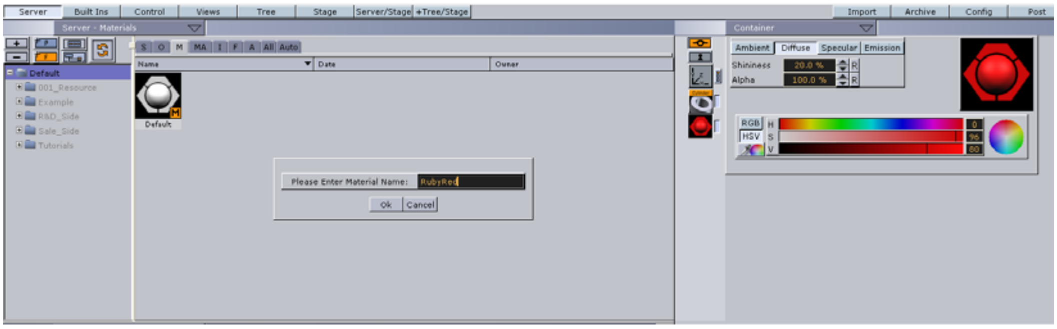
5. Adjust the Color Properties and Schemes and as necessary.

6. Adjust the color as necessary. See Changing Color.

Note: So far, the material only exists in this container. You can use it in this scene by dragging it onto another container.

7. In the upper right quadrant, click the material preview icon and drag it into the Server area.

   Alternatively, drag material icon directly from the container into the Server area.



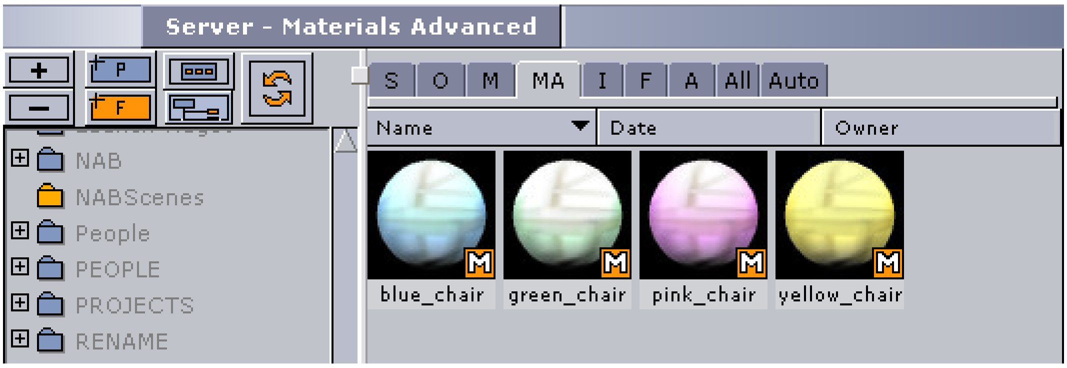
Please Enter Material Name: RubyRed

8. Enter a name for the material and click **OK**.

6.2.2 Working with Advanced Materials

An advanced material is an element that combines shader, image, or material settings. Existing advanced materials are stored in the database and available through the Server area, see To view Advanced Materials in a Project/Folder.

Figure 59: Server area showing all advanced materials in selected folder



To add an advanced material as a container item (part of another container), drag the advanced material from the Element List onto a the container in the scene tree.

This section contains information on the following procedures:

- To create an Advanced Material
- To edit an Advanced Material
- To rename an Advanced Material
- To delete an Advanced Material
- To view Advanced Materials in a Project/Folder

To create an Advanced Material

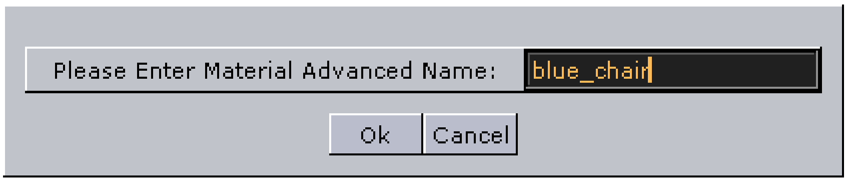
1. Drag a material from the Element List to a container in the scene tree.
2. Drag an image from the Element List to the same container in the scene tree.
3. Drag a shader from the Built Ins section to the same container in the scene tree.

Note: It is not necessary to add all three items, but a combination of two items (material, shader, or image) must be applied in order to create an advanced material.

4. Right-click the container in the scene tree.
5. In the menu that appears, select **Merge to Material Advanced**.

   The material, shader, and image icons on the container will then be merged into one combined material advanced icon.

6. Select the project/folder where the advanced material should be placed.
7. Drag the material advanced icon from the container in the scene tree to the Element List.



Please Enter Material Advanced Name: blue_chair

8. In the dialog box that opens, type a descriptive name for the advanced material.

Note: Special characters, such as space, are not allowed in element names.

9. Click the **Ok** button.

To edit an Advanced Material

1. Select the project/folder where the existing advanced material is placed.
2. Make sure to open the material advanced view of the selected folder, see To view Advanced Materials in a Project/Folder.
3. Drag the advanced material that should be modified to a container in the scene tree.
4. Right-click the material advanced icon on the container.
5. In the menu that appears, select **Split Material Advanced**.

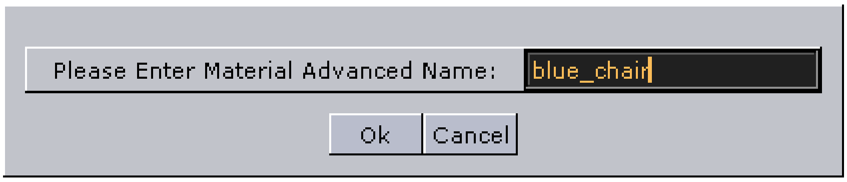
   The material advanced icon on the container will then be replaced by separate icons for any material, shader, or image that the advanced material consists of.

6. Modify the material, shader, or image settings.
7. Right-click the container in the scene tree.
8. In the menu that appears, select **Merge to Material Advanced**.

   The material, shader, and image icons on the container will then be merged into one combined material advanced icon again.

9. Select the project/folder where the modified advanced material should be placed.

10. Make sure to open the material advanced view of the selected folder, see To view Advanced Materials in a Project/Folder.
11. Drag the material advanced icon from the container in the scene tree to the Element List.


Please Enter Material Advanced Name: blue_chair

12. In the dialog box that opens, type a descriptive name for the advanced material.

Note: Special characters, such as space, are not allowed in element names.

13. Click the **Ok** button.

To rename an Advanced Material

1. Select the project/folder where the existing advanced material is placed.
2. Make sure to open the material advanced view of the selected folder, see To view Advanced Materials in a Project/Folder.
3. Select the advanced material that should be renamed.
4. Press **F2**.
5. In the text field that appears, type a new descriptive name.

Note: Special characters, such as space, are not allowed in element names. Also, if renaming a linked element, only the name of the selected element will change, not the linked one. For more information, see To link an element.

Alternatively, perform the following steps:

1. Select the project/folder where the existing advanced material is placed.
2. Make sure to open the material advanced view of the selected folder, see To view Advanced Materials in a Project/Folder.
3. Right-click the advanced material that should be renamed.
4. In the menu that appears, select **Rename**.
5. In the text field that appears, type a new descriptive name.

To delete an Advanced Material

Tip: Multiple advanced materials can be deleted at once.

1. Select the project/folder where the existing advanced material is placed.
2. Make sure to open the material advanced view of the selected folder, see To view Advanced Materials in a Project/Folder.
3. Drag the advanced material that should be deleted to the trash can.
4. In the dialog box that opens, confirm the delete operation.

Note: If deleting a linked element, only the link to the project/folder will be removed. The element itself will remain in the database. For more information, see To link an element. Also, referenced elements cannot be deleted. Referenced elements are used by other elements.

Alternatively, perform the following steps:

1. Select the project/folder where the existing advanced material is placed.
2. Make sure to open the material advanced view of the selected folder, see To view Advanced Materials in a Project/Folder.
3. Right-click the advanced material that should be deleted.
4. In the menu that appears, select **Delete**.
5. In the dialog box that opens, confirm the delete operation.

To view Advanced Materials in a Project/Folder

To view all advanced materials available in a specific project/folder, perform the following steps:

1. Select a project/folder in the server tree.
2. Select the **MA** tab of the Content Switcher.

Alternatively, perform the following steps:

1. Select a project/folder in the server tree.
2. Select **Material Advanced** from the Server Menu.

Alternatively, perform the following steps:

1. Select a project/folder in the server tree.
2. Press the keyboard shortcut **CTRL+4** while holding the pointer over the Server area.

6.2.3 Color Properties and Schemes

- **Shininess** The shininess defines how spread out the highlighted area is. The lower the value, the more spread out the shine; the higher the value, the more focused the shine on a single point. However, if Specular is black (all values 0) then shininess does not have any visible effect.
- **Alpha** The alpha value can be set directly on a material to adjust the level of transparency.

Viz Artist uses schemes to describe the colors of a material.

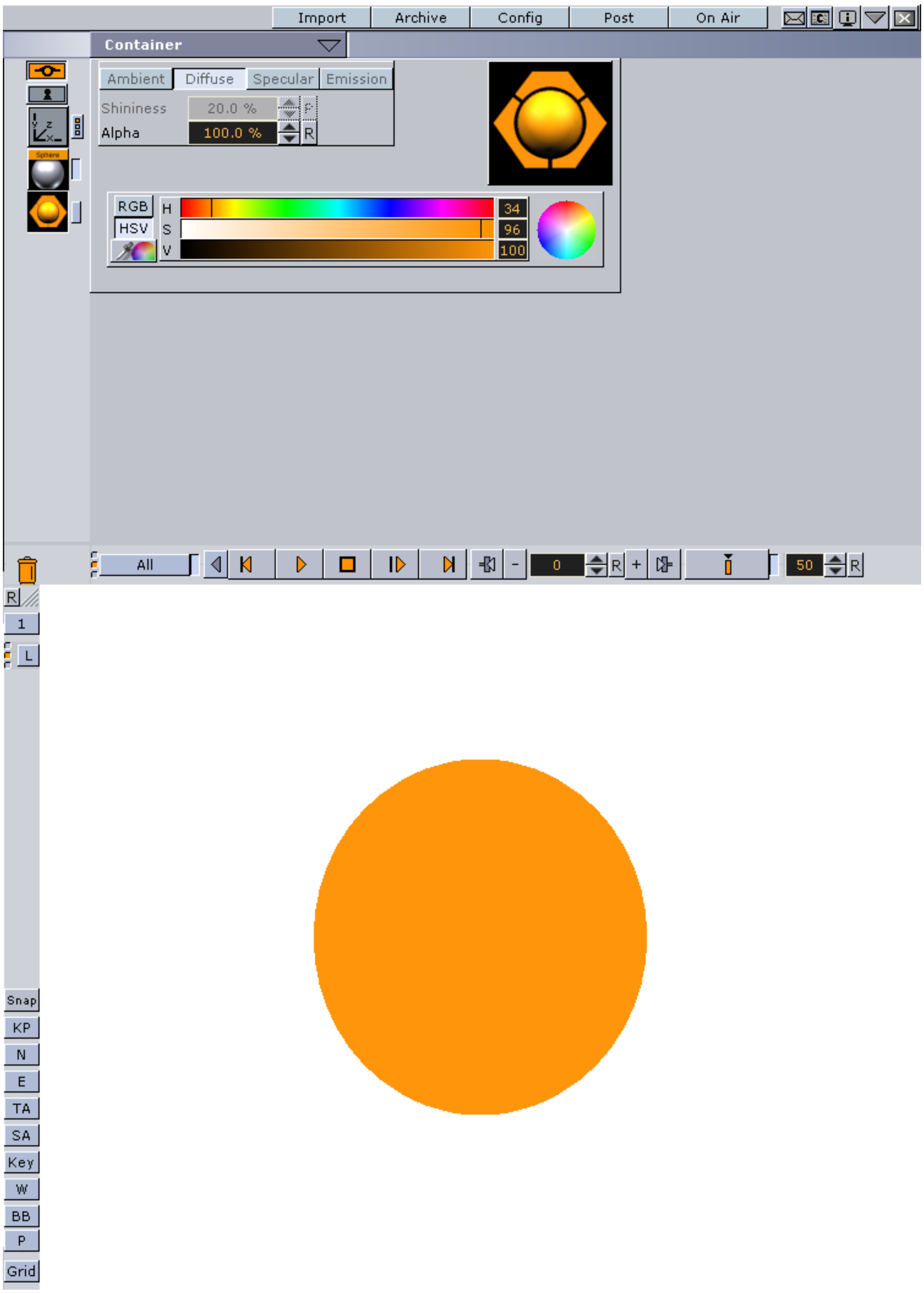
- **Ambient** The area which is not directly lit by any light source. Ambient light surrounds the object and the color looks the same on every side. Since the light does not come from any direction, but instead from “all” directions, the color/light spreads evenly over the object.
- **Diffuse** The object’s “core color” observed under “normal” lightening conditions. Shows the color of an object when illuminated by a directional light source. It spreads evenly over the object. Light normally has some level of diffusion.

- **Specular** Shows the color that appears when light comes from a particular direction and bounces back from the surface in a mirrored direction. The specular color is in fact the direct reflection of a directional light source. This is the typical effect of shiny metal and plastic. Specular color can be understood as a kind of shininess color.
- **Emission** Emissive color is produced by an object's natural glow. The object itself is the light source illuminating the color. The emissive color is unaffected by any other light sources and does not produce any light for other objects in the scene.

Base Color

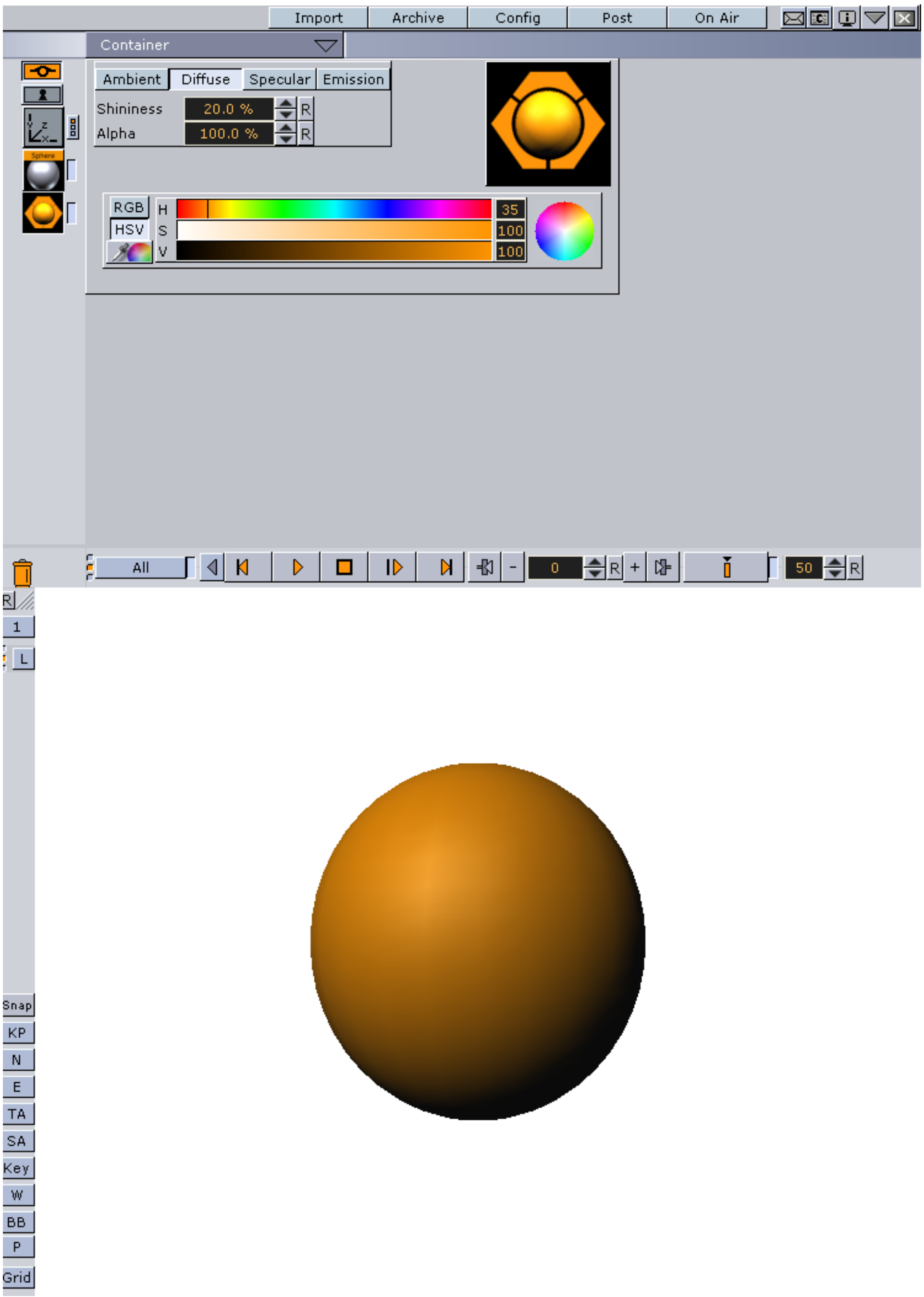
Disabling the material icon in the Property area causes only the object's base color to be shown. This is a color without any shading and it is not influenced by light, which means that it is visible even if there are no active light sources. Its initial color is created through a conversion of the material RGB components, but the values can be modified afterwards.

Figure 60: Material disabled



Shininess 20.0 %
Alpha 100.0 %
H 34
S 96
V 100
0
50

Figure 61: Material enabled



Shininess 20.0 %

Alpha 100.0 %

H 35

S 100

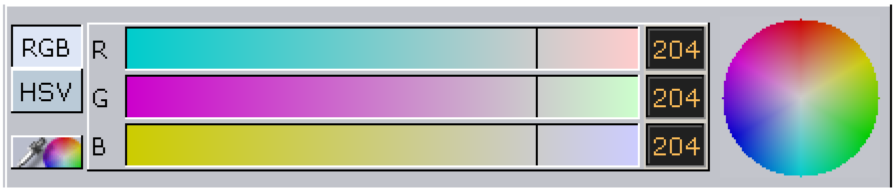
V 100

0

50

For information about the various methods that can be used to define the color, see Changing Color.

6.2.4 Changing Color



To modify a color, select the color type (RGB or HSV), and then use one of the following methods:

- **Sliders** Click the slider, and then hold the left mouse button pressed while dragging the pointer sideways. You can also click on a single point in the slider.

Tip: For RGB only, right-clicking will lock all three sliders, so that all values can be changed at once.

- **Manual Numeric Edit** Click the input box and enter a numeric value.

Tip: Click the input box and use the up arrow/down arrow keys to adjust the value by a factor of 1. Pressing and holding the <Alt> key while using the up arrow/down arrow keys will adjust the value by a factor of 10.

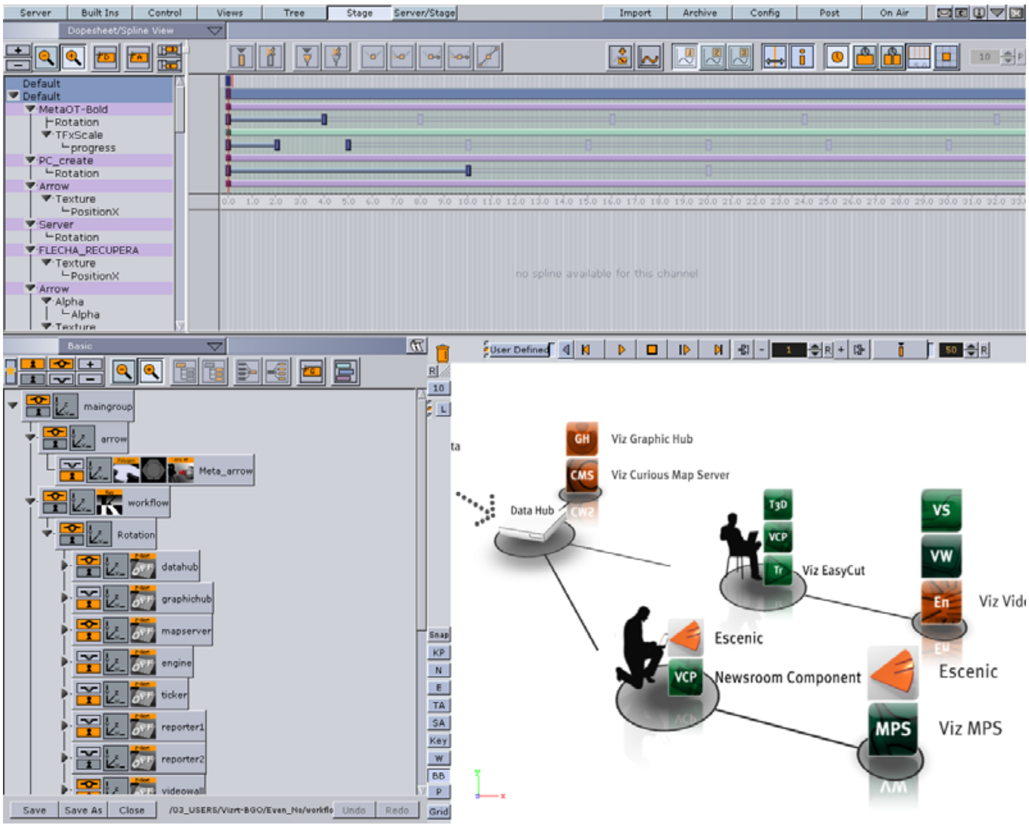
- **Color Circle** In the color circle, click a color to select it.
- **Color Picker** After clicking the Color Picker button below the RGB and HSV buttons, a color can be picked from anywhere on the screen.
- **Material** You can drag a material from the Server area into Properties area, to the left of the sliders.

7 Animation Stage

The interactive user interface for creating animations is the Stage. This section contains descriptive information on all the parts of the Stage.

- Stage Tree Area
- Stage Editor Area
- Timeline Editor
- Timeline Marker
- Director Control Panel
- Director Editor
- Actor Editor
- Channel Editor
- Action Channel Editor
- Dopesheet Editor
- Spline Editor
- Stage Object Editor
- Keyframe Editor
- Stops, Tags and Pause Editor

Figure 62: Stage displayed together with scene tree and render output



The Stage is divided in two; the Stage Tree Area (left side) and Stage Editor Area (right side). The Stage Tree Area shows a representation of all animated elements in the current scene. The Stage Editor Area displays the animated elements as bars along the timeline.

Tip: To resize the two areas, drag the vertical separator.

To display the stage

- From the main menu select **Stage**.

To display the stage together with the server area and render output

- From the main menu, select **Server/Stage**.

To show the Stage together with the scene tree, server area, and render output

- From the main menu, select **+Tree/Stage**.

Note: You must have a screen resolution with a width of greater than 1500 for this feature to be enabled.

To resize the stage

1. Straddle the mouse pointer over the bottom edge of the stage.
   The mouse pointer becomes a two-sided arrow.
2. Click on the border and drag up or down to resize.

See Also

- Scene Editor Animation Controls and Shortcuts
- Stage Controls and Shortcuts

7.1 Stage Tree Area

The Stage Tree area shows a representation of all animated elements in the current scene.

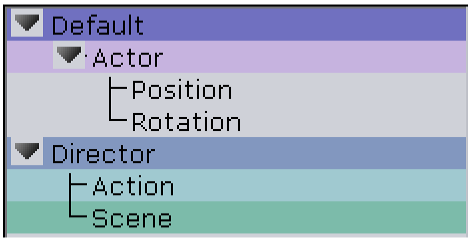
This section contains information on the following topics:

- Stage Tree
- Stage Tree Bar
- Stage Tree Menu

7.1.1 Stage Tree

The stage tree shows an overview of all directors, actors, and channels that are used to create the animations in the current scene.

Figure 63: Stage tree with directors, actor, channels, action, and scene



```
▼ Default
   ▼ Actor
       ├Position
       └Rotation
▼ Director
    ├Action
    └Scene
```

This section contains information on the following procedures:

- To expand a stage tree branch
- To expand all stage tree branches
- To collapse a stage tree branch
- To collapse all stage tree branches

To expand a stage tree branch

▶

- Click the **Expand** icon to the left of the branch name.

To expand all stage tree branches

- Click the **Expand** button in the Stage Tree Bar or press <Alt> and click the Expand icon to the left of the branch name.

To collapse a stage tree branch

▼

- Click the **Collapse** icon to the left of the branch name.

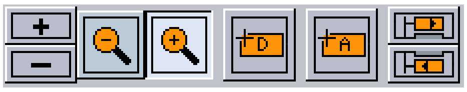
To collapse all stage tree branches

- Click the **Collapse** button in the Stage Tree Bar.

7.1.2 Stage Tree Bar

The Stage Tree bar is located above the Stage Tree.

Figure 64: Stage Tree bar



The Stage Tree bar contains the following options:

⊞

- **Expand**: Expands all branches in the stage tree. See To expand a stage tree branch.

- **Collapse**: Collapses all branches in the stage tree. See To collapse a stage tree branch.

- **Zoom Out**: Scales the Stage to provide a better overview of complex animations.

- **Zoom In**: Scales the Stage back to standard display.

- **Add New Director**: Adds a new director to the stage tree. See To create a director.

- **Add New Action**: Adds a new action to a director, see To create an action channel.

- **Jump to Next Animation**: If an actor (container) has multiple animation channels placed on various directors, the next channel in the stage tree will be selected.

- **Jump to Previous Animation**: If an actor (container) has multiple animation channels placed on various directors, the previous channel in the stage tree will be selected.

7.1.3 Stage Tree Menu

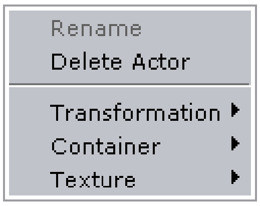
Right-clicking a director in the stage tree opens the Stage Tree menu. For more information about directors, see Working with Directors in the Stage Tree.

Figure 65: Stage Tree menu when right-clicking a director

- **Rename**: Makes it possible to rename the director, see To rename a director.
- **Delete [Director]**: Makes it possible to delete the director, see To delete a director.
- **Add Scene**: Adds a new scene to the director.

Right-clicking an actor in the stage tree opens the Stage Tree menu. For more information about actors, see Working with Actors in the Stage Tree.

Figure 66: Stage Tree menu when right-clicking an actor



- **Delete [Actor]**: Deletes the actor. See To delete an actor.
- **Transformation**:
  - **Position**: Creates a position channel for the actor, and automatically adds two default keyframes.
  - **Rotation**: Creates a rotation channel for the actor, and automatically adds two default keyframes.
  - **Scaling**: Creates a scaling channel for the actor, and automatically adds two default keyframes.

Note: If a transformation channel is disabled (grayed out), the channel is already available for the actor in this director.

- **Container**:
  - **Active**: Creates an active channel for the actor, and automatically adds two default keyframes. The active status can be modified in the Keyframe Editor. If an active keyframe is set to Value=**Off**, the whole actor (container) including all animations will be hidden in the render output. The action remains hidden until another keyframe appears in the timeline with Value=**On**.
  - **Material**: Creates a material channel for the actor, and automatically adds two default keyframes. The material can be modified in the Keyframe Editor. For more information, see Working with Materials.

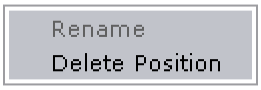
Note: The material option is only available if a material is added to the container.

- **[Texture]**: This option varies according to the element that is available on the actor.

Example: If an image is added to the container, various texture options become available. If a built-in geometry is added, for example a cube, various cube options become available. If a font is added to the container, various text options become available. For details about options for the various elements, see the respective editor sections, for example Image Editor and Fontstyle Editor.

Right-clicking a channel in the stage tree opens the Stage Tree menu. For more information about channels. See Working with Channels in the Stage Tree.

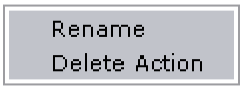
Figure 67: Stage Tree menu when right-clicking a channel



- **Delete [Channel]**: Deletes a channel. See To delete a channel.

Right-clicking an action channel in the stage tree opens the Stage Tree Menu. For more information about actions, see the Action Channel Editor.

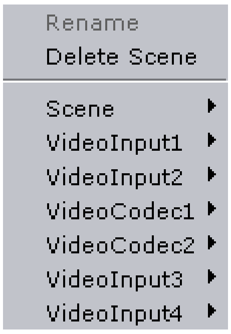
Figure 68: Stage Tree menu when right-clicking an action



- **Rename**: Makes it possible to rename an action channel. See To rename an action channel.
- **Delete [Action Channel]**: Deletes an action channel. See To delete an action channel.

Right-clicking a scene in the stage tree opens the Stage Tree Menu.

Figure 69: Stage Tree menu when right-clicking a scene



- **Delete Scene**: Deletes a scene channel inside the stage.
- **Scene**: The Scene options refer to various scene settings.
- **VideoInput1-4**: The VideoInput options allow you to control live video input parameters such as Alpha, Audio etc. Note that VideoInput1-4 allows you to add Clip, but they will not work. Use VideoCodec1-2.
- **VideoCodec1-2**: Similar to VideoInput1-4, but in this case you control clips.

See Also

- Working with Clip Keyframes

7.2 Stage Editor Area

The Stage Editor area displays the animated elements as bars and splines along the timeline.

This section contains information on the following topics:

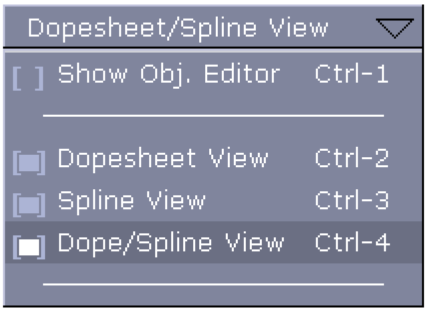
- Stage Editor Menu

- Stage Editor Bar
- Timeline Editor
- Timeline Marker
- Spline Editor
- Stage Object Editor

7.2.1 Stage Editor Menu

Above the Stage Tree Area is a menu that defines the appearance of the Stage Editor Area.

Figure 70: Stage Editor menu

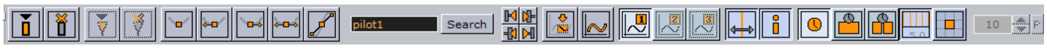


- **Show Obj. Editor**: Enables/disables the Stage Object Editor at the right side of the Stage. see To enable/disable the stage object editor.
- **Dopesheet View**: Shows only the Dopesheet Editor.
- **Spline View**: Shows only the Spline Editor.
- **Dope/Spline View**: Shows both the Dopesheet Editor and Spline Editor.

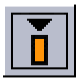
7.2.2 Stage Editor Bar

The Stage Editor bar is located above the Dopesheet Editor/Spline Editor.

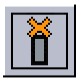
Figure 71: Stage Editor bar



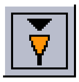
The Stage Editor bar contains the following options:



- **Add Keyframe**: Adds a keyframe at the current time/frame/field. See Working with Keyframes.



- **Delete Keyframe**: Deletes the selected keyframe. See To delete a keyframe.



- **Add Stop/Tag**: Adds a stop/tag to the selected director at the current time/frame/field. See To create a stop, tag or pause.

- **Delete Stop/Tag**: Deletes the selected stop/tag, see To delete a stop/tag/ pause.

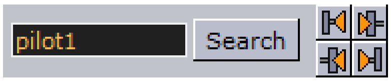
- **Linear Left and Right**: Sets the left and right spline handles in the selected keyframe to linear.

- **Smooth Left and Linear Right**: Sets the spline handles in the selected keyframe to smooth left and linear right.

- **Linear Left and Smooth Right**: Sets the spline handles in the selected keyframe to linear left and smooth right.

- **Smooth Left and Right**: Sets the left and right spline handles in the selected keyframe to smooth.

- **Handles Tangential**: Sets the left and right spline handles in the selected keyframe to tangential.



pilot1

Search

- **Search**: Searches the name of a keyframe. The arrow buttons can be used to search for the first, next, previous and last keyframe matching the search criteria.

- **Zoom to Standard**: Zooms the Dopesheet Editor and Spline Editor to standard view.

- **Zoom to Whole Spline**: Zooms the Dopesheet Editor and Spline Editor to fit the animation timeline of the current director.

- **Show Spline 1, 2, or 3**: Displays spline 1, 2, or 3 in the Spline Editor. The splines refer to animations performed on various axes. If only one axis is animated, only the spline 1 option is available.

Example: When animating a rotation, each spline describes the rotation around one axis. Spline 1 shows the rotation around axis X, spline 2 shows axis Y, and spline 3 shows axis Z.

- **Animate Timeline Marker**: If enabled, the Timeline Marker moves along the Dopesheet Editor and Spline Editor while playing the animation.

Tip: When computing performance should be reduced to a minimum, this option should be disabled.

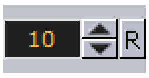
- **Show Info**: If enabled, holding the pointer over an item in the Dopesheet or Spline Editor, for example a keyframe, will display timeline information as a tool tip.

- **Seconds/Frames/Fields**: Switches the time scaling between seconds, frames, and fields. The time scaling is available in the bar separating the Dopesheet and Spline Editor.

- **Show Grid**: If enabled, shows a grid of vertical lines in the Dopesheet and Spline Editor, making it easier to position items accurately along the timeline.

- **Snap to Grid**: If enabled, shows the snap to grid feature in the Dopesheet and Spline Editor. This feature shows vertical lines at the field interval defined in the grid width option. By using this feature, keyframes and other items can be dragged precisely along the timeline, as it is not possible to drop items between the grid lines.

Example: If the snap to grid feature is enabled and the grid width is set to 10, it is only possible to drag keyframes to field 10, 20, 30, and so on. If a keyframe should be placed at field 23 while the snap to grid feature is enabled, this must be manually entered in the Keyframe Editor.



- **Grid Width**: Defines the width of the snap to grid in fields.

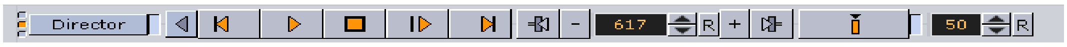
See Also

- Stage Controls and Shortcuts

7.3 Timeline Editor

The Timeline editor is located above the render output. It provides various options for controlling animations.

Figure 72: Timeline editor

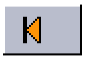


- **[Layer Indicator]**: The three vertical indicators show the current layer in the Director Control Panel. The three layers are Front, Main and Back.
- **[Director]**: Defines the selected (active) director and opens the Director Control Panel.

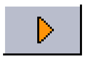
Note: The settings in the Director Control Panel are not saved with the scene. If only one director is selected, the button will be labelled with the director name. If multiple directors are selected, the button will be labelled *User Defined*. If all directors are selected, the button will be labelled *All*.

- **[Lock Selection]**: When activated, the director(s) of the selected container in the scene tree will be active in the stage. When deactivated, locks the selected director(s) so that no matter where else you click in the scene tree, only the selected director will be controlled by the buttons in the Timeline Editor.

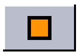
- **Reverse**: If enabled, the animation is played in reverse.



- **Go to Start**: The timeline jumps to the start of the current director(s). See To jump to start of an animation.

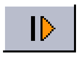


- **Start Animation**: Plays the current director(s) in the current layer from the beginning. See To play all animations in a director from the start of the timeline.

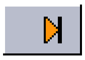


- **Stop Animation**: Manually stops the currently playing animation at the current point in the timeline. Right-clicking stops the currently selected director(s). See To stop an animation.

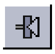
Note: The animation is automatically stopped when it reaches the end of the timeline, or a stop point along the timeline. For more information, see Locked vs Unlocked Keyframes.



- **Continue Animation**: Continues the animation from where it was stopped. See To continue an animation.

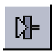


- **Go to End**: Jumps to the end of the animation.

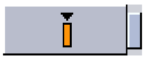


- **Keyframe Previous**: Jumps to the previous keyframe of the selected container in the scene tree.

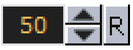
- **Timeline**: The input box shows the current position on the timeline in fields.



- **Keyframe Next**: Jumps to the next keyframe of the selected container in the scene tree.



- **Set Key**: Creates a keyframe at the defined timeline.
- **[Lock Timeline Position]**: When activated, the timeline jumps automatically after a Set Key operation by the defined Increment Value (see below). When deactivated, the timeline does not jump after a Set Key operation.



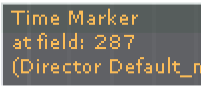
- **Increment Value**: The input box defines the step value (in fields) to be incremented when creating a keyframe.

7.4 Timeline Marker

The timeline marker defines the current position along the time axis (X) in the Dopesheet and Spline Editor. The timeline marker is displayed as a thin red vertical line.

A separate timeline marker is created for each director in the stage tree. Dragging a timeline marker shows the animation in the render output accordingly.

Figure 73: Timeline marker tool tip



Time Marker
at field: 287
(Director Default_r

If the Show Info option is enabled in the Stage Editor Bar, a tool tip with information about the current time and belonging director will appear when holding the pointer over the timeline marker.

If the Animate Timeline Marker option is enabled in the Stage Editor Bar, the timeline marker is animated in the Dopesheet while playing the animation.

This section contains information on the following procedures:

- To move the timeline marker (Alternative 1)
- To move the timeline marker (Alternative 2)
- To move the timeline marker (Alternative 3)
- To move the timeline marker (Alternative 4)
- To move the timeline marker five fields on the timeline
- To move the timeline marker one field on the timeline (Alternative 1)
- To move the timeline marker one field on the timeline (Alternative 2)
- To move the timeline marker one field on the timeline (Alternative 3)

To move the timeline marker (Alternative 1)

- Drag the timeline marker along the timeline in either the Dopesheet Editor or Spline Editor.

To move the timeline marker (Alternative 2)

- Click an empty area of the Dopesheet Editor or Spline Editor.

Note: In complex animations with many directors, channels, and actors, there might not be an empty area to click. If this is the case, press `<Alt>` while clicking.

To move the timeline marker (Alternative 3)

- In the Dopesheet Editor, enter the time (in fields) in the **Timeline** input box.

To move the timeline marker (Alternative 4)

1. Enable the Stage Object Editor.
2. Select the director in the Stage Tree or Dopesheet Editor.
3. In the Director Editor that appears, enter the time (in fields) in the **Timeline** text box.

To move the timeline marker five fields on the timeline

- Press the left or right arrow key on the keyboard.

To move the timeline marker one field on the timeline (Alternative 1)

- Press `<Ctrl>` and the left or right arrow key on the keyboard.

To move the timeline marker one field on the timeline (Alternative 2)

- Press `<Alt>` and the left or right arrow key on the keyboard.

To move the timeline marker one field on the timeline (Alternative 3)

- Press the Timeline up/down buttons in the Dopesheet Editor or Director Editor.

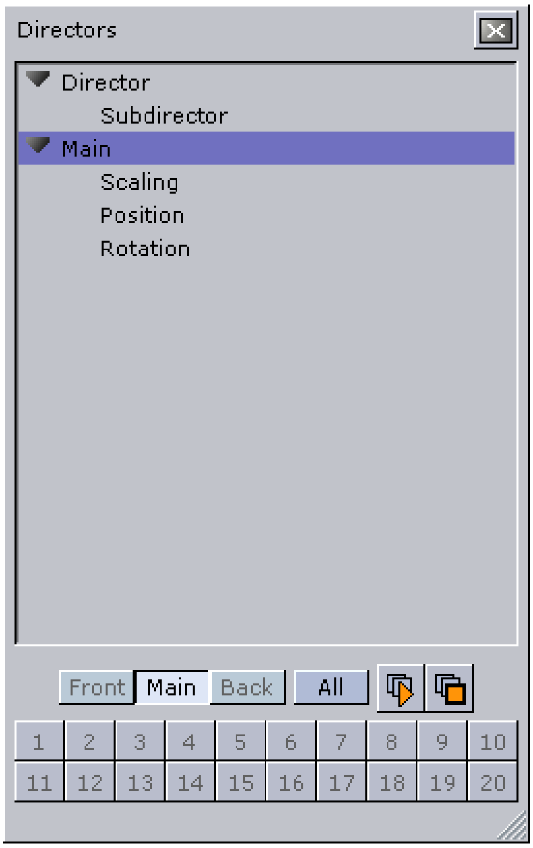
See Also

- To jump to start of an animation
- To jump to end of an animation

7.5 Director Control Panel

Clicking the director option in the Timeline Editor opens the Director Control Panel.

Figure 74: Director Control Panel showing two directors at root level with various subdirectors



The Director Control Panel lists all available Directors of the selected layer.

It is possible to select one or more directors in the list.

- **Front**: Lists the directors belonging to the front layer scene.
- **Main**: Lists the directors belonging to the main scene.

- **Back**: Lists the directors belonging to the back layer scene. This button is only enabled if a back layer has been defined in the Layer Manager.
- **All**: If activated, selects all directors in the list.
- **Start Animation**: Plays the current director(s) of all layers from the beginning. See Basic Animation Functions.
- **Stop Animation**: Clicking manually stops the currently playing animation at the current point in the timeline. Right-clicking stops only the selected directors. See Basic Animation Functions.

Note: The animation is automatically stopped when it reaches the end of the timeline, or a stop point along the timeline. For more information, see Locked vs Unlocked Keyframes.

- **[numbers]**: You can save up to 20 combinations of selected directors per layer by selecting them and dragging them onto one of the numbers. Afterward, when you click a number, the combination of saved directors is then selected. Note that these selections are on a per-layer basis and also cannot be saved.

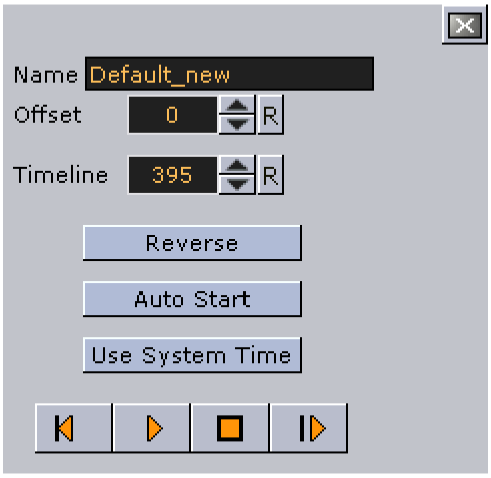
See Also

- Director Editor
- Working with Directors in the Stage Tree
- Working with Directors in the Dopesheet Editor

7.6 Director Editor

The Director editor shows details about the selected director.

Figure 75: Director editor



Name Default_new

Offset 0

Timeline 395

- **Name**: Defines the name of the director. See To rename a director.
- **Offset**: Sets the offset in fields for the whole director from the start of the timeline. This is the easiest way to shift all animations within the director. See To delay animations in a director using offsets.
- **Timeline**: Defines the timeline of the director in fields (each director has a separate timeline). Changing the value in this box will move the Timeline Marker in the Timeline Editor, and the other way around.

- **Reverse**: If enabled, the animations in this director will run backwards. For more information, see To play an animation in reverse.
- **Auto Start**: If enabled, the animation will start automatically when it is loaded from the database.
- **Use System Time**: This function plays seamless looping directors, if you have multiple scenes with the same background animation. As an example, if you have multiple scenes for elections but each of the scenes has the same background animation/director and you set the Use System Time to **Active** for this director, it will play the background without dropouts or breaking the loop while loading the other scenes, as long as they have the same director where the system time is activated.

⏮

- **Go to Start**: Jumps to the start of the timeline, see To jump to start of an animation.

▷

- **Start Animation**: Starts the animation from the beginning, see Basic Animation Functions.

□

- **Stop Animation**: Manually stops the animation at the current timeline, see To stop an animation.

Note: The animation is automatically stopped when it reaches the end of the timeline, or a stop point along the timeline. For more information, see Locked vs Unlocked Keyframes.

I▷

- **Continue Animation**: Continues the animation after it has been stopped, see To continue an animation.

To enable the director editor

1. Perform the procedure To enable/disable the stage object editor.
2. Click a director in the Stage Tree or in the Dopesheet Editor.

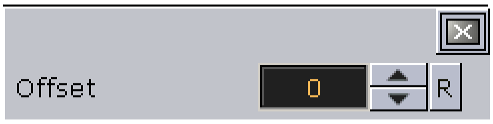
See Also

- Director Control Panel
- Working with Directors in the Stage Tree
- Working with Directors in the Dopesheet Editor
- Stage Controls and Shortcuts

7.7 Actor Editor

The Actor editor shows details about the selected actor.

Figure 76: Actor editor



Offset 0

- **Offset**: Sets the offset in fields for the whole actor from the start of the timeline. This is the easiest way to shift all animations within the actor.

To enable the actor editor

1. Perform the procedure To enable/disable the stage object editor.
2. Click an actor in the Stage Tree or in the Dopesheet Editor.

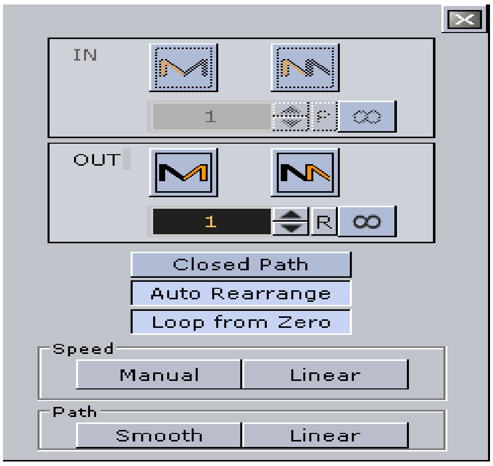
See Also

- Working with Actors in the Stage Tree
- Working with Actors in the Dopesheet Editor

7.8 Channel Editor

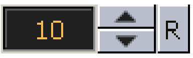
The Channel editor shows details about the selected channel.

Figure 77: Channel editor



- **Swing In**: Enables swing in.

- **Loop In**: Enables loop in.



10

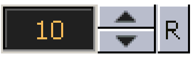
- **Number of Swings/Loops**: Defines the number of swings/loops in.

∞

- **Infinite**: Sets number of swings/loops in to infinite (from the start of the animation).

- **Swing Out**: Enables swing out.

- **Loop Out**: Enables loop out.



- **Number of Swings/Loops**: Defines the number of swings/loops out.

∞

- **Infinite**: Sets the number of swings/loops out to infinite (animation never ends).

To enable the channel editor

1. Perform the procedure To enable/disable the stage object editor.
2. Click a channel in the Stage Tree or in the Dopesheet Editor.

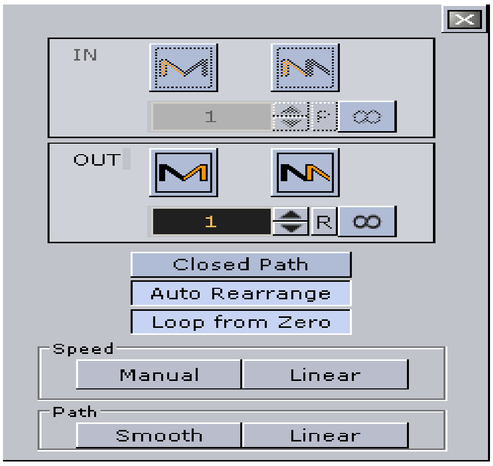
See Also

- Working with Channels in the Stage Tree
- Working with Channels in the Dopesheet Editor

7.9 Action Channel Editor

The action editor shows details about the selected action channel.

Figure 78: Action channel editor



- **Swing In**: Enables swing in.

- **Loop In**: Enables loop in.

10 R

- **Number of Swings/Loops**: Defines the number of swings/loops in.

∞

- **Infinite**: Sets number of swings/loops in to infinite (from the start of the animation).

- **Swing Out**: Enables swing out.

- **Loop Out**: Enables loop out.

10 R

- **Number of Swings/Loops**: Defines the number of swings/loops out.

∞

- **Infinite**: Sets the number of swings/loops out to infinite (animation never ends).

The Action Channel editor also has the following buttons:

- **Closed Path**: When activated, the animation path with go from the last keyframe to the first. When deactivated, the animation path will remain open.

- **Auto Rearrange**: By default, this setting is enabled, ensuring that the order of the keyframes in the dopesheet editor is the same as in the Scene Editor so that the animation only ever continues forward. When disabled, if you change the order of the keyframes in the dopesheet editor, the spline will remain the same, but the animation may return from the previously last keyframe to the newly added keyframe now in last position.
- **Loop from Zero**: When activated, if there is a gap before an animation loop, the gap gets included in the loop (like 2.x behavior).
- **Speed**
  - **Manual**: Allows you to change the speed between different keyframes.
  - **Linear**: Ensures the same speed between all keyframes.
- **Path**
  - **Smooth**: Allows for a curving animation.
  - **Linear**: Allows only an linear animation.

Note: These settings are global for the entire animation in question.

Figure 79: Smooth path

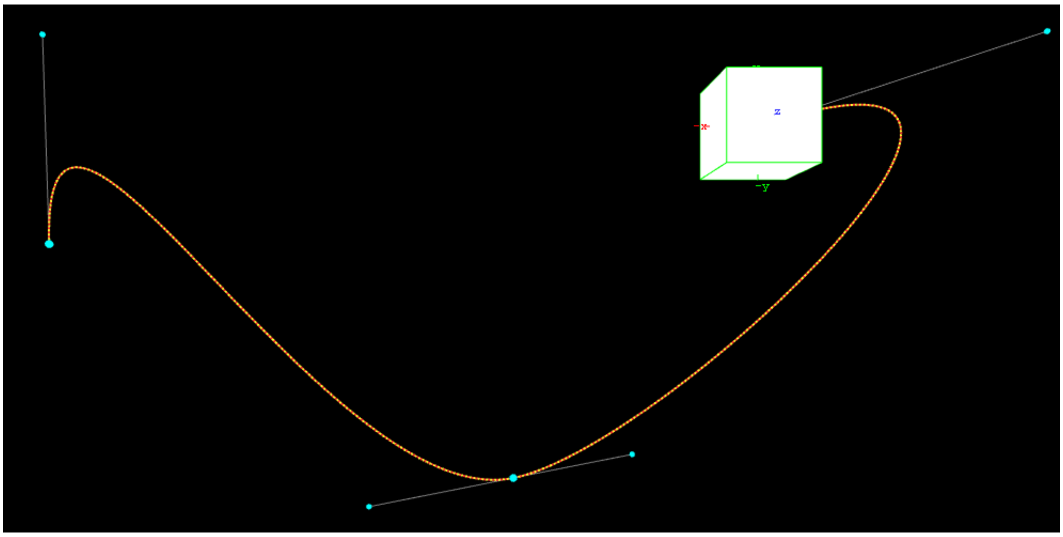
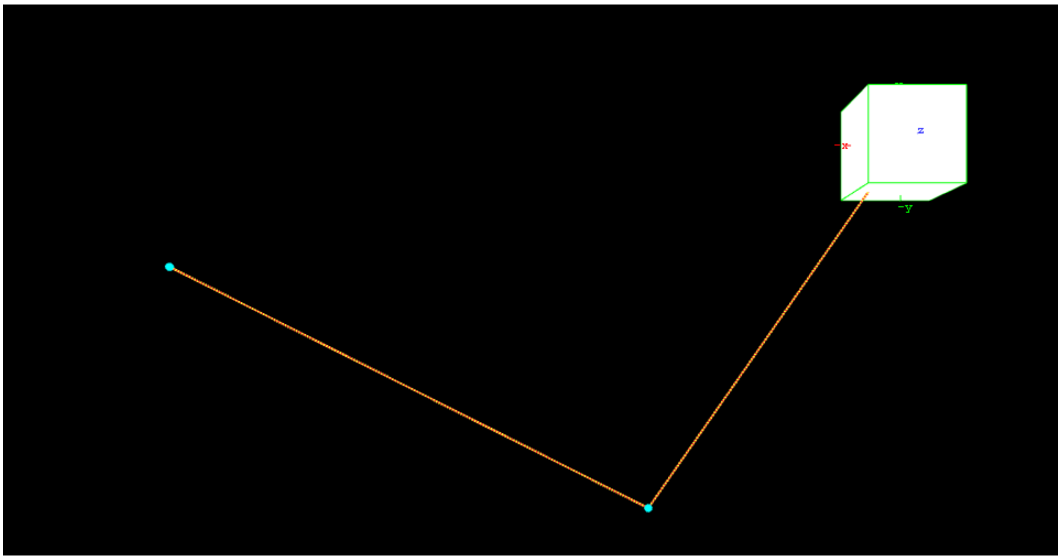


Figure 80: Linear path



To enable the action channel editor

1. Perform the procedure To enable/disable the stage object editor.
2. Click an action channel in the Stage Tree or in the Dopesheet Editor.

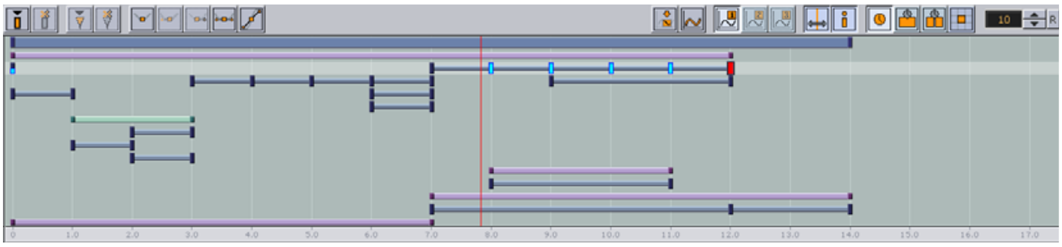
See Also

- Working with Action Channels in the Stage Tree
- Working with Action Channels in the Dopesheet Editor

7.10 Dopesheet Editor

The Dopesheet editor displays the items in the stage tree as bars along the timeline.

Figure 81: Dopesheet editor

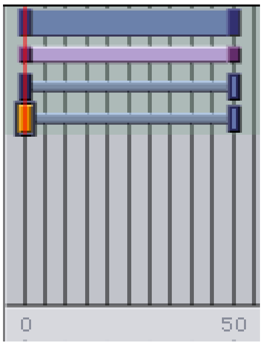


The items are presented vertically in the Dopesheet editor according to the structure in the stage tree. The horizontal scaling represents the time value.

Tip: The time value can be set to either seconds, frames, or fields from the Stage Editor Bar.

The Timeline Marker defines the current time, and the render output shows the animation accordingly.

Figure 82: Dopesheet editor with Snap to Grid option enabled with field width set to 5



The Grid and Snap to Grid options that can be enabled in the Stage Editor Bar make it possible to position items in the Dopesheet editor more accurately.

Figure 83: Channel tool tip

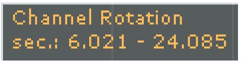
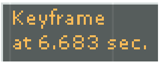

Channel Rotation
sec.: 6.021 - 24.085

Figure 84: Keyframe tool tip

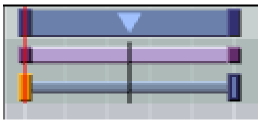

Keyframe
at 6.683 sec.

If the Show Info option is enabled in the Stage Editor Bar, a tool tip with time information will appear when holding the pointer over an item in the Dopesheet editor.

The Dopesheet editor can contain bars for various stage items: directors, actors, channels, and so on. Similar to the Stage Tree, directors are displayed with a gray blue color, container actors with light purple, while channels and actions are displayed with a gray color.

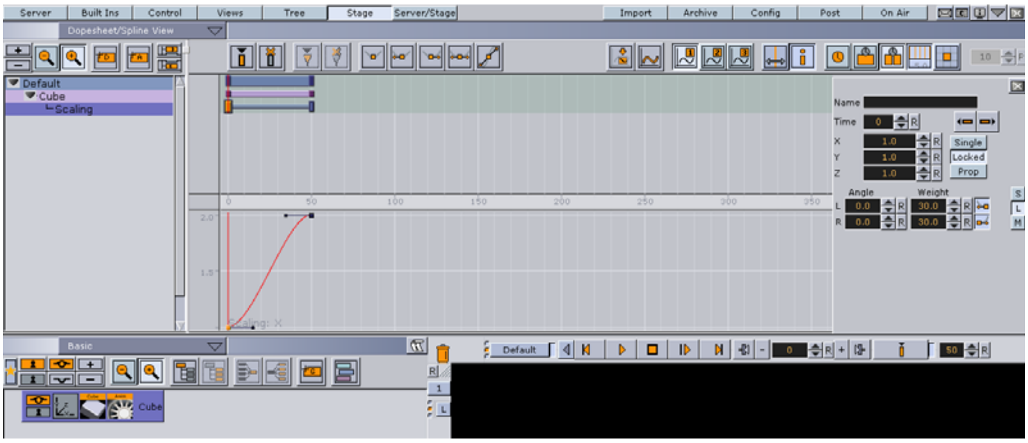
To build up an animation, keyframes and stops/tags can be added to the scene. A keyframe defines a property at a given time. Multiple keyframes belonging to the same channel are connected in the Dopesheet editor with one or more bars.

Figure 85: Example of animation where stop/tag has been added between two keyframes



A stop/tag can be added to a director to end or pause an animation at a given time. See Working with Stops, Tags and Pauses.

Figure 86: Example of animation where Cube container is scaled at field 0 and 50



Name
Time 0
X 1.0
Y 1.0
Z 1.0
Single
Locked
Prop
Angle
Weight
L 0.0 30.0
R 0.0 30.0

Example: If the scaling of a cube is 1.0 at field 0 and 2.0 at field 50, one keyframe will be created at field 0 and another at field 50. A bar will connect the two keyframes in the Dopesheet editor.

If the Stage Object Editor is enabled, clicking an item in the Dopesheet editor will open the corresponding editor. See To enable/disable the stage object editor.

Example: Clicking a director will open the Director Editor, clicking a keyframe will open the Keyframe Editor, and so on.

To enable/disable the dopesheet editor

- From the Stage Editor Menu select **Dopesheet View**.

  Alternatively while holding the pointer over the Stage press the keyboard shortcut `<Ctrl>+<2>`.

To enable the dopesheet editor together with the spline editor

- From the Stage Editor Menu select **Dope/Spline View**.

  Alternatively while holding the pointer over the Stage press the keyboard shortcut `<Ctrl>+<4>`.

To resize the height of the dopesheet editor

- Drag the horizontal separator between the Dopesheet editor and Spline Editor.

To resize the width of the dopesheet editor

- Drag the vertical separator between the Stage Tree Area and Stage Editor Area.

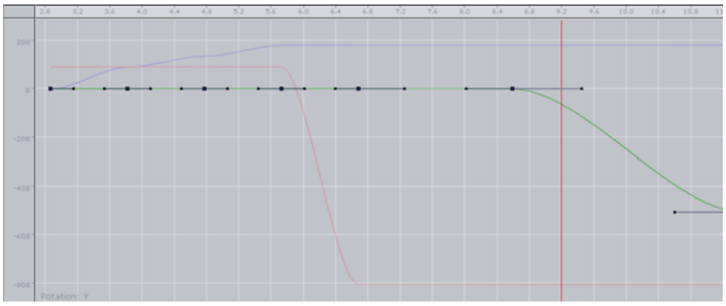
See Also

- Working with Directors in the Dopesheet Editor
- Working with Actors in the Dopesheet Editor
- Working with Channels in the Dopesheet Editor
- Working with Action Channels in the Dopesheet Editor
- Working with Keyframes

7.11 Spline Editor

The Spline editor displays the keyframes on a channel as splines along the timeline.

Figure 87: Spline editor



The keyframes are presented vertically in the Spline editor according to the properties they possess. The horizontal scaling represents the time value.

Tip: The time value can be set to either seconds, frames, or fields from the Stage Editor Bar.

The Timeline Marker defines the current time, and the render output shows the animation accordingly.

It is possible to drag keyframes in the Spline Editor both horizontally (changing the time) and vertically (changing the value) at once.

This section contains information on the following topics:

- To enable/disable the spline editor
- To enable the spline editor together with the dopesheet editor
- To resize the height of the spline editor
- To resize the width of the spline editor
- To switch between the various splines

- To move keyframes in the spline editor
- Position in the Spline Editor
- Rotation in the Spline Editor
- Scale in the Spline Editor
- Material in the Spline Editor
- Additional Best Practices for the Spline Editor

To enable/disable the spline editor

- From the Stage Editor Menu select **Spline View.**
  Alternatively while holding the pointer over the Stage press the keyboard shortcut `<Ctrl>+<3>`.

To enable the spline editor together with the dopesheet editor

- From the Stage Editor Menu select **Dope/Spline View.**
  Alternatively while holding the pointer over the Stage press the keyboard shortcut `<Ctrl>+<4>`.

To resize the height of the spline editor

- Drag the horizontal separator between the Dopesheet editor and Timeline Marker.

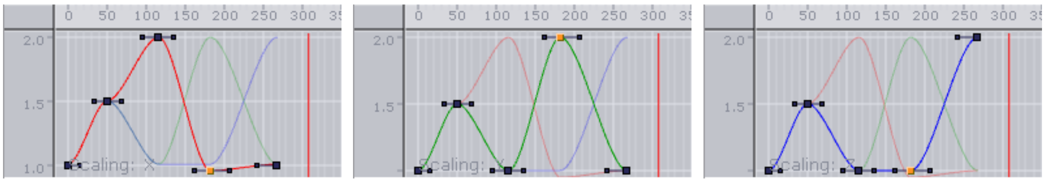
To resize the width of the spline editor

- Drag the vertical separator between the Stage Tree Area and Stage Editor Area.

To switch between the various splines

- Use the Show Spline 1,2, and 3 buttons from the Stage Editor Bar
  Alternatively use the keyboard shortcuts `<Ctrl>+<Z>`, `<Ctrl>+<X>` and `<Ctrl>+<C>` .

Figure 88: Example of axes X, Y, and Z in the Spline editor



To move keyframes in the spline editor

- Select the desired keyframe and drag it around.
- To change only the position on the timeline or the value, press `<Shift>` to lock the axes. Keep `<Ctrl>` pressed while you drag the keyframe if you want to snap to grid along the timeline.

You can move a keyframe in the spline editor, so either the position on the timeline (left and right) or the value stored (up and down) in the keyframe will be changed.

7.11.1 Editing Handles

As each keyframe on the spline has two handles, moving them is the most easy way to edit the curve. To edit a handle left click it and drag it around. You will see that the spline will reflect your movement.

By default you move the handles proportional this way. This means if you change the handle on one side, the one on the other side will move too. If you want to only edit the handle on one side, you need to left click the desired handle. Keep the left button pressed and press `<Ctrl>`. If you drag the handle, only the one you drag right now will move and the other one will stay. This “single handle mode” will be enabled for the keyframe as long as you press `<Ctrl>`.

Furthermore you can use the buttons in the main menu to edit the handles at a keyframe. To do so select the keyframe you want to change the handles in and select the action in the main menu. For ease of use multiple keyframes can be selected.

- set left and right spline handles in selected keyframe to linear.
- set spline handles in selected keyframe to smooth left and linear right.
- set spline handles in selected keyframe to linear left and smooth right.
- set left and right spline handles in selected keyframe to smooth.
- set left and right spline handles in selected keyframe to tangential.

See Also

- Scene Editor Animation Controls and Shortcuts

7.11.2 Position in the Spline Editor

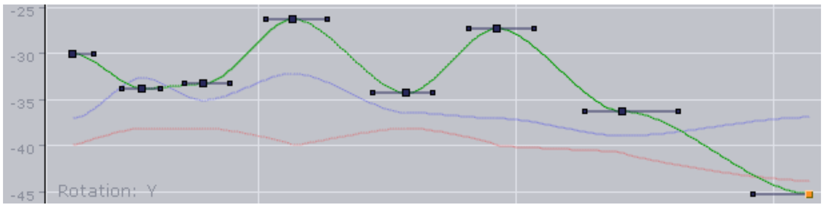
If you animate a position, there are much more considerations to make.

See Also

- Path Control
- Spline Control
- Working with Position Keyframes

7.11.3 Rotation in the Spline Editor

If you have animated the rotation, you can rotate about up to all three axes at the same time. In the spline editor three different splines will be shown, one for every axes. The red one is for the X-axis, the green one for the Y-axis and the blue one for the Z-axis. For ease of use in the lower left corner you will find indicated which axis is currently selected.

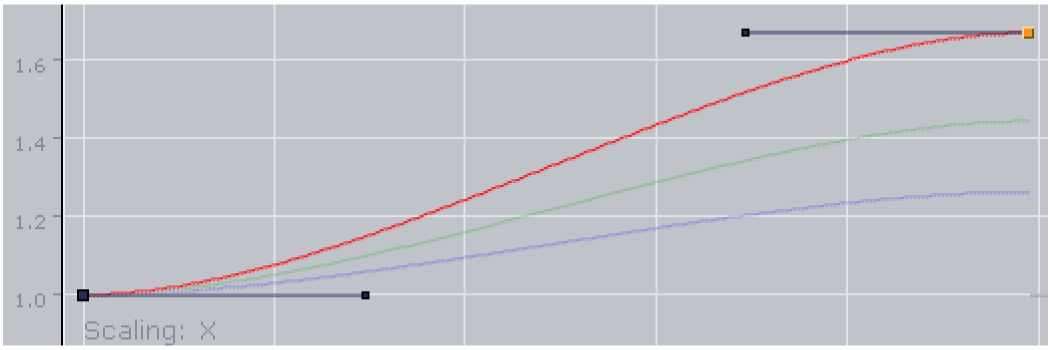


See Also

- To create keyframes for rotation
- To modify the rotation settings
- Working with Rotation Keyframes

7.11.4 Scale in the Spline Editor

If you have animated the scale, you can scale along up to all three axes. In the spline editor three different splines will be shown (when **Prop** or **Single** is selected), one for every axis. The red one is for the X-axis, the green one for the Y-axis and the blue one for the Z-axis. For ease of use in the lower left corner you will find indicated which axis is currently selected.

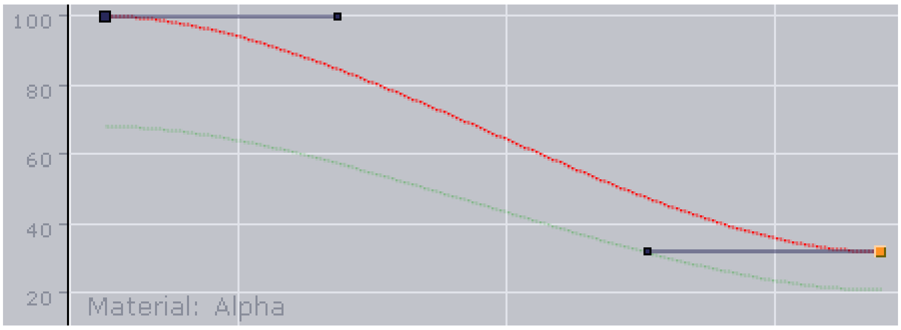


See Also

- Working with Scale Keyframes

7.11.5 Material in the Spline Editor

If you have animated the material, in the spline editor two different splines will be shown. The red one is for the alpha value, the green one for the shininess. For ease of use in the lower left corner you will find indicated which parameter is currently selected.



See Also

- To add and animate a material
- Animating Material Keyframes

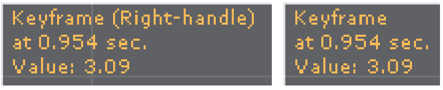
7.11.6 Additional Best Practices for the Spline Editor

Normally for each channel only one spline is shown in the spline editor. But there are animation channels where more splines are available for editing. As you can only edit the keyframes/handles of one spline, you must use the number keys in the Stage Editor Bar to select the desired spline.

Pressing `<Shift>` while moving a keyframe in the Spline Editor enables you to drag it only horizontally or vertically, but not both directions at the same time. Pressing `<Ctrl>` while moving a keyframe will enable the Snap to Grid function, allowing the keyframe to be placed only at certain timeline intervals. For more information, see Stage Editor Bar.

When animating certain channels, such as scaling or rotation, up to three splines will be displayed in the Spline editor, each describing the state along one axis, either X, Y, or Z. For a better overview, different colors are used; red for X-axis, green for Y-axis, and blue for Z-axis.Every spline contains various keyframes with handles.

Figure 89: Example of keyframe and handle tool tips


Keyframe (Right-handle)
at 0.954 sec.
Value: 3.09

Keyframe
at 0.954 sec.
Value: 3.09

If the Show Info option is enabled in the Stage Editor Bar, placing the pointer over a keyframe or handle will display a tool tip with status information about timeline and value.

7.12 Stage Object Editor

The Stage Object editor is positioned at the right side of the Stage Editor Area. The editor has different views depending on the particular editor it is displaying.

Example: The Director Editor shows information about directors, the Actor Editor shows information about actors, the Channel Editor shows information about channels, the Actor Editor shows information about actions, the Keyframe Editor shows information about keyframes, the Stops, Tags and Pause Editor shows information about stops/tags, and so on.

Clicking an object in the Dopesheet Editor or selecting an item from the stage tree will open the corresponding editor in the Stage Object Editor.

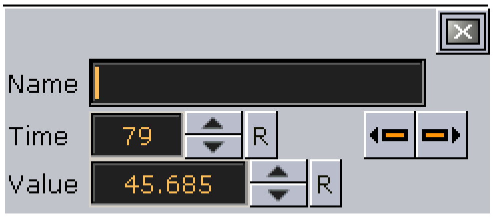
To enable/disable the stage object editor

- From the Stage Editor Menu select **Show Obj. Editor.**
  Alternatively, while holding the mouse pointer over the Stage, press `<Ctrl>+<1>`.

7.13 Keyframe Editor

The Keyframe editor shows details about the selected keyframe.

Figure 90: Keyframe editor



Name:
Time: 79 R
Value: 45.685 R

- **Name**: Defines the name of the keyframe. This name is used for external commands or scripts. For more information, see how To rename a keyframe.
- **Time**: Defines where along the timeline (in fields) the keyframe is placed.
- **Previous**: If the property has more than one keyframe, the previous keyframe along the timeline is selected.
- **Next**: If the property has more than one keyframe, the next keyframe along the timeline is selected.
- **[Value]**: The rest of the Keyframe editor options vary according to which property is being animated.

Example: For scaling, it is possible to define the size of an object in direction X, Y, and Z.

To enable the keyframe editor

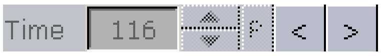
1. Perform the procedure To enable/disable the stage object editor.
2. Click an keyframe in the Stage Tree or in the Dopesheet Editor.

This section contains information on the following topics:

- Locked vs Unlocked Keyframes
- Working with Position Keyframes
- Working with Rotation Keyframes
- Working with Scale Keyframes
- Animating Material Keyframes

7.13.1 Locked vs Unlocked Keyframes

By default the keyframes in the position animation are locked. They store information about the position of the container along the X-, Y- and Z-axis, as well as the time the keyframe will be passed by in the animation.



An unlocked keyframe stores only information of the position along X-, Y- and Z-axis, without the information at which time the keyframe should be passed. To reflect this, in the object editor for a locked keyframe the **Time** is grayed out. Furthermore, as the spline editor shows time and path-percentage, the unlocked keyframes will not be shown there. In the animation editor unlocked keyframes will be shown at half height of locked keyframes. Viz Artist calculates the time the keyframe will be passed to fit with the handles set in the next locked keyframe before and after the unlocked keyframe.

To better understand the difference between locked and unlocked keyframes, create a simple position animation. You will see the container moving with different speeds between the keyframes. Next unlock all keyframes and play the animation once more. The container will move evenly during the whole animation.

Using unlocked keyframes you can define the path to fit your needs without editing the time for every keyframe. This will save you a great amount of time.

7.13.2 Working with Position Keyframes

A position keyframe contains information about the position of a container at a specific time. The parameters of position keyframes are:

- Path Control
- Speed Control

Path Control

An animation’s path can be either smooth or linear.

Figure 91: Smooth path

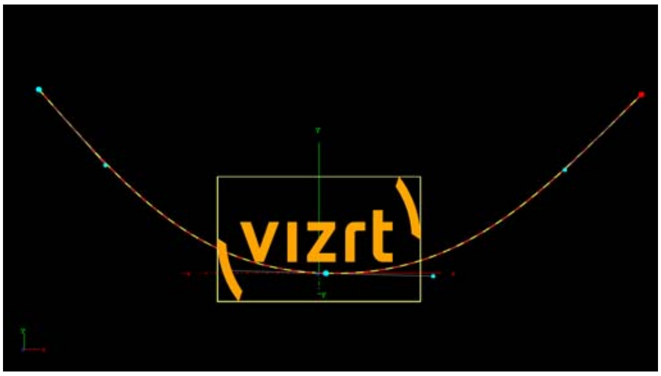
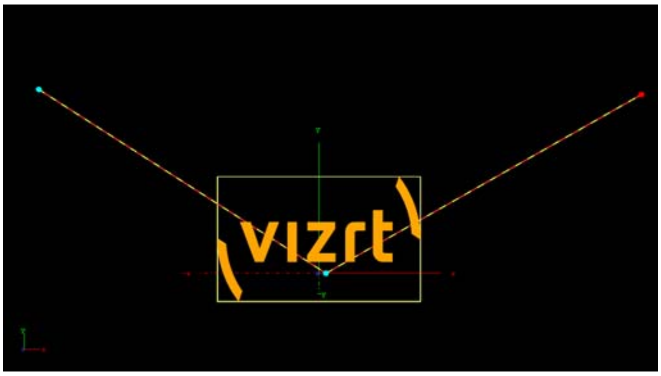
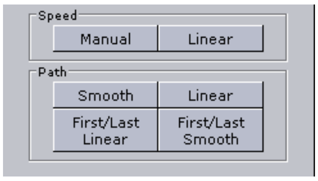


Figure 92: Linear path



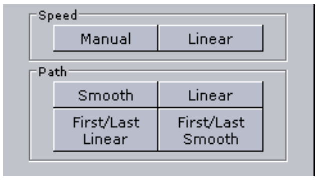
The path (and speed) control editor allows you to set the path.



- **Smooth**: Sets the animation path to be smooth.
- **Linear**: Sets the animation path to be linear.
- **First/Last Linear**: Sets the animation to be linear for the first and last keyframe, and smooth for the rest.
- **First/Last Smooth**: Sets the animation to be smooth for the first and last keyframe, and linear for the rest.

Speed Control

The (path and) speed control editor allows you to configure speed options.



- **Manual Speed**: locks all keyframes. There is no linear speed with this option. The duration between each keyframe can be different, depending on the timing value and the transformation values between the different keyframes. The result is that you will not have a continuous animation speed.

- **Linear Speed**: unlocks all keyframes. The keyframes between the first and last keyframes are locked. If you change the value of either the first or last keyframe, or any of the keyframes in between, the timing will change automatically for each keyframe. The result is an animation with a continuous speed between all keyframes.

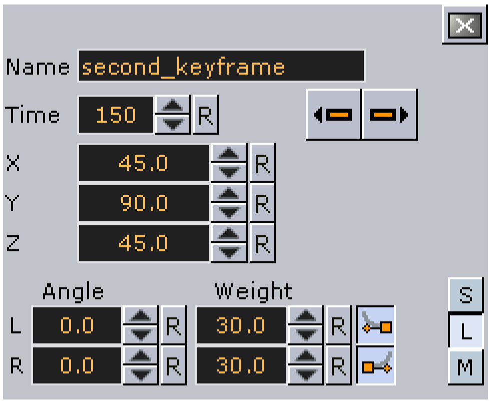
See Also

- Locked vs Unlocked Keyframes

7.13.3 Working with Rotation Keyframes

A rotation keyframe contains information regarding container rotation. In addition to the default options in the Keyframe Editor, for a rotation keyframe it is possible to define specific rotation options.

Figure 93: Keyframe editor with rotation options



Name second_keyframe

Time 150

X 45.0

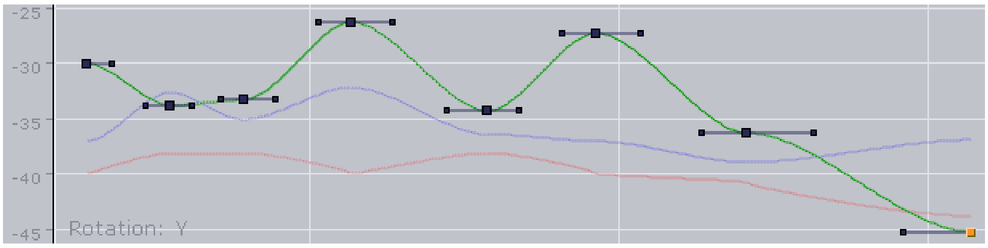
Y 90.0

Z 45.0

	Angle	Weight
L	0.0	30.0
R	0.0	30.0

- **X**: Defines the rotation along the X axis.
- **Y**: Defines the rotation along the Y axis.
- **Z**: Defines the rotation along the Z axis.
- **Angle**: Defines the angle of the handles for the path or speed.
- **Weight**: Defines the weight of the handles for the path or speed.
- **Smooth Left/Right**: Allows the handles for the path to be smooth.
- **Single (S)**, **Locked (L)** and **Mirror (M)**: Defines the way handles for the path or speed should be modified.

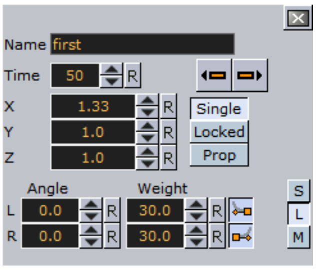
Furthermore in the spline editor three different splines will be shown, one for every axes. The red one is for the X-axis, the green one for the Y-axis and the blue one for the Z-axis. As you can only edit the currently selected spline, you must use three spline buttons at the top of the Dopesheet Editor (or shortcuts `<Ctrl>+<Z>`, `<Ctrl>+<X>` and `<Ctrl>+<C>`) to select the spline you want to edit. For ease of use in the lower left corner you will find indicated which axis is currently selected.



7.13.4 Working with Scale Keyframes

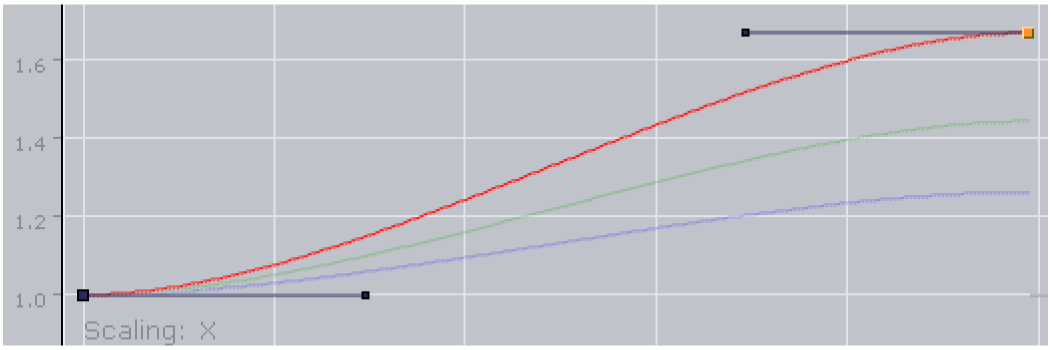
Scale keyframes have to do with the scaling of the object in the animation. For a scale keyframe, in addition to the default options in the Keyframe Editor, it is possible to define the scaling along the X, Y, and Z axes in the three input boxes.

Figure 94: Keyframe editor with scale options


Name first
Time 50
X 1.33
Y 1.0
Z 1.0
Angle L 0.0 Weight L 30.0
Angle R 0.0 Weight R 30.0

If you have animated the scale, you can scale along up to all three axes. To change all three scale values, in the object editor you will find an input box for **X**, one for **Y** and one for **Z**. There you can change if the values should be **Locked** or if you change them proportional (**Prop**) or each **Single** value.

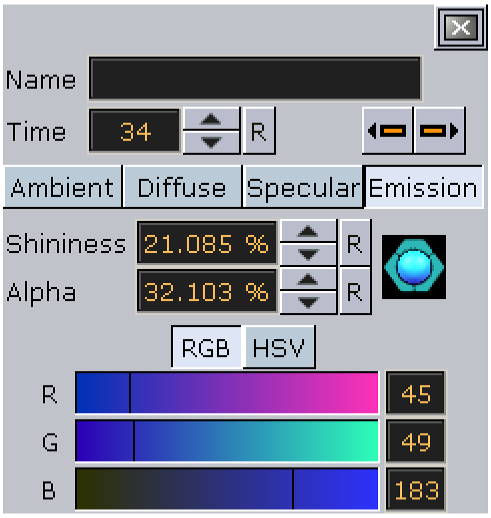
Furthermore in the spline editor three different splines will be shown (when **Prop** or **Single** is selected), one for every axis. The red one is for the X-axis, the green one for the Y-axis and the blue one for the Z-axis. As you can only edit the currently selected spline, you must use three spline buttons at the top of the Dopesheet Editor (or shortcuts `<Ctrl>+<Z>`, `<Ctrl>+<X>` and `<Ctrl>+<C>`) to select the spline you want to edit. For ease of use in the lower left corner you will find indicated which axis is currently selected.



7.13.5 Animating Material Keyframes

For a material keyframe, in addition to the default options in the Keyframe Editor, it is possible to define color.

Figure 95: Keyframe editor with material options



Name

Time 34 R

Ambient | Diffuse | Specular | Emission

Shininess 21.085 % R

Alpha 32.103 % R

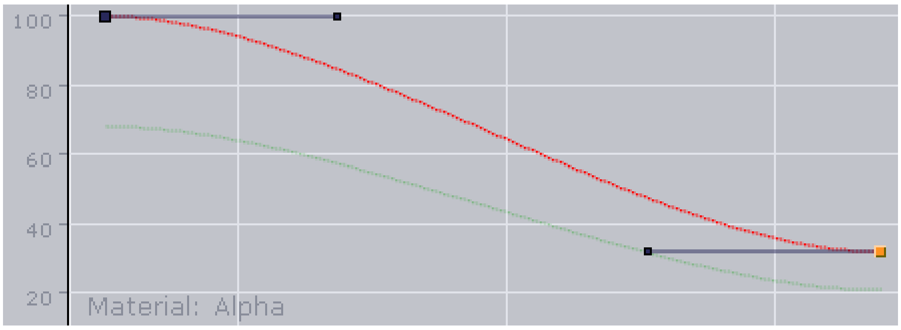
RGB | HSV

R 45

G 49

B 183

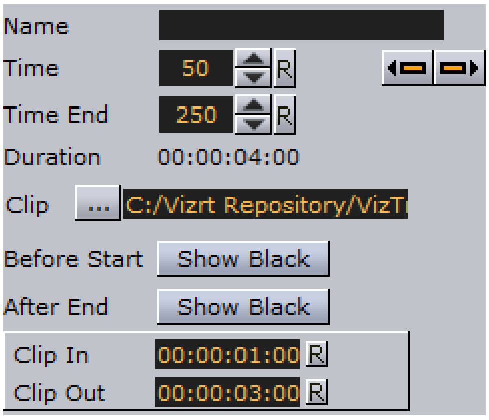
If you have animated the material, the object editor for the keyframes shows a Material editor. This allows you to conveniently the change the material. Furthermore in the spline editor two different splines will be shown. The red one is for the alpha value, the green one for the shininess. As you can only edit the keyframes/handles of one spline, you must use the number keys in the Stage Editor Bar to select the desired spline.For ease of use in the lower left corner you will find indicated which parameter is currently selected



.

7.13.6 Working with Clip Keyframes

Figure 96: Keyframe editor with clip options



Name

Time 50 R

Time End 250 R

Duration 00:00:04:00

Clip ... C:/Vizrt Repository/VizTi

Before Start Show Black

After End Show Black

Clip In 00:00:01:00 R

Clip Out 00:00:03:00 R

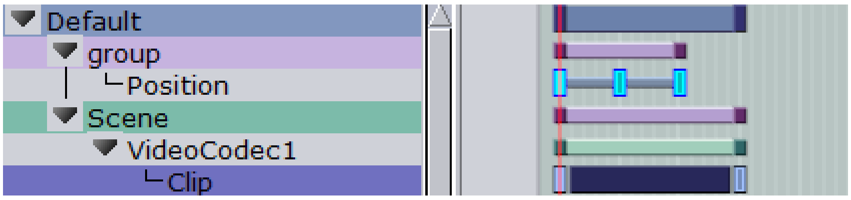
Time and **TimeEnd** define the starting and ending time in fields of the key frame. **Clip In** and **Clip Out** mark the corresponding times within the clip itself.

Example:

- Time = 50
- Time End = 250
- Clip Length = 4 sec (50i clip)
- Clip In = 00:00:01:00
- Clip Out = 00:00:03:00

The clip playback starts 1 second after the director starts. The clip starts at frame 25 and ends at frame 75. Because the key frame duration is twice as long the clip part will be played twice.

To add a clip to the stage

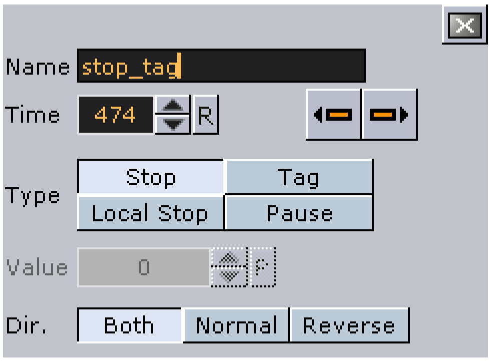


1. Right-click the director where you want your clip and select **Add Scene** from the appearing context menu
2. Right click the Scene item and from the appearing context menu select, for example, **VideoCodec1** (or 2) and then **Clip**
   - To open the clip's keyframe editor click the keyframe in the dopesheet

7.14 Stops, Tags and Pause Editor

The Stop-Tag-Pause editor shows details about the selected stop/tag.

Figure 97: Stop-Tag-Pause editor



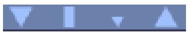
Name stop_tag

Time 474

- **Name**: Defines the name of the stop/tag. This name is used for external commands or scripts. For more information, see how To rename a stop/tag/pause.
- **Time**: Defines where along the timeline (in fields) the stop/tag is placed. For more information, see how To move a stop, tag or pause.
- **Previous**: If the director has more than one stop/tag, the previous stop/tag along the timeline is selected.

- **Next**: If the director has more than one stop/tag, the next stop/tag along the timeline is selected.
- **Type**:
  - **Stop**: Stops the animation until being continued. A stop applies for the director and all subdirectors.
  - **Tag**: Indicates a position along the timeline. A tag does not stop the animation.
  - **Local Stop**: Stops the animation until being continued. A local stop only applies for the director it is placed on, not its subdirectors.
  - **Pause**: Makes it possible to stop the animation for a certain amount of time (in frames).

Figure 98: Example of stop, tag, local stop, and pause on a director in the Stage



- **Dir**:
  - **Both**: Sets the direction the stop/tag will be valid to both normal and reverse.
  - **Normal**: The stop/tag is only valid when the animation direction is normal.
  - **Reverse**: The stop/tag is only valid when the animation direction is reversed.

To enable the Stop/Tag/Pause editor

1. Perform the procedure To enable/disable the stage object editor.
2. Click a stop/tag/pause in the Dopesheet Editor.

8 Creating Animations

By creating an animation, the state of an object (for example its position, rotation, or scaling) is defined at various points in time. Keyframes are used to define these states, and the animation system interpolates between them to create a continuous animation.

This section contains information on the following topics:

- Working with Directors in the Stage Tree
- Working with Directors in the Dopesheet Editor
- Working with Stops, Tags and Pauses
- Working with Actors in the Stage Tree
- Working with Actors in the Dopesheet Editor
- Working with Channels in the Stage Tree
- Working with Channels in the Dopesheet Editor
- Working with Action Channels in the Stage Tree
- Working with Action Channels in the Dopesheet Editor
- Working with Keyframes
- Basic Animation Functions
- Creating a Basic Animation
- Creating an Advanced Animation
- Further Advanced Animation Topics

See Also

- Scene Editor Animation Controls and Shortcuts
- Stage Controls and Shortcuts

8.1 Working with Directors in the Stage Tree

When creating an animation, there is at least one director in the stage tree. Directors are used to group a certain set of stage elements, such as actors and channels in the stage tree. It is possible to create directors at the root level of the stage tree, or as subdirectors. A director can hold as many items as desired, but to improve usability it is recommended to group certain items in different directors, especially when working with complex animations. In addition to improved overview in the stage tree, multiple directors make it possible to show only the animations contained within each director. This opens the possibility to for example have multiple animation channels on one object driven by multiple directors. Each director can also hold controlling elements such as stop points and actions that only affect the animations within the director. For more information, see Basic Animation Functions.

Directors are displayed in the stage tree with a gray blue background color.

This section contains information on the following procedures:

- To create a director

- To move a director
- To copy a director
- To rename a director
- To delete a director

See Also

- Working with Directors in the Dopesheet Editor
- Director Control Panel
- Director Editor

To create a director

- Click the **Add New Director** button in the Stage Tree Bar.

Alternatively, for greater control of where the (sub)director is created:

- Drag and drop the Add Director icon at the right part of the director label in the stage tree. (Dropping the icon at the left part of the label will create a director at the same hierarchy level.)

To move a director

- Drag one or more directors to a different place in the stage tree hierarchy.

To copy a director

- Press <Ctrl> while dragging one or more directors to a different place in the stage tree hierarchy.

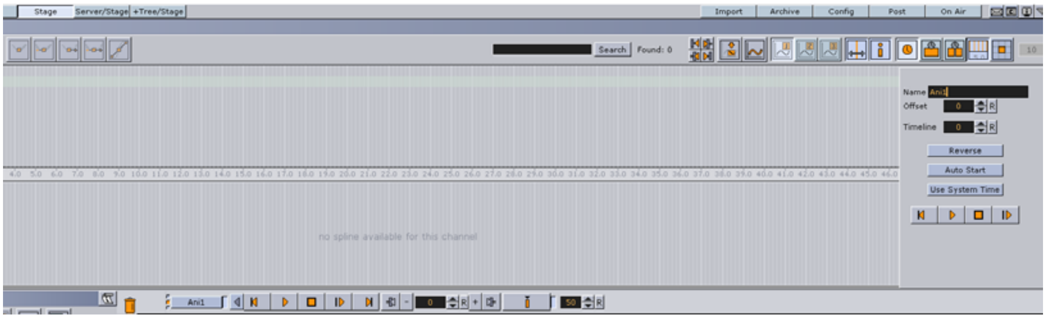
To rename a director

1. Right-click a director in the stage tree.
2. In the Stage Tree Menu that appears, select **Rename**.
3. In the text box that appears, type a new descriptive name, and then press <Enter>.

Note: Special characters, such as space, are not allowed in director names.

Alternatively:

1. Enable the Director Editor.
2. Select the director in the stage tree.
3. In the editor that appears, type a descriptive name in the **Name** text box, and then press <Enter>.



To delete a director

1. Right-click a director in the stage tree.
2. In the Stage Tree Menu that appears, select **Delete [Director]**.

Alternatively:

- Drag the director to the trash can.

8.2 Working with Directors in the Dopesheet Editor

Most actions on actors are done when Working with Directors in the Stage Tree. Actions affecting the timeline are performed directly in the dopesheet editor. Working with directors in the dopesheet editor enables the Director Editor.

Directors are displayed in the Dopesheet Editor with a gray-blue color.

This section contains information on the following topics:

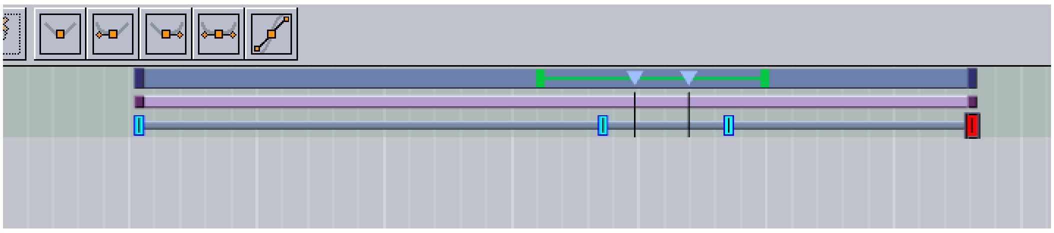
- To mark a region within a director
- To move a director in the timeline
- To speed a director up
- To slow a director down

See Also

- Working with Directors in the Stage Tree
- Director Control Panel
- Director Editor

To mark a region within a director

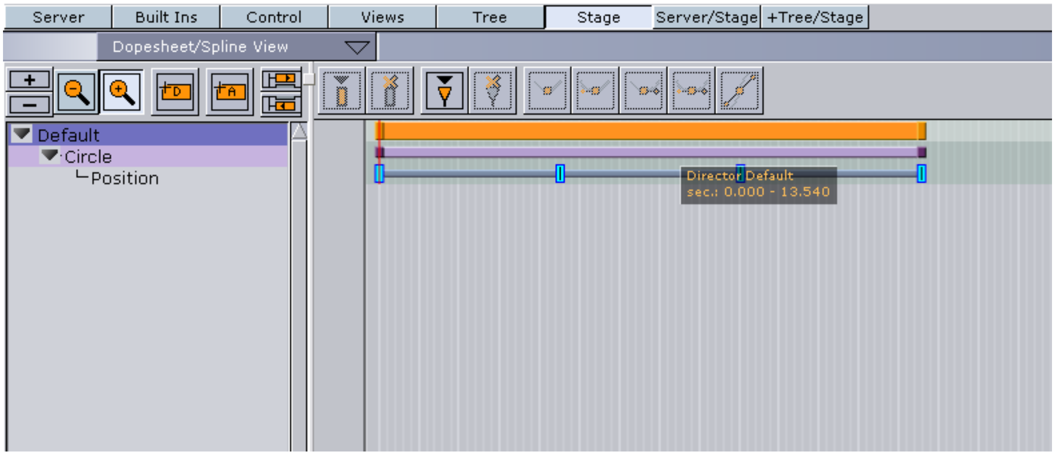
1. In the Dopesheet Editor, press <T> and click a starting point within a director.
2. Press <T> again and click an ending point within the director.



Dragging the green knobs to lengthen or shorten the selected region within the director affects only the keyframes inside this region.

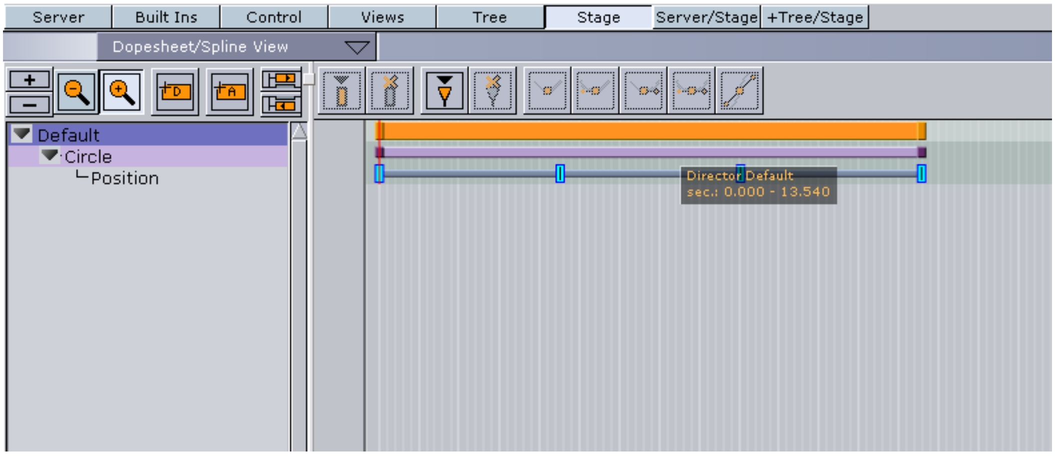
To move a director in the timeline

- Drag the director to a different time.



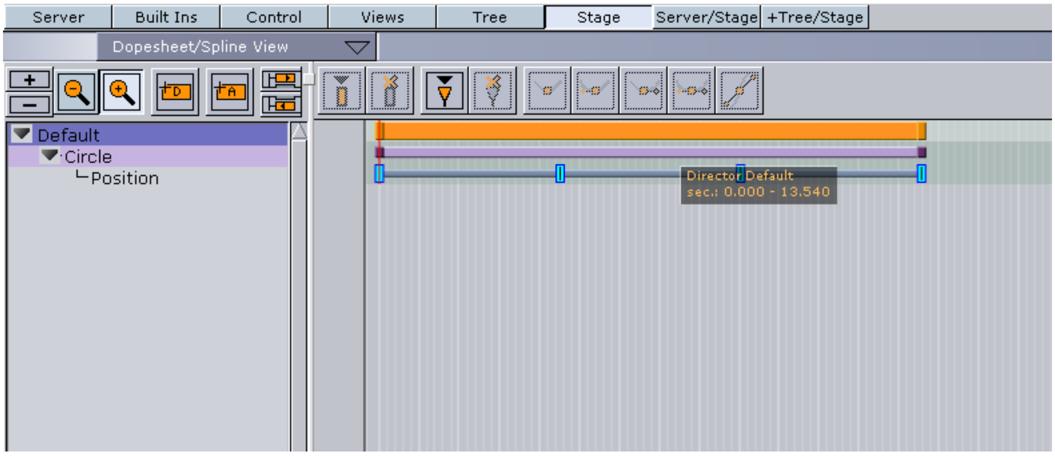
To speed a director up

- At either end of the director, click the knob and drag it to shorten the director.



To slow a director down

- At either end of the director, click the knob and drag it to lengthen the director.



8.3 Working with Stops, Tags and Pauses

It is possible to add stops, tags and pauses to a director. At a stop, the director will stop and wait until it is continued. A tag is used only to indicate a position along the timeline. A pause is where the director stops for a given interval before continuing automatically.

This section contains information on the following procedures:

- To create a stop, tag or pause
- To change a stop to a tag, pause or local stop (and vice versa)
- To move a stop, tag or pause
- To rename a stop/tag/pause
- To delete a stop/tag/pause

See Also

- Locked vs Unlocked Keyframes

To create a stop, tag or pause

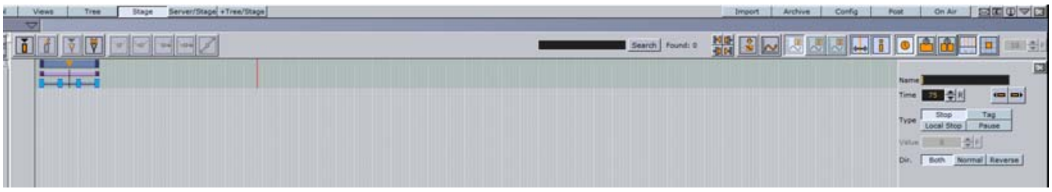
1. Select the director in the Stage Tree or Dopesheet Editor where the stop/tag is to be added.
2. Set the Timeline Marker to the point where the stop/tag is to be added.
3. Click the **Add Stop/Tag** button in the Stage Editor Bar.

Alternatively:

- Press **S** while clicking a director at the desired timeline in the Timeline Editor.

To change a stop to a tag, pause or local stop (and vice versa)

1. Enable the Stops, Tags and Pause Editor.
2. Select the stop/tag/pause you intend to change.



3. In the Stop-Tag-Pause editor, click a button to change status. Your options are:
   - **Stop**
   - **Tag**
   - **Local Stop**
   - **Pause**
4. If you select **Pause**, in the **Value** input box, define the number of frames that the animation should halt before continuing.

To move a stop, tag or pause

1. Select one or more stops/tags/pauses.
2. Drag the stops/tags/pauses along the director in the Dopesheet Editor.

Alternatively:

1. Enable the Stops, Tags and Pause Editor.
2. Select a stop/tag/pause.
3. In the Stop-Tag-Pause editor, enter the time (in fields) in the **Time** input box.

To rename a stop/tag/pause

1. Enable the Stops, Tags and Pause Editor.
2. Select a stop/tag/pause.
3. In the Stop-Tag-Pause editor, enter a descriptive stop/tag/pause name in the **Name** text box, and then press <Enter>.

Note: Special characters, such as space, are not allowed in stop/tag/pause names.

To delete a stop/tag/pause

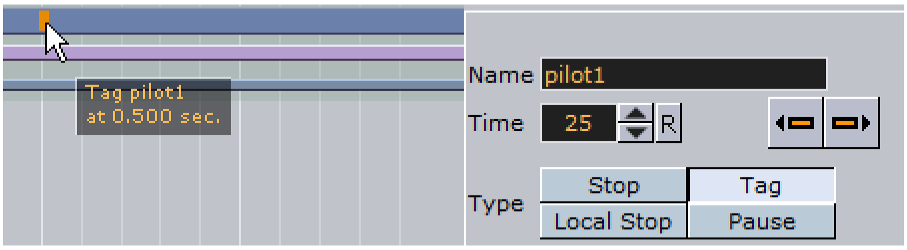
1. Select one or more stops/tags/pauses.

2. Click the **Delete Stop/Tag** button in the Stage Editor Bar.

Alternatively:

1. Select one or more stops/tags.
2. Press <Delete>.

To create a preview point for Viz Content Pilot



Name pilot1

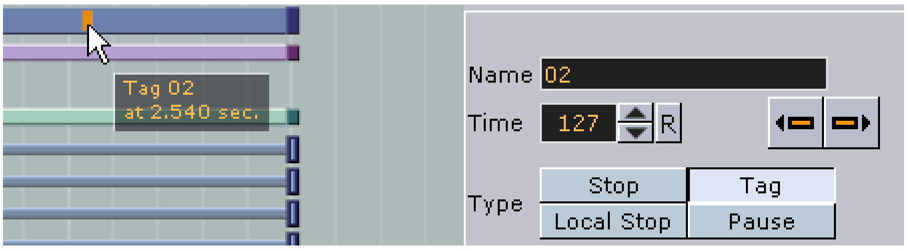
Time 25

1. Open the Stage, and select the **Default** director.
2. **Position** the **Time Marker** from where you would like the scene to be previewed.
3. Click the **Add a Stop/Tag** button.
4. In the appearing editor enter Name as **pilot1** (case sensitive), and set Type to Tag.
   - The **pilot1** tag is the default preview point for a standalone scene or a foreground scene in a transition logic scene.
   - Custom tags can be used as well; however, those tags are only selectable within the Newsroom Component.

To create a preview point for Viz Trio

- Repeat the procedure on how To create a preview point for Viz Content Pilot.

To create a preview point for Viz Trio with local preview



Name 02

Time 127

1. Open the Stage, and select the **Default** director.
2. **Position** the **Time Marker** from where you would like the scene to be previewed.
3. Click the **Add a Stop/Tag** button.
4. In the appearing editor enter Name as **01**, **02**, **03**, etc. according to the value of the corresponding Field identifiers (01, 02, 03, etc.).
   - Setting these points will, when selecting a tab field in Viz Trio (01, 02, 03, etc.), preview the scene in the local Viz Trio preview window.

See Also

- Tables and Preview Points

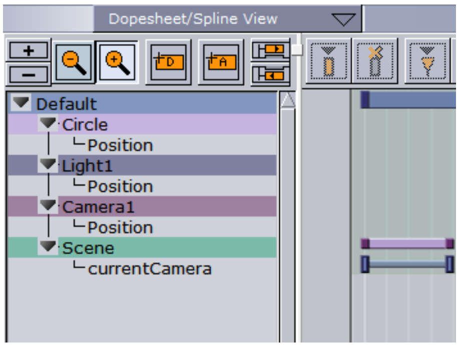
8.4 Working with Actors in the Stage Tree

A container is represented as an actor in the stage tree. The actor carries the same name in the stage tree as the corresponding container in the scene tree. If you rename the container, the actor will reflect the change.

When adding an animation to a container, the corresponding actor and channel will automatically appear in the stage tree inside a director. The actor for a container holds the information of all animated channels: scene, camera, or light.

As it is possible to place different animation channels of an actor in separate directors, the same actor can be available in several places in the stage tree.

Container actors are displayed in the stage tree in light purple. Scene actors are shown in green, camera actors in dark purple, and light actors in dark gray.



This section contains information on the following procedures:

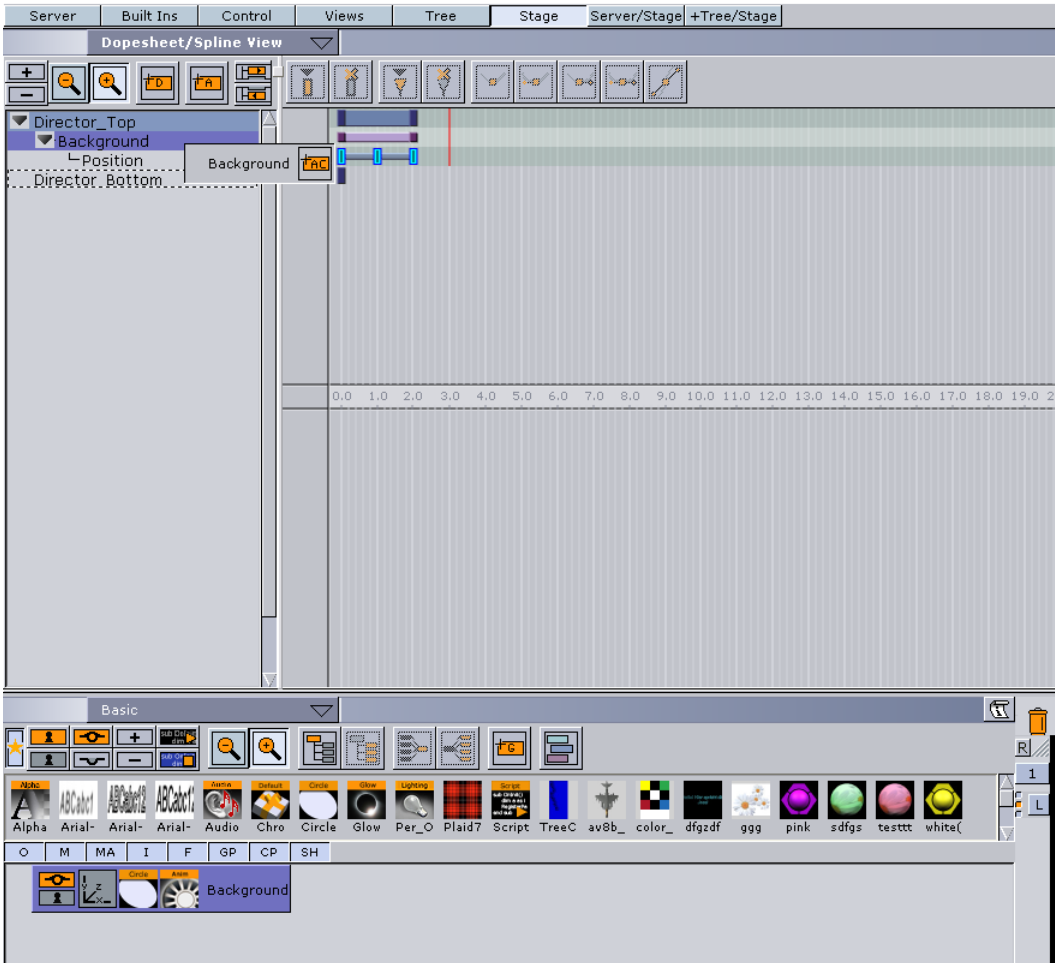
- To move an actor
- To copy an actor
- To delete an actor

See Also

- Working with Actors in the Dopesheet Editor
- Working with Channels in the Stage Tree
- Actor Editor

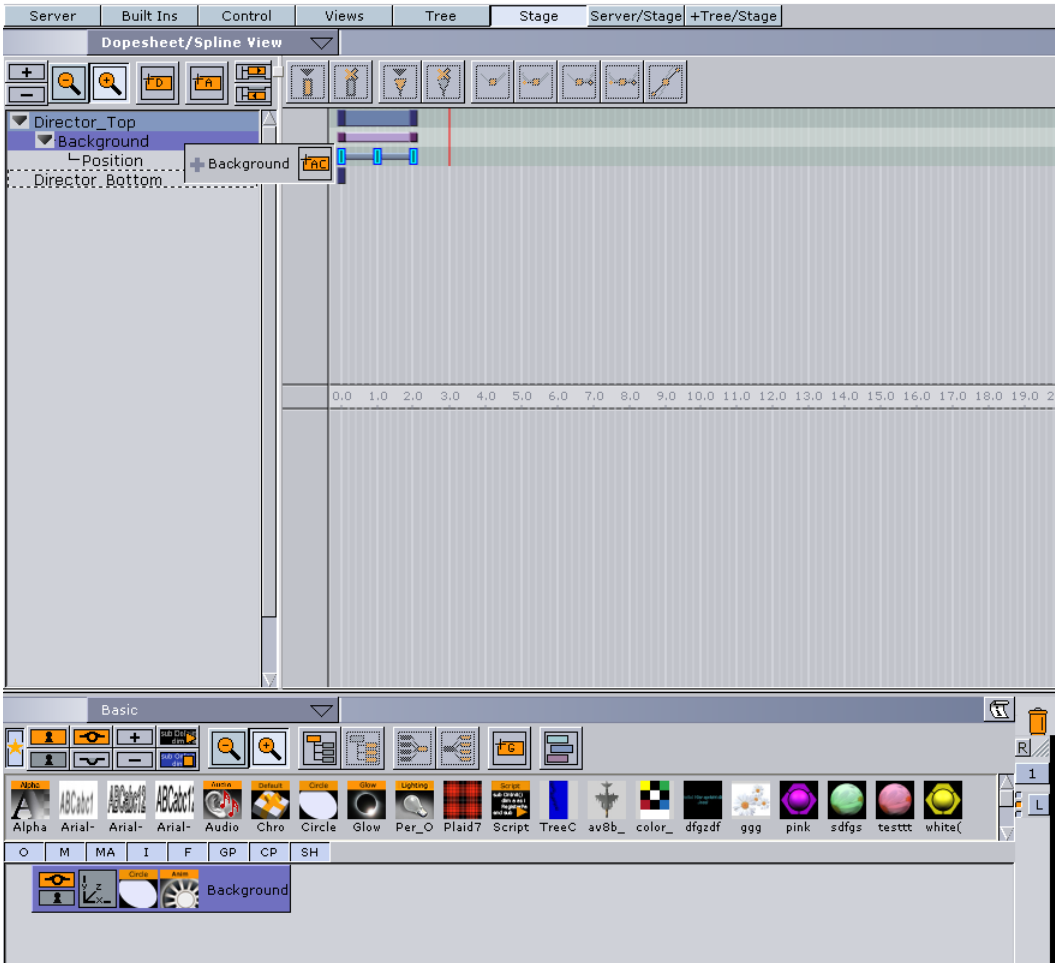
To move an actor

- Drag one or more actors from a director in the stage tree to the target director.



To copy an actor

- Press <Ctrl> while dragging one or more actors from a director in the stage tree to the target director.



To delete an actor

1. Right-click an actor in the stage tree.
2. In the Stage Tree Menu that appears, select **Delete [Actor]**.

Alternatively:

- Drag the actor to the trash can.

8.5 Working with Actors in the Dopesheet Editor

Most actions on actors are done when Working with Actors in the Stage Tree. Actions affecting the timeline are performed directly in the dopesheet editor. Working with actors in the dopesheet editor enables the Actor Editor.

This section contains information on the following procedures:

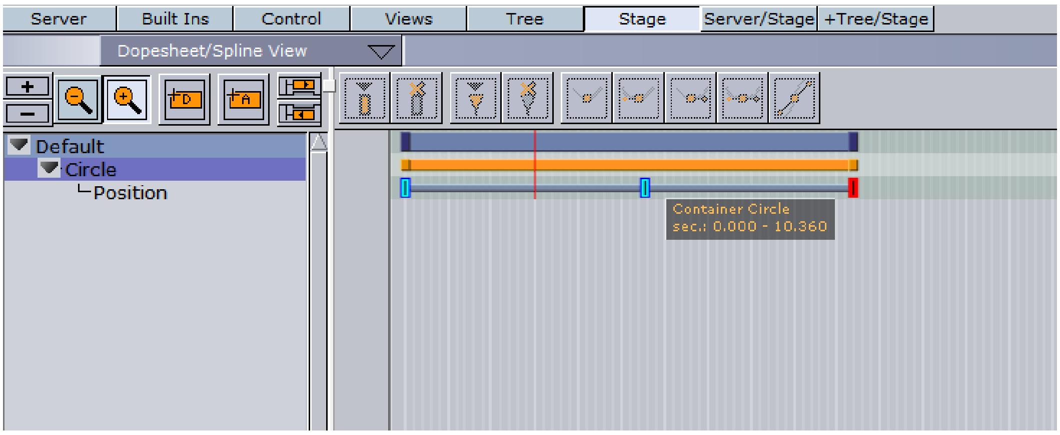
- To move an actor in the timeline
- To speed an actor up
- To slow an actor down
- To delay all the animations in an actor

See Also

- Working with Actors in the Stage Tree
- Working with Channels in the Stage Tree
- Actor Editor

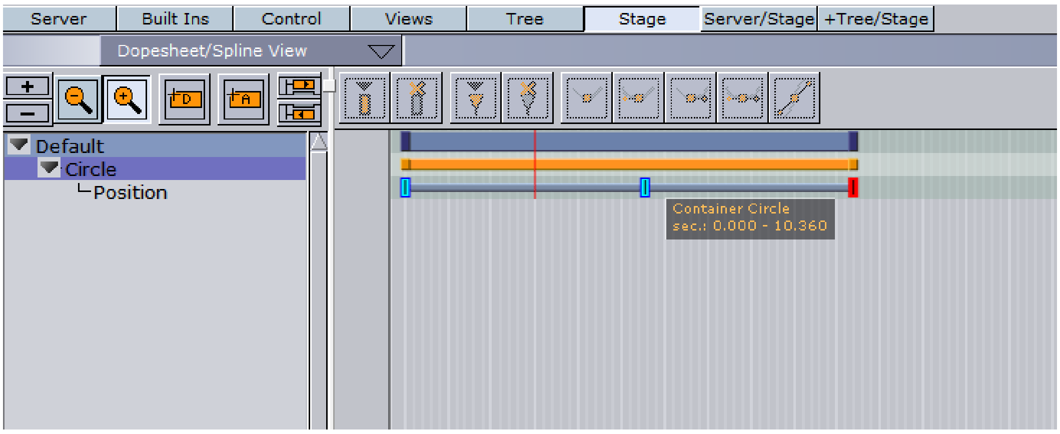
To move an actor in the timeline

- Drag the actor to a different time.



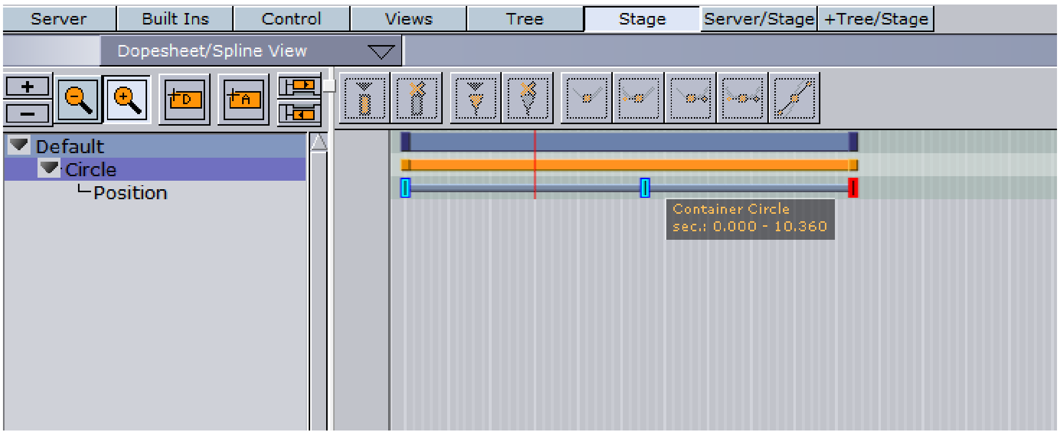
To speed an actor up

- At either end of the actor, click the knob and drag it to shorten the actor.



To slow an actor down

- At either end of the actor, click the knob and drag it to lengthen the actor.



To delay all the animations in an actor

1. Enable the Actor Editor.
2. Select an actor.
3. In the Actor Editor, set the **Offset** value (in fields).

Alternatively, it is possible to move the actor (horizontally along the timeline) including all its animations directly in the Dopesheet Editor, although this will not actually add the offset value.

8.6 Working with Channels in the Stage Tree

Every animated property of an actor is stored in a separate channel. Separate channels are created for position, rotation, scaling, and so on. The channel is named automatically according to the property being animated. It is not possible to rename a channel.

Along the channel, keyframes are placed. Keyframes hold information about the value of the property *at the time they are placed on the timeline*. If keyframes are *not* connected by a channel in the Dopesheet Editor, the information about the property is identical, and there is no animation of this property between these keyframes.

This section contains information on the following procedures:

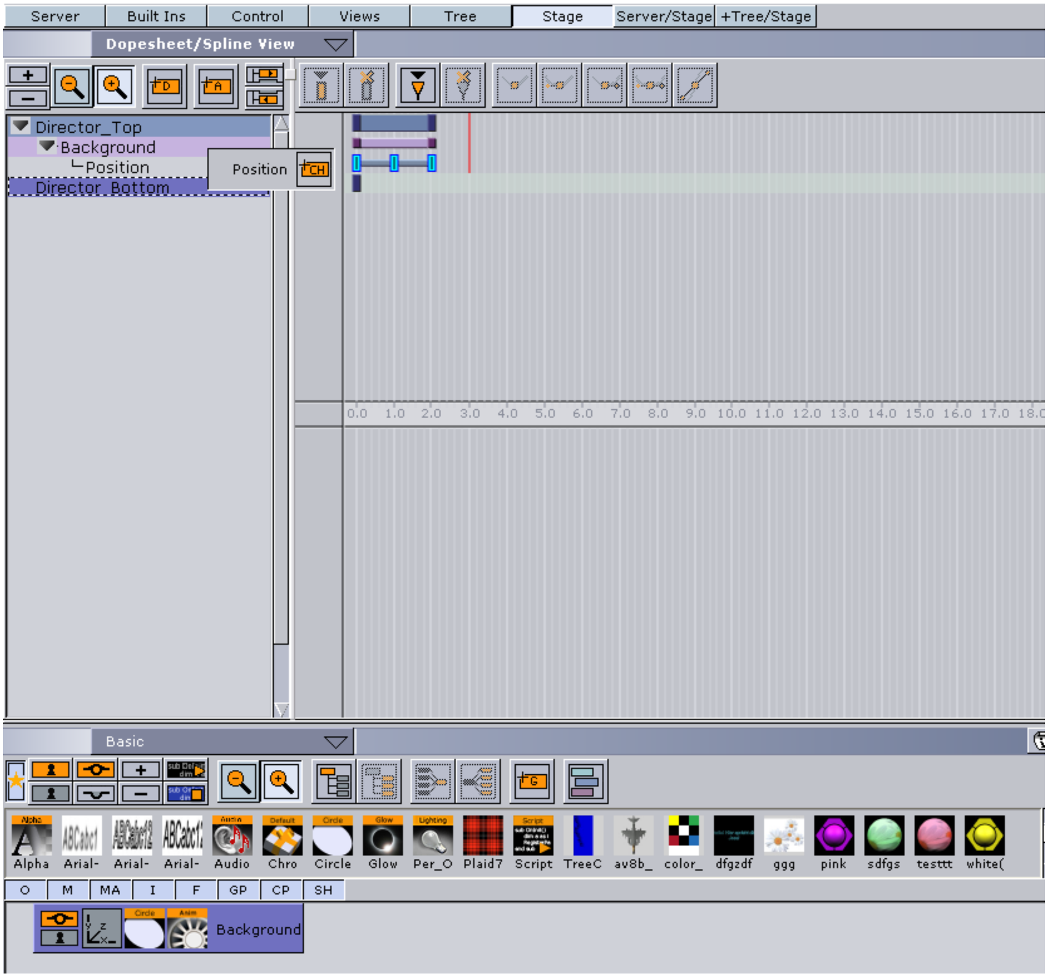
- To move a channel
- To copying a channel
- To select a channel
- To delete a channel

See Also

- Working with Channels in the Dopesheet Editor
- Working with Actors in the Stage Tree
- Working with Keyframes
- Channel Editor

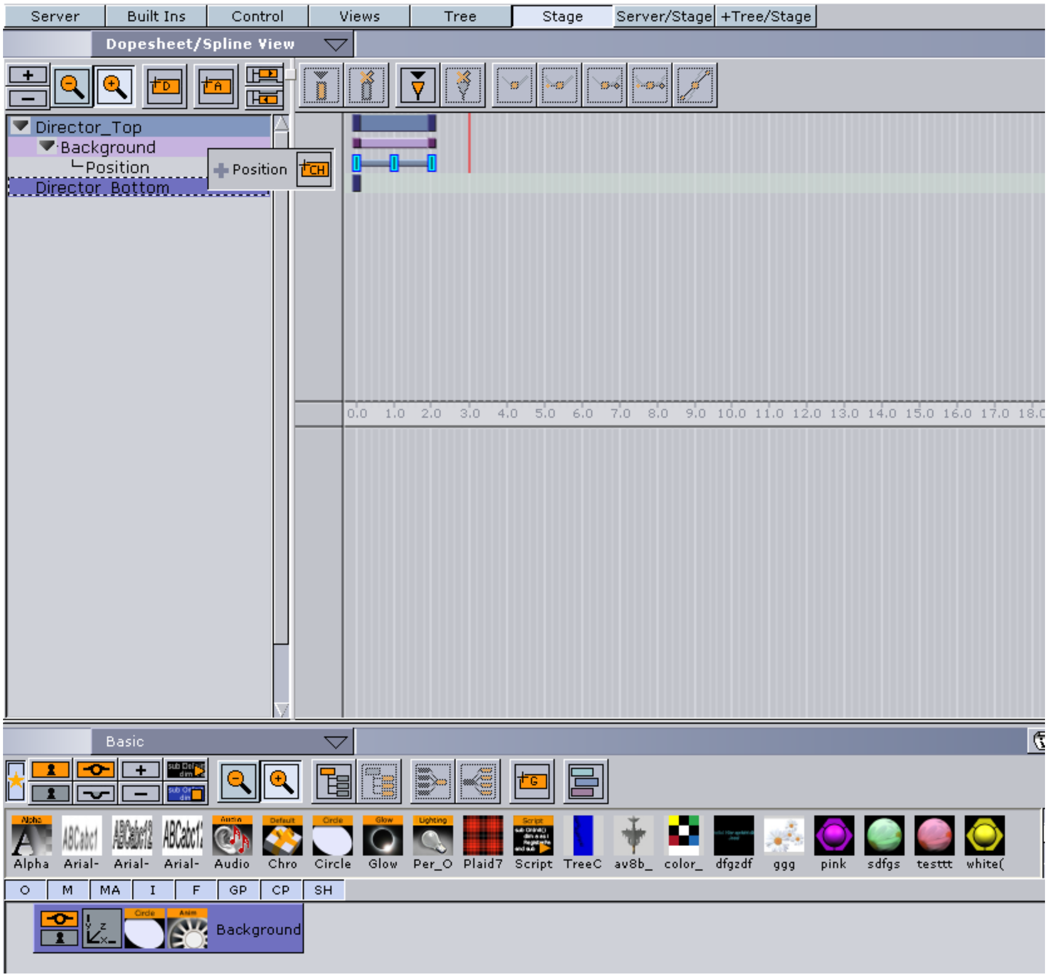
To move a channel

- Drag one or more channels from a director in the stage tree to the target director.



To copying a channel

- Press <Ctrl> while dragging one or more channels from a director in the stage tree to the target director.



To select a channel

- Click a channel in the Dopesheet Editor.

Alternatively:

- Click one or more channels in the stage tree.

To delete a channel

1. Right-click a channel in the stage tree.
2. In the Stage Tree Menu that appears, select **Delete [Channel]**.

Alternatively:

- Drag the channel to the trash can.

See Also

- Animating Channels
- Dopesheet Editor
- Spline Editor

8.7 Working with Channels in the Dopesheet Editor

Most actions on channels are done when Working with Channels in the Stage Tree. Actions affecting the timeline are performed directly in the dopesheet editor. Working with channels in the dopesheet editor enables the Channel Editor.

This section contains information on the following topics:

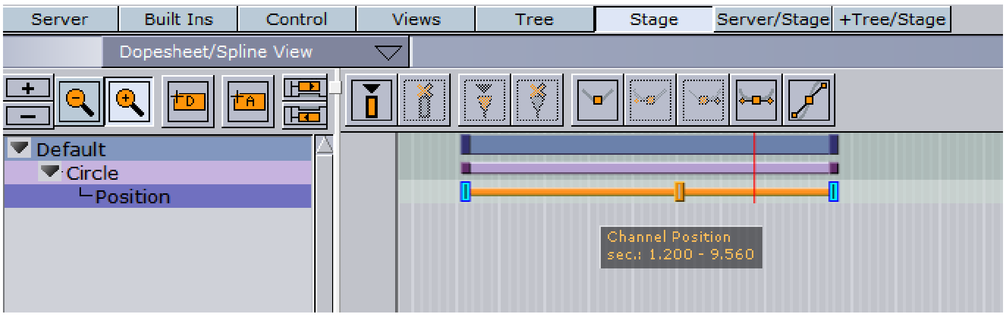
- To move a channel in the timeline
- To speed a channel up
- To slow a channel down

See Also

- Working with Channels in the Stage Tree
- Working with Actors in the Stage Tree
- Animating Channels
- Channel Editor

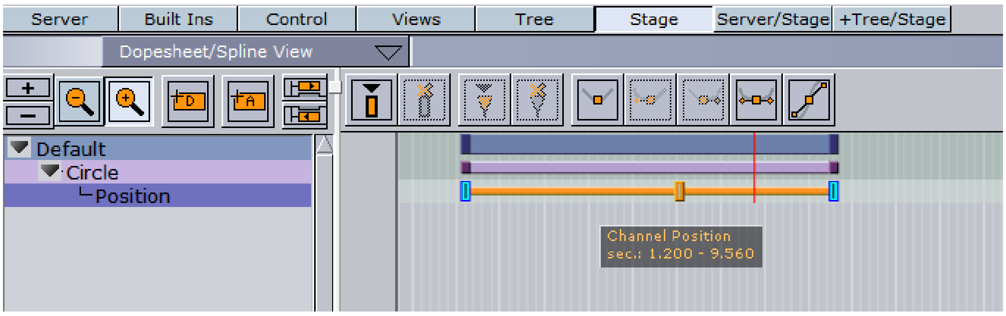
To move a channel in the timeline

- Drag the channel to a different time.



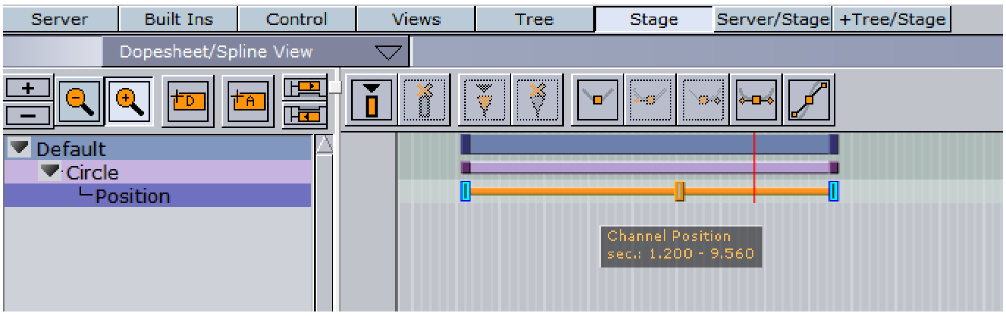
To speed a channel up

- At either end of the channel, click the knob and drag it to shorten the channel.



To slow a channel down

- At either end of the channel, click the knob and drag it to lengthen the channel.



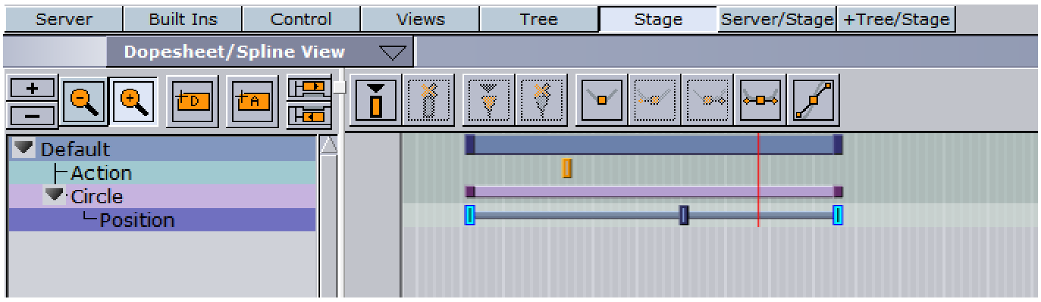
8.8 Working with Action Channels in the Stage Tree

Actions are controlling elements performing special tasks when the timeline passes them.

Regular channels are typically basic actions that can be created through the use of the Viz Artist GUI. For example, the rotation of a cube can be defined in the Transformation Editor of the cube container. For more special scenarios, such as audio animations, it may not be possible to create the animations based on the standard options. In such cases it is necessary to create actions with custom scripts that suit the situation.

To create an action, it is necessary to first create an action channel. In order to create the actual action itself, first create an action keyframe. Finally, define the action keyframe in the Action Channel Editor. The action will be executed when the director timeline equals the action keyframe.

Actions are displayed in the stage tree in cyan.



This section contains information on the following procedures:

- To create an action channel
- To move an action channel
- To copy an action channel
- To rename an action channel
- To delete an action channel

See Also

- Working with Action Channels in the Dopesheet Editor
- Action Channel Editor
- Working with Keyframes

To create an action channel

1. Select a director in the stage tree.
2. Click the **Add New Action** button in the Stage Tree Bar.
3. In the dialog box that opens, type a new descriptive name for the action channel, and then press <Enter> or click the **OK** button.

Note: Special characters, such as space, are not allowed in action channel names.

Alternatively:

1. Drag the **Add New Action** icon in the Stage Tree Bar to a director in the stage tree.
2. In the dialog box that opens, type a descriptive name for the action channel, and then press <Enter> or click the **OK** button.

To move an action channel

- Drag an action channel from a director in the stage tree to the target director.

To copy an action channel

- Press <Ctrl> while dragging an action channel from a director in the stage tree to the target director.

To rename an action channel

1. Right-click an action channel in the stage tree.
2. In the Stage Tree Menu that appears, select **Rename**.
3. In the text box that appears, type a new descriptive name, and press <Enter>.

Note: Special characters, such as space, are not allowed in action channel names.

To delete an action channel

1. Right-click an action channel in the stage tree.
2. In the Stage Tree Menu that appears, select **Delete [Action Channel]**.

Alternatively:

- Drag the action channel to the trash can.

8.9 Working with Action Channels in the Dopesheet Editor

Most procedures on action channels are done when Working with Action Channels in the Stage Tree. Procedures affecting the timeline are performed directly in the dopesheet editor. Working with action channels in the dopesheet editor enables the Action Channel Editor.

This section contains information on the following procedures:

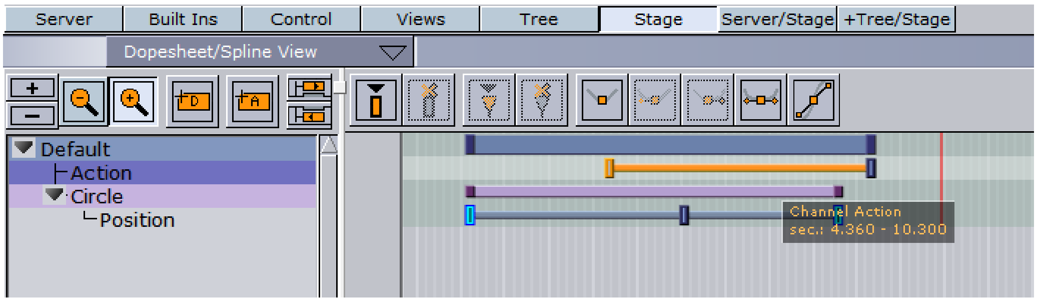
- To move an action channel in the timeline
- To speed an action channel up
- To slow an action channel down

See Also

- Working with Action Channels in the Stage Tree
- Action Channel Editor

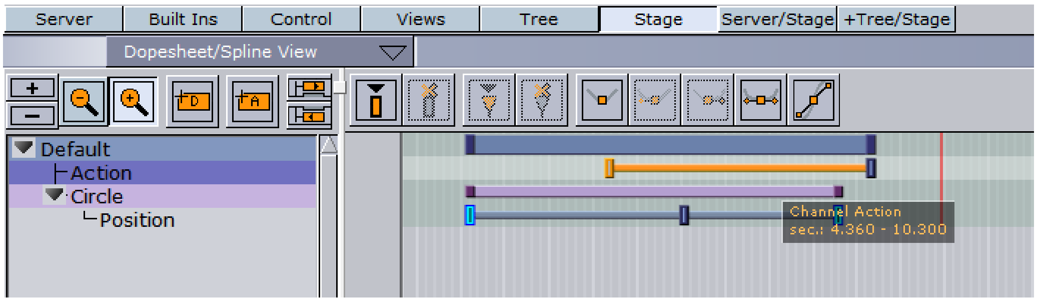
To move an action channel in the timeline

- Drag the action to a different time.



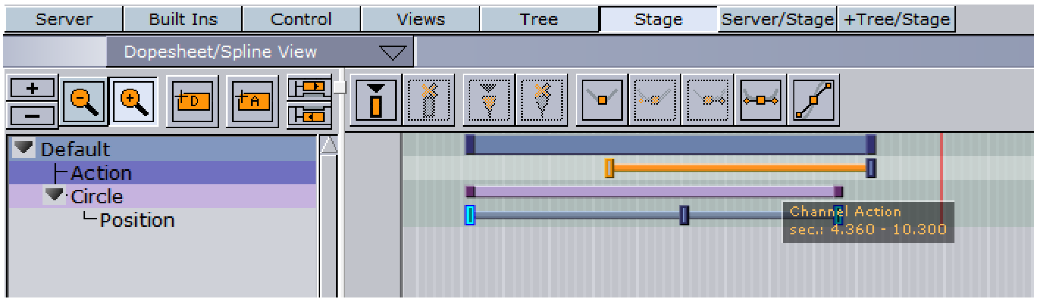
To speed an action channel up

- At either end of the action, click the knob and drag it to shorten the action.



To slow an action channel down

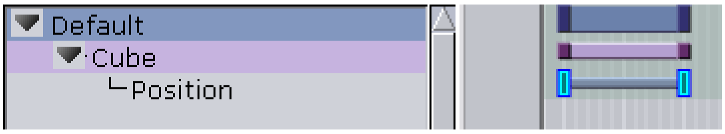
- At either end of the action, click the knob and drag it to lengthen the action.



8.10 Working with Keyframes

A keyframe stores information about a property at a given time. Keyframes can be added to channels and action channels. If two keyframes next to each other hold different values, the keyframes are connected with a bar in the Dopesheet Editor. When selecting a keyframe in the dopesheet editor or Spline Editor, the corresponding keyframe in the other editor will be selected too. Working with keyframes in the dopesheet enables the Keyframe Editor.

Figure 99: Example of cube with animated position; two keyframes and bar between



Example: The position of a cube can be at X=0 at field 0 and X=100 at field 50. In such a scenario, two keyframes will be created, one at field 0 and one at field 50. Since the position of the cube is different in the two keyframes, a bar is added between them, indicating that an animation takes place when the timeline moves from keyframe A to keyframe B.

This section contains information on the following procedures:

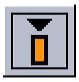
- To add a keyframe (Alternative 1)
- To add a keyframe (Alternative 2)
- To add a keyframe (Alternative 3)
- To add a keyframe (Alternative 4)
- To move one or more keyframes
- To move a keyframe to a specific time
- To copy a keyframe
- To rename a keyframe
- To delete a keyframe
- To reverse the order of keyframes

See Also

- Keyframe Editor
- Stage Editor Bar

To add a keyframe (Alternative 1)

1. Select the channel or action channel in the stage tree where the keyframe should be added.
2. Set the Timeline Marker to when the keyframe should be added.



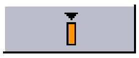
3. Click the **Add Keyframe** button in the Stage Tree Bar.

See Also

- Working with Channels in the Stage Tree
- Working with Channels in the Dopesheet Editor
- Working with Action Channels in the Stage Tree
- Working with Action Channels in the Dopesheet Editor

To add a keyframe (Alternative 2)

1. Select the channel or action channel in the stage tree where the keyframe should be added.
2. Set the Timeline Marker to when the keyframe should be added.



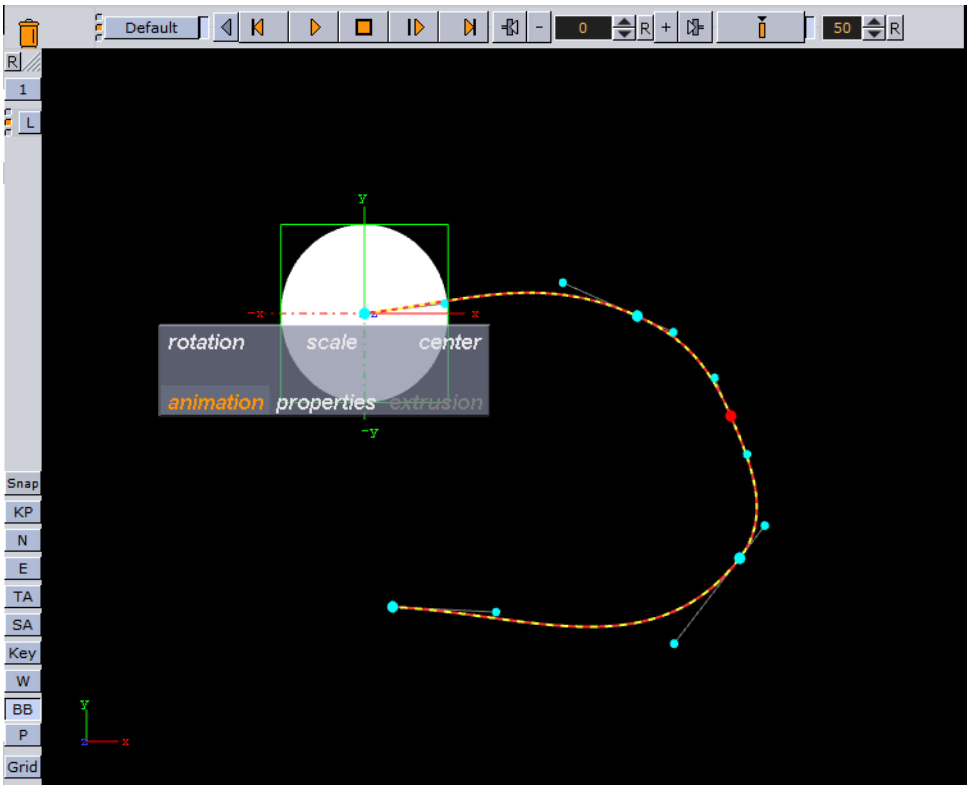
3. Do *one* of the following:
   - Click the **Set Key** button from the Timeline Editor.
   - Ensure the mouse pointer is over the render output and press <Enter>.

To add a keyframe (Alternative 3)

1. Select the channel or action channel in the stage tree where the keyframe should be added.
2. In the Spline Editor, make sure to hold the pointer above the spline at the desired timeline location (the spline should turn orange if doing this correctly).
3. Click the spline.

To add a keyframe (Alternative 4)

1. In either the scene tree or the scene editor, click the object whose animation path you want to view.
2. In the scene editor, right-click and select **Animation**.



3. Hold down the <Insert> key and click the path.
   A new locked keyframe is added.

To move one or more keyframes

1. Select one or more keyframes.

Tip: You can also hold down the <S> key and use the rubber band to select multiple keyframes.

2. Drag the keyframes along the timeline in the Dopesheet Editor.

Alternatively:

1. Select one or more keyframes.
2. Drag the keyframes along the timeline (horizontally) in the Spline Editor.

Note: Dragging a keyframe vertically in the Spline Editor will change the value of the keyframe, not the timeline value.

To move a keyframe to a specific time

1. Enable the Keyframe Editor.
2. Select a keyframe.
3. In the keyframe editor, enter the time (in fields) in the **Time** input box.

To copy a keyframe

1. Select one or more keyframes.

2. Press <C> while dragging the keyframes to the new position in the Dopesheet Editor.

To rename a keyframe

1. Enable the Keyframe Editor.
2. Select a keyframe.
3. In the keyframe editor, enter a descriptive keyframe name in the **Name** text box, and press <Enter>.

Note: Special characters, such as space, are not allowed in keyframe names.

To delete a keyframe

1. Select one or more keyframes.

2. Click the **Delete Keyframe** button in the Stage Editor Bar.

Alternatively:

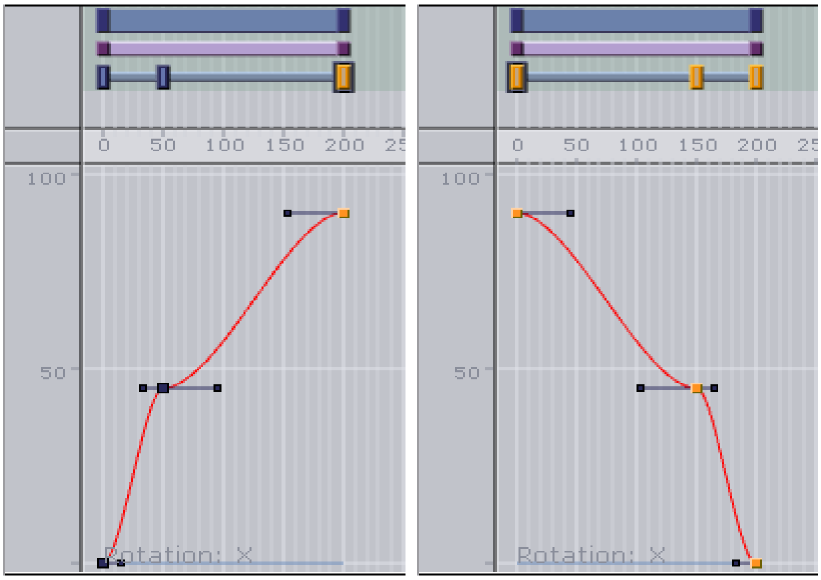
1. Select one or more keyframes.
2. Press <Delete>.

8.10.1 Reversing the Order of Selected Keyframes

Reversing the order of keyframes is akin to mirroring the keyframes.

Example: In Figure 100: Animation with normal and reversed keyframe order, the image to the left illustrates a rotation animation, with keyframes placed at fields 0, 50, and 200. The rotation value at field 0 is X=0, at field 50 X=45, and at field 200 X=90. After the keyframes have been rotated, the image to the right illustrates the animation with keyframes placed at fields 0, 150, and 200. The rotation value at field 0 is X=90, at field 50 X=45, and at field 200 X=0.

Figure 100: Animation with normal and reversed keyframe order



To reverse the order of keyframes

1. Select two or more keyframes in the Dopesheet Editor.
2. While holding the mouse pointer over the Stage, Press <Ctrl>+<R> .

8.11 Basic Animation Functions

This section contains information on the following procedures:

- To play all animations in a director from the start of the timeline
- To play all animations in a scene
- To play an animation in reverse
- To stop a director
- To stop an animation
- To stop all animations
- To continue an animation
- To jump to start of an animation
- To jump to end of an animation
- To delay animations in a director using offsets
- To scale animations in a director
- To scale a region within a director

To play all animations in a director from the start of the timeline

1. Select the desired director.
2. Click the **Start Animation** button in the Timeline Editor.

Alternatively:

1. Enable the Director Editor.
2. Select the desired director.

3. In the director editor, click the **Start Animation** button.

Alternatively:

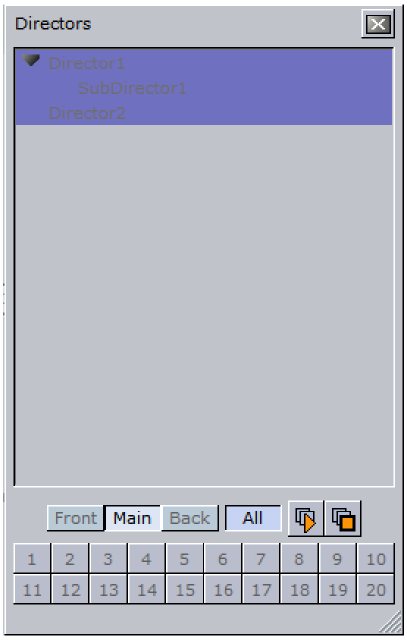
1. Ensure the mouse pointer is in the dopesheet.
2. Press <Enter>

Alternatively:

1. Select the desired director.
2. Ensure the mouse pointer is in the dopesheet and press the <Space Bar>.

To play all animations in a scene

1. Enable the Director Control Panel.
2. Click **All**.



3. Do *one* of the following:
   - Ensure the mouse pointer is in the dopesheet and press <Enter>.
   - Ensure the mouse pointer is in the dopesheet and press the <Space Bar>.
   - Click the **Start** button in the Timeline Editor.
   - Click the **Start** button in the Director Control Panel. (Clicking here will also start the animations in the other layers if they are available).

To play an animation in reverse

1. In the Timeline Editor, click the Reverse Play button.

◁

2. Do *one* of the following:
   - Ensure the mouse pointer is in the dopesheet and press <Enter>.
   - Click the **Start** button in the Timeline Editor.
   - Click the **Start** button in the Director Control Panel. (Clicking here will also start the animations in the other layers if they are available).

Alternatively:

1. Select the desired director.
2. Ensure that the mouse pointer is in the dopesheet.
3. Press <Shift> + <Enter>.

To stop a director

1. Select one or more directors.
2. In the Timeline Editor, *right-click* the **Stop** button.

Alternatively:

1. Enable the Director Editor.
2. In the director editor, click the **Stop** button.

Alternatively:

1. Select the desired director.
2. Ensure the mouse pointer is in the dopesheet and press the <Space Bar>.

To stop an animation

- In the Timeline Editor, click the **Stop** button.

Note: It is also possible to add one or more stop points to a director, so that each time the timeline passes a certain frame, the animations stop automatically. For details on how to create such a stop point, see Working with Stops, Tags and Pauses.

To stop all animations

1. Enable the Director Control Panel.
2. Click the **Stop** button.
   This stops all animations in all layers.

To continue an animation

1. Select the desired director.
2. Click the **Continue Animation** button in the Timeline Editor.

Alternatively:

1. Select the desired director.
2. Ensure the mouse pointer is in the dopesheet and press the <Space Bar>.

Alternatively:

1. Enable the Director Editor.
2. Select the desired director.
3. In the director editor, click the **Continue Animation** button.

To jump to start of an animation

1. Select the desired director.
2. Click the **Go to Start** button in the Timeline Editor.

Alternatively:

1. Select the desired director.
2. Press <Ctrl> + <Home>.

To jump to end of an animation

1. Select the desired director.
2. Click the **Go to End** button in the Timeline Editor.

Alternatively:

1. Select the desired director.
2. Press <Ctrl> + <End>.

To delay animations in a director using offsets

1. Enable the Director Editor.
2. Select the desired director.
3. In the director editor, set the **Offset** value (in fields).

Alternatively, it is possible to move the director (horizontally along the timeline) including all its animations directly in the Timeline Editor, although this will not actually add the offset value.

To scale animations in a director

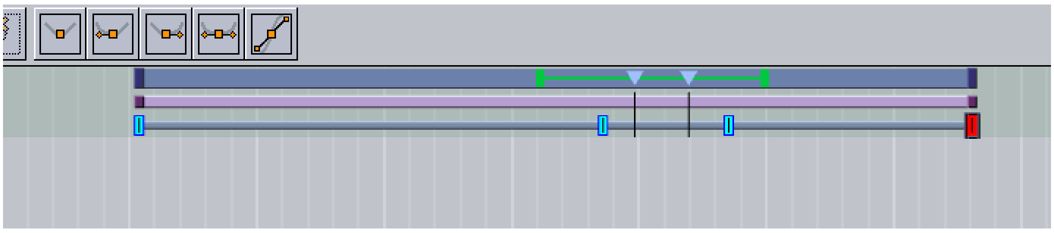
- Drag the knob at either the beginning or end of a director in the Dopesheet Editor.

Note: If dragging the beginning marker, it is not possible to drag it past the offset value.

Scaling a director is the easiest way to change the total running time of the animations in the director. Scaling is done proportionally and includes all subdirectors.

To scale a region within a director

1. Follow the procedure To mark a region within a director.
2. Drag the green knob at either the beginning or end of the marked region.



Scaling is done proportionally and includes all subdirectors.

8.12 Creating a Basic Animation

To demonstrate how to create basic animations, the following procedures make up a simple use case whereby a cube is created and then moved to the right.

This use case contains the following procedures:

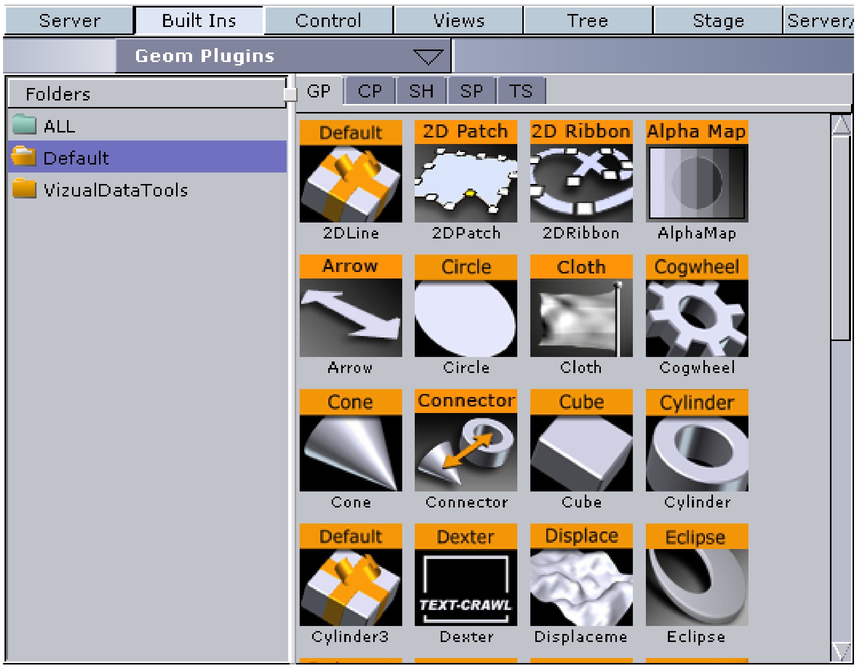
- To create the new scene
- To create a container in the scene tree
- To create the first keyframe
- To create the second keyframe (Alternative 1)
- To create the second keyframe (Alternative 2)

To create the new scene

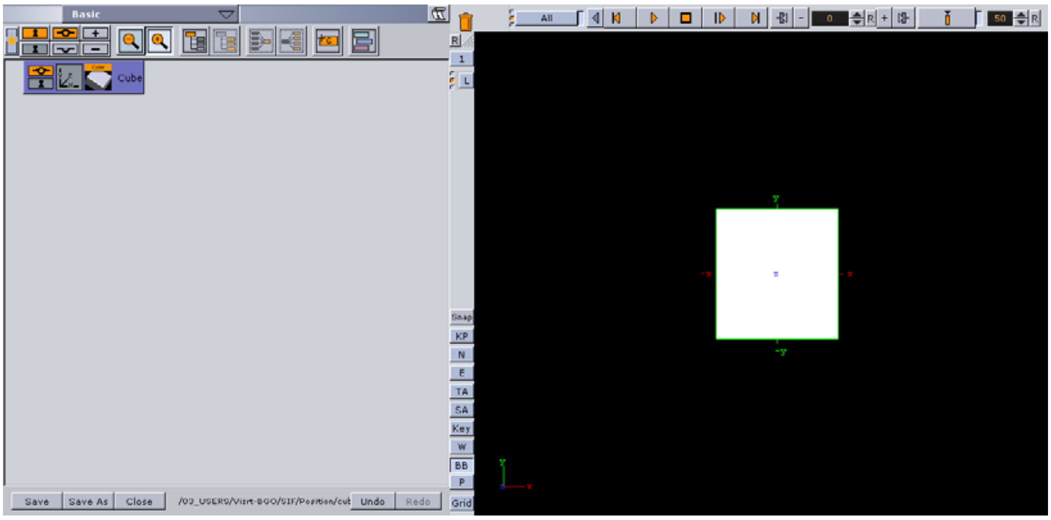
1. Follow the procedure for To create a Scene.
2. Open the scene.

To create a container in the scene tree

1. From the main menu, click **Built Ins**.
2. Click the **GP** (Geometry Plugin) tab to show all available geometry plug-ins.
3. In the Folders pane, click the **Default** folder.
4. Drag the **Cube** icon from the plug-in list into the scene tree.



The scene tree and render output should then look like this:



To create the first keyframe

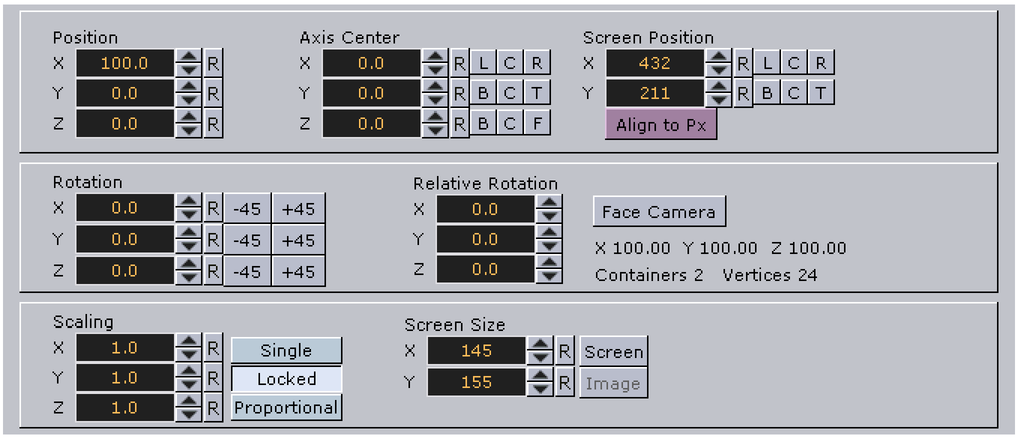
1. Set the timeline to field *0*, as in the procedure To move the timeline marker (Alternative 1).
2. Perform the procedure To add a keyframe (Alternative 2).

To create the second keyframe (Alternative 1)

1. In the render output, drag the cube rightward, so that it shifts to a different position.
2. Create the other keyframe, by performing the procedure To add a keyframe (Alternative 2).

To create the second keyframe (Alternative 2)

1. Click the **Transformation** icon on the **Cube** container in the scene tree.



Position
X 100.0 R
Y 0.0 R
Z 0.0 R

Axis Center
X 0.0 R L C R
Y 0.0 R B C T
Z 0.0 R B C F

Screen Position
X 432 R L C R
Y 211 R B C T
Align to Px

Rotation
X 0.0 R -45 +45
Y 0.0 R -45 +45
Z 0.0 R -45 +45

Relative Rotation
X 0.0
Y 0.0
Z 0.0

Face Camera

X 100.00 Y 100.00 Z 100.00
Containers 2 Vertices 24

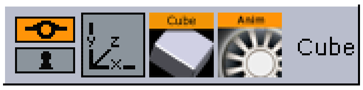
Scaling
X 1.0 R
Y 1.0 R
Z 1.0 R
Single
Locked
Proportional

Screen Size
X 145 R Screen
Y 155 R Image

2. In the Transformation Editor that appears in the upper right part of the screen, set the **Position** to X = *100*.
3. Create the other keyframe, by performing the procedure To add a keyframe (Alternative 2).

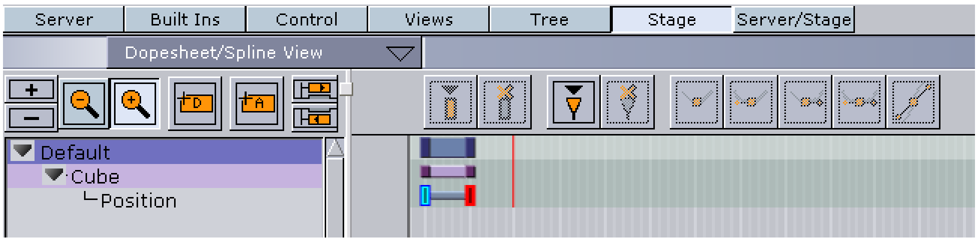
We have now created a basic position animation. On the cube container in the scene tree, an Anim icon has been automatically added, indicating that one or more properties on this container has been animated.

Figure 101: Cube container with Animation icon



It is also possible to see a representation of the animation in the Stage.

Figure 102: Stage showing the animation of the Cube container’s position



See Also

- To save a scene

8.13 Creating an Advanced Animation

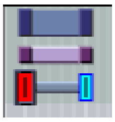
Once you have mastered Creating a Basic Animation, where the position of a cube is moved, you can move on to more advanced tasks. This section provides some additional procedures, so that a cube can continue its journey rightward, rotate while moving, and gain a nicer appearance by adding color to it.

This section contains the following procedures:

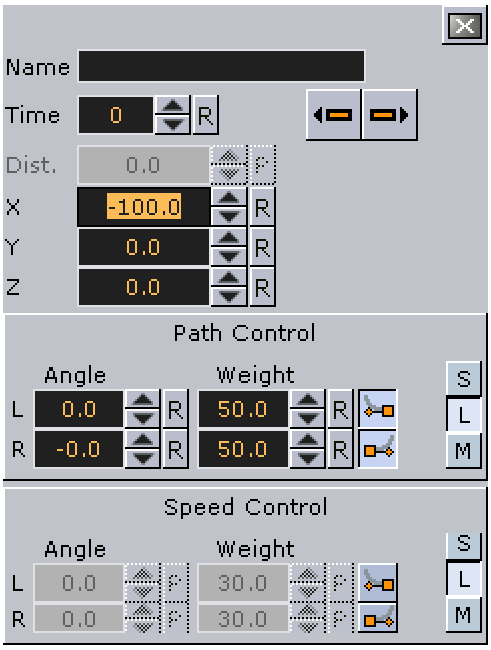
- To edit the first keyframe
- To edit the second keyframe
- To create an additional keyframe
- To add a stop point
- To create keyframes for rotation
- To modify the rotation settings
- To run the animations independently
- To add and animate a material

To edit the first keyframe

1. Enable the Keyframe Editor.



2. Select the keyframe at the far left in the Dopesheet Editor.



Name
Time 0
Dist. 0.0
X -100.0
Y 0.0
Z 0.0

Path Control

	Angle	Weight
L	0.0	50.0
R	-0.0	50.0

Speed Control

	Angle	Weight
L	0.0	30.0
R	0.0	30.0

3. In the Keyframe Editor, set the **Position** to X = -100.

To edit the second keyframe

1. Select the keyframe at the right in the Dopesheet Editor.
2. In the Keyframe Editor, set the **Position** to X = 0.

Tip: Set the position either by typing in the input box, using the up/down buttons, or clicking the R (reset/default setting) button.

When playing the animation now, the cube will still slide rightward, but from a point at the left part of the render output to the center.

3. Click the **Play** button in the Timeline Editor.

To create an additional keyframe

1. In the Timeline Editor, set the timeline field to 100.
2. In the render output, drag the cube rightward, so that it shifts to a different position.
3. Perform the procedure To add a keyframe (Alternative 2).
4. Click the **Play** button in the Timeline Editor.

When playing the animation now, the cube will slide rightward from a point at the left part of the render output, through the center, and all the way to the right part.

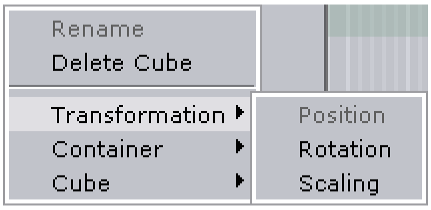
To add a stop point

1. Select the desired director.
2. In the Timeline Editor, set the timeline field to 50.
3. Create a stop by performing the procedure To create a stop, tag or pause.
4. Click the **Play** button in the Timeline Editor.

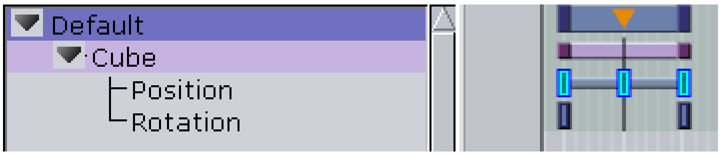
When playing the animation now, the cube will slide rightward until it reaches the center of the render output, and then come to a halt. To continue playing the rest of the animated journey, see To continue an animation.

To create keyframes for rotation

1. Right-click the **Cube** actor in the stage tree.



2. In the Stage Tree Menu that appears, click **Transformation** -> **Rotation.**
   A rotation channel and two keyframes are added.



To modify the rotation settings

1. Select the keyframe at the right in the Dopesheet Editor.

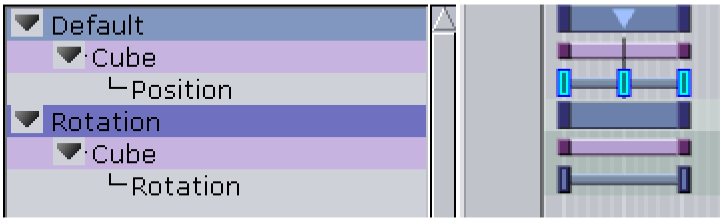
2. In the Keyframe Editor, set the **Rotation** to X = 360.
3. Click the **Play** button in the Timeline Editor.

   When playing the animation now, the cube will slide rightward until it reaches the center of the render output, while rotating all the while before coming to a halt.
4. To continue playing the rest of the animated journey, see To continue an animation.

To run the animations independently

1. Create a director at the root level of the stage tree. See the procedure To create a director.
2. Rename the new director to **Rotation**. See the procedure To rename a director.
3. Drag the rotation channel to the new director. See the procedure To move a channel.

   The stage tree and should now look like this:



When playing the animations now, the output depends on which director is selected. If the **Default** director is selected, only the position animation will be played. If the **Rotation** director is selected, only the rotation animation will be played. Selecting both/all directors will play both the position and rotation animations.

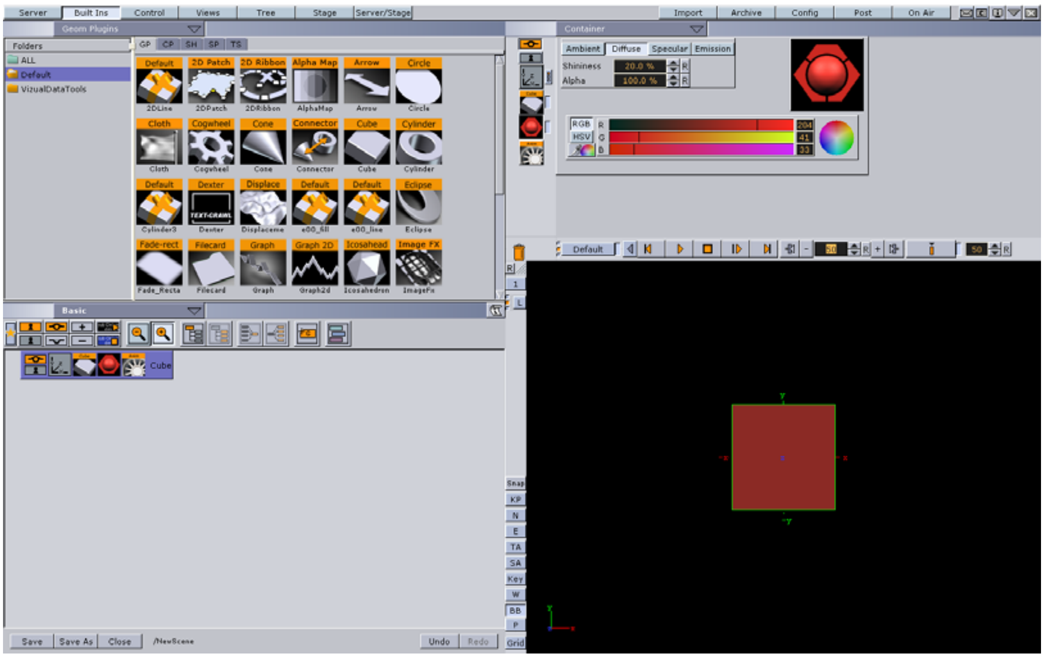
See Also

- To play all animations in a director from the start of the timeline
- To play all animations in a scene

To add and animate a material

1. Perform the procedure To create a material in the Server area and define its color.
2. Drag the material to the container in the scene tree.
3. In the scene tree, click the material in the container.

   The material editor is displayed.



4. In the Timeline Editor, set the time when you want the keyframe to be created.
5. In the Timeline Editor, click the **Set Key** button.
6. In the Material editor, change the material’s color.
7. To create the next keyframe, in the Timeline Editor, click the **Set Key** button.
8. Click the **Play** button in the Timeline Editor.

The cube now has a color that changes as part of the animation.

8.14 Further Advanced Animation Topics

This section will guide you through creating a scene and using the more advanced animation features in Viz Artist. The following topics are covered:

- Animating Channels
- Path Control
- Spline Control
- Animating Audio
- Animating Cameras
- Animating Light
- Merging Animations

As your basis, you can take the scene created in Creating a Basic Animation and improved on in Creating an Advanced Animation.

8.14.1 Animating Channels

This section contains information on the following topics:

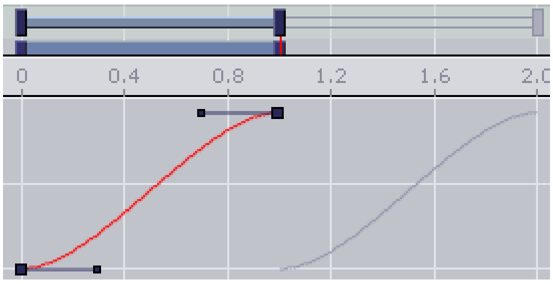
- Repeating Animation Channels (Loop)
- Repeating Animation Channels in Reverse (Swing)

- Combining Swing and Loop

Repeating Animation Channels (Loop)

The loop operation makes it possible to repeat the animation of a channel either at the beginning or end of the channel timeline.

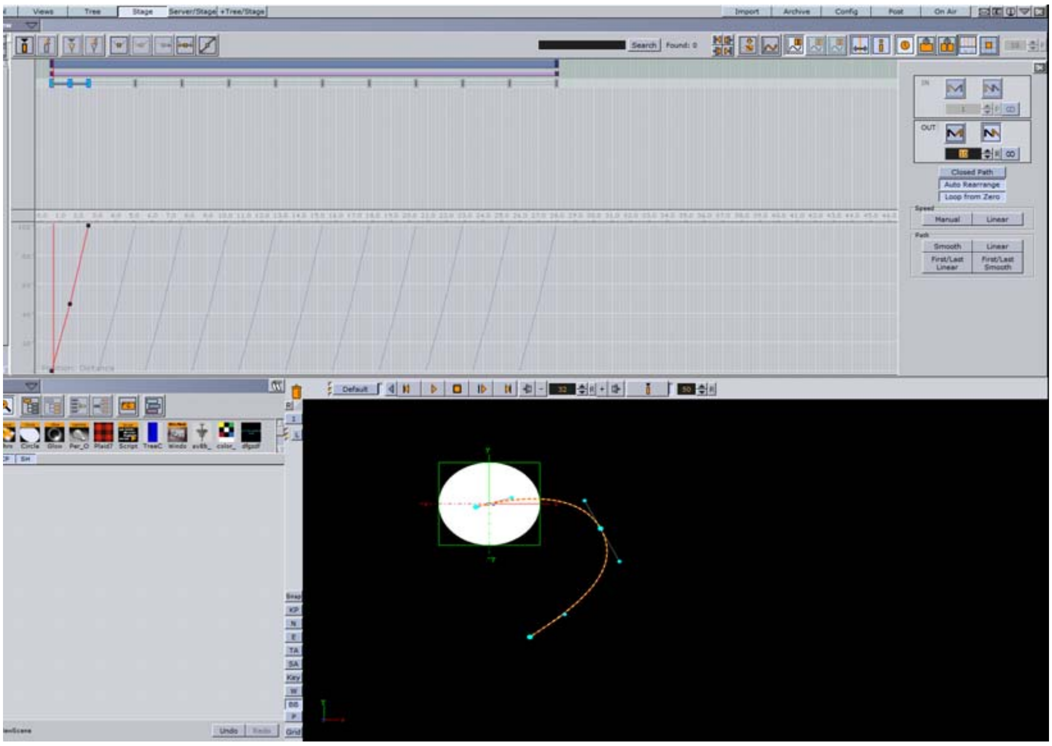
Figure 103: Channel with loop operation enabled



To repeat the animation of a channel

1. Select a channel.
2. In the Channel Editor, enable the **Loop In/Out** buttons.

Note: Loop at the beginning is only available if the belonging actor has a defined offset value.



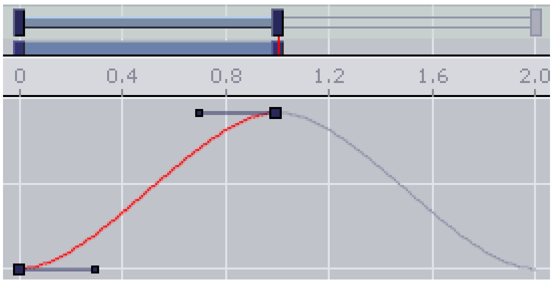
3. Define how many times the loop should run before coming to a halt. Either enter a number in the field or click the **Infinite** button.

When enabling loop, the looped animation appears as gray in the in the Dopesheet Editor

Repeating Animation Channels in Reverse (Swing)

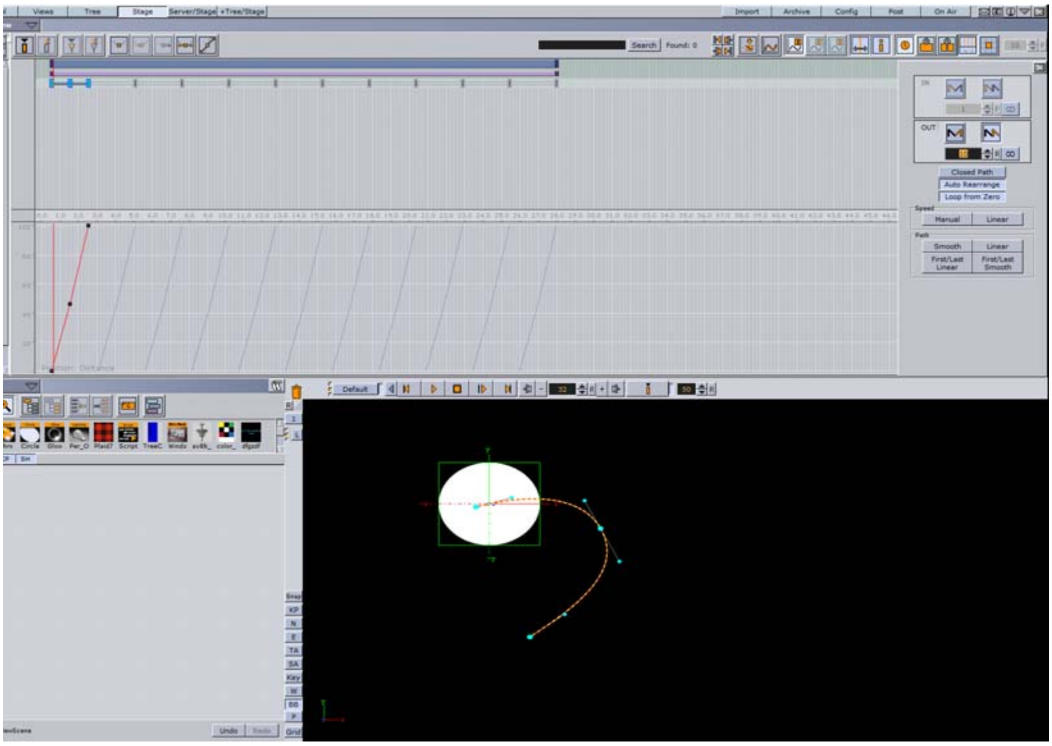
The swing operation makes it possible to repeat the animation of a channel in reverse play either at the beginning or end of the channel timeline.

Figure 104: Channel with swing operation enabled



To create a swinging animation

1. Select a channel
2. In the Channel Editor, enable the **Swing In/Out** buttons.

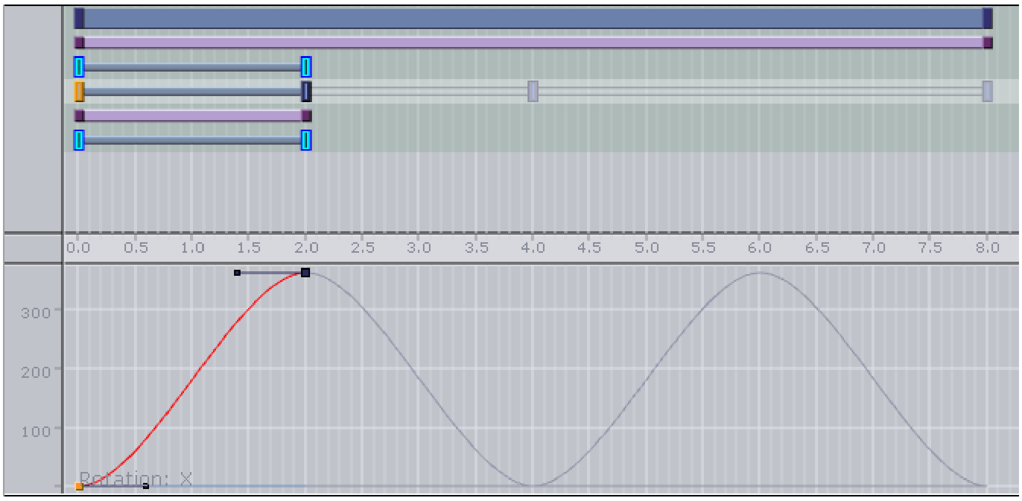


When enabling swing, the animation appears as gray in the in the Dopesheet Editor.

Combining Swing and Loop

It is possible to combine the swing and loop operations on a channel. The animation will then first be added in reverse play (swing) and then started once again from the beginning to the end (loop). This will save you a great amount of time if in need of a recurring animation, for example a bouncing ball.

Figure 105: Channel with swing and loop operations enabled



To combine swing and loop

1. Follow the procedure To repeat the animation of a channel.
2. Follow the procedure To create a swinging animation.

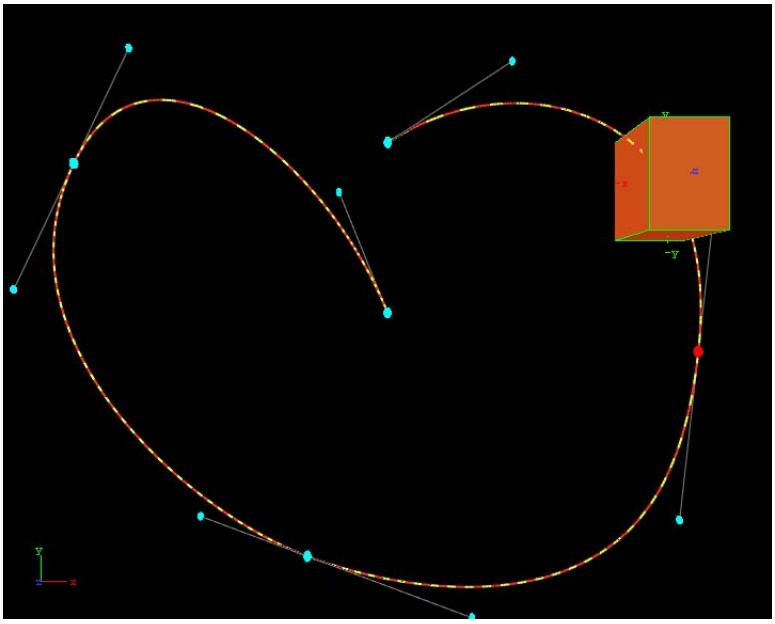
Animations with swings and loops appear as gray in the in the Dopesheet Editor.

8.14.2 Path Control

The position animation follows a path you define by the keyframes. A spline will be interpolated from the keyframes. You can view and edit this spline in the render output.

To edit the spline in the render output

1. Enable the bounding box by clicking the **BB** button to the left of the render output.
   Alternatively, ensure the mouse pointer is in the render output, and press `<B>`.
2. In the scene tree, select the animated container.
3. Right-click in the render output and select **Animation**.
   The path will be shown as a spline with the keyframes marked on it.



4. Drag the keyframes around to modify the animation trajectory.

The red and yellow markings on the spline provide information about the speed the object is moving along its trajectory. One segment represents a field, the longer the segments are, the faster the object will travel.

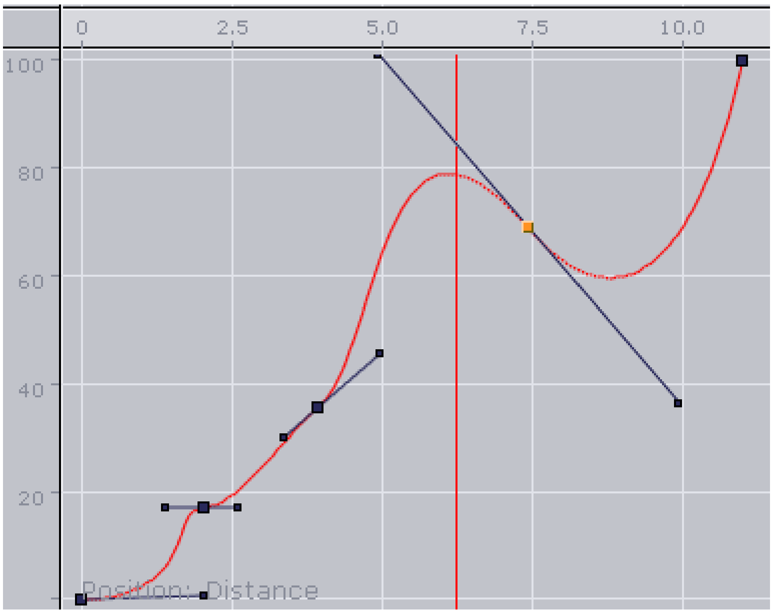
As every keyframe has handles (seen in blue) you can use them to modify the keyframe’s trajectory. To do so select a handle and drag it around.

By default the handles are set to be locked and proportional. If you want to switch to a mirrored mode, press <Shift> while you drag the handle around. Pressing `<Ctrl>` will allow you to edit the single handle only.

An other option to set the handles is the object editor. Select the keyframe you want to change the handles in and set either **Single**, **Locked** or **Mirrored** and change the values in **Weight L** and **R**.

8.14.3 Spline Control

In the spline editor the distance spline is shown, which sets the current position of the container along the path. The y-axis in the spline editor is scaled to use the percentage of the trajectory, while the x-axis still uses the time. In the spline editor you can only change the position of a keyframe along the timeline, but not its percentage of the total trajectory.



By default the handles at every keyframe on the spline are set to be linear.

To edit the spline

- Drag the handles around to change their positions.
  By default the handles are set to move proportionally. So as you move the handle of a keyframe the other one will move to.
- Press `<Ctrl>` while you drag a handle to modify only the selected one.
  This “single handle mode” will be enabled for the keyframe as long as you do not reset it by a click the keyframe icons in the Stage Editor Bar.

Example: If you, for example, modify the spline in such a way that the value decreases, the container will move backwards.

To move keyframes in the spline

- Select the desired keyframe and drag it.
  Multiple keyframes can be selected to move at once.

To insert keyframes from the spline

1. Moving the cursor on the spline exactly where you would like to add the keyframe.
2. Click on the desired location to set the keyframe.

To delete a keyframe from the spline

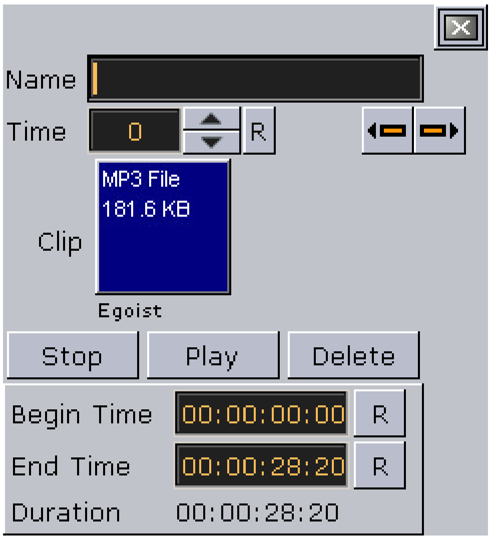
1. Select the desired keyframe
2. Press `<Del>` or click Delete Keyframe button in the Stage Editor Bar.
   Multiple keyframes can be selected to be deleted at once.

8.14.4 Animating Audio

An audio animation plays an audio clip in your animation.

To enable audio in an animation

1. Apply the audio container function to a container in your scene.
2. Add an actor for this container to the stage tree, if it is not present yet. The most easy way to do is, by dragging the container from the scene tree onto the stage tree.
3. Open the shortcut menu and select **Audio** followed and then **Clip**.
4. Select a keyframe in this channel to open the object editor.
5. Switch to the **Server**/**Stage** view and drag an audio clip from the server view onto the **Clip** drop zone in the object editor.
6. In the object editor for the keyframes of the AudioClip channel, configure the following parameters, as necessary.



Name

Time 0 R

Clip

MP3 File
181.6 KB

Egoist

Begin Time 00:00:00:00 R

End Time 00:00:28:20 R

Duration 00:00:28:20

- **Name** Sets a name for the keyframe. This name is for use only for external commands or scripts.
- **Time** Shows the time value in fields the keyframe is placed. Editing this value will change to position of the keyframe.
- **Clip** Drag an audio clip from the server view onto this drop zone.
- **Stop** Stops playback.
- **Play** Plays the clip.
- **Delete** removes the clip.
- **Begin Time** Sets the begin time within the clip. Everything before this time will not be played in the animation. Another way to achieve this is to drag the keyframe at the beginning of the clip.
- **End Time** Sets the end time within the clip. Everything after this time will not be played in the animation. Another way to achieve this is to drag the keyframe at the end of the clip.
- **Duration** Shows the duration to be played.

8.14.5 Animating Cameras

You can animate cameras by adding an actor for a camera to the stage tree is by setting keyframes in the render output.

To animate a camera

1. Set the Timeline Marker to when the keyframe should be added to start with the animation of the camera.
2. Click the **Add Keyframe** button in the Stage Tree Bar
3. Change one of the selected cameras' properties and set the next keyframe. This will add an actor in the stage tree for the camera whose properties you have changed.

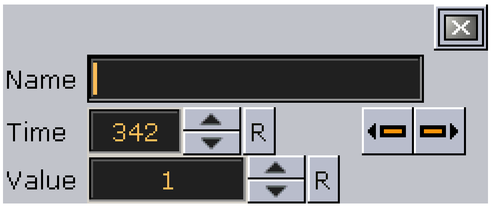
Another way to animate a camera is to enable position and/or direction tracking in the camera editor.

See Also

- Camera Editor
- Working with Actors in the Dopesheet Editor
- Working with Keyframes

To animate the current Camera

1. Set a keyframe in the cu rent Camera channel.
2. In the object editor edit the following parameters for the keyframe as necessary.



Name

Time 342 R

Value 1 R

- **Name** Set a name for the keyframe. This name is for use only for external commands or scripts.
- **Time** Shows the time value in fields the keyframe is placed. Editing this value will change to position of the keyframe. If you click arrow buttons in the Stage Editor Bar the previous or next keyframe will be selected.
- **Value** Sets the camera you want to switch to.

The animation of the current Camera allows you to switch the camera through which to view at a given time.

To create a zoom animation

If you click Spline button Stage Editor Bar or press *Shift* + *Q*, the whole spline which is currently displayed will be zoomed to.

If you have selected an actor or a director, the zoom will be adjusted to fit this entry.

When zooming the whole spline, both the time-axis for animation editor and the spline editor will be zoomed, as well as the value-axis in the spline editor.

Another way to zoom the time-axis is by pressing *X* to zoom in and *Y* or *Z* to zoom out. As an alternative you can use the mouse wheel.

If you press *Shift* + *Y* you will set the zoom along the time-axis to zoom the whole spline while *Shift* + *X* will set the zoom along the time-axis to show the whole animation.

In the spline editor you can zoom the value axis too. This is done by pressing *A* to zoom in or *S* to zoom out. If you press *Shift* + *A* you will zoom the whole spline along the value axis.

You can zoom also if you press one of the “zoom keys” and move the cursor around with the middle button pressed.

If you have changed the zoom level, you can switch back to standard zoom at any time by a click designated button in the Stage Editor Bar or if you press *Shift* + *W*.

As you maybe zoomed the view in such away you are no longer able to edit your spline comfortably, you can pan by moving around the cursor while the middle button is pressed.

Zooming along the time axis will zoom both the animation editor and the spline editor.

8.14.6 Animating Light

The only way to add an actor for a light to the stage tree is by setting keyframes in the render output. To do so set the timeline value to the time you want to start with the animation of the light and set the first keyframe. Now change one of the selected lights properties and set the next keyframe. This will add an actor for the light which property you have changed to the stage tree.

An easier way the animate a light is to enable position and/or direction tracking in the light editor. To learn more about the light editor, see Working with Light.

8.14.7 Merging Animations

In Viz Artist, it is possible to merge container groups into one single compound object. If an animation has been defined for at least one of the containers, on merging a new director will be created, given the name of the merged object in the scene tree. There will be a subdirector by the name of the original director the animations were created in. The animations created for the single containers will be kept, but cannot be changed anymore unless you split the object into the containers it has been created from.

The animations will be kept also if you save the newly created object in the database.

Please be aware that you cannot add subdirectors to the directors created automatically by merging.

9 Cameras

You can control the “camera” displaying your Viz Artist scene much the same way that you would control a real camera. Cameras are used by assigning two tracking objects. The first is the camera itself, and the second is the object being tracked.

This section contains information on the following topics:

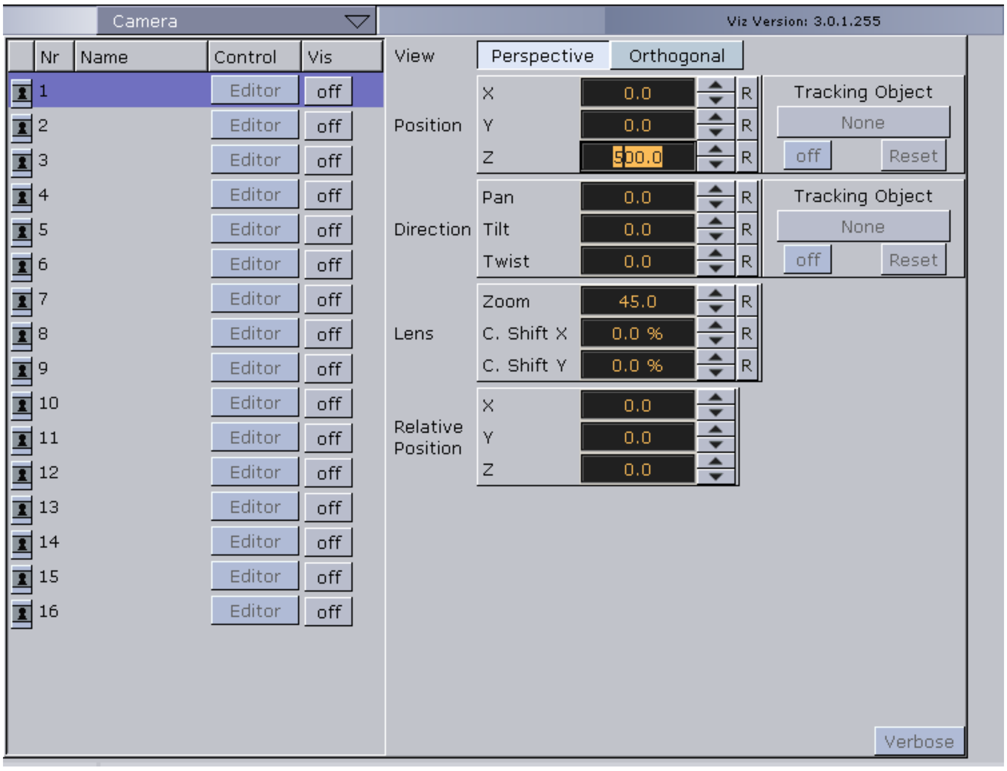
9.1 Camera Editor

Cameras are configured via the Camera Editor. The Camera editor makes it possible to modify the properties for the 16 editable cameras that can be used in Viz Artist.

To open the camera editor

- From the Scene Setup menu, select **Camera.**

To open the Camera editor, select Camera from the Scene Setup menu.



Nr	Name	Control	Vis
1		Editor	off
2		Editor	off
3		Editor	off
4		Editor	off
5		Editor	off
6		Editor	off
7		Editor	off
8		Editor	off
9		Editor	off
10		Editor	off
11		Editor	off
12		Editor	off
13		Editor	off
14		Editor	off
15		Editor	off
16		Editor	off

View: Perspective | Orthogonal

Position	X	0.0
	Y	0.0
	Z	500.0
Direction	Pan	0.0
	Tilt	0.0
	Twist	0.0
Lens	Zoom	45.0
	C. Shift X	0.0 %
	C. Shift Y	0.0 %
Relative Position	X	0.0
	Y	0.0
	Z	0.0

The camera editor opens up in the top right portion of the screen. The left panel of the camera editor shows a list of all 16 editable cameras, and the right panel shows the parameters for the currently selected camera.

Tip: The selected camera will automatically be set as viewing **Perspective** in the Output area (see Configurable Parameters).

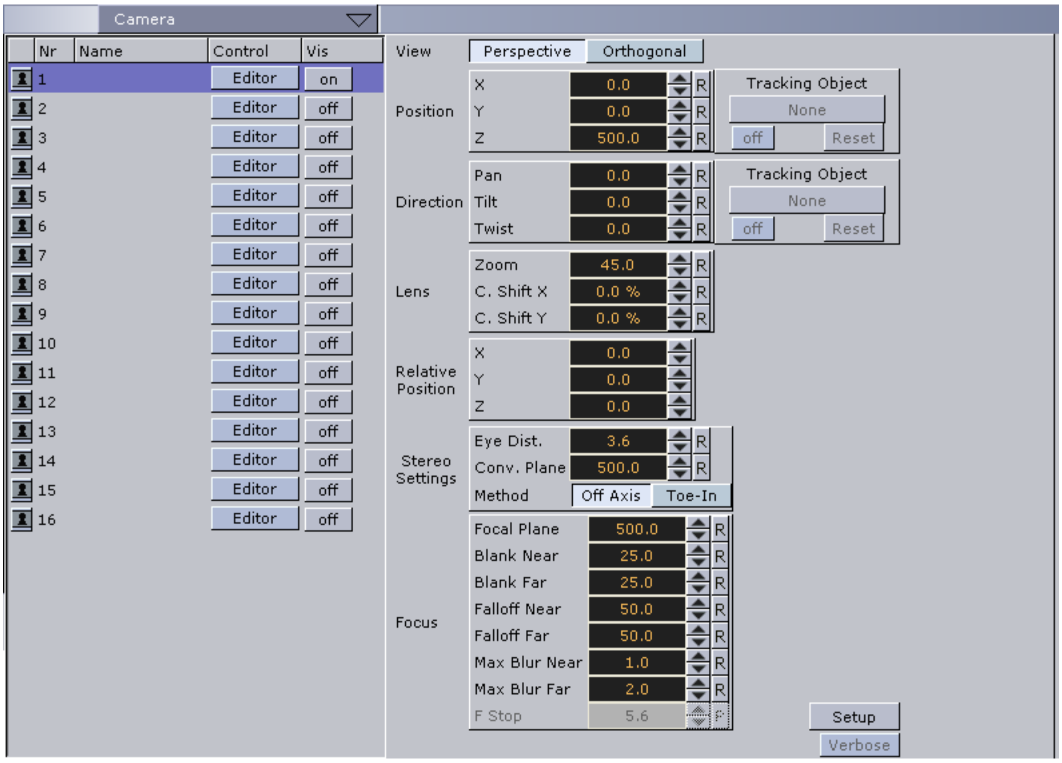
9.2 Configurable Parameters

All the cameras' configurable parameters can be accessed by clicking **Server** and then selecting **Camera** in the pull-down menu.

This section has information on the following topics:

- Left Pane Parameters
- Right Pane Parameters

Figure 106: Camera parameters



Camera

Nr	Name	Control	Vis
1		Editor	on
2		Editor	off
3		Editor	off
4		Editor	off
5		Editor	off
6		Editor	off
7		Editor	off
8		Editor	off
9		Editor	off
10		Editor	off
11		Editor	off
12		Editor	off
13		Editor	off
14		Editor	off
15		Editor	off
16		Editor	off

View: Perspective | Orthogonal

Position: X 0.0, Y 0.0, Z 500.0 — Tracking Object: None, off, Reset

Direction: Pan 0.0, Tilt 0.0, Twist 0.0 — Tracking Object: None, off, Reset

Lens: Zoom 45.0, C. Shift X 0.0 %, C. Shift Y 0.0 %

Relative Position: X 0.0, Y 0.0, Z 0.0

Stereo Settings: Eye Dist. 3.6, Conv. Plane 500.0, Method Off Axis | Toe-In

Focus: Focal Plane 500.0, Blank Near 25.0, Blank Far 25.0, Falloff Near 50.0, Falloff Far 50.0, Max Blur Near 1.0, Max Blur Far 2.0, F Stop 5.6

9.2.1 Left Pane Parameters

- **Lock/Unlock** Clicking the Keyhole icon left to the camera number will lock/ unlock the camera. The properties of a locked camera can not be modified in the Output area.
- **Nr** Shows the camera number, from 1 to 16.
- **Name** Shows the camera name. Right-click or press <F2> to assign a new camera name.
- **Control** Can toggle between **Editor** and **Remote**.
  - **Remote** allows you to receive tracking data from a real camera.
  - **Editor** allows you to configure camera settings.
- **Vis** Shows the status of the camera:
  - **Off** The camera will not be shown in the six predefined orthogonal views. Displaying a camera in a predefined orthogonal view can help to change the properties of a camera, by dragging the parameters in those views.
  - **On** Shows the camera.
  - **Inf** The camera will be displayed with infinite lines. This is a good aid for visualizing the scope of a camera that is located far away.

9.2.2 Right Pane Parameters

The following parameters can be configured from the camera editor’s right pane.

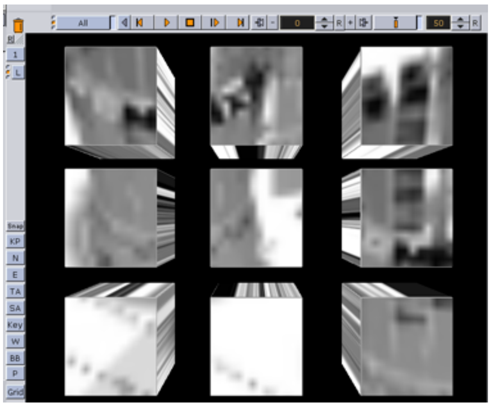
This section has information on the following topics:

- View
- Position

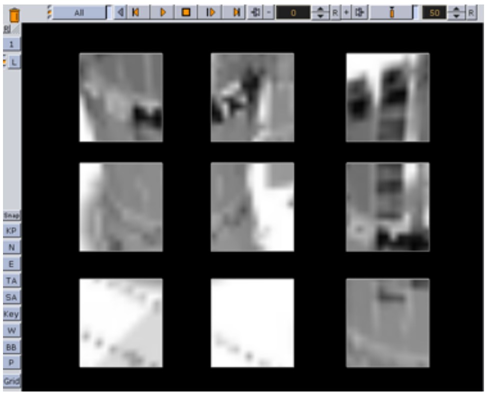
- Direction
- Lens
- Relative Position
- Verbose
- Stereo Settings
- Focus

View

There are three views:



- **Perspective** The basic view of a virtual camera is a normal depth perspective, showing three dimensions. All other parameters can be edited in when the view is set to Perspective.



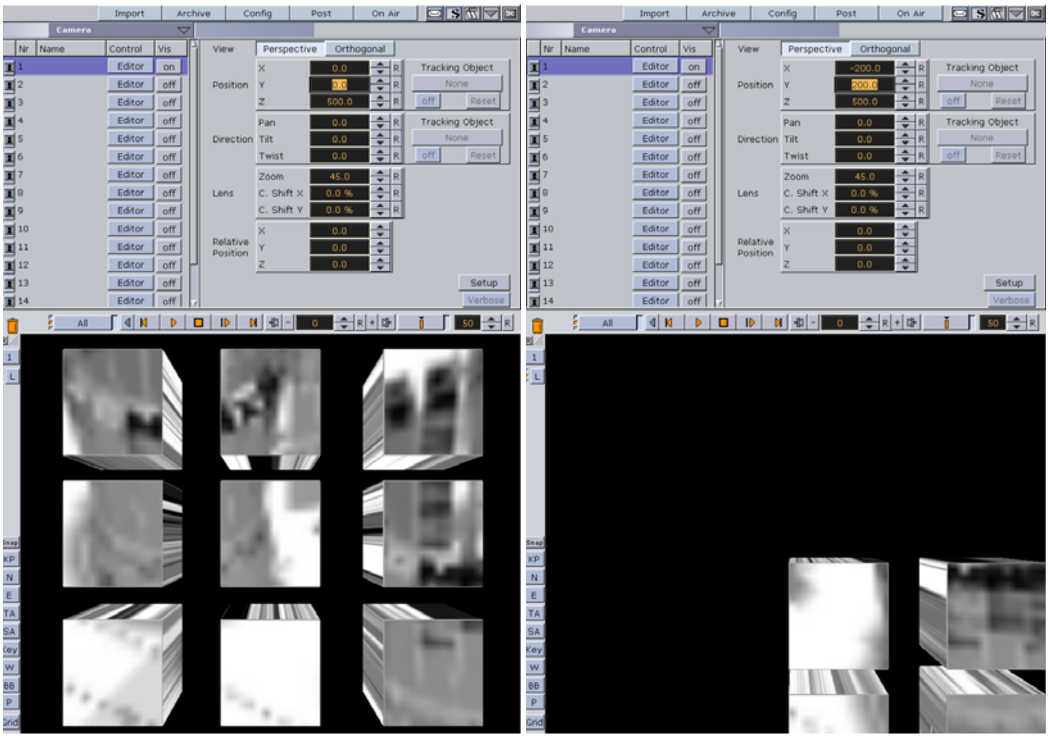
- **Orthogonal** Shows a camera view without perspective, so that objects will not change their size when being moved towards or away from the camera. All parameters, except Position X and Y, will be disabled when the camera is set to orthogonal. Other parameters cannot be edited when the view is set to Orthogonal. Changing Camera Parameters in Orthogonal Views.

- **Window** Shows the camera view like looking through a real window. The graphics, through a virtual camera, are adapted according to the real camera movement to produce this behavior. Specify the physical position and rotation of the display unit in Window size, Window centre and Window rotation.

Position

Sets the position of the camera along the X, Y, and Z axis.

Figure 107: Default position settings (L) and modified settings (R)



View: Perspective | Orthogonal

Position X 0.0 Y 0.0 Z 500.0

Direction Pan 0.0 Tilt 0.0 Twist 0.0

Lens Zoom 45.0 C. Shift X 0.0 % C. Shift Y 0.0 %

Relative Position X 0.0 Y 0.0 Z 0.0

Tracking Object None off Reset

View: Perspective | Orthogonal

Position X -200.0 Y 200.0 Z 500.0

Direction Pan 0.0 Tilt 0.0 Twist 0.0

Lens Zoom 45.0 C. Shift X 0.0 % C. Shift Y 0.0 %

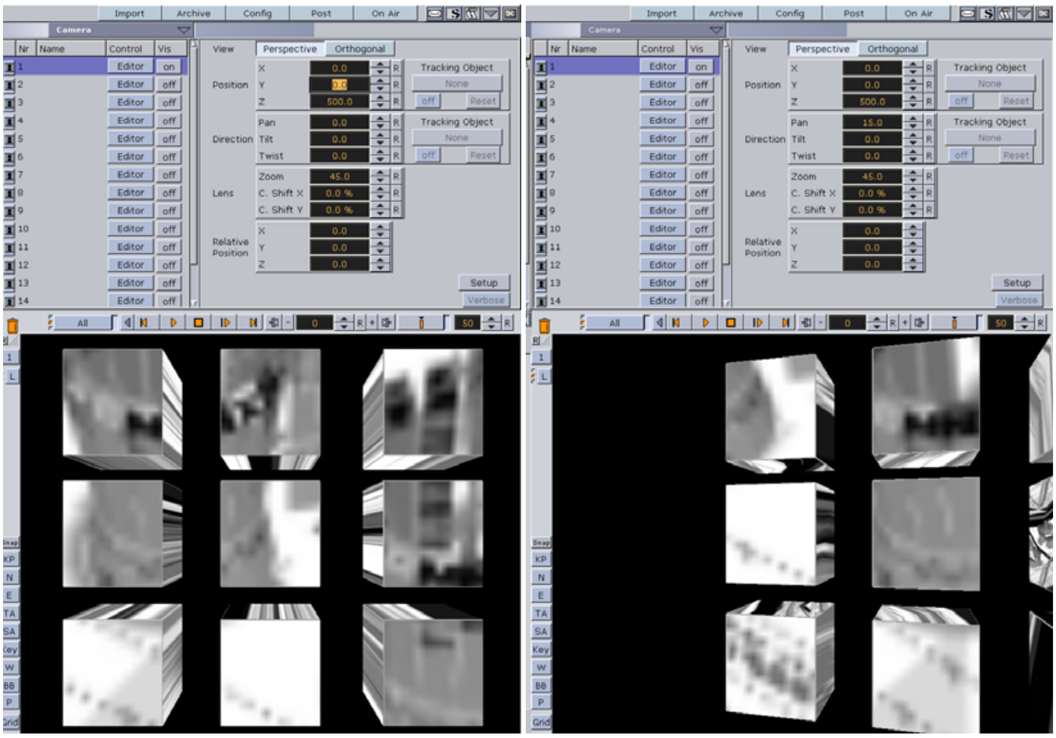
Relative Position X 0.0 Y 0.0 Z 0.0

Tracking Object None off Reset

Direction

Sets the values for Pan, Tilt, and Twist.

Figure 108: Direction: pan set to 0 (L) and 15 (R)



View: Perspective | Orthogonal

Position X 0.0, Y 0.0, Z 500.0

Direction Pan 0.0, Tilt 0.0, Twist 0.0

Lens Zoom 45.0, C. Shift X 0.0 %, C. Shift Y 0.0 %

Relative Position X 0.0, Y 0.0, Z 0.0

Tracking Object None

View: Perspective | Orthogonal

Position X 0.0, Y 0.0, Z 500.0

Direction Pan 15.0, Tilt 0.0, Twist 0.0

Lens Zoom 45.0, C. Shift X 0.0 %, C. Shift Y 0.0 %

Relative Position X 0.0, Y 0.0, Z 0.0

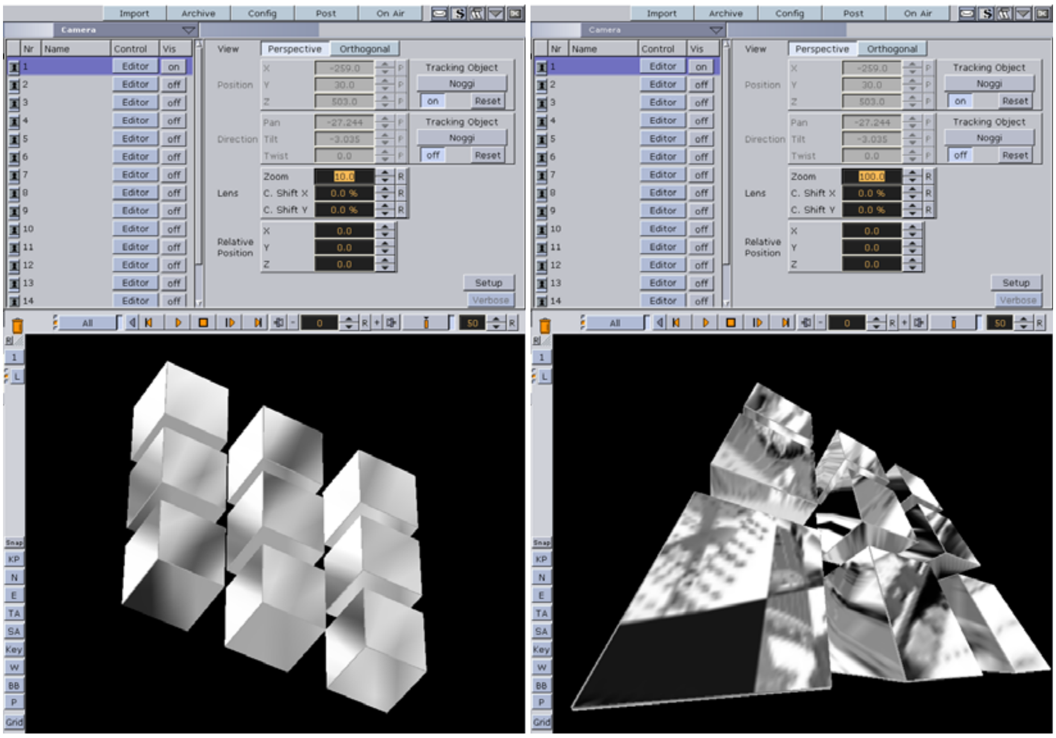
Tracking Object None

Lens

There are three lens parameters:

- **Zoom** Zooms in and out. From a design perspective, you get a much sharper view of the object when you use the camera zoom rather than zooming the object itself. Using the camera zoom increases object perspective.

Figure 109: Zoom set to 10 (L) and 100 (R)

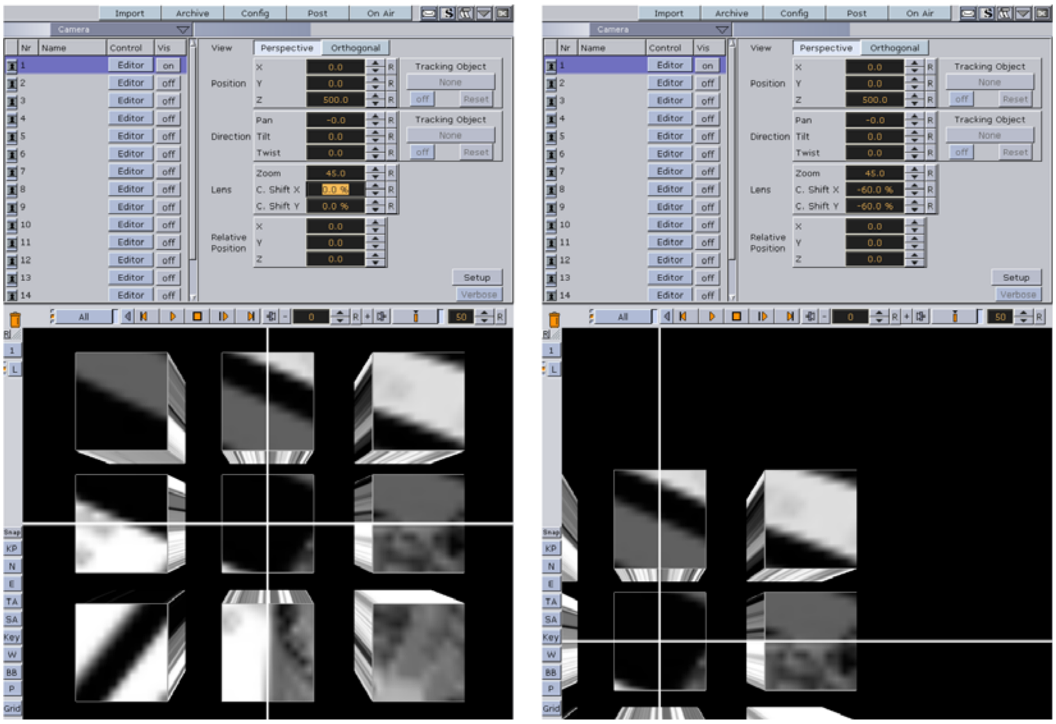


Left: Zoom 10.0, C. Shift X 0.0 %, C. Shift Y 0.0 %

Right: Zoom 100.0, C. Shift X 0.0 %, C. Shift Y 0.0 %

- **C. Shift X** Shifts the center of the lens according to the X axis.
- **C. Shift Y** Shifts the center of the lens according to the Y axis.

Figure 110: Left is default (0); right is C Shift X = -60 and C Shift Y= -60



Position X 0.0 Y 0.0 Z 500.0
Direction Pan -0.0 Tilt 0.0 Twist 0.0
Lens Zoom 45.0 C. Shift X 0.0 % C. Shift Y 0.0 %
Relative Position X 0.0 Y 0.0 Z 0.0

Position X 0.0 Y 0.0 Z 500.0
Direction Pan -0.0 Tilt 0.0 Twist 0.0
Lens Zoom 45.0 C. Shift X -60.0 % C. Shift Y -60.0 %
Relative Position X 0.0 Y 0.0 Z 0.0

Relative Position

Sets the position of the camera along the X, Y, and Z axis, relative to a real camera. Entering a value in for X, Y and Z axis for the Relative Position changes the actual camera position values, but relative to its orientation.

Verbose

This button in the lower right corner of the Camera editor is linked to cameras in virtual sets. For more information about the Virtual Set option, see the Virtual Set documentation.

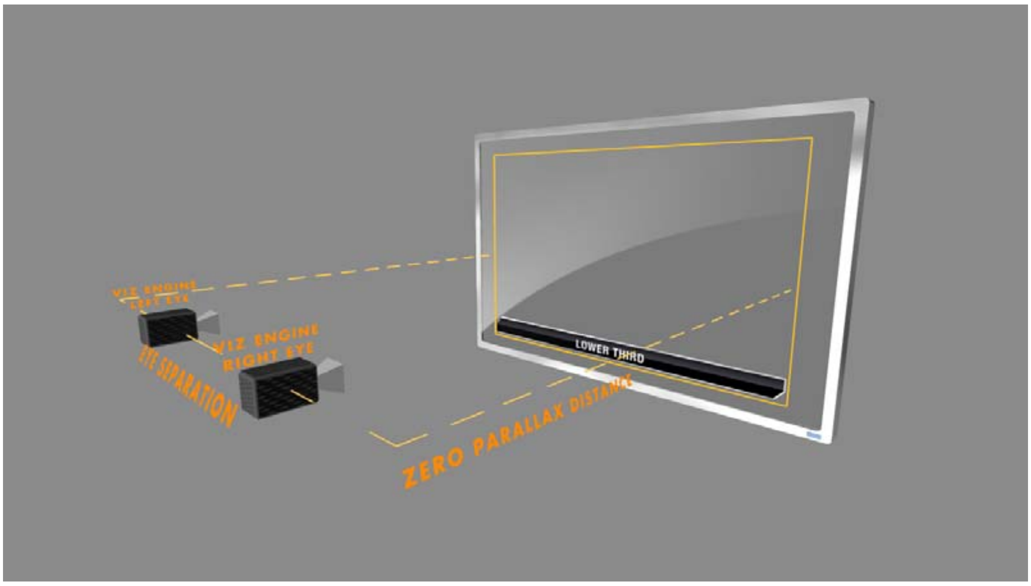
Stereo Settings

Any Viz Artist animation that was created for 2D broadcast can also be rendered as a stereo pair.

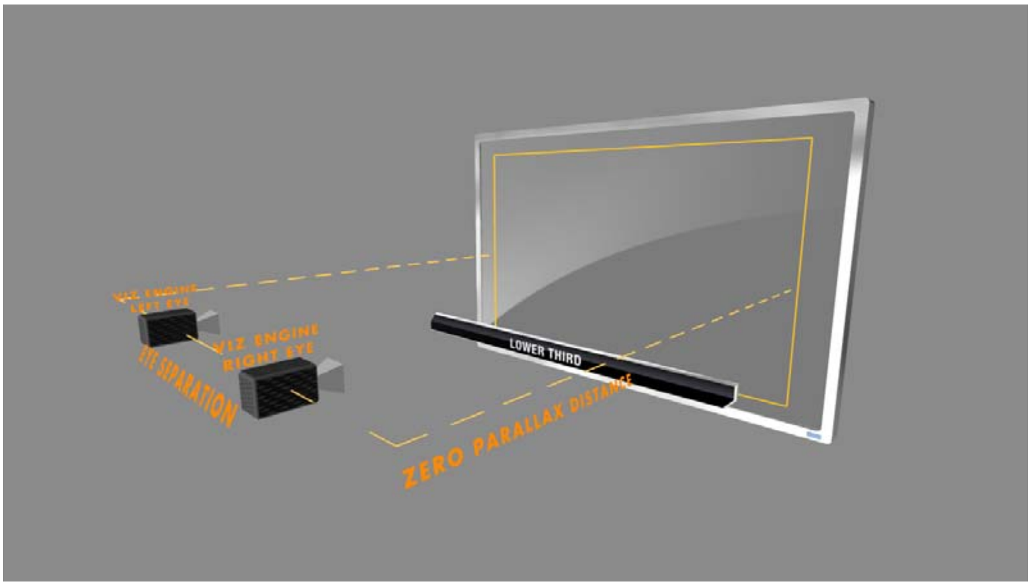
There are two parameters which define how each image is rendered. Both parameters are part of the camera configuration, but they can be changed remotely on the fly as well.

Zero parallax means that the pixels for the left and the right eye are identical. An object which would be positioned at zero parallax distance from the camera would be identical for the left and right eye. The eye separation defines the distance of the cameras for the left and right eye. When looking at the drawing the cameras are separated by the same amount as the eyes of the viewer. A different scaling might be used in the scene and the eye separation must be adjusted accordingly.

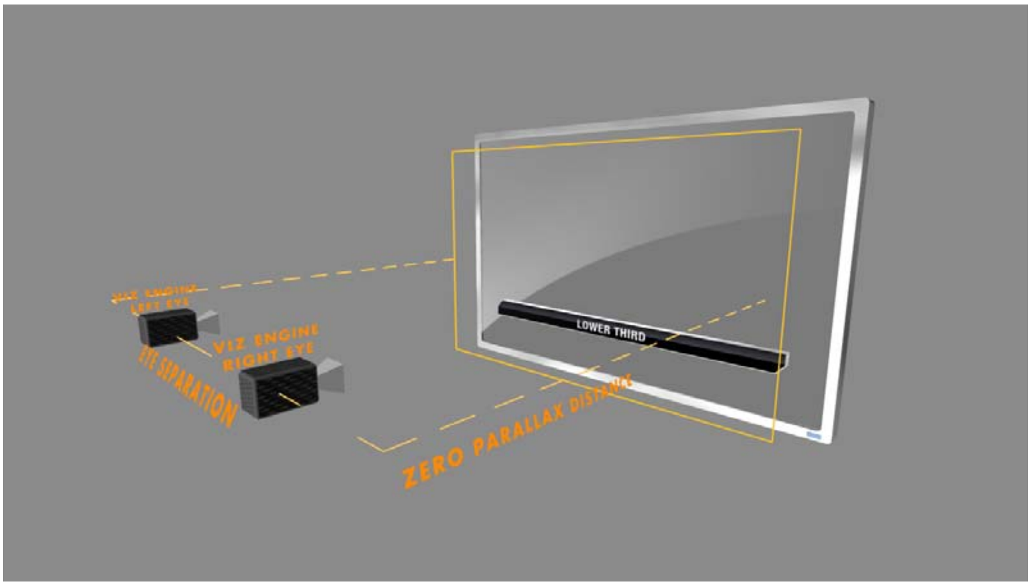
If the scene is built in the same scale as the monitor, with the camera at the same distance from the object as the viewer, and the eyes separated at the same amount an object positioned at the zero parallax, the distance appears to be flat with the screen’s surface.



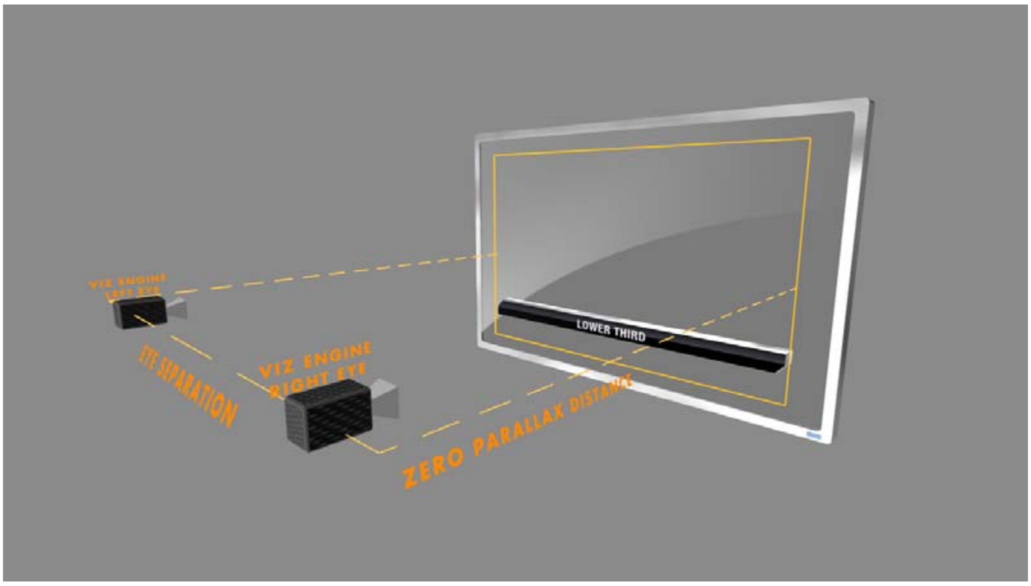
If the zero parallax distance is increased the object moves to the front.



If the distance is decreased the object moves away.

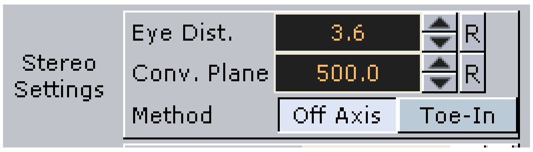


The eye separation defines how flat or deep an object appears. If we increase the eye separation, the scene appears deeper.



A very large eye separation parameter results in an unreadable 3D representation, as the brain can not mix the images for the left and right eye anymore. Zero eye separation results in a flat representation, like a 2D image.

Figure 111: Stereo settings


Stereo Settings

Eye Dist. 3.6 R

Conv. Plane 500.0 R

Method [✓ Off Axis] Toe-In

- **Eye Dist.** Fine-tunes for individual users. A higher value for will result in more depth whereas a lower value will flatten the image.
- **Conv. Plane** A higher value for the zero parallax distance means that objects will come closer to the viewer, a lower one will push the graphics to the back.

- **Method**:
  - **Off Axis** Creates a distortion-free stereo pair. This is the preferred option.
  - **Toe-In** The cameras point slightly inwards towards each other so that the lens axes converge at a single point (“vergence point”). Objects in the same plane as the vergence point will appear to be close to the plane of the image, with other objects seeming to be behind or in front of this point.

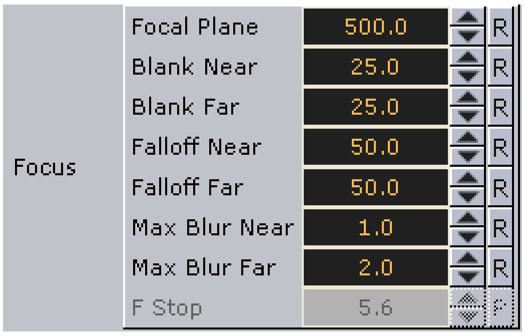
These values must be set during production in accordance with the convergence values used for the cameras.

See Also

- Stereoscopy Best Practices

Focus

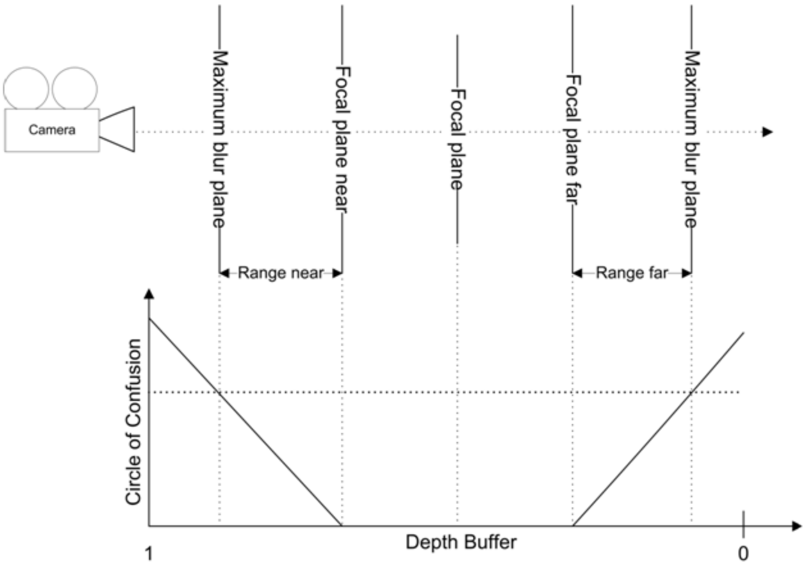
Focus settings are enabled once the Depth of Field parameter is set to Active. See Rendering.



Focus	Focal Plane	500.0	R
	Blank Near	25.0	R
	Blank Far	25.0	R
	Falloff Near	50.0	R
	Falloff Far	50.0	R
	Max Blur Near	1.0	R
	Max Blur Far	2.0	R
	F Stop	5.6	R

The focus settings are as follows.

- **Focal Plane** Defines the distance between the camera and objects which are in focus.
- **Blank Near** Adjusts the space between the camera and objects nearest to the camera in focus.
- **Blank Far** Adjusts the space between the camera and objects furthest from the camera in focus.
- **Fall-off Near** Decreases the space between objects with the maximum blur and objects in focus.
- **Fall-off Far** Increases space between objects with the maximum blur and objects in focus.
- **Max. Blur Near** Sets the size of maximum blur nearest to the camera. For example a value of 20 signify that a pixel is enlarged to a size of 20 pixels.
- **Max. Blur Far** Sets the size of maximum blur furthest from the camera. For example a value of 20 signify that a pixel is enlarged to a size of 20 pixels.
- **F Stop** Only for CGP Panasonic cameras. Consult the relevant documentation.



To avoid artifacts, avoid using maximum blur levels which are higher than 30 pixels. A blur radius of 30px creates rather blurry results which should be sufficient. It is possible to increase this maximum radius, but this would also increase the rendering time.

Set the clipping planes in the range you really need for the scene. So if your scene has only a depth of 200, set the far clipping plane to 250 or a similar value. Since the depth of field shader takes the depth from the depth buffer, this causes more accurate results.

Try to avoid high blur levels for objects before the focal plane. This can cause undesired effects.

You can increase performance, but artifacts would appear for blur levels lower then 30px.

9.3 Stereoscopy Best Practices

The following is a list of what to do and not to do in order to get the best results.

What to do

- Think of it as a window you are looking through.
- Make sure that single elements which form a new element, like rows of a table, are all at the same distance and not scaled to appear the same size in 2D.
- Make sure that all graphics which belong to one show are positioned roughly the same distance away from the camera. This way you do not need to adjust the convergence parameters all the time.
- With alioscopy, keep the floating objects not too far from the 0 plane, as this will make them have artefacts.
- For best effect have objects animate slowly forward and then fade to 0 alpha before hitting the screen edge.

- With alioscopy, for text that is supposed to be readable, keep it big. Dropshadow is a very good effect to enhance the sense of depth. 3DS productions use HD signals.
- For a 24" screen the best viewing distance is about 2.8 meters. For a 40" it is about 4.4 meters.
- With alioscopy, look at the scene from the top view and space things out like in a theatre. Depth and various objects in different Z positions will make it more believable: foreground, middle ground and background.
- Wobble effects are fine as long as they do not exaggerate the effect, otherwise you get the paper bag effect. This again depends on how far or close to the camera the object is positioned.

What not to do

- Do not use transparency where it goes over video. It's OK if a transparent graphics object is over another graphical element if the distance is not too large.
- Do not use too much depth. Try to keep the elements separated not too far from each other. The Z space available for graphics is limited.
- Do not create animations which come in from the side, especially when the final result will be floating in front of the screen (this is very disturbing).
- Do not use moving flairs or effects.
- Do not have any objects pulsating forth and back in Z space as a part of a loop as this will lead to the paper bag effect.

See Also

- Stereo Settings
- Stereoscopic Output Using Shutter Glasses

9.4 Stereoscopic Output Using Shutter Glasses

It is very easy to set up a working 3D environment for a designer to start creating stereoscopic scenes. The demands on the hardware are moderate.

Hardware requirements

- Any workstation
- Any NVIDIA Quadro graphics card with a 3-pin DIN connector
- NVIDIA 3D vision or equivalent
- Monitor with a refresh rate >= 100Hz

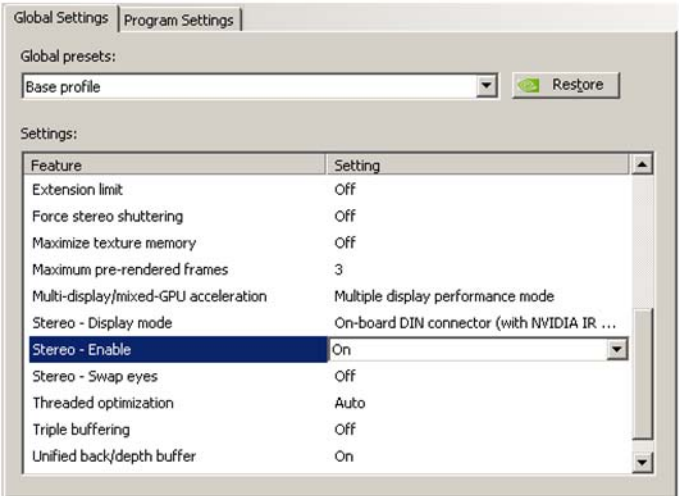
Software requirements

- Viz Artist / Viz Engine 3.3 or later
- The display must be set to a refresh rate >= 100Hz

To set up a 3D working environment

1. In the Windows video card 3D settings, switch **Stereo - Enable** to **On**
2. Select the appropriate **Stereo - Display mode** for your stereo devices.

Note: For NVIDIA's 3D Vision this is “On-board DIN connector”.

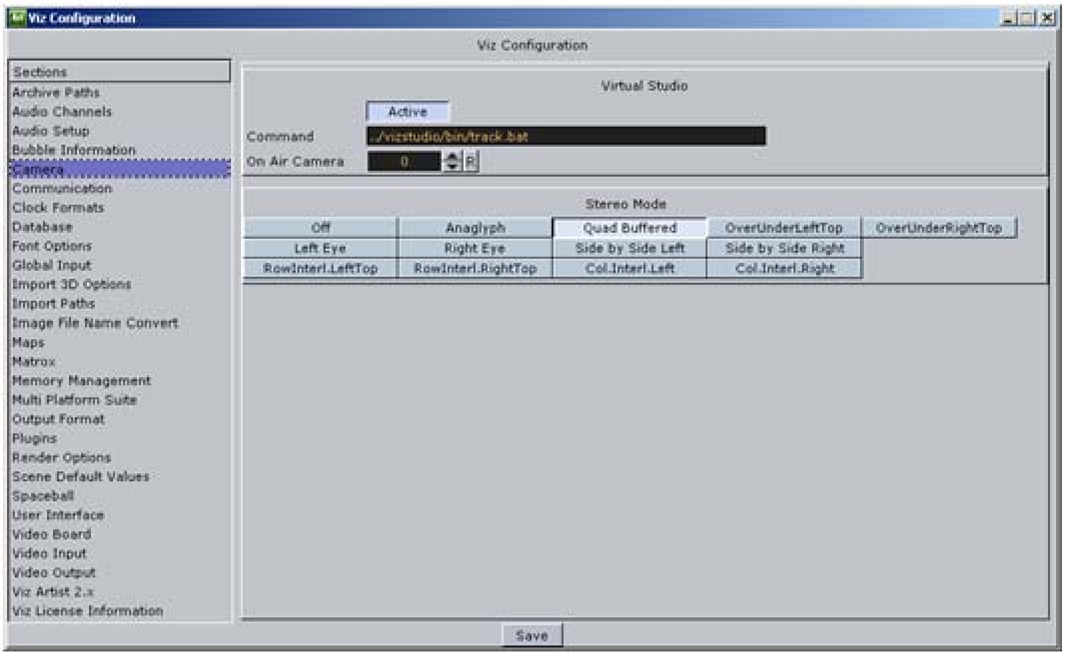


Global presets: Base profile

Settings:

Feature	Setting
Extension limit	Off
Force stereo shuttering	Off
Maximize texture memory	Off
Maximum pre-rendered frames	3
Multi-display/mixed-GPU acceleration	Multiple display performance mode
Stereo - Display mode	On-board DIN connector (with NVIDIA IR ...
Stereo - Enable	On
Stereo - Swap eyes	Off
Threaded optimization	Auto
Triple buffering	Off
Unified back/depth buffer	On

3. Set the Stereo Mode to **Quad Buffered** in the Viz Config Camera settings.



Command ../vizstudio/bin/track.bat

On Air Camera 0

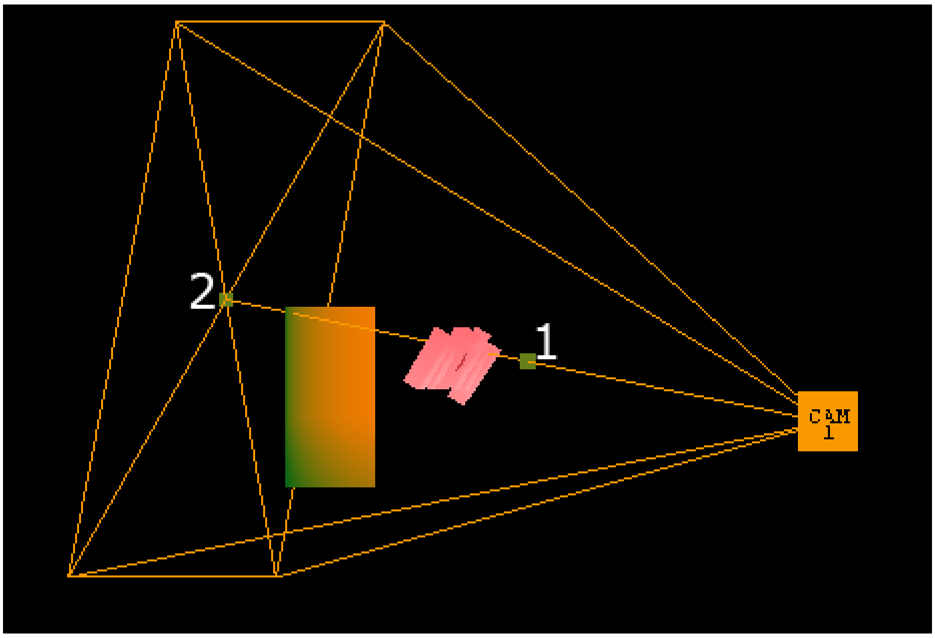
Alternatively it is possible to use an autostereoscopic monitor that supports sub-fields or side-fields, such as Miracube's c190 series. The drawback with these displays is that you need to go on-air to get a full-screen image, which is halved in resolution.

9.5 Changing Camera Parameters in Orthogonal Views

To view the predefined orthogonal views

- Click **Views**.

The camera is shown as an orange square symbol in the predefined orthogonal views. In addition, two small green square shaped handles makes it possible to modify the position along the X, Y and Z axis of the camera, and also the values for pan and tilt.



- Dragging the camera icon will alter the position of the camera while handle (2) will be fixed. The values for pan and tilt will also be altered.
- Dragging handle (1) will alter the position of the camera. The camera icon and handle (2) will be dragged along, so that pan and tilt will not be altered.
- Dragging handle (2) will alter pan and tilt while the position of the camera will be fixed.

9.6 Tracking Objects with a Camera

A virtual camera can be set up to track containers both regarding position and direction. Tracking position means that the position of the camera will be in the center of the container. Tracking direction means that the camera will look at the center of the container, for example throughout an animation.

There are three options for configuring parameters view, depending on your number of tracking objects:

- Zero tracking objects: You can only modify the Position X and Y.
- One tracking object: When dragging a container into the Position drop zone, all the parameters will be greyed out. In the Scene Setup menu, switch from Camera to **Container** to modify Positions X, Y and Z.
- Two tracking objects: You can only modify the Lens and Relative Position parameters.

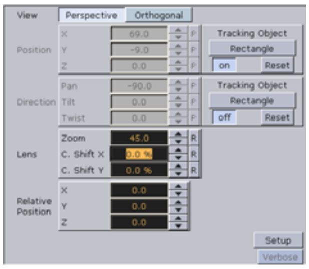
For the relevant camera parameter details, see Right Pane Parameters.

To track an object

1. In your scene tree, create a group with two child primitive objects.

Tip: It is better to use objects with a low polygon, like a rectangle, as your tracking object. High polygon tracking objects, such as a sphere, make your scene heavy and slow.

2. Drag the first container into the Position drop zone.



View: Perspective | Orthogonal

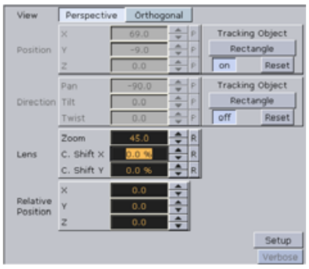
Position: X 69.0, Y -9.0, Z 0.0 — Tracking Object: Rectangle, on, Reset

Direction: Pan -90.0, Tilt 0.0, Twist 0.0 — Tracking Object: Rectangle, off, Reset

Lens: Zoom 45.0, C. Shift X 0.0 %, C. Shift Y 0.0 %

Relative Position: X 0.0, Y 0.0, Z 0.0

3. Drag the second container into the Direction drop zone.



View: Perspective | Orthogonal

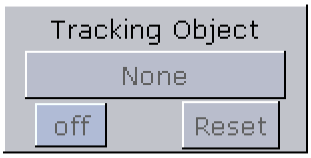
Position: X 69.0, Y -9.0, Z 0.0 — Tracking Object: Rectangle, on, Reset

Direction: Pan -90.0, Tilt 0.0, Twist 0.0 — Tracking Object: Rectangle, off, Reset

Lens: Zoom 45.0, C. Shift X 0.0 %, C. Shift Y 0.0 %

Relative Position: X 0.0, Y 0.0, Z 0.0

4. Modify camera parameters. Select the Position and/or Direction tracking object in the scene tree, and change the position and/or direction in the Container properties (select Container from the Scene Setup menu).



Tracking Object: None

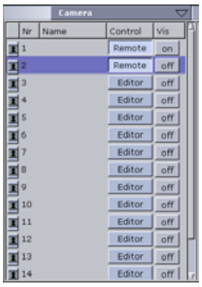
- **On/Off** By dragging a container to the drop zone, the camera tracking is automatically activated. To inactivate the tracking, click the On/Off button.
- **Reset** Disables the camera tracking.

9.7 Receiving Tracking Data from a Real Camera

In virtual set scenarios, as the real cameras moves, the virtual camera should move in the same manner.

To receive tracking data from a real camera

- In the Left Pane Parameters, in the Control column, click the **Editor** button to set it to **Remote**.



Camera

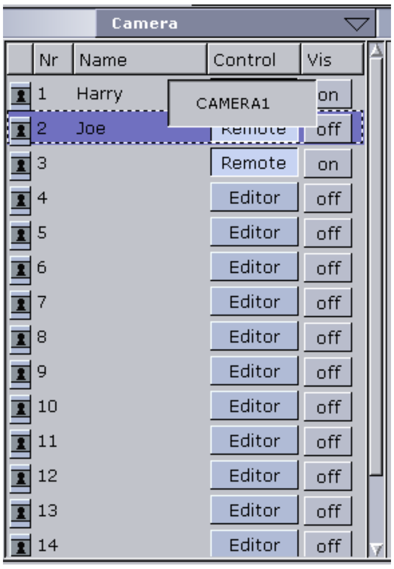
Nr	Name	Control	Vis
1		Remote	on
2		Remote	off
3		Editor	off
4		Editor	off
5		Editor	off
6		Editor	off
7		Editor	off
8		Editor	off
9		Editor	off
10		Editor	off
11		Editor	off
12		Editor	off
13		Editor	off
14		Editor	off

For more information about the Virtual Set option, see the Virtual Set documentation.

9.8 Copying Properties from one Camera to another

To copy camera properties

- In the Left Pane Parameters, click the source camera and drag it onto the target camera.



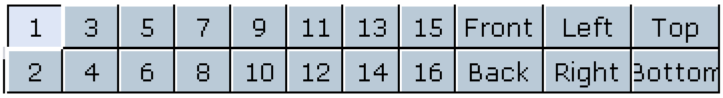
Camera

Nr	Name	Control	Vis
1	Harry		on
2	Joe	Remote	off
3		Remote	on
4		Editor	off
5		Editor	off
6		Editor	off
7		Editor	off
8		Editor	off
9		Editor	off
10		Editor	off
11		Editor	off
12		Editor	off
13		Editor	off
14		Editor	off

The properties of the source camera will be copied onto the target camera.

9.9 Camera Selection

The scene can be viewed from various angles.



1	3	5	7	9	11	13	15	Front	Left	Top
2	4	6	8	10	12	14	16	Back	Right	Bottom

16 editable cameras and six predefined orthogonal views are available. The label on the Camera Selection button indicates which camera or view is currently selected and used in the render output.

To view the scene from a different angle, click the Camera Selection button, and then from the Camera menu that opens, select a camera or view. To close the Camera menu, click the Camera Selection button again.

- 1: 16 Editable cameras.

Note: The position and direction of the 16 editable cameras can be defined in the scene setup. For more information, see Containers.

- **Front** Displays output through orthogonal camera in front of the scene.
- **Back** Displays output through orthogonal camera behind the scene.
- **Left** Displays output through orthogonal camera at the left side.
- **Right** Displays output through orthogonal camera at the right side.
- **Top** Displays output through orthogonal camera above the scene.
- **Bottom** Displays output through orthogonal camera below the scene.

9.10 Camera Animation

All camera parameters can be animated in the same way as Creating Animations in general. The process is roughly as follows:

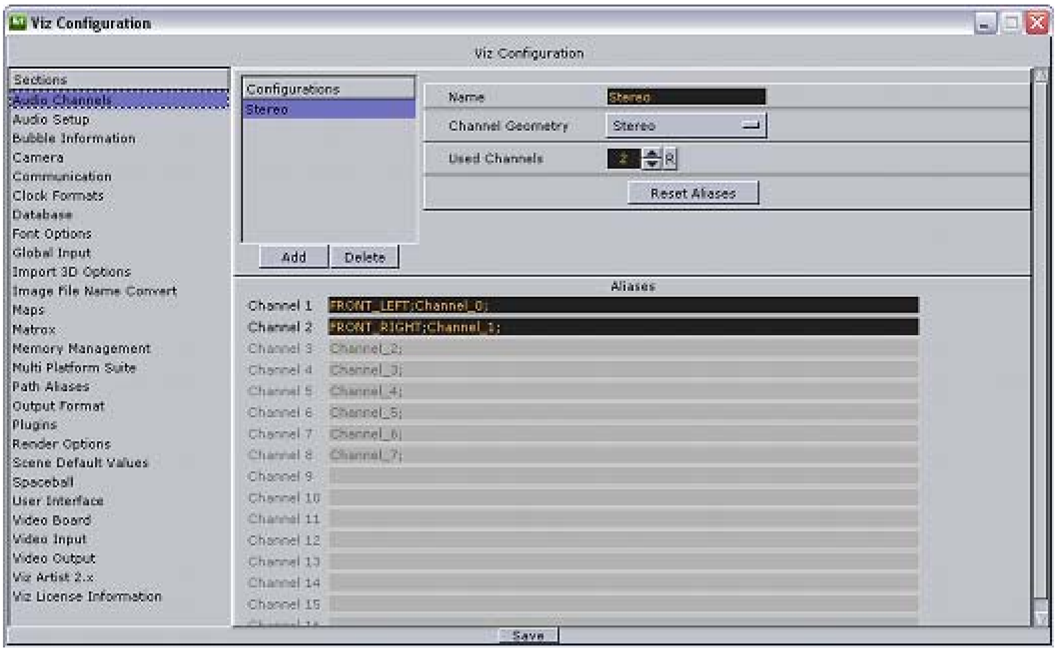
- Set a value in the Camera editor
- Add a keyframe
- Change the value
- Add a keyframe again.

Animations can also easily be created in one of the orthogonal views. There the cameras appear as symbols and by repositioning them and adding keyframes, animation sequences are created. For more information, see Changing Camera Parameters in Orthogonal Views.

More advanced camera animations are normally created with the camera tracking feature, as in Tracking Objects with a Camera.

10 Configuring Viz

Figure 112: Viz Configuration Interface, Video



Name: Stereo
Channel Geometry: Stereo
Used Channels: 2

Channel 1	FRONT_LEFT;Channel_0;
Channel 2	FRONT_RIGHT;Channel_1;
Channel 3	Channel_2;
Channel 4	Channel_3;
Channel 5	Channel_4;
Channel 6	Channel_5;
Channel 7	Channel_6;
Channel 8	Channel_7;
Channel 9	
Channel 10	
Channel 11	
Channel 12	
Channel 13	
Channel 14	
Channel 15	

Viz Config is the configuration interface for Viz Engine and other applications that integrate with Viz Engine.

Basically the user interface is divided in two; sections on the left and settings on the right. Various parameters regarding the program functionality can be set in the configuration; however, additional and more advanced settings can be set using the configuration file (not recommended).

Viz is mainly configured using Viz Config; however, all configuration settings are stored in a file found under the install directory. The configuration file uniquely identifies the machine Viz is installed on by using its hostname (e.g. *Viz-<hostname>-0-0.cfg*).

Any changes to the hostname will affect the configuration of Viz. If a hostname is changed, a new configuration file is created with a default setup. The old configuration file is not deleted, but left unused. It is possible to reassign the old configuration file using the Viz command -g <config file>.

CAUTION! Make sure to save any changes before closing Viz Config or else they might be discarded. Hence, it is recommended to restart the software to make sure that the changes take effect.

This section contains information on the following topics:

- To start Viz Config
- Audio Settings
- Camera

- Communication
- Clock Formats
- Database
- Font Options
- Global Input
- Import 3D Options
- Image File Name Convert
- Maps
- Matrox
- Memory Management
- Multi Platform Suite
- Path Aliases
- Output Format
- Plugins
- Render Options
- Scene Default Values
- Spaceball
- User Interface
- Video Board
- Video Clip
- Video Input
- Video Output
- Viz Artist 2.x
- Viz License Information

To start Viz Config

1. Start Viz Config from *Start* > *All Programs* > *Vizrt* > *Viz 3* > *Viz Config 3*, or
2. if Viz Artist 3 is running, click *Config* on the main menu.

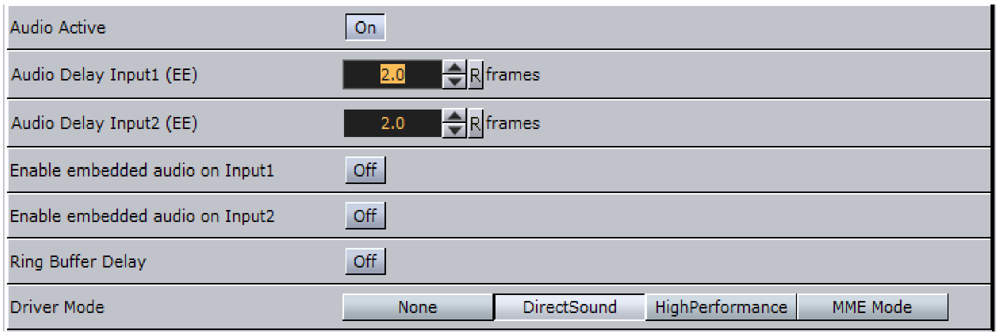
10.1 Audio Settings

This section contains information on the following topics:

- Properties and Parameters
- Dual Channel Configuration
- To add new audio channels
- To add multi-language audio channels
- To add multiple audio channel configurations
- To map audio output for a dual channel setup
- To delete audio channels
- To manually activate an audio device

Properties and Parameters

Figure 113: Various tab



Audio Active	On
Audio Delay Input1 (EE)	2.0 R frames
Audio Delay Input2 (EE)	2.0 R frames
Enable embedded audio on Input1	Off
Enable embedded audio on Input2	Off
Ring Buffer Delay	Off
Driver Mode	None DirectSound HighPerformance MME Mode

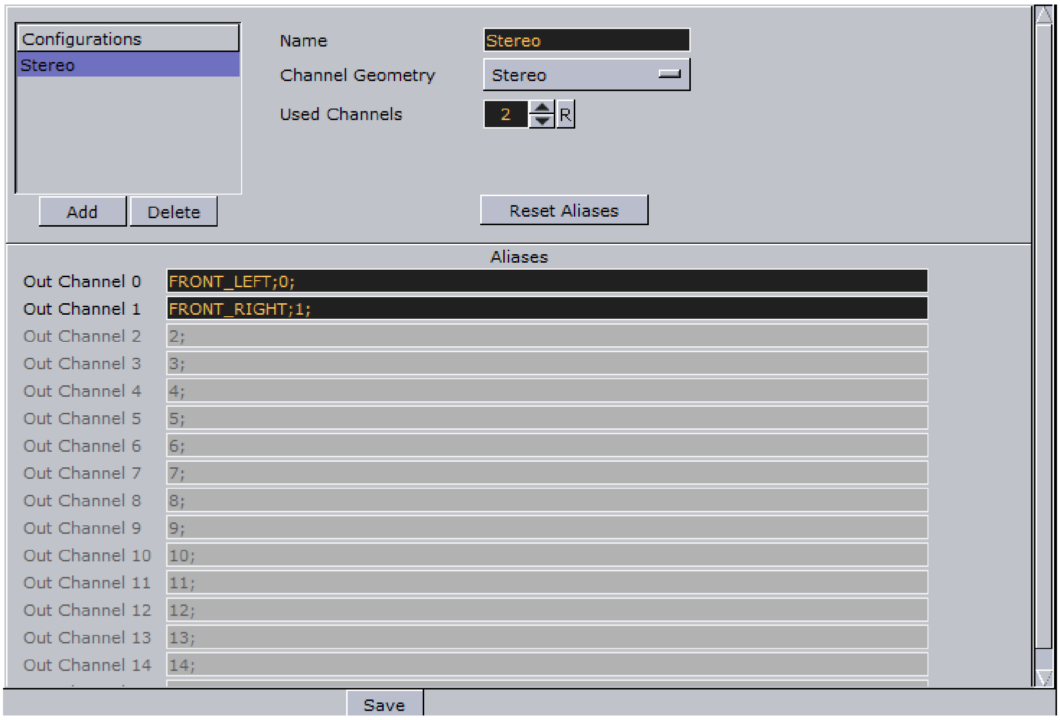
- **Audio Active**: Makes audio active or inactive.
- **Audio Delay Input1 (EE)**: Delay can be adjusted to synchronize the inputs at the output. This only affects Targa boards.
    - For Matrox, see the configuration file and SECTION MATROX_CONFIG's *Matrox0.AudioIn1.AudioDelayDVE* setting.
- **Audio Delay Input2 (EE)**: Delay can be adjusted to synchronize the inputs at the output. This only affects Targa boards.

Note: To enable audio mixing on the video board, these two settings must be turned on. Otherwise the audio data from Input 1 to Input 4 are not mixed with the values for the stage and played out.

Note: Targa boards do not distinguish between DVE and texture delays.

- **Enable embedded audio on Input1**: Enables embedded audio for use with video as texture or DVE. Input 1 in the Video Input section must be enabled in order to use embedded audio output.
- **Enable embedded audio on Input2**: Enables embedded audio for use with video as texture or DVE. Input 2 in the Video Input section must be enabled in order to use embedded audio output.
- **Ring buffer delay**: Turn **On** to enable the audio system to compensate for the ring buffer delay automatically.
- **Driver mode**: Select **None** for no sound card output. Select **DirectSound** for direct sound. Select **High Performance** for high performance MME mode (this only works for Wave driver cards. Select **MME** mode for that mode only.

Figure 114: Channels tab



Stereo

Name: Stereo

Channel Geometry: Stereo

Used Channels: 2

Out Channel 0: FRONT_LEFT;0;

Out Channel 1: FRONT_RIGHT;1;

Out Channel 2: 2;

Out Channel 3: 3;

Out Channel 4: 4;

Out Channel 5: 5;

Out Channel 6: 6;

Out Channel 7: 7;

Out Channel 8: 8;

Out Channel 9: 9;

Out Channel 10: 10;

Out Channel 11: 11;

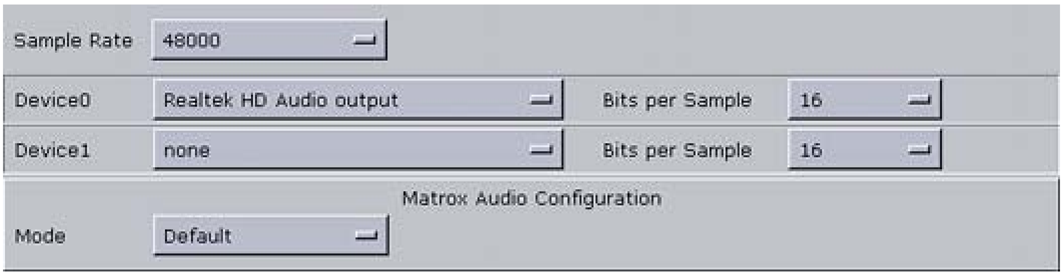
Out Channel 12: 12;

Out Channel 13: 13;

Out Channel 14: 14;

- **Configurations**: Displays a list of channel configuration(s).
- **Add**: Adds a new channel configuration to the Configurations list.
- **Delete**: Deletes the selected channel configuration from the Configurations list.
- **Name**: Sets the name of the selected channel configuration.
- **Channel Geometry**: Sets the channel geometry. Available options are: Mono, Stereo, 5.1, 7.1 and Quad.
- **Used Channels**: Sets the number of configurable channel alias fields that can be mixed by the internal channels in Viz Engine (software). On a Matrox system this number must be equal to the number of configured input channels (hardware). This setting is independent of the Channel Geometry setting.
- **Reset Aliases**: Sets the channel aliases to the default option; however, the custom entry is not removed.
- **Out Channel 0-15**: The Out Channels represent the internal Viz audio channels, which are mixed to the output device one by one. Alias names are not only useful for creating multilingual systems, but also define the channel geometry for the Default and FX audio clip mix modes. Aliases are separated by a semicolon. The Audio plug-in only uses the channel aliases to find the correct speakers for Pan and 3D sound effects in FX mode. With this functionality you are able to use any of the 16 internal audio channels to play any audio geometry.

Figure 115: Setup tab



Sample Rate 48000

Device0 Realtek HD Audio output Bits per Sample 16

Device1 none Bits per Sample 16

Matrox Audio Configuration

Mode Default

In this section all installed DirectSound audio cards are listed displaying the available audio inputs that can be mixed to one audio output by Viz Engine. It is currently only possible To manually activate an audio device by editing the configuration file.

Note: The Matrox audio-extension board is not configurable through this user interface.

- **Sample Rate**: *Reserved for later use*. Displays the sample rate. Default sample rate is 48kHz (48000) which is the maximum allowed.
- **Device** *n*: Displays the name of the audio card.
- **Bits per Sample**: *Reserved for later use*. Displays the number of bits used per sample. Default value is 16 Bit per sample rate.
- **Mode**: Refers to the audio mode of the Matrox card. Available modes are Embedded, AES, Embedded AES, Loop and Default.
  - **Embedded**: Forces the Matrox card to capture audio from the video signal and outputting embedded audio.
  - **AES**: Forces the Matrox card to capture and output audio through the AES/EBU connectors.
  - **Embedded AES**: Forces the Matrox card to capture audio from the video signal and outputting it to the AES/EBU connectors.
  - **AES Embedded**: Forces the Matrox card to capture audio from the AES connectors and outputting the audio as streams in the video signal.
  - **Loop**: Moves the audio from the inputs to the outputs without being down-mixed by Viz Engine.
  - **Default**: The default device(s) are used (usually the built-in sound board).

Note: Matrox X.mio1 only supports balanced audio through XLR connectors. X.mio2 only supports unbalanced audio through BNC connectors.

Dual Channel Configuration

In a dual channel setup AES channels can be configured to use up to 4 channels per Viz Engine for an X.mio1 and up to 8 channels per engine for an X.mio2. If 8/16 channels are set for one Viz Engine the other Viz Engine will overwrite these settings as the maximum of AES channels is 8/16 for X.mio1/X.mio2.

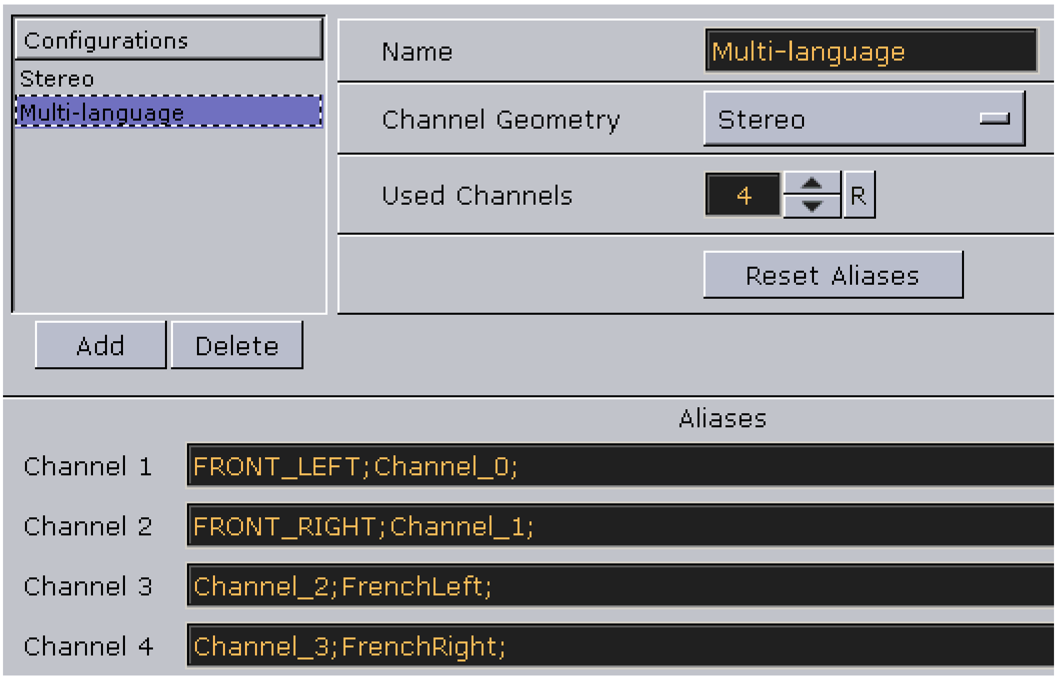
For correct mapping both Viz Engines should be set to an equal number of used AES channels.

To add new audio channels

1. Click the **Add** button.
2. Select the new entry from the **Configurations** list, and enter the new name in the **Name** field.
3. Set the **Channel Geometry**, and the number of **Used Channels**.
4. Click **Save**.

To add multi-language audio channels

Figure 116: Multi-language audio setup



Configurations
Stereo
Multi-language

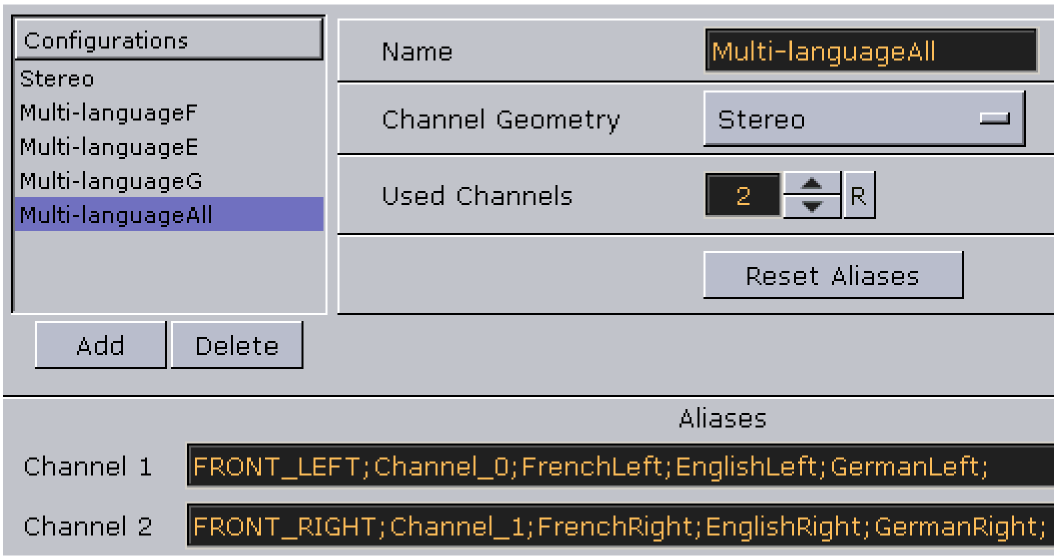
Name: Multi-language
Channel Geometry: Stereo
Used Channels: 4

Aliases
Channel 1: FRONT_LEFT;Channel_0;
Channel 2: FRONT_RIGHT;Channel_1;
Channel 3: Channel_2;FrenchLeft;
Channel 4: Channel_3;FrenchRight;

1. Click the **Add** button.
2. Select the new entry from the **Configurations** list, and enter the new name in the **Name** field.
3. Set the Channel Geometry to **Stereo**, and then click **Reset Aliases**.
4. Increase the number of **Used Channels** to 4
   - This will improve the performance.
5. In the **Channel 3** field add the alias **FrenchLeft**.
6. In the **Channel 4** field add the alias **FrenchRight**.
   - Add the same configuration for English (EnglishLeft, EnglishRight) and German (GermanLeft, GermanRight) on the other machines.
7. Click **Save**.

To add multiple audio channel configurations

Figure 117: Multi-languageAll setup



Name: Multi-languageAll

Channel Geometry: Stereo

Used Channels: 2

Channel 1: FRONT_LEFT;Channel_0;FrenchLeft;EnglishLeft;GermanLeft;

Channel 2: FRONT_RIGHT;Channel_1;FrenchRight;EnglishRight;GermanRight;

1. See how To add multi-language audio channels, and repeat the process such that the Audio Settings configuration has 3 setups for the three languages.
2. Name the configurations the following way;
   - Multi-languageF
   - Multi-languageE
   - Multi-languageG
3. Add a Multi-languageAll configuration (see Figure 117: Multi-languageAll setup).
4. In the **Channel 1** field add the alias **FrenchLeft**, **EnglishLeft** and **GermanLeft.**, and do the same for **Channel 2**.

```
FRONT_LEFT;Channel_0;FrenchLeft;EnglishLeft;GermanLeft;
FRONT_RIGHT;Channel_0;FrenchRight;EnglishRight;GermanRight;
```

5. Click **Save**.

To map audio output for a dual channel setup

1. In a dual channel environment there are two configuration files, one for each Viz Engine.
2. For the first Viz Engine (1) the configuration of the audio output channel mappings should look like this:

```
Matrox0.AudioOut1.MapToVizChannel = 0
Matrox0.AudioOut2.MapToVizChannel = -1
```

3. Note that the audio output for the first Viz Engine (1) should be according to the video output channel:

```
Matrox0.VideoOut1.MapToVizChannel = 0
Matrox0.VideoOut2.MapToVizChannel = -1
```

4. For the second Viz Engine (2) the configuration should look like this:

```
Matrox0.AudioOut1.MapToVizChannel = -1
Matrox0.AudioOut2.MapToVizChannel = -1
```

5. Mind the difference to the video output settings:

```
Matrox0.VideoOut1.MapToVizChannel = -1
Matrox0.VideoOut2.MapToVizChannel = 0
```

6. All other audio output channels should be turned off:

```
Matrox0.AudioOut3.MapToVizChannel = -1
Matrox0.AudioOut4.MapToVizChannel = -1
```

To delete audio channels

1. Select a configuration entry from the **Configurations** list, and click the **Delete** button.
2. Click **Save**.

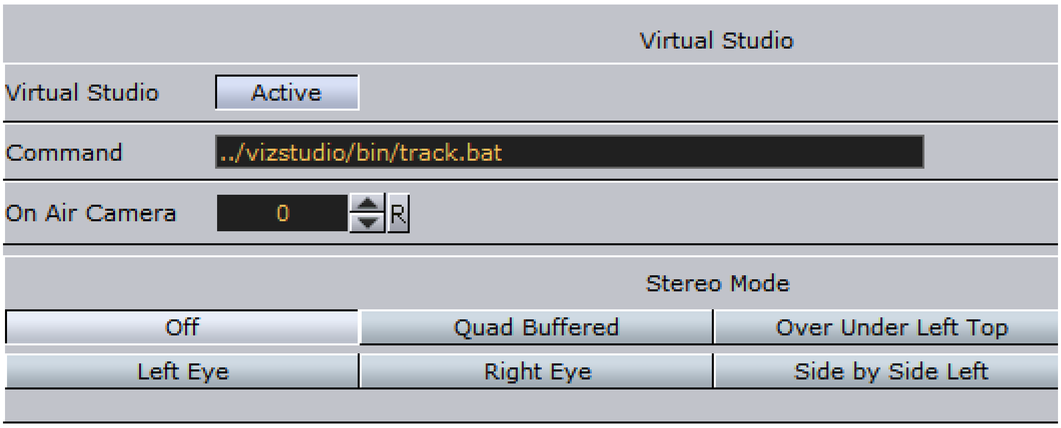
To manually activate an audio device

1. Open the configuration file.
2. Go to **SECTION AUDIO_CONFIG** and locate the **Available0** setting.
3. Activate the identified audio device (**Available0**) by adding its name to the **AudioDevice0** setting.
   - For multiple outputs, more than one device can be added as AudioDevice1, AudioDevice2 and so on; however, this is not very common.
4. Save the configuration file.

Example: AudioDevice0 = Realtek HD Audio output

10.2 Camera

Figure 118: Camera



Virtual Studio

Virtual Studio: Active

Command: ../vizstudio/bin/track.bat

On Air Camera: 0

Stereo Mode

In this section, special camera behavior settings that are used for virtual studio setups, can be defined. Viz IO is used as the studio configuration and calibration tool for enabling connectivity and control between all required studio devices such as cameras, routers, VTRs, video servers, audio mixers and other studio equipment.

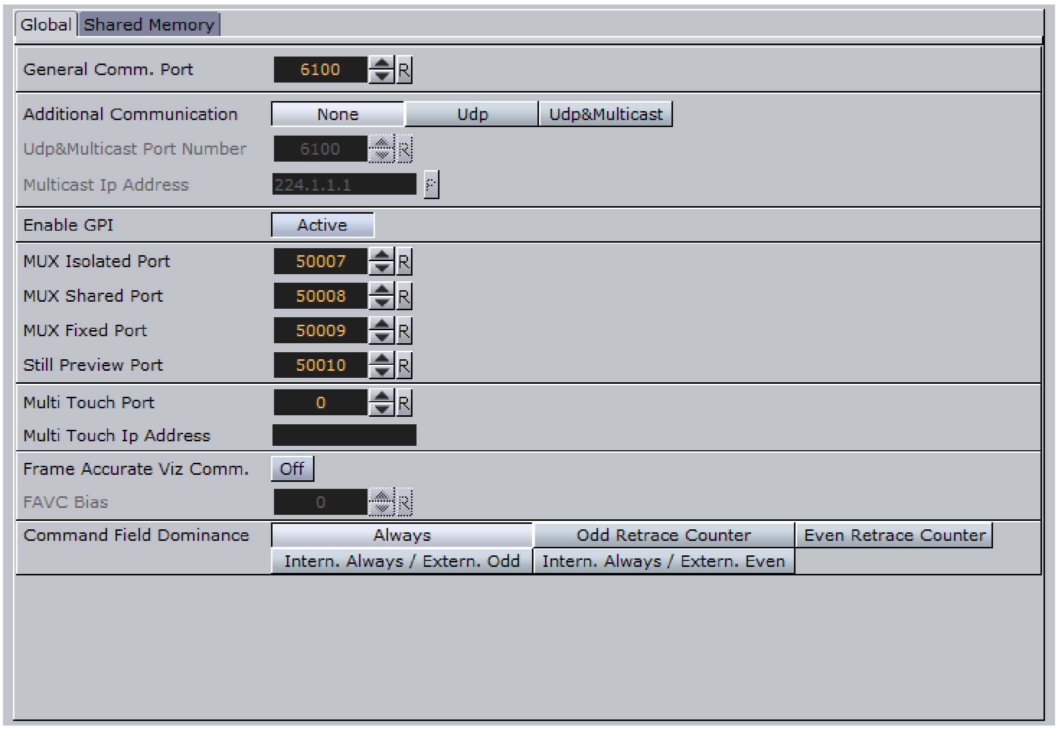
- **Virtual Studio**: When set to *Active* the tracking process will be started when Viz Engine is started.
- **Command**: Sets the path to the .bat file that will start the tracking process. If a valid path is given, Viz Engine will start Viz IO during startup.
- **On-air camera**: Activates the selected camera that should be used when setting the scene in on air mode. The camera is controlled by an external tracking device.

Stereo Mode settings will only take effect on air, during artist you won't see this rendering behavior!

- **Quad Buffered**: When using nVisio glasses together with a monitor with a frequency >100Hz the image for the left/right eye will be shown alternately and the glasses will darken the other eye. This configuration provides the full resolution.
- **Over Under Left Top/Right Top**: Both images will be drawn beneath each. Either side can be drawn first. The image height will be halved, providing half resolution
- **Left/Right Eye**: Either the left or the right eye image will be rendered for dual-channel setups in combination with a Video Wall / Stereo Distributor. Full resolution.
- **Side by Side Left/Right**: Both images will be rendered side by side, either left or right first. The image width will be halved, providing half resolution.

IMPORTANT! If you want to playout 3D Stereo Clips, you need to start Viz Engine with the -3D parameter. This sets Viz Engine in a special mode where a side by side clip will be split and played out on channel A and B on the Matrox board. Note that this is for clips only!

10.3 Communication



General Comm. Port	6100
Additional Communication	None Udp Udp&Multicast
Udp&Multicast Port Number	6100
Multicast Ip Address	224.1.1.1
Enable GPI	Active
MUX Isolated Port	50007
MUX Shared Port	50008
MUX Fixed Port	50009
Still Preview Port	50010
Multi Touch Port	0
Multi Touch Ip Address	
Frame Accurate Viz Comm.	Off
FAVC Bias	0
Command Field Dominance	Always Odd Retrace Counter Even Retrace Counter Intern. Always / Extern. Odd Intern. Always / Extern. Even

In this section, network connections can be set. Any external control software, for example Viz Content Pilot, uses TCP/IP network connections to send commands to the Viz Engine renderer engine. Viz Artist expects the commands at the port that is defined here. For a single pipe (one display) system, the default value (6100) should normally not be changed.

Global

- **General Communication Port**: Sets a general communication port for receiving external commands when in On Air mode. Changes to this setting must be reflected on the client side. Default is port 6100.
- **Additional Communication**: Enables commands to be sent to Viz Engine on UDP and Multicast or a combination of the two. This is used by Viz Video Wall.
- **Udp&multicast Port Number**: Sets the port number for the computers that share the same virtual IP address.
- **Multicast IP Address**: Sets the shared virtual IP address.

IMPORTANT! Viz 3.2 and later allows up to 255 connections; however, the number of connections is also limited to the available main memory and texture memory on the graphics card (see how To limit the number of TCP connections).

- **Enable GPI**: GPIO device control. Enables frame accurate triggering of commands via GPI (general purpose interface). We support Sealevel GPI devices for GPI input.
- **MUX Isolated Port**: Port number for isolated sessions - no data shared (NLE).
- **MUX Shared Port**: Port number for shared sessions - shared data (NLE).
- **MUX Fixed Port**: Port number for fixed sessions - shared data, no reference counting (NLE).
- **Still Preview Port**: Port number for still preview.
- **Multi Touch Port**: Port number for listening to multi touch server.
- **Multi Touch IP Address**: IP address for listening to multi touch server.
- **Frame Accurate Viz Comm.**: Frame accurate commands via TCP (commands delayed by ring buffer and specified FAVC bias). Enable only for external control that supports special frame accurate command execution.
- **FAVC Bias**: Delay fields (in addition to ring buffer size) for frame accurate commands via TCP or GPI. This is the bias in frames for the commands if Frame Accurate Viz Command is turned on. Could be negative as well.
- **FAVC Field Dominance**: States when to handle the commands sent to the engine. E.g when set to “Odd retrace counter” all commands will be handled on odd fields. Not valid in progressive modes. Options are:
  - Always
  - Odd Retrace Counter: Execute commands at an odd retrace counter.
  - Even Retrace Counter: Execute commands at an even retrace counter.
  - Intern. Always/Extern. Odd: Execute commands internally always and externally at an odd retrace counter.
  - Intern. Always/Extern. Even: Execute commands internally always and externally at an even retrace counter.

Shared Memory

- **Multicast IP Address**: Sets the address for synchronizing distributed shared memory map without Viz Graphics Hub.
- **Multicast Port**: Synchronizes shared memory between all Viz Engines listening to the multicast.
- **UDP Port**: Sets the UDP listening port for shared memory input.
- **TCP Port**: Sets the TCP listening port for shared memory input.

- **Master Engine IP Address**: Sets the IP address of the master Viz Engine which holds the complete shared memory map (i.e. loaded during startup of Viz Engine).
- **Master Engine Port**: Sets the initializing port for the shared memory on startup (i.e. command port of the master Viz Engine).
- **Enable Master Poll**: When enabled Viz Engine will load the shared memory map from the defined master Viz Engine.

To limit the number of TCP connections

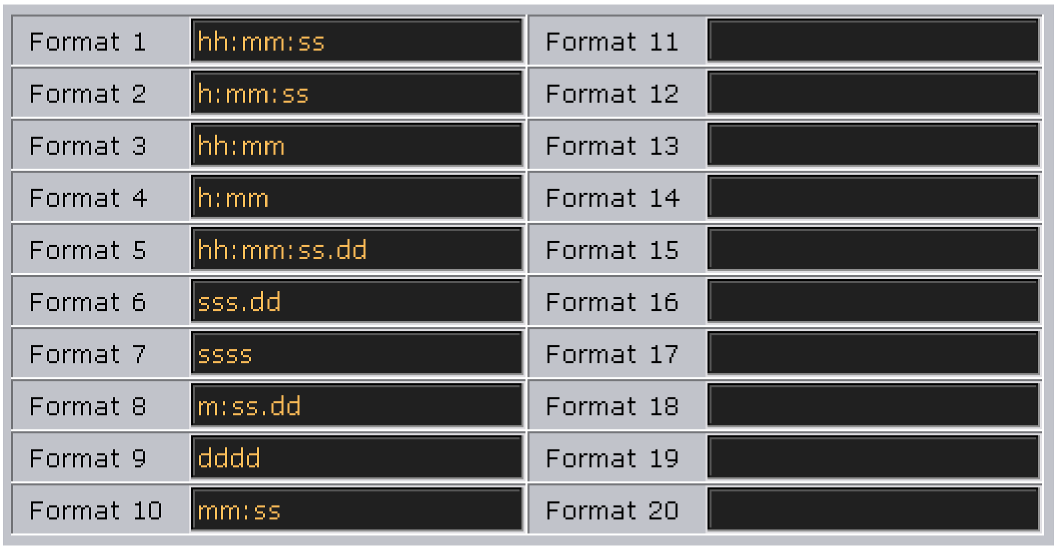
Viz 3.3 and later allows you to limit the number of TCP connections to Viz.

1. Open the Viz configuration file (i.e. *Viz-<hostname>-0-0.cfg*) found in the Viz3 program folder.
2. Under **SECTION COMMUNICATION** set `max_tcp_connections` to the number of TCP connections required.
    - If the number is set to 1 the first control application connecting to Viz will get exclusive control over Viz.

Note: This setting applies to the default port (6100) and the Multiplexing Ports.

10.4 Clock Formats

Figure 119: Clock Formats



Format 1	hh:mm:ss	Format 11	
Format 2	h:mm:ss	Format 12	
Format 3	hh:mm	Format 13	
Format 4	h:mm	Format 14	
Format 5	hh:mm:ss.dd	Format 15	
Format 6	sss.dd	Format 16	
Format 7	ssss	Format 17	
Format 8	m:ss.dd	Format 18	
Format 9	dddd	Format 19	
Format 10	mm:ss	Format 20	

In this section, twenty (20) various digital date and time formats can be set.

- **Format 1-20**: Sets and enables the clock formats that may be selected in Viz Artist during scene design.

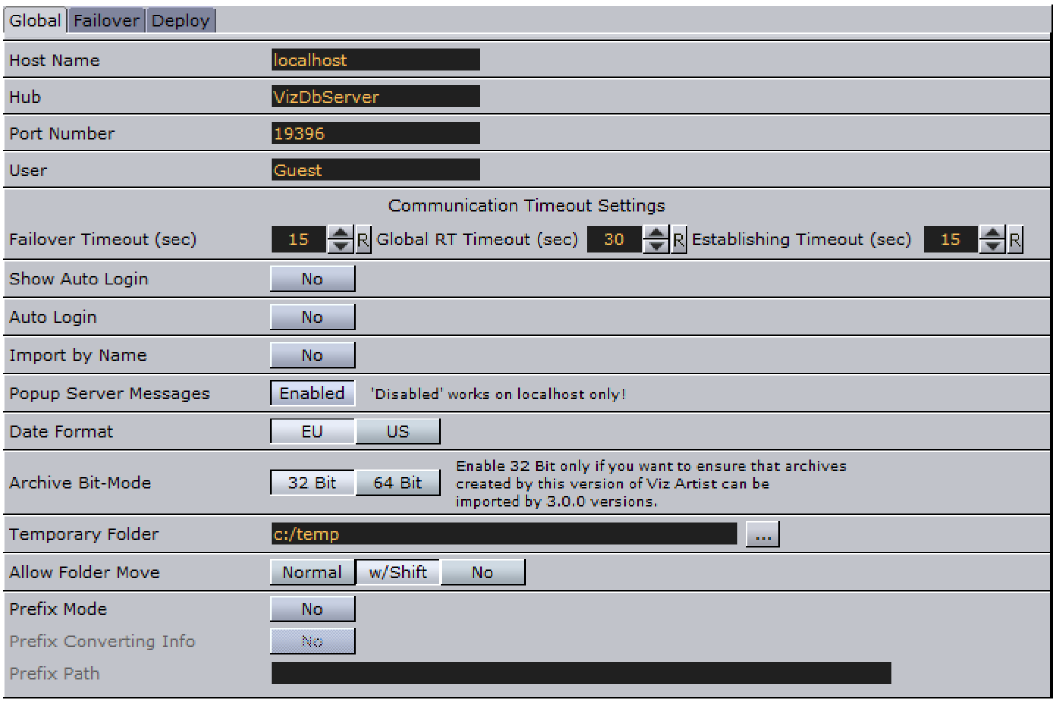
Tip: Add a font GEOM in Viz Artist to see how the clock formats can be used.

10.5 Database

The Database section has the following tabs for setting connections to e.g. Viz Graphics Hub, failover servers and deploy server.

- Global
- Failover
- Deploy

Global



Host Name: localhost
Hub: VizDbServer
Port Number: 19396
User: Guest
Communication Timeout Settings
Failover Timeout (sec): 15 R | Global RT Timeout (sec): 30 R | Establishing Timeout (sec): 15 R
Show Auto Login: No
Auto Login: No
Import by Name: No
Popup Server Messages: Enabled 'Disabled' works on localhost only!
Date Format: EU US
Archive Bit-Mode: 32 Bit 64 Bit Enable 32 Bit only if you want to ensure that archives created by this version of Viz Artist can be imported by 3.0.0 versions.
Temporary Folder: c:/temp ...
Allow Folder Move: Normal w/Shift No
Prefix Mode: No
Prefix Converting Info: No
Prefix Path

In this section, the Viz Graphics Hub database connection settings are configured.

- **Host Name**: Sets the name of the Viz Graphics Hub naming service. The naming service will always be a one to one map to the hostname of the machine running Viz Graphics Hub.
- **Hub**: Sets Viz Graphics Hub server name.
- **Port Number**: Sets the listener port number for Viz Graphics Hub. The default port number is 19396, and should normally not be changed..
- **User**: Sets the default user.
- **Communication Timeout Settings**:
  - **Failover Timeout (sec)**: Sets the maximum waiting time before a fail over is initiated from the main to the replication Viz Graphics Hub.
  - **Global RT Timeout (sec)**: Sets the maximum response time for any request to Viz Graphics Hub.
  - **Establishing Timeout (sec)**: Sets the maximum waiting time to establish a connection to Viz Graphics Hub.
- **Show Auto Login**: Enables (Yes) the user to check the Auto login check-box in the Viz Graphics Hub login window. When disabled (No), the Auto login check-box is hidden.

- **Auto Login**: Enables or disables automatic login to Viz Graphics Hub. This will effectively disable the login screen for Viz Engine and Viz Artist.
- **Import by Name**: Set to **Yes** to check for existing objects by name rather than by UUID. Set to **No** to check by UUID.
- **Popup Server Messages**: Enables or disables popup server messages. Disabled only works on local host.
- **Date Format**: Sets the date format to EU (DD.MM.YYYY 13:54) or US (MM/DD/YY 01:54).
- **Archive Bit-Mode**: Sets the bit-mode in which the archive will be saved.

Note: For compatibility, 32-bit should be enabled if scenes will be imported to Viz Artist 3 versions prior to build 2310.

- **Temporary Folder**: Sets the location of the temporary folder that is used to save temporary plug-in data. The folder can be a local drive, mapped drive or a Universal Naming Convention (UNC) path.

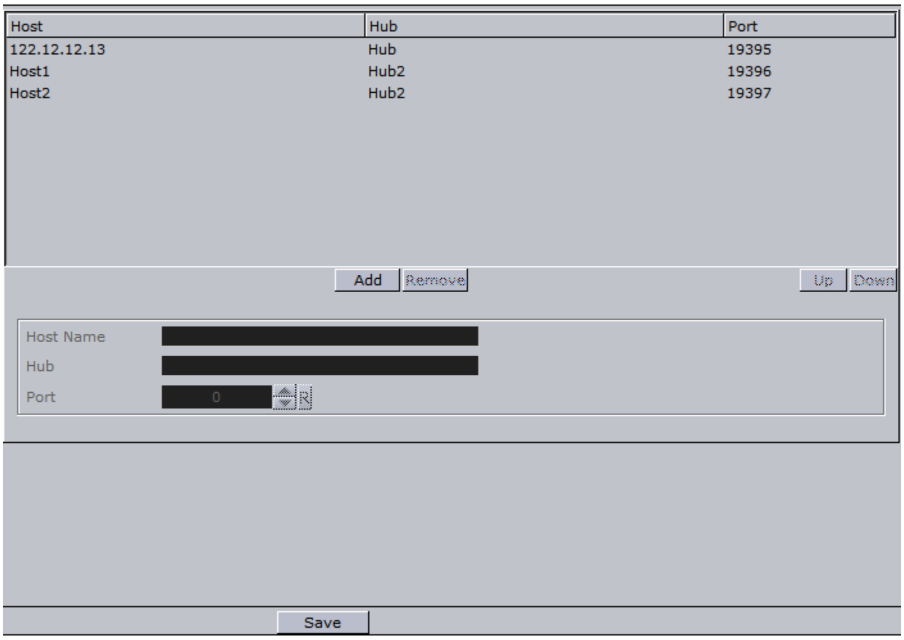
IMPORTANT! Make sure the Temporary Folder is configured with read and write access rights.

- **Allow Folder Move**: Allows or restricts the users ability to move/organize projects and folders in Viz Graphics Hub. Options are Normal, with Shift or No. Default is Normal which allows the user to freely move projects and folder.

The following prefix settings are needed if you are using an external control application that sends commands containing certain path locations, but where the path of your files is a different one on your Viz Graphics Hub (e.g. because you deployed them to a specific location).

- **Prefix Mode**: Activates the prefix mode.
- **Prefix Converting Info**: Shows the prefixed (final) paths in the console (this behaves like the Viz Artist 2.x debug mode).
- **Prefix Path**: Contains the prefix path string that is used for incoming commands containing path parameters.

Failover



Host	Hub	Port
122.12.12.13	Hub	19395
Host1	Hub2	19396
Host2	Hub2	19397

Host Name

Hub

Port 0

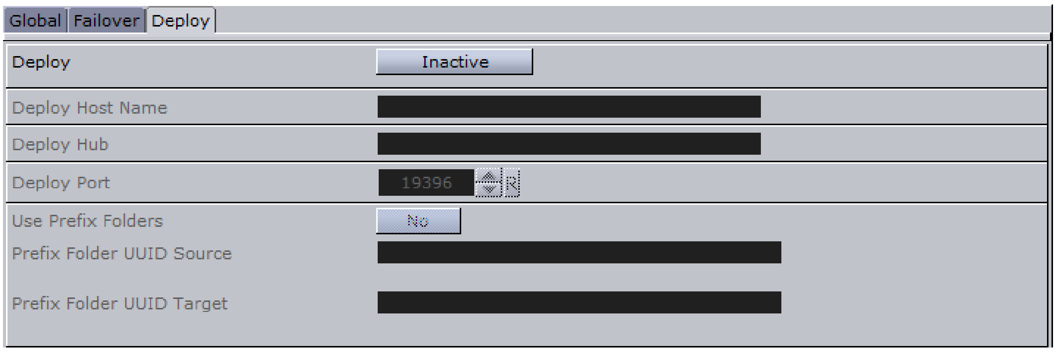
To add redundant servers to the failover list

1. Enter the host name.
2. Enter the Graphics Hub.
3. Enter the port number.
4. Click **Add**.

Tip: You can raise and/or lower a database's priority in the event of failover using the **Up** and **Down** buttons in the database failover list.

5. When finished all modifications, click **Save**.

Deploy



Deploy Inactive

Deploy Host Name

Deploy Hub

Deploy Port 19396

Use Prefix Folders No

Prefix Folder UUID Source

Prefix Folder UUID Target

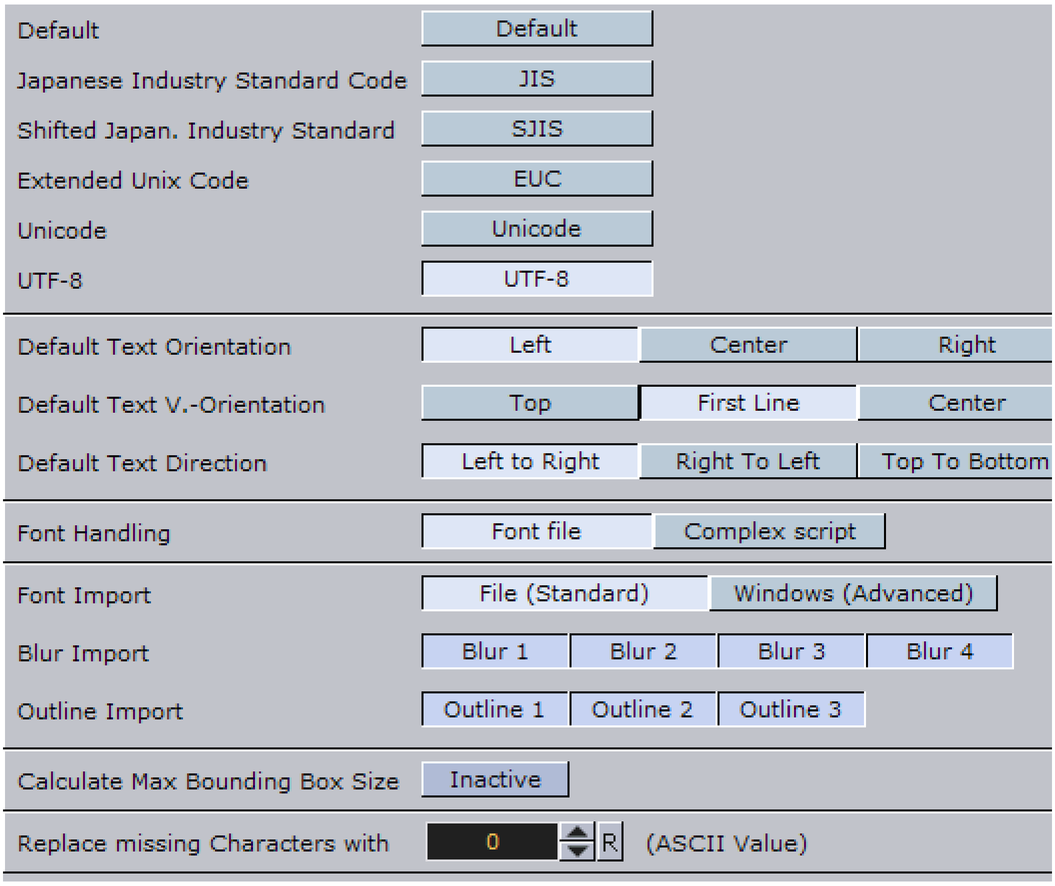
- **Deploy**: Enable or disable deployment of graphics data. When activated you will get a Deploy tab/button in the Viz Artist user interface.
- **Deploy Host Name**: Sets the hostname of the Viz Graphics Hub machine you deploy your files to.
- **Deploy Hub**: Sets the name of the Viz Graphics Hub instance you deploy your files to.

- **Deploy Port**: Sets your Viz Graphics Hub's listener port number.
- **Use Prefix Folders**: Enables you to define a prefix source and target.
- **Prefix Folder Uuid Source**: Sets the source folder's Uuid of the Viz Graphics Hub you deploy from.
- **Prefix Folder Uuid Target**: Sets the target folder's Uuid of the Viz Graphics Hub you deploy to.

Note: To use correct Uuids you must have both Viz Graphics Hub systems running.

10.6 Font Options

Figure 120: Font Options



Default	Default
Japanese Industry Standard Code	JIS
Shifted Japan. Industry Standard	SJIS
Extended Unix Code	EUC
Unicode	Unicode
UTF-8	UTF-8
Default Text Orientation	Left, Center, Right
Default Text V.-Orientation	Top, First Line, Center
Default Text Direction	Left to Right, Right To Left, Top To Bottom
Font Handling	Font file, Complex script
Font Import	File (Standard), Windows (Advanced)
Blur Import	Blur 1, Blur 2, Blur 3, Blur 4
Outline Import	Outline 1, Outline 2, Outline 3
Calculate Max Bounding Box Size	Inactive
Replace missing Characters with	0 R (ASCII Value)

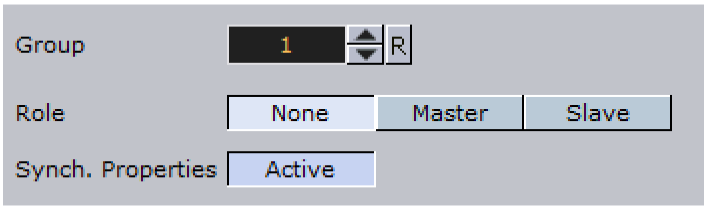
In this section, the font encoding can be configured.

- **Font Options: Sets the font encoding. Available options are:**
  - **Default**: Sets the font encoding to single character interpretation (limited to 255).
  - **Japanese Industry Standard Code (JIS)**: Sets Japanese industry standard code character encoding.
  - **Shifted Japanese Industry Standard Code (SJIS)**: Sets the newer Shift JIS character encoding standard which sets aside certain character codes to signal the start of a two-character sequence.

    - **Extended Unix Code (EUC)**: Sets Extended Unix Code (EUC) character encoding that is a multi byte character encoding system used primarily for Japanese, Korean, and simplified Chinese.
    - **Unicode**: Sets the Unicode character encoding where every two characters are interoperated as one (not widely used).
    - **UTF-8**: Sets UTF-8 (8-bit UCS/Unicode Transformation Format) character encoding that is a variable-length character encoding for Unicode.
- **Default Text Orientation**: Sets the default horizontal text orientation. Available options are; Left, Center and Right.
- **Default Text V. Orientation**: Sets the default vertical text orientation. Available options are; Top, First Line, Center and Bottom.
- **Default Text Direction**: Sets the default text direction. Available options are; Left to Right, Right to Left and Top to Bottom.
- **Font Handling**: Sets the font handling. Available options are Font file or Complex script.
    - **Font file**: Sets the font kerning to Font file that is mainly used for languages written from left to right.
    - **Complex script**: Sets the font kerning to Complex script. Complex script is mainly used for languages written from right to left and/or that one character can be are composed of one or several Glyphs.
- **Font Import**: Sets the Font import. Available options are; File (Standard) and Windows (Advanced).
    - **File (Standard)**: Imports and stores fonts on the database.
    - **Windows (Advanced)**: Stores only the font name on the database. For this to work the font must be installed on the Windows system where Viz Engine resides.
- **Blur Import**: Enables blur levels for fonts used in Viz 2.x scenes. These options will make sure that fonts that are imported to Viz 3.x will be recreated with all blur levels used in Viz 2.x. Available options are; Blur 1-4.
- **Outline Import**: Enables outline levels for fonts used in Viz 2.x scenes. These options will make sure that fonts that are imported to Viz 3.x will be recreated with all outline levels used in Viz 2.x. Available options are; Outline 1-3.
- **Calculate Max Bounding Box Size**: Enables Viz 3.x to calculate the bounding boxes as they were calculated in Viz 2.x. In Viz 3.x a text object's bounding box height will increase if a capital character is entered (e.g. an umlaut (double dots)). In viz 2.x the bounding box height was always the same and independent of the characters in the text object. Available options are Active and Inactive. Default is Inactive (false).
- **Replace missing Characters with**: Replaces a missing font character in a font file with a default font character. The Unicode value refers to the decimal value of the replacement character in the Unicode table (valid values are 0-65553). Normal usage would select a * (42) or _ (95).

10.7 Global Input

Figure 121: Global Input



Group 1 R

Role None Master Slave

Synch. Properties Active

The Global Input settings influence the generation and handling of *Six Degrees of Freedom* (6DoF) messages that can be distributed to several Viz Engines.

6DoF is used when working in 3D space in combination with special input devices such as a mouse. A mouse uses two coordinates (xy) which Viz is able to translate into three coordinates (xyz) based on a grid.

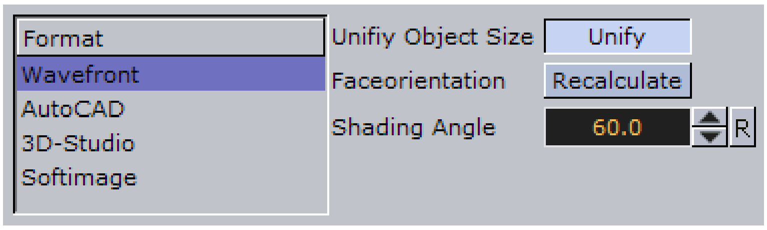
- Group: Defines which multicast group the generated or received messages belong to. If more than one group is to be defined, a unique Group number must be set for each group.
- Role: Defines how messages are generated and processed. Alternatives are; None, Master and Slave.
  - None: Messages are only generated and processed on the local Viz Engine.
  - Master: A Master creates messages for himself and the defined group.
  - Slave: A Slave reads and processes 6DOF messages but is not allowed creating them.
- Synch. Properties: Activates/deactivates synchronization of Viz Engine scene properties.

To synchronize multiple Viz Engines

1. Start Viz Config on all involved render machines.
2. Set the same Group ID for all engines, and set Synch. Properties to Active.
3. Save and close all Viz Configs.
4. Open the Control Panel on all render machines, and make sure that you deactivate all unused network connections.
   - Viz Engine always uses the first network connection setup by the Windows operating system.
   - Synchronized engines work within the same network segment only because it is using multicasts, hence, it is important to use the right connection.
   - You can also determine the first connection by setting a manual metric in Windows: http://support.microsoft.com/kb/299540.
5. Start all Viz Engines again.
6. Create a simple test scene with a geometry and the Synchronized Properties plug-in (Built Ins > Container > Global) on the same container.
7. Save the scene and open it on all involved engines.
8. Move the geometry on one engine.
   - All the other engines will show the same object movement.

10.8 Import 3D Options

Figure 122: Import 3D Options



In this section, parameters influencing the import of 3D objects can be configured. There are four different formats; Wavefront, AutoCAD, 3D-Studio, and Softimage.

The Wavefront, AutoCAD, and 3D-Studio formats have three available controls:

- **Unify Object Size**: If enabled, all vertices are recalculated during import in a way that the object centre is moved to the origin (0,0,0), and the size of the object is 100 cm in its largest extent. If disabled, all vertices retain their values as defined in the original file. An object could appear invisible in a Viz Artist 3 scene because the object is translated a lot from the origin or is scaled up or down a lot. It could be necessary to deactivate the unification to be able to recombine several separately imported objects that must keep their size and relative position.
- **Face Orientation**: Polygonal 3D models often do not have a consistent face orientation, but for performance reasons, the Viz Artist 3 renderer expects that all faces of an object point to the same direction. If enabling this function, Viz Artist 3 tries to rearrange the orientation of the object faces during import.
- **Shading Angle**: If the 3D object has no normal vector information, Viz Artist automatically recalculates the normal’s from the geometry to make lighting possible. This recalculation is influenced by the shading angle, which acts as a threshold between sharp and soft edges.

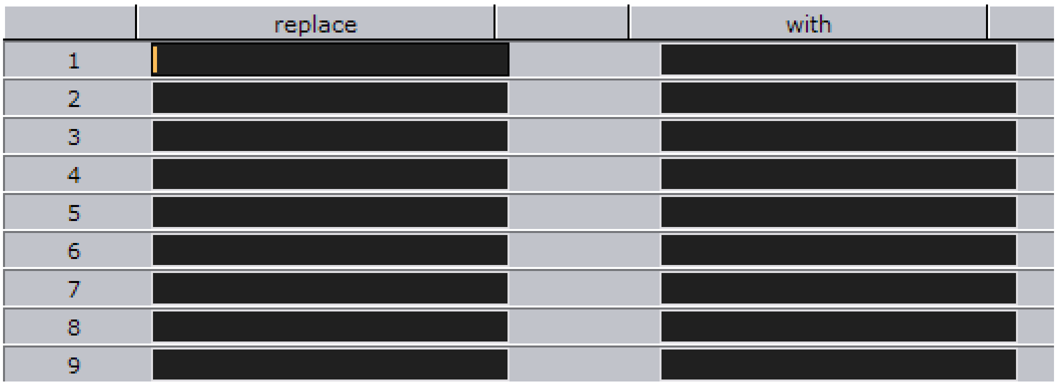
Note: A shading angle value of 60 means that an edge between two faces is considered to be a soft edge for angles below 60 and a sharp edge above this level. 60 is the default shading angle.

The Softimage format has two available controls:

- **Classic Scaling**: Changes Softimage rotation order to conform to the internal rotation order of Viz.
- **Enable Animations**: Allows import of saved Softimage animations.

10.9 Image File Name Convert

Figure 123: Image File Name Convert

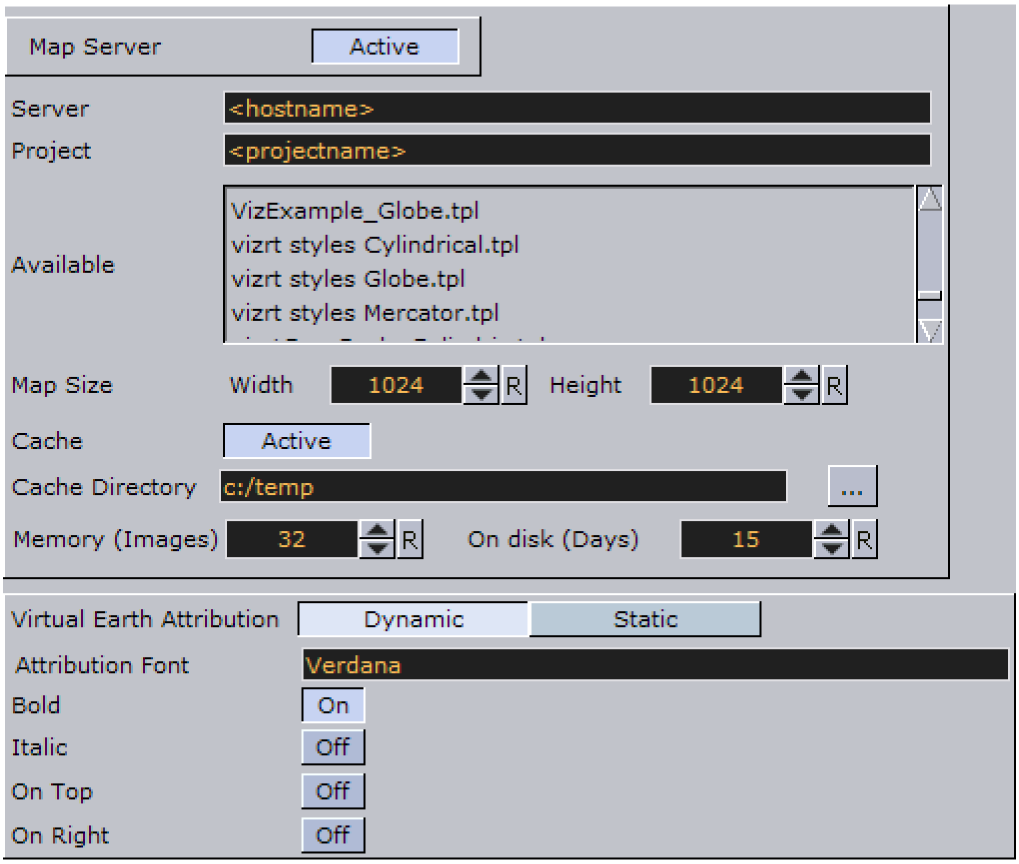


	replace	with
1		
2		
3		
4		
5		
6		
7		
8		
9		

In this section, nine (9) different replacement configurations can be set for image file names. For example; If an image file name starts with the string *Replace* this part of the file name can be replaced by the string *With*.

- **Replace**: Defines the string that will be replaced.
- **With**: Defines the replacement string.

10.10 Maps



Map Server: Active

Server: <hostname>

Project: <projectname>

Available:
VizExample_Globe.tpl
vizrt styles Cylindrical.tpl
vizrt styles Globe.tpl
vizrt styles Mercator.tpl

Map Size: Width 1024 R Height 1024 R

Cache: Active

Cache Directory: c:/temp ...

Memory (Images): 32 R On disk (Days): 15 R

Virtual Earth Attribution: Dynamic Static

Attribution Font: Verdana

Bold: On

Italic: Off

On Top: Off

On Right: Off

- **Map Server**: Enables or disables Viz World Server (WoS) connection for Viz World Client (WoC).
- **Server**: Sets the Viz World Server host.
- **Project**: Sets the default map project that will be opened with the client application.
- **Available**: Lists all available Viz World Server projects.
- **Map size**: Sets the default map size that will be used with the client application.
- **Cache**: Enables caching of maps for faster preview and fetching of maps. Especially useful for journalists and operators using Newsroom Component and Viz Trio respectively.
- **Cache Directory**: Sets the cache directory for cached maps which can be a local drive, mapped drive or a Universal Naming Convention (UNC) path.

IMPORTANT! Make sure the Cache Directory folder is configured with read and write access rights.

- **Memory (Images)**: Sets the number of images to keep in memory.
- **On disk (Days)**: Sets the number of days to save images on disk.
- **Virtual Earth Attribution**: Adds an attribution to the map. Alternatives are; Static and Dynamic.
  - **Dynamic**: Displays the attribution when a Microsoft Virtual Earth image is in view and disappears when the image is out of view.

- **Static**: Displays the attribution as long as there is a Microsoft Virtual Earth image in the scene.
- **Attribution Font**: Sets the font for the Virtual Earth attribution.
- **Bold**: Sets the Virtual Earth attribution font to **bold**.
- **Italic**: Sets the Virtual Earth attribution font to *italic*.
- **On Top**: Places the Virtual Earth attribution image to the top in the screen. Default is bottom.
- **On Right**: Places the Virtual Earth attribution image to the right in the screen. Default is left.

10.11 Matrox

In Viz Artist 3.3 it is possible to assign Matrox' in and out channels to Viz Artist's in and out channels.

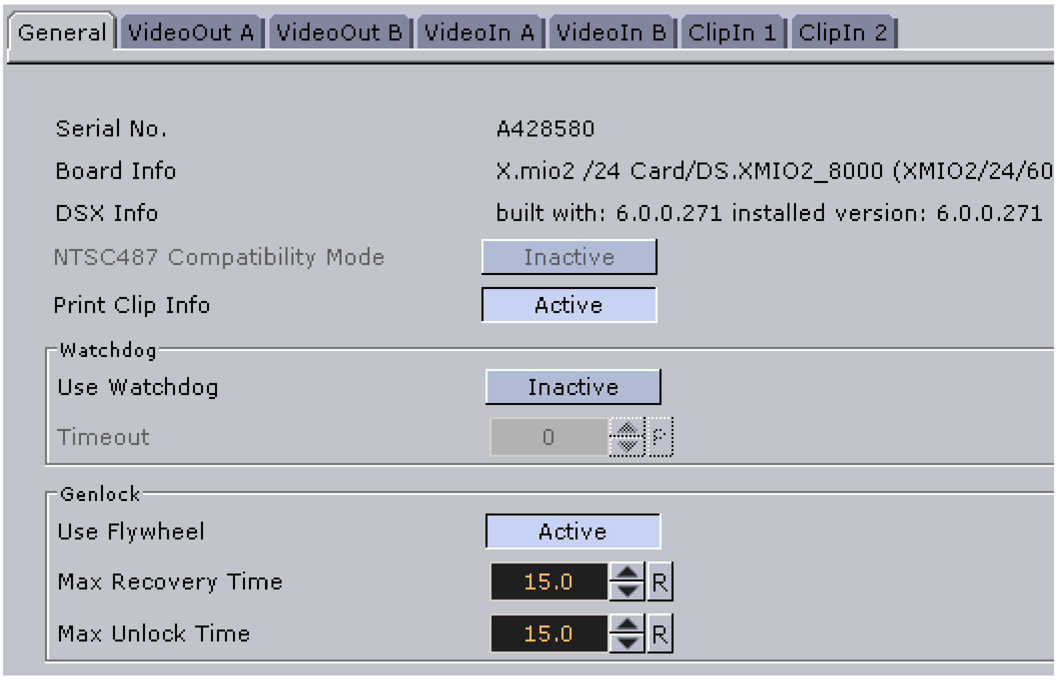
The GUI shows a tab for General and, according to the installed hardware, tabs for VideoOut, VideoIn, and ClipIn. Depending on your installed hardware there might be different tabs available.

This section contains information on the following topics:

- General
- VideoOut
- VideoIn
- ClipIn
- ClipOut

10.11.1 General

Figure 124: General Matrox configuration



Serial No. A428580

Board Info X.mio2 /24 Card/DS.XMIO2_8000 (XMIO2/24/60

DSX Info built with: 6.0.0.271 installed version: 6.0.0.271

NTSC487 Compatibility Mode Inactive

Print Clip Info Active

Watchdog

Use Watchdog Inactive

Timeout 0

Genlock

Use Flywheel Active

Max Recovery Time 15.0

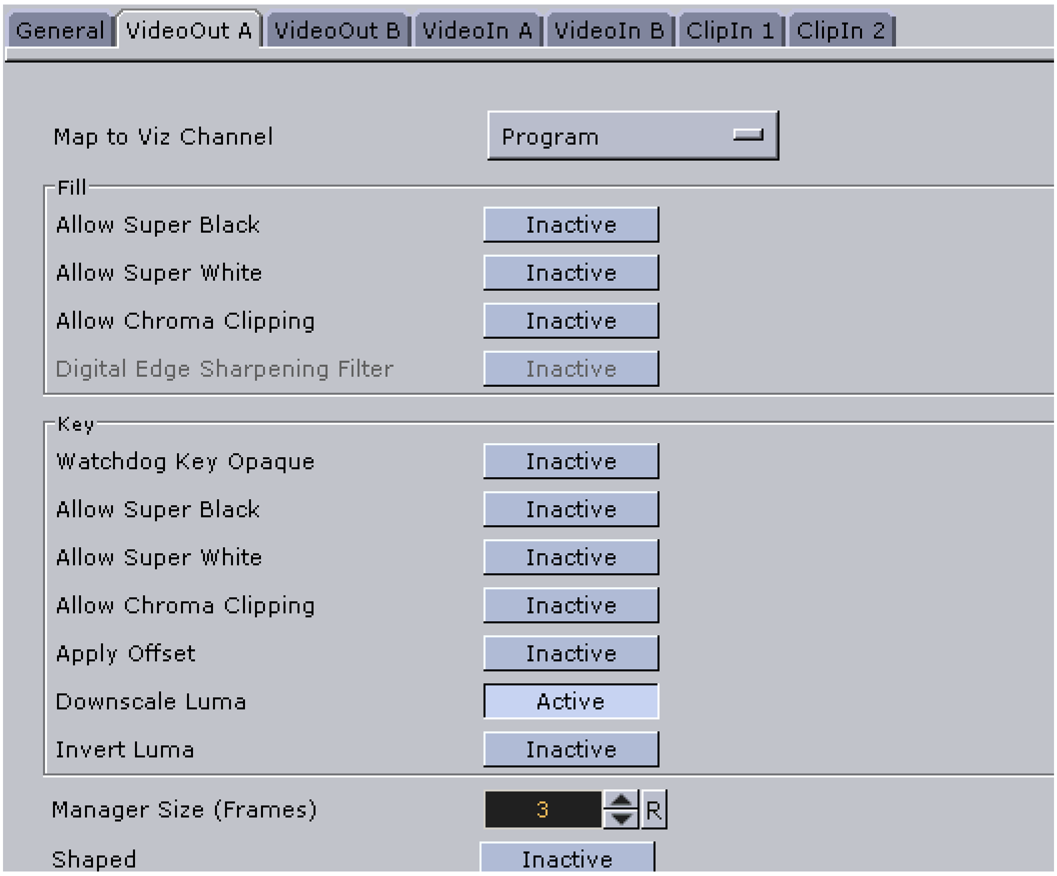
Max Unlock Time 15.0

The General tab shows information about the installed hardware.

- **Serial No.**: Shows the serial number of the installed Matrox board.
- **Board Info**: Shows the model and type of the Matrox board.
- **DSX Info**: Shows the software version and driver version.
- **NTSC487 Compatibility Mode**: *For future use*. Enables the NTSC 487 compatibility mode. Default mode is Inactive.
- **Print Clip Info**: When activated this setting enables printing of clip information to the console; however, such information may cause the render loop to stall. Default mode is Inactive.
- **Use Watchdog**: When activated this setting enables the Matrox X.mio watchdog feature. It defines what kind of key is produced when the watchdog takes over control. Default mode is Inactive. When the watchdog is enabled, turning off the video out will activate the hardware bypass after a given timeout (see next setting).
- **Timeout**: Defines the time in milliseconds until the watchdog takes over control. This value should not be smaller than the time of two fields/frames. Default value is 999 milliseconds.
- **Use Flywheel**: When activated the Matrox card adopts a tracking mode if the genlock signal is interrupted or lost that maintains the signal frequency until the source genlock signal is regained. Default mode is Active.
- **Max Recovery Time**: Represents the time in milliseconds (ms) provided to the flywheel to attempt to regain the genlock before an abrupt jump to the locked state is performed. Default value is 15.
- **Max Unclock Time**: Represents the time in milliseconds (ms) provided to the flywheel to remain in the unlocked state before switching to the free running state. Default value is 15.

10.11.2 VideoOut

Figure 125: VideoOut configuration



Map to Viz Channel	Program
Fill	
Allow Super Black	Inactive
Allow Super White	Inactive
Allow Chroma Clipping	Inactive
Digital Edge Sharpening Filter	Inactive
Key	
Watchdog Key Opaque	Inactive
Allow Super Black	Inactive
Allow Super White	Inactive
Allow Chroma Clipping	Inactive
Apply Offset	Inactive
Downscale Luma	Active
Invert Luma	Inactive
Manager Size (Frames)	3
Shaped	Inactive

The VideoOut tab displays a simple and an Advanced Properties and Parameters view. The simple view shows the mapped Viz Artist channel. In the advanced view every setting of the output channel can be controlled. The switch between the simple and the advanced view is done with the small black triangle on the right hand side.

- **Map to Viz Channel**: Decides which Viz Artist video out channel is mapped onto this Matrox video out channel. The drop-down gives a choice between the available channels. Only the channels not already taken are shown.
  - **Unused**: Do not use this Matrox channel for output.
  - **Program**: Plays out the Program signal of Viz Artist.
  - **Preview**: Plays out the Preview signal of Viz Artist.

On a single channel configuration VideoOutA is usually mapped to Program and VideoOutB to Preview, whereas on a dual channel configuration usually the first channel maps VideoOutA to Program and the second channel VideoOutB to Program.

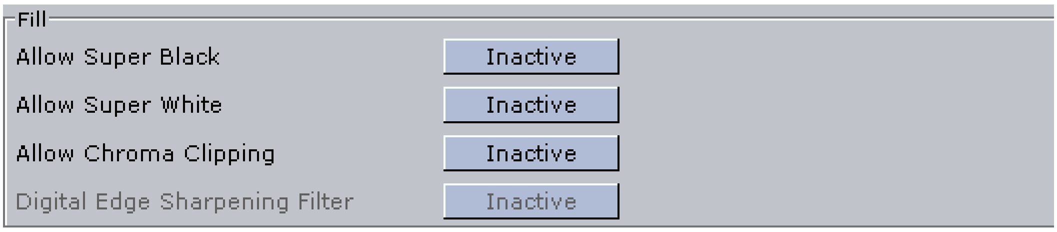
This section also contains information on the following topics:

- Advanced Properties and Parameters
  - VideoOut fill settings
  - VideoOut key settings

- Manager size, shaped and repeat mode settings
- VideoOut VBI settings

Advanced Properties and Parameters

Figure 126: VideoOut fill settings



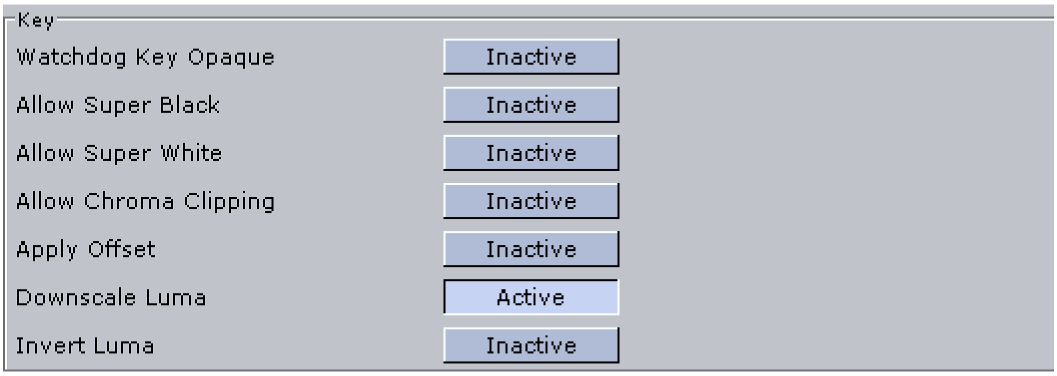
Fill

Allow Super Black	Inactive
Allow Super White	Inactive
Allow Chroma Clipping	Inactive
Digital Edge Sharpening Filter	Inactive

Group for fill component of the video output.

- **Allow Super Black**: Determines whether or not to clip an output video signal that is under 7.5 IRE units. Default mode is Inactive.
- **Allow Super White**: Determines whether or not to clip an output video signal that is over 100 IRE units. Default mode is Inactive.
- **Allow Chroma Clipping**: Determines whether or not to clip over-saturated chroma levels in the active portion of the output video signal. Default mode is Inactive.
- **Digital Edge Sharpening Filter**: Applies an edge sharpening filter to digital output video. Default mode is Inactive. SD configurations only.

Figure 127: VideoOut key settings



Key

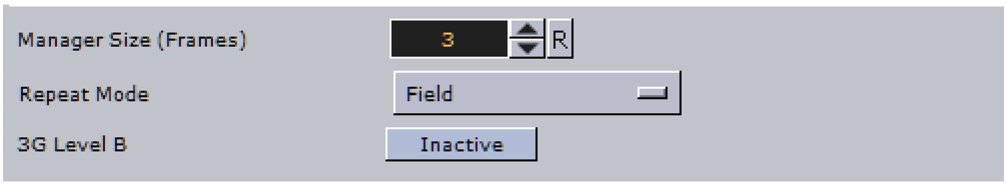
Watchdog Key Opaque	Inactive
Allow Super Black	Inactive
Allow Super White	Inactive
Allow Chroma Clipping	Inactive
Apply Offset	Inactive
Downscale Luma	Active
Invert Luma	Inactive

Group for key component of the video output.

- **Watchdog Key Opaque**: Specifies if the output key must be opaque or transparent when the watchdog unit activates. Default mode is Inactive.
- **Allow Super Black**: Determines whether or not to clip an output video signal that is under 7.5 IRE units. Default mode is Inactive.
- **Allow Super White**: Determines whether or not to clip an output video signal that is over 100 IRE units. Default mode is Inactive.
- **Allow Chroma Clipping**: Determines whether or not to clip over-saturated chroma levels in the active portion of the output video signal. Default mode is Inactive.
- **Apply Offset**: Applies an offset to the luminance values such that the inverted result still falls within the 16-235 range. Default mode is Inactive.
- **Downscale Luma**: Compresses the luminance range of the output key signal from 0-255 to 16-235. Default mode is Active.

- **Invert Luma**: Inverts the luminance part of the output key signal (inverts the key). Default mode is Inactive.

Figure 128: Manager size, shaped and repeat mode settings



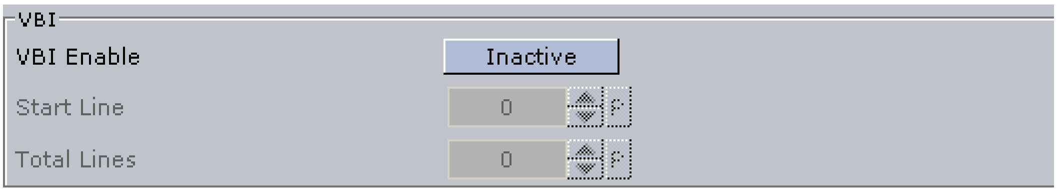
Manager Size (Frames) 3

Repeat Mode Field

3G Level B Inactive

- **Manager Size (frames)**: Sets the number of frames available in the on-board memory for output. A too high value may cause memory problems on the Matrox card. Default value is 3.
- **Repeat Mode**: Defines the way the output should be repeated if Viz Engine is stalled and does not update the output. Default mode is Field. Available modes are:
  - **None**: Does not repeat. Output goes black.
  - **Field**: Repeats the last played field.
  - **Frame**: Repeats the last played frame.
- **3G Level B**: Activates Level B for 3G mode in 1080p 50/60/60M (default mode is Level A).

Figure 129: VideoOut VBI settings



VBI

VBI Enable Inactive

Start Line 0

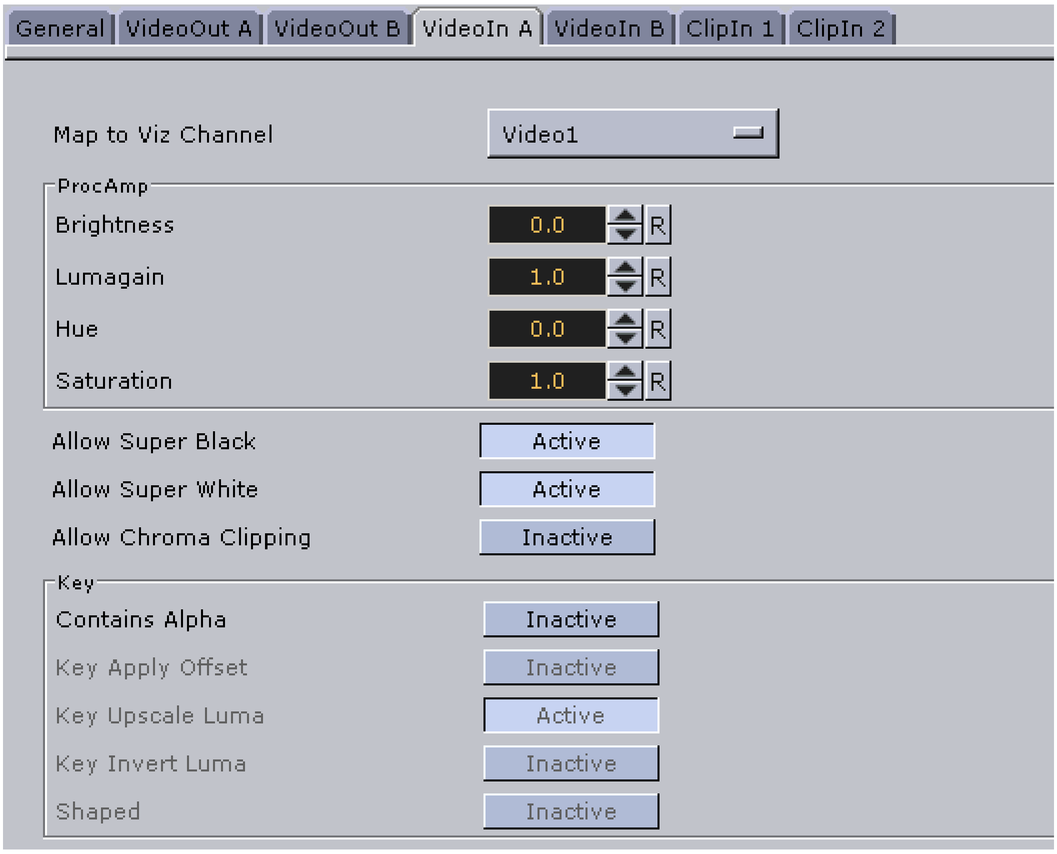
Total Lines 0

Group for Vertical Blanking Interval (VBI) related settings.

- **VBI Enable**: Enable/disable VBI output.
- **Start Line**: Defines at which line on the output the VBI section should start. Default value is 0 (Off). The minimum VBI values are:
  - NTSC: 7
  - PAL: 6
  - 720p: 7
  - 1080i: 6
- **Total Lines**: Defines how many lines the VBI section in the output should have in total. Default value is 0 (Off). The maximum VBI values are:
  - NTSC: 32
  - PAL: 34
  - 720p: 19
  - 1080i: 30

10.11.3 VideoIn

Figure 130: VideoIn configuration



Map to Viz Channel: Video1

ProcAmp

Brightness	0.0
Lumagain	1.0
Hue	0.0
Saturation	1.0

Allow Super Black	Active
Allow Super White	Active
Allow Chroma Clipping	Inactive

Key

Contains Alpha	Inactive
Key Apply Offset	Inactive
Key Upscale Luma	Active
Key Invert Luma	Inactive
Shaped	Inactive

The VideoIn tab comes with a simple and an Advanced Properties and Parameters view. The simple view shows only the mapped Viz Artist channel. In the advanced view every setting of the input channel can be controlled. The switch between the simple and the advanced view is done with the small black triangle on the right hand side.

- **Map to Viz Channel**: decides which Viz Artist video in channel is mapped onto this Matrox video in channel. The drop-down gives a choice between the available channels. Only the channels not already taken are shown.
  - **Unused**: Do not use this Matrox channel for video input
  - **Video1**: Captured input is available in Viz Artist Video1.

On a single channel configuration VideoInA is usually mapped to Video1 and VideoInB to Video2 and so on whereas on dual channel configuration usually the first channel maps VideoInA to Video1 and the second channel maps VideoInB to Video1. In this case both, the first and the second channel, have one video input configured, namely Video1.

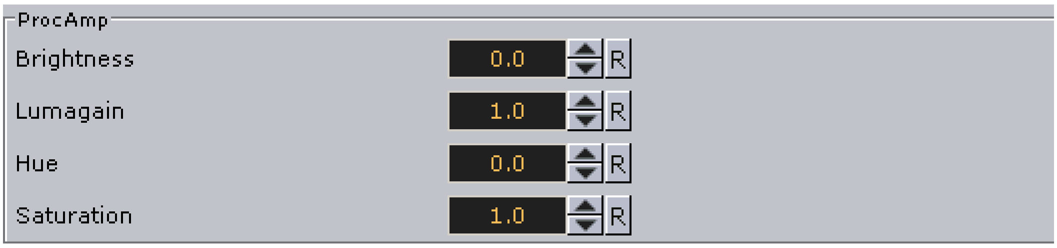
This section also contains information on the following topics:

- Advanced Properties and Parameters
  - VideoIn ProcAmp settings
  - VideoIn super black, white and chroma clipping settings
  - VideoIn key settings

- VideoIn video settings
- VideoIn VBI settings
- VideoIn audio settings

Advanced Properties and Parameters

Figure 131: VideoIn ProcAmp settings



ProcAmp

Brightness 0.0 R

Lumagain 1.0 R

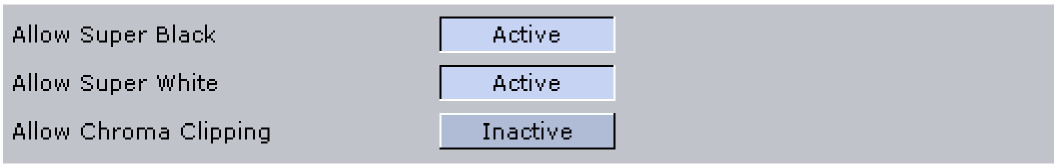
Hue 0.0 R

Saturation 1.0 R

Group for signal related settings.

- **Brightness**: Sets the relative offset on the luminance component of the incoming video (min./max. values are dynamic and determined by the hardware). Default value is 0 (Off).
- **Lumagain**: Sets the gain on the luminance component of the incoming video (min./max. values are dynamic and determined by the hardware). Default value is 0 (Off).
- **Hue**: Sets the color shift on the chrominance component of the incoming video (min./max. values are dynamic and determined by the hardware). Default value is 0 (Off).
- **Saturation**: Sets the gain on the chrominance component of the incoming video (min./max. values are dynamic and determined by the hardware). Default value is 1 (On).

Figure 132: VideoIn super black, white and chroma clipping settings



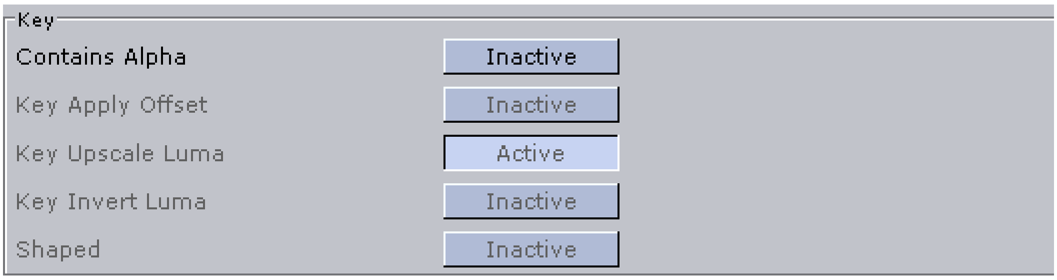
Allow Super Black Active

Allow Super White Active

Allow Chroma Clipping Inactive

- **Allow Super Black**: Determines whether or not to clip an input video signal that is under 7.5 IRE units. Default mode is Active.
- **Allow Super White**: Determines whether or not to clip an input video signal that is over 100 IRE units. Default mode is Active.
- **Allow Chroma Clipping**: Determines whether or not to clip over-saturated chroma levels in the active portion of the input video signal. Default mode is Inactive.

Figure 133: VideoIn key settings



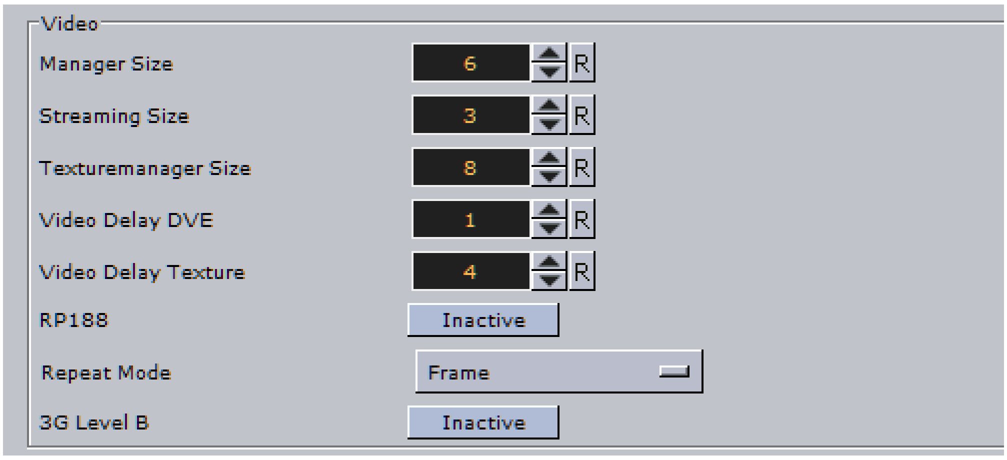
Key

Contains Alpha	Inactive
Key Apply Offset	Inactive
Key Upscale Luma	Active
Key Invert Luma	Inactive
Shaped	Inactive

Group for key component of the video input.

- **Contains Alpha**: Enables/Disables use of alpha component. This switch is only enabled on input channels where capture with alpha is supported.
- **Key Apply Offset**: Enables/Disables key offset.
- **Key Upscale Luma**: Enables/Disables key upscale luma.
- **Key Invert Luma**: Enables/Disables key invert luma.
- **Shaped**: Enables/Disables capture in shaped format.

Figure 134: VideoIn video settings



Video

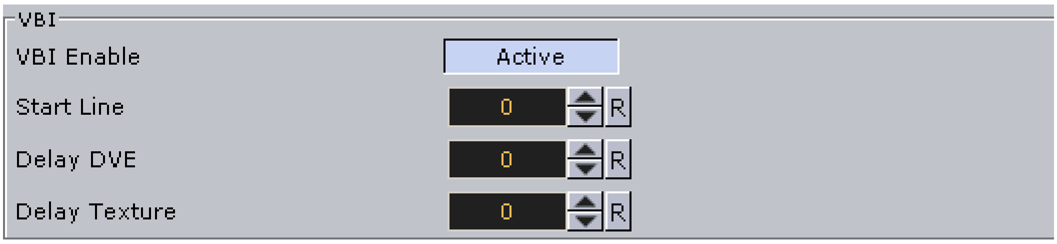
Manager Size	6
Streaming Size	3
Texturemanager Size	8
Video Delay DVE	1
Video Delay Texture	4
RP188	Inactive
Repeat Mode	Frame
3G Level B	Inactive

Group for fill component of the video input.

- **Manager Size:** Sets the number of frames available in the on-board memory for capturing. This value is influenced by the input delays specified in the parameters below and will automatically be adjusted if it is too low. A too high value may cause memory problems on the Matrox card. Default value is 6.
- **Streaming Size**: *Not in use*.
- **Texturemanager Size**: Defines the size of the texture buffer in frames. Default value is 8.
- **Video Delay DVE**: Sets the number of frames the live input should be delayed before it can be used as a DVE layer. Similar to the *LoopthroughDelay* for Targa boards. Default value is 0 (Off).
- **Video Delay Texture**: Sets the number of frames the live input should be delayed before it can be used as a texture in the scene. Default value is 4.
- **RP188 Enable**: Enables capturing of SMPTE RP 188 extra information such as LTC and VITC. Default is Inactive.
- **Repeat Mode**: Determines the behavior of the video input in case of capture drops. The drop-down gives a choice between
  - **None**: Does not repeat. Input goes black.

    - **Field**: Repeats the last field.
    - **Frame**: Repeats the last frame.
- **3G Level B**: Activates Level B for 3G mode in 1080p 50/60/60M (default mode is Level A).

Figure 135: VideoIn VBI settings



VBI

VBI Enable: Active

Start Line: 0

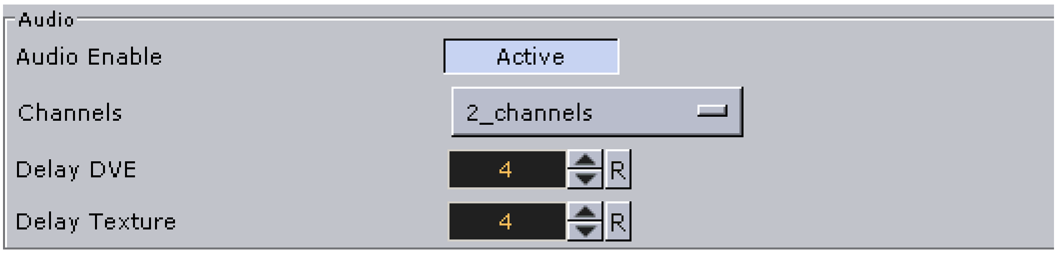
Delay DVE: 0

Delay Texture: 0

Group for Vertical Blanking Interval (VBI) related settings.

- **VBI Enable**: When set to Active this setting will enable VBI for this channel. Default is Inactive. Note that if the input resolution is different from the output resolution it is impossible to activate VBI since VBI can only be inserted if the resolutions are the same.
- **Start Line**: Defines at which line on the input the VBI section will start. Default value is 0 (Off). The minimum VBI values are (as for VideoOut):
    - NTSC: 7
    - PAL: 6
    - 720p: 7
    - 1080i: 6
- **Delay DVE**: Sets the number of frames the VBI should be delayed before the clip can be used in DVE mode. Default for Viz 3.3 is 0 (Off). For Viz 3.2 the value is 1.
- **Delay Texture**: Sets the number of frames the VBI should be delayed before the clip can be used in texture mode. Default is 0 (Off).

Figure 136: VideoIn audio settings



Audio

Audio Enable: Active

Channels: 2_channels

Delay DVE: 4

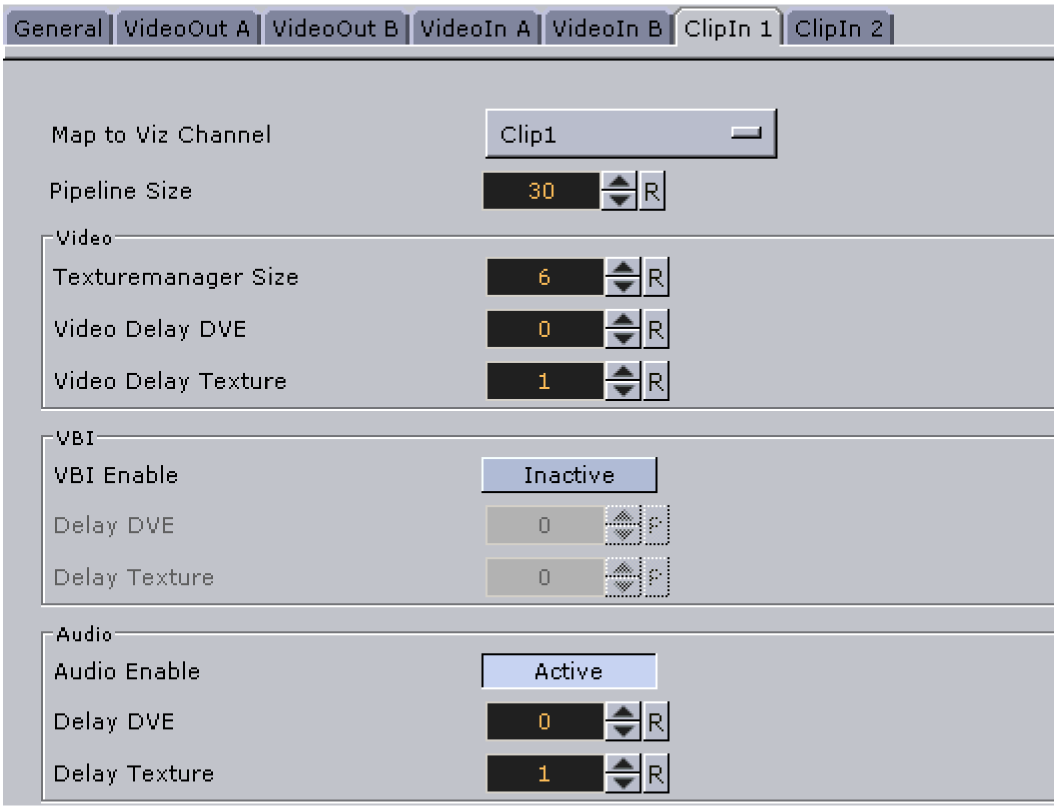
Delay Texture: 4

Group for audio related settings.

- **Audio**: When set to Active this setting enables audio capturing on this channel. Default mode is Active.
- **Channels**: Sets the number of audio channels to capture. Default number of channels are 2. Available channel options for AES on X.mio are None, 1, 2, and 4. For AES on X.mio2 and for Embedded the channel options are None, 1, 2, 4, 8, and 16.
- **Delay DVE**: Sets the number of frames the audio should be delayed in DVE mode before it can be mixed to the output. Default value is 4.
- **Delay Texture**: Sets the number of frames the audio should be delayed in texture mode before it can be mixed to the output. Default value is 4.

10.11.4 ClipIn

Figure 137: ClipIn configuration



Map to Viz Channel: Clip1

Pipeline Size: 30

Video

Texturemanager Size: 6

Video Delay DVE: 0

Video Delay Texture: 1

VBI

VBI Enable: Inactive

Delay DVE: 0

Delay Texture: 0

Audio

Audio Enable: Active

Delay DVE: 0

Delay Texture: 1

The ClipIn tab comes with a simple and an advanced view. The simple view shows only the mapped Viz Artist channel. In the advanced view every setting of the input channel can be controlled. The switch between the simple and the advanced view is done with the small black triangle on the right hand side.

- **Map to Viz Channel**: Decides which Viz Artist clip in channel is mapped onto this Matrox clip in channel. The drop-down gives a choice between the available channels. Only the channels not already taken are shown.
  - **Unused**: Do not use this Matrox channel for clip input.
  - **Clip1**: Clip playback is available in Viz Artist Clip1.

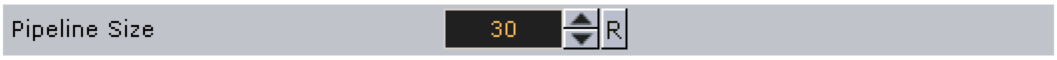
Usually Matrox' ClipIn1 is mapped to Viz' Clip1 and ClipIn2 to Clip2 and so on.

This section also contains information on the following topics:

- Advanced Properties and Parameters
  - ClipIn pipeline size setting
  - ClipIn video settings
  - ClipIn VBI settings
  - ClipIn audio settings
  - ClipIn key settings

Advanced Properties and Parameters

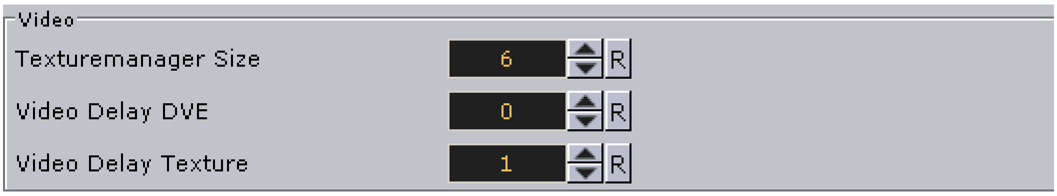
Figure 138: ClipIn pipeline size setting



Pipeline Size 30

- **Pipeline Size**: Defines the number of frames Matrox' internal clip reader buffer should buffer in advance. Default value is 30.

Figure 139: ClipIn video settings



Video

Texturemanager Size 6

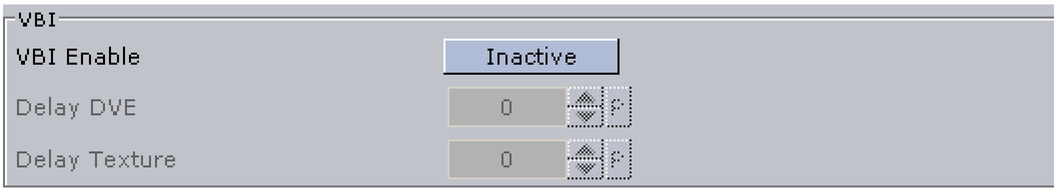
Video Delay DVE 0

Video Delay Texture 1

Group for fill component of the clip in channel

- **Texuremanager Size**: Texture download buffer size.
- **Video Delay DVE**: DVE Delay of video when used as DVE.
- **Video Delay Texture**: Texture Delay of video when used as Texture.

Figure 140: ClipIn VBI settings



VBI

VBI Enable Inactive

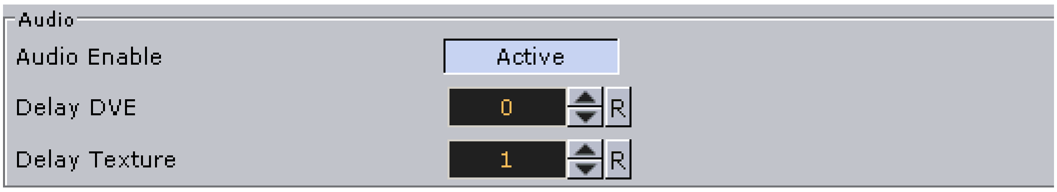
Delay DVE 0

Delay Texture 0

Group for Vertical Blanking Interval (VBI) related settings.

- **VBI Enable**: Defines whether VBI should be used for this channel. Default mode is Inactive.
- **Delay DVE**: Sets the number of frames the VBI should be delayed before the clip can be used in DVE mode. Default value is 0 (Off).
- **Delay Texture**: Sets the number of frames the VBI should be delayed before the clip can be used in texture mode. Default value for Viz 3.3 is 1. For Viz 3.2 the value is 2.

Figure 141: ClipIn audio settings



Audio

Audio Enable Active

Delay DVE 0

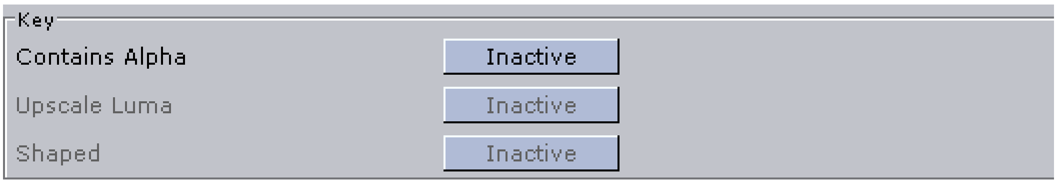
Delay Texture 1

Group for audio related settings.

- **Audio Enable**: When activated, this setting enables audio for this channel. When inactive audio is disabled. Default value is Activated.

- **Delay DVE**: Sets the number of frames the audio clip should be delayed in DVE mode before it can be mixed to the output. Default value is 4.
- **Delay Texture**: Sets the number of frames the audio clip should be delayed in texture mode before it can be mixed to the output. Default value is 4.

Figure 142: ClipIn key settings



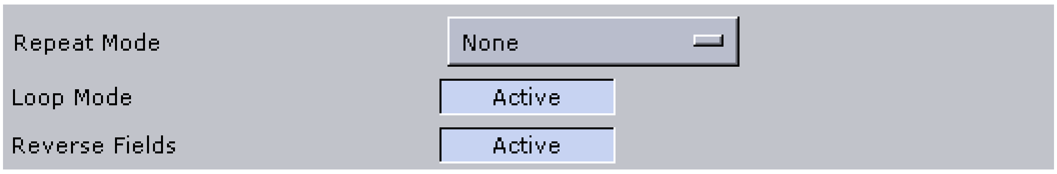
Key

Contains Alpha	Inactive
Upscale Luma	Inactive
Shaped	Inactive

Group for key component of the clip in channel

- **Contains Alpha**: Enables/disables playback of clips with alpha.
- **Upscale Luma**: Enables/disables the default for upscale luma. Per scene setting of this value is set per clip channel under Scene Settings and Video clip options.
- **Shaped**: Defines whether the fill from this channel, when the channel is used in DVE mode, should be interpreted as shaped video during DVE compositing. Default value is Inactive.

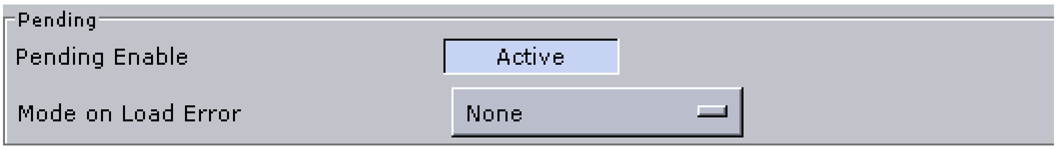
Figure 143: Repeat and loop mode and reverse fields



Repeat Mode	None
Loop Mode	Active
Reverse Fields	Active

- **Repeat Mode**: Determines the behavior of the video input in case of capture drops. Available options are None, Field and Frame.
  - **None**: Does not repeat. Input goes black.
  - **Field**: Repeats the last field.
  - **Frame**: Repeats the last frame.
- **Loop Mode**: Enables/disables default for loop mode. Per scene setting of this value is set per clip channel under Scene Settings and Video clip options.
- **Reverse Fields**: Swaps fields when playing interlaced clips with negative playback speed (default setting). Per scene setting of this value is set per clip channel under Scene Settings and Video clip options.

Figure 144: ClipIn pending settings



Pending

Pending Enable	Active
Mode on Load Error	None

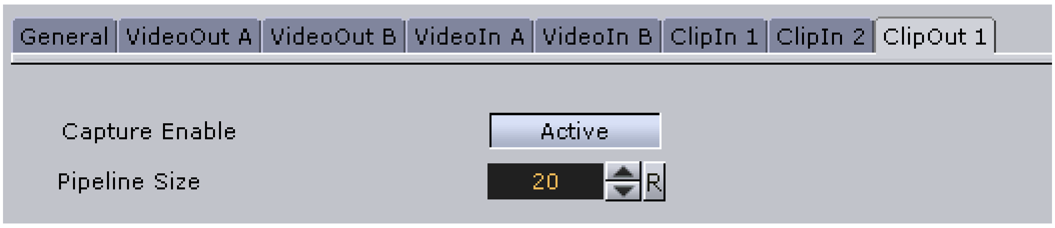
Group for pending support.

- **Pending Enable**: Enables/disables pending clip player for this channel. The pending clip player allows clip loading of another clip while the clip channel is still using the current clip.
- **Mode on Load Error**: Determines the behavior of the current clip when loading of the pending clip fails.

- **None**: Current clip mode is not changed.
- **Stop**: Performs a Stop command on the current clip.
- **Pause**: Current clip enters pause mode.
- **Flush**: Unloads the current clip.

10.11.5 ClipOut

Figure 145: Clipout settings



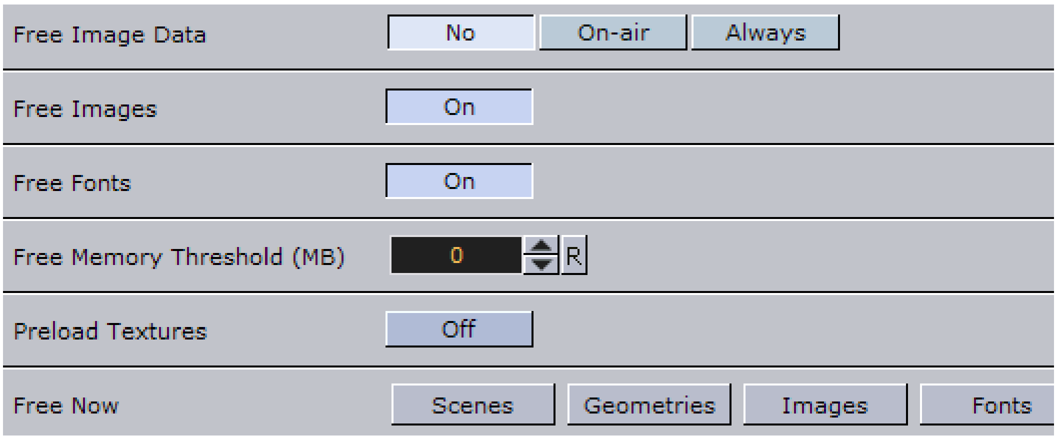
Capture Enable: Active

Pipeline Size: 20

- **Capture Enable**: Enables or disables the clip writer functionality. The main use is to give you control over host memory resources. When the clip writer functionality is not needed then the clip out channel does not need to be allocated.
- **Pipeline Size**: Gives control over the number of frames that the clip writer uses for file handling, similar to the setting in the ClipIn channels.

10.12 Memory Management

Figure 146: Memory Management



Free Image Data: No / On-air / Always

Free Images: On

Free Fonts: On

Free Memory Threshold (MB): 0

Preload Textures: Off

Free Now: Scenes / Geometries / Images / Fonts

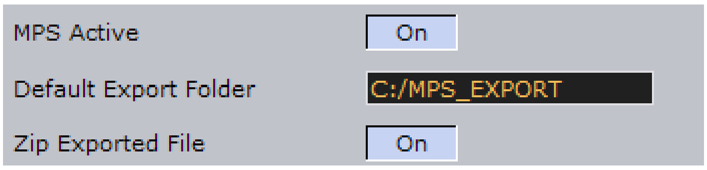
- **Free Image Data**: Loads images into the main memory, and if those images are rendered too, they will be loaded as texture to the graphics card memory as well. Free Image Data gives the possibility to free the image data from the main memory after texture creation. Available options are No, On-air and Always.

Note: If there are modifications done to an image then its texture will be rebuild a lot faster if the data already lies in the main memory (instead of re-loading it from the database).

  - **No**: Disables the Free Image Data option. This option is faster, but needs a lot of memory.
  - **On-air**: Frees image data when in On Air mode, but not in Viz Artist. mode.
  - **Always**: Frees image data every time after the texture was created. This option saves a lot of memory but is slower in case of texture rebuilds.
- **Free Images**: Removes unused images (not referenced in a loaded scene) from the Image Pool (main memory and graphics card memory).
- **Free Fonts**: Removes unused fonts from the Font Pool.
- **Free Memory Threshold (MB)**: If set to greater than zero (>0) then Viz Engine tries to automatically unload unused Pool objects until the specified amount of main memory is free again.
- **Preload Textures**: When enabled (On), then all images which will be loaded with a scene (they do not need to be rendered) are loaded as textures to the graphics card too. This eliminates the texture creation time during rendering afterwards (e.g. useful when initializing a show or a playlist). Default is disabled (Off).
- **Free Now**: Frees the selected unused Pool objects (Scenes, Geometries, Images, Fonts or all) from the memory.
- **Preload Textures**: If this option is active, then all Images which will be loaded with a scene (they do not have to be rendered) are loaded as texture to the graphics card too. This eliminates the texture creation time during rendering afterwards (useful when initializing Viz Trio shows).

10.13 Multi Platform Suite

Figure 147: Multi Platform Suite



MPS Active: On

Default Export Folder: C:/MPS_EXPORT

Zip Exported File: On

In this section, Viz Multi Platform Suite (MPS) is enabled before MPS scenes are published. Enabling MPS for Viz Artist will display an MPS button (upper right) for publishing MPS files. For more information on how to enable MPS and create MPS files in Viz Artist, see the *Viz Multi Platform Suite user's guide*.

- **MPS Active**: Enables and disables MPS from publishing MPS scenes.
- **Default Export Folder**: Sets the default export folder for MPS scenes which can be a local drive, mapped drive or a Universal Naming Convention (UNC) path.
- **Zip Exported File**: Packages the MPS files to a compressed ZIP file.

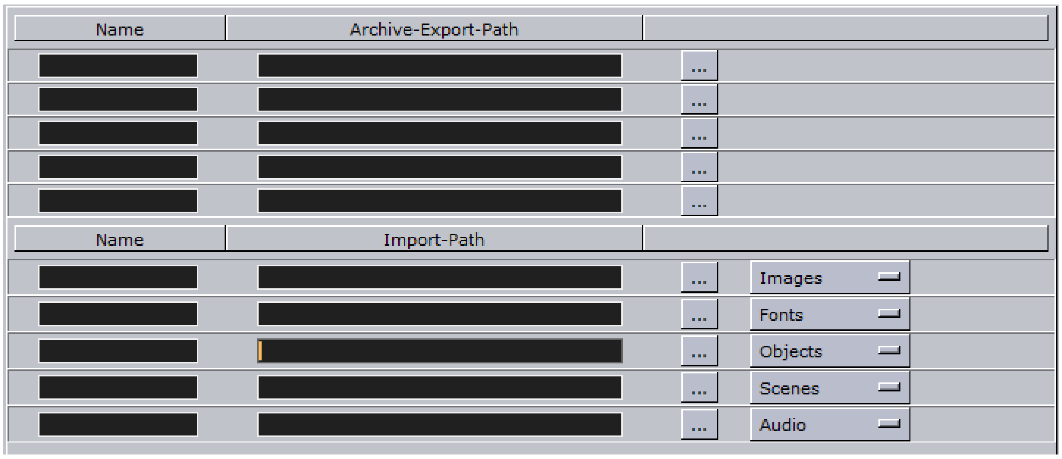
IMPORTANT! Make sure the Default Export Folder is configured with read and write access rights.

To enable Viz Multi Platform Suite in Viz Artist

1. **Enable** the MPS Active button (*On*).
2. Set the **Default Export Folder** field.
3. Click **Save**.

10.14 Path Aliases

Figure 148: Path aliases



In this section, five favorite archive and import paths can be set.

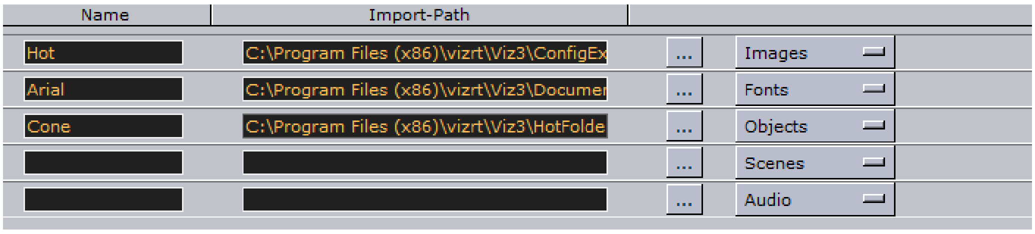
- **Name**: Sets the path alias name for the archive or import path.
- **Path**: Sets the archive or import path which can be a local drive, mapped drive or a Universal Naming Convention (UNC) path.

IMPORTANT! Make sure the archive folder is configured with read and write access rights.

Tip: Path aliases are available in the Viz Artist's Archive and Import panes.

To add a path

1. Enter a descriptive name in the **Name** field
2. Enter a path in the **path** field, or click the **Browse** button to navigate and select a path.
3. Assign a type. Your options are:
   - Images
   - Fonts
   - Objects
   - Scenes
   - Audio

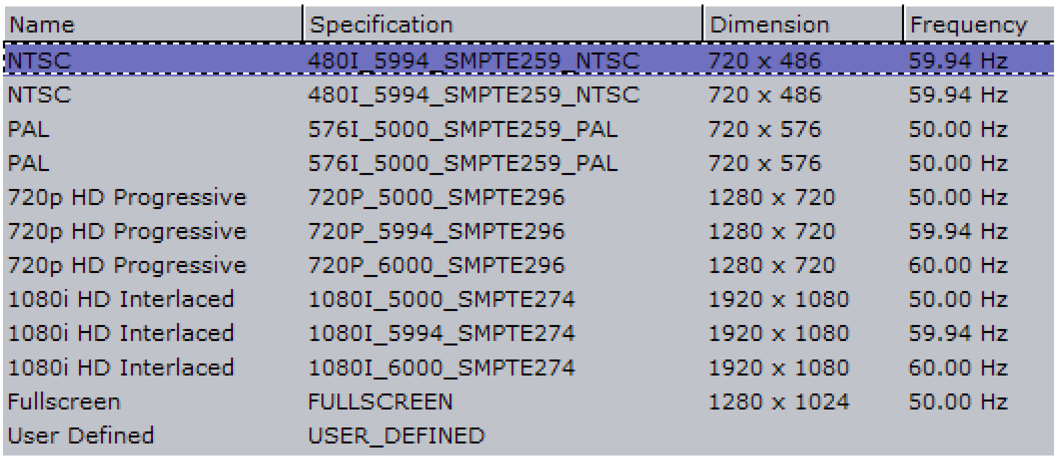


Name	Import-Path		
Hot	C:\Program Files (x86)\vizrt\Viz3\ConfigEx	...	Images
Arial	C:\Program Files (x86)\vizrt\Viz3\Documer	...	Fonts
Cone	C:\Program Files (x86)\vizrt\Viz3\HotFolde	...	Objects
		...	Scenes
		...	Audio

   In this way, when you click on an alias before Importing Files, it automatically takes you to the designated folder and switches to the assigned type.
4. Click **Save**.

10.15 Output Format

Figure 149: Output Format

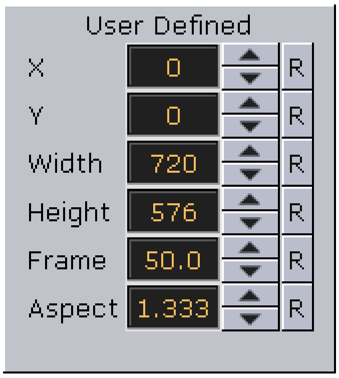


Name	Specification	Dimension	Frequency
NTSC	480I_5994_SMPTE259_NTSC	720 x 486	59.94 Hz
NTSC	480I_5994_SMPTE259_NTSC	720 x 486	59.94 Hz
PAL	576I_5000_SMPTE259_PAL	720 x 576	50.00 Hz
PAL	576I_5000_SMPTE259_PAL	720 x 576	50.00 Hz
720p HD Progressive	720P_5000_SMPTE296	1280 x 720	50.00 Hz
720p HD Progressive	720P_5994_SMPTE296	1280 x 720	59.94 Hz
720p HD Progressive	720P_6000_SMPTE296	1280 x 720	60.00 Hz
1080i HD Interlaced	1080I_5000_SMPTE274	1920 x 1080	50.00 Hz
1080i HD Interlaced	1080I_5994_SMPTE274	1920 x 1080	59.94 Hz
1080i HD Interlaced	1080I_6000_SMPTE274	1920 x 1080	60.00 Hz
Fullscreen	FULLSCREEN	1280 x 1024	50.00 Hz
User Defined	USER_DEFINED		

In this section, the output format of the rendering engine can be set. All video hardware configurations are hooked to the video standard set as output format. This setting defines the frequency (frame rate) at which Viz Engine is running.

For PAL and NTSC, the aspect ratio of the format can be set to 4:3 (standard TV) and 16:9 (wide screen TV).

Clicking **Fullscreen** sets the output format to the screen size of the current machine. Note that Fullscreen only allows you to customize the *frame rate* and *aspect ratio* setting.



User Defined

X	0
Y	0
Width	720
Height	576
Frame	50.0
Aspect	1.333

Clicking the **User Defined** sets the user defined output format. It can be used to fit the requirements of multi-pipe systems (for example video walls or visionariums). The multi-pipe settings can be configured in the right part of the editor.

Setup of multi-pipe systems should be performed by experienced system engineers. For more information, please contact your local Vizrt representative.

- **X**: Sets the horizontal alignment in pixels on the screen. Value is calculated from top left of the screen.
- **Y**: Sets the vertical alignment in pixels on the screen. Value is calculated from top left of the screen.
- **Width**: Sets the width in pixels.

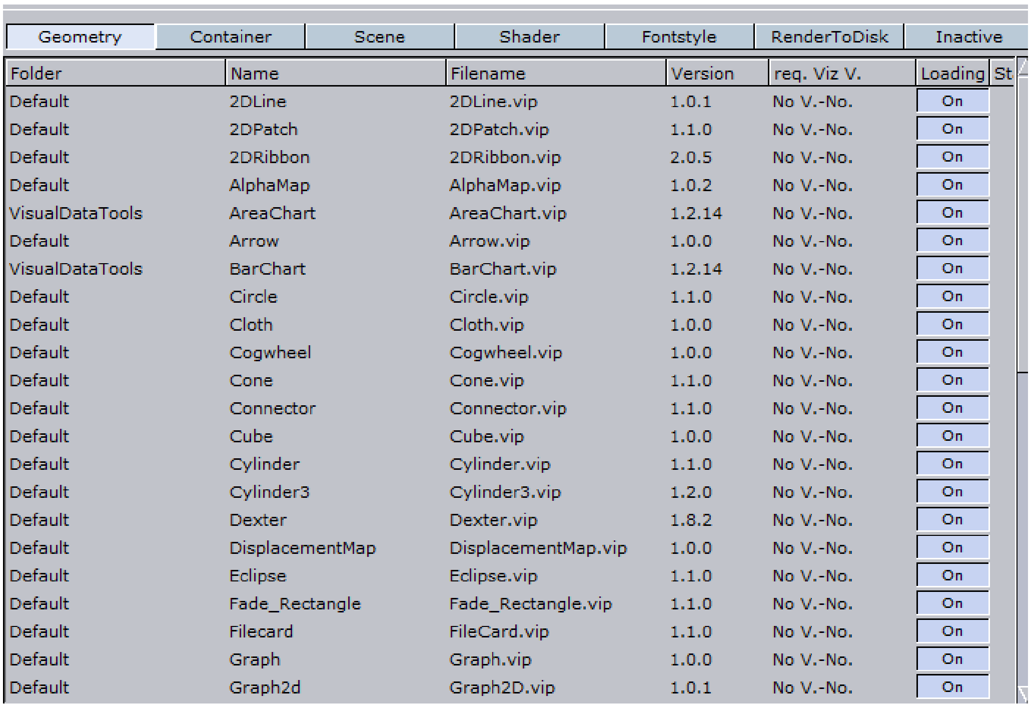
- **Height**: Sets the height in pixels.
- **Frame**: Sets the refresh rate/frequency per frame in hertz (Hz).
- **Aspect**: Sets the aspect ratio. For example 1.778:1 which is 16:9 or 1.333:1 which is 4:3.

Note: Make sure that the physical refresh rate of the graphics hardware and the video hardware is configured correspondingly.

There are 3 frequency groups/families; 50, 59,94 and 60 Hz. This defines the output format, and how fast Viz Engine operates. The frequency is the same as frames per second. This will also define the input format that is allowed; hence, an NTSC SD input cannot produce a PAL SD output, but an HD input with the same frequency as the SD output would work.

10.16 Plugins

Figure 150: Plug-ins



Folder	Name	Filename	Version	req. Viz V.	Loading	St
Default	2DLine	2DLine.vip	1.0.1	No V.-No.	On	
Default	2DPatch	2DPatch.vip	1.1.0	No V.-No.	On	
Default	2DRibbon	2DRibbon.vip	2.0.5	No V.-No.	On	
Default	AlphaMap	AlphaMap.vip	1.0.2	No V.-No.	On	
VisualDataTools	AreaChart	AreaChart.vip	1.2.14	No V.-No.	On	
Default	Arrow	Arrow.vip	1.0.0	No V.-No.	On	
VisualDataTools	BarChart	BarChart.vip	1.2.14	No V.-No.	On	
Default	Circle	Circle.vip	1.1.0	No V.-No.	On	
Default	Cloth	Cloth.vip	1.0.0	No V.-No.	On	
Default	Cogwheel	Cogwheel.vip	1.0.0	No V.-No.	On	
Default	Cone	Cone.vip	1.1.0	No V.-No.	On	
Default	Connector	Connector.vip	1.1.0	No V.-No.	On	
Default	Cube	Cube.vip	1.0.0	No V.-No.	On	
Default	Cylinder	Cylinder.vip	1.1.0	No V.-No.	On	
Default	Cylinder3	Cylinder3.vip	1.2.0	No V.-No.	On	
Default	Dexter	Dexter.vip	1.8.2	No V.-No.	On	
Default	DisplacementMap	DisplacementMap.vip	1.0.0	No V.-No.	On	
Default	Eclipse	Eclipse.vip	1.1.0	No V.-No.	On	
Default	Fade_Rectangle	Fade_Rectangle.vip	1.1.0	No V.-No.	On	
Default	Filecard	FileCard.vip	1.1.0	No V.-No.	On	
Default	Graph	Graph.vip	1.0.0	No V.-No.	On	
Default	Graph2d	Graph2D.vip	1.0.1	No V.-No.	On	

In this section, all installed plug-ins that are identified by Viz as valid plug-ins are listed. Click the Geometry, Container, Scene, Shader, Fontstyle, RenderToDisk or Inactive buttons to view plug-ins per category.

All plug-ins can individually be activated or deactivated. If a plug-in is inactive it will not be loaded during startup. All inactive plug-ins are listed under the Inactive pane.

Note: Some unlicensed plugins will not be loaded while others will; however, in the latter case a watermark will be shown.

Note: You must open the Config section in Viz Artist itself to display all information. In the Viz Config standalone application, the plugins are not actually loaded. However you can still enable or disable the loading state.

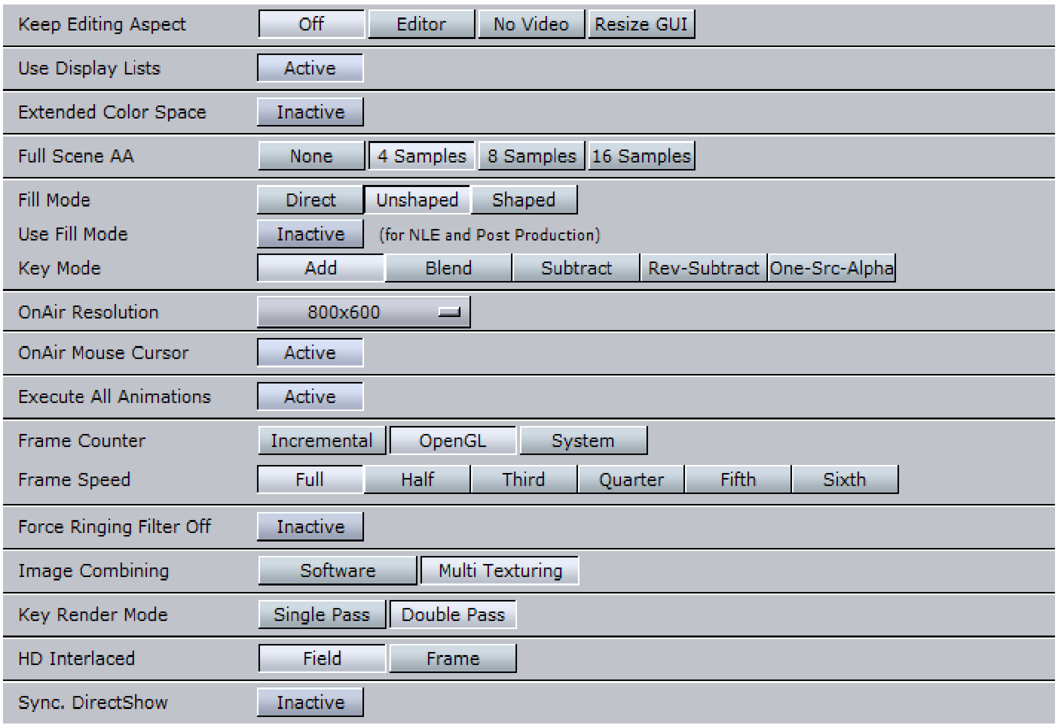
WARNING! Viz does not load inactive plugins during run-time; hence, if a scene uses a plug-in that is deactivated Viz is unable to activate it.

See Also

- Viz License Information.

10.17 Render Options

Figure 151: Render Options

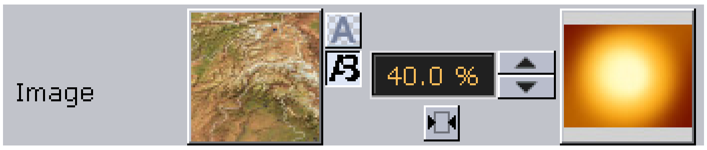


Keep Editing Aspect	Off	Editor	No Video	Resize GUI		
Use Display Lists	Active					
Extended Color Space	Inactive					
Full Scene AA	None	4 Samples	8 Samples	16 Samples		
Fill Mode	Direct	Unshaped	Shaped			
Use Fill Mode	Inactive	(for NLE and Post Production)				
Key Mode	Add	Blend	Subtract	Rev-Subtract	One-Src-Alpha	
OnAir Resolution	800x600					
OnAir Mouse Cursor	Active					
Execute All Animations	Active					
Frame Counter	Incremental	OpenGL	System			
Frame Speed	Full	Half	Third	Quarter	Fifth	Sixth
Force Ringing Filter Off	Inactive					
Image Combining	Software	Multi Texturing				
Key Render Mode	Single Pass	Double Pass				
HD Interlaced	Field	Frame				
Sync. DirectShow	Inactive					

In this section, the following render options can be set:

- **Force Sleep**: Enables Viz Engine to sleep on low performance renderer machines (for example a laptop) when Viz Engine is running inside another application such as Viz Trio.
- **Keep Editing Aspect**: Influences scene designs in Viz Artist mode. Alternatives are; Off, Editor, No Video and Resize GUI.
  - **Off**: Scenes are only shown in Anamorphic widescreen in the 4:3 VGA render window.
  - **Editor**: Scenes are displayed using a letter-box format during scene editing giving designers the option to set a user defined **camera aspect** ratio (under Scene Settings -> Rendering). On-Air and Viz Engine modes are not affected.

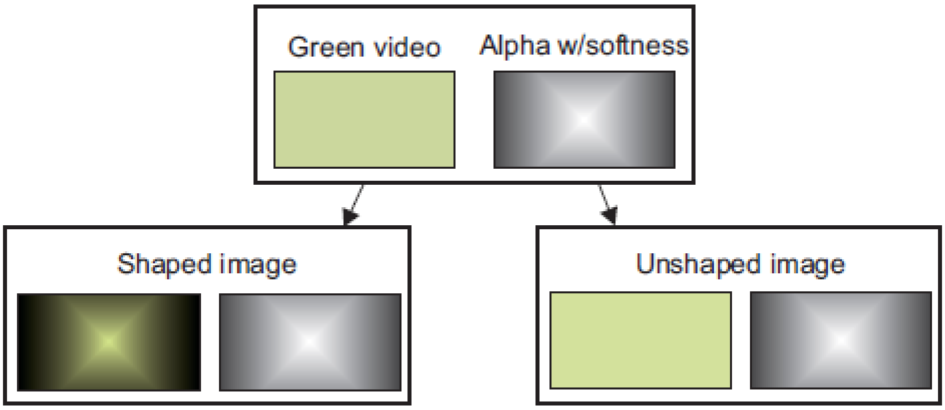
  - **No Video**: Scenes are shown using a letter-box format as long as the video out is inactive. If video out is active scenes are shown in Anamorphic widescreen in Viz Artist mode.
  - **Resize GUI**: Increases the renderer window when editing 16:9 scenes to 16:9 format. On-Air and Viz Engine modes are not affected.
- **Use Display Lists**: Activates or deactivates the use of display lists in the Open GL engine.
- **Extended Color Space**: *Not in use*.
- **Full Scene AA**: Sets the hardware Anti aliasing (provided by the graphics card). Alternatives are; None, 4, 8, and 16 Samples.
- **Fill Mode**: **Direct**: Unmodified fill output. **Unshaped**: Brightened fill when AutoKey is enabled. **Shaped**: Fill is premultipled with key. Shaped video versus unshaped video.
- **Use Fill Mode**: Determines whether the *Fill Mode* setting should also be applied to post rendering and NLE. If *Fill Mode* is inactive, and *Shaped* setting is active, then the output looks brighter, as it is required for keyers. This option will remove inconsistencies between full frames and fields with post rendering and NLE.
- **Key Mode**: *Not in use*.
- **RGB to YUV**: Enables color conversion either in the Shader or on the Matrox card.
- **On Air Resolution**: Sets the DVI output resolution for Viz Video Wall. Viz Video Wall must activate DVI Output in the Video Output section for the On Air Resolution to take effect. Alternatives are; Desktop Resolution, 800x600, 1024x768, 1280x1024, 1600x900, and 1600x1200.
- **On Air Mouse Cursor**: Enables a mouse cursor when in On Air mode and using interactive applications. Should be disabled for Viz Video Wall and when DVI out is enabled.
- **Execute All Animations**: When deactivated this setting will enable Viz Engine to only animate visible objects. Default is Active.
- **Use NVIDIA Frame Counter**: When activated this settings takes effect for Standard-PC versions that use NVIDIA cards and drivers that do not return valid frame counter values. When deactivated, an internal timer is used as a timing trigger.
- **Frame Speed**: Enables Viz to run at a slower rate than the actual refresh rate (determined by the screen speed with the use of a g-sync card). This is relevant for video wall applications when the scene (e.g. interactive scenes) cannot run real time (resource intensive), hence, all participating computers are synched to a lower speed. Running at 30Hz (screen speed 60Hz divided by 2) can be acceptable, however, this will affect the animation quality (i.e. will not be as smooth).



- **Image Combining**: In a texture editor (see Viz Artist) it is possible to set a second texture which will be used for the image combining. The texture editor offers two possible modes: The first mode uses the second image as an alpha channel, whereas the second mode defines a blend between the two textures. Configuring Image Combining to **Software** enables the combination to be calculated entirely on the CPU. Configuring Image Combining to **Multi Texturing** enables the combination to be calculated on the graphics card for combining or blending the two images. In this case the texture creation is faster and memory will be saved as well. Default is Software. If there are performance or memory issues, especially with scenes imported from Viz 2.x, it is recommended to change this setting to Multi Texturing.

- **Key Render Mode**: This configuration is used when the Key Render Mode under Scene Settings in Viz Artist is set to Config, and determines how the key should be rendered:
  - **Double Pass**: Uses two rendering steps as in older 3.x versions. Double Pass should be used for old 3.x scenes for not breaking compatibility, for new scenes Single Pass should be used as it is faster.
  - **Single Pass**: Uses one rendering step as in 2.8 versions.

Figure 152: Shaped video versus unshaped video

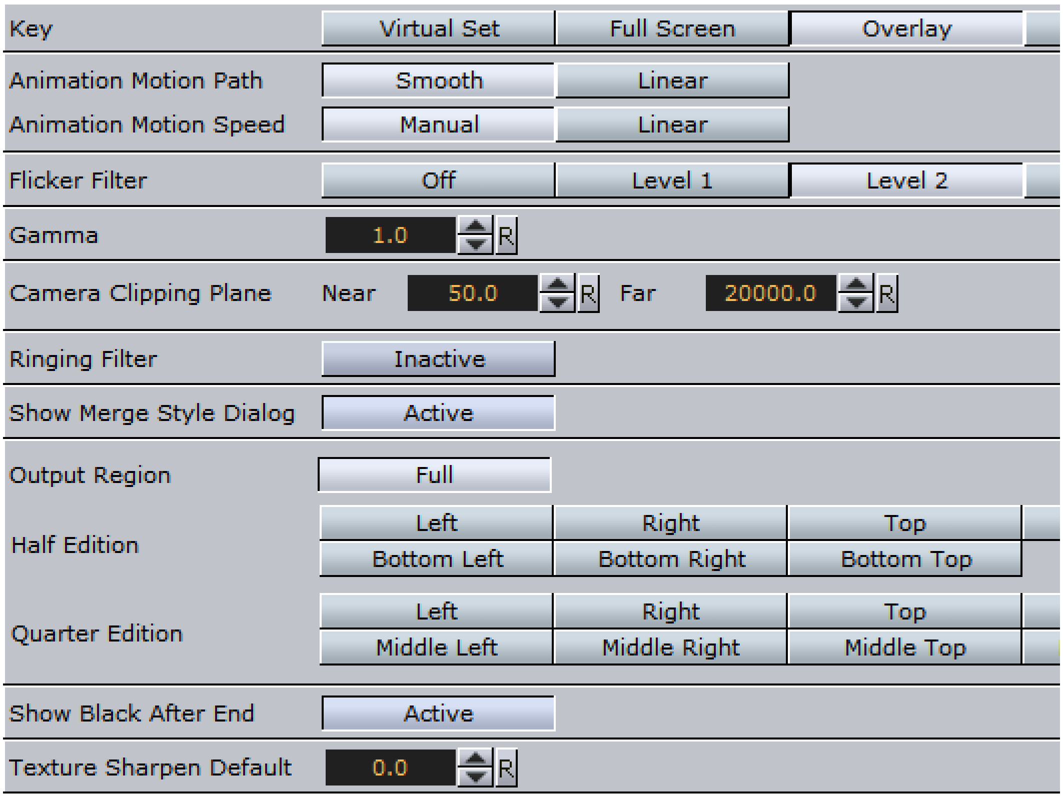


A shaped video image has its video data multiplied by its alpha component while the video data of an unshaped image remains untouched. Shaped images are also referred to as *pre-multiplied alpha images*.

- **HD Interlaced**: Specifies if Viz Engine should render frames or fields for interlaced HD formats (i.e. 1080i). Rendering frames improves the rendered output but has a higher performance cost. Available options are: Field and Frame.
- **Sync. DirectShow**: Synchronize DirectShow clip playback with renderer (may cause video jumps and audio cracks).
- **NLE Antialiasing**: Antialiasing for small images snapshots and NLE frames.
- **Image Load Error**: If an image load error occurs you may configure Viz to keep the old image or clear the image (i.e. not showing anything).

10.18 Scene Default Values

Figure 153: Scene Default Values



Key: Virtual Set | Full Screen | Overlay

Animation Motion Path: Smooth | Linear

Animation Motion Speed: Manual | Linear

Flicker Filter: Off | Level 1 | Level 2

Gamma: 1.0 R

Camera Clipping Plane: Near 50.0 R Far 20000.0 R

Ringing Filter: Inactive

Show Merge Style Dialog: Active

Output Region: Full

Half Edition: Left | Right | Top; Bottom Left | Bottom Right | Bottom Top

Quarter Edition: Left | Right | Top; Middle Left | Middle Right | Middle Top

Show Black After End: Active

Texture Sharpen Default: 0.0 R

This section configures the default values for new scenes.

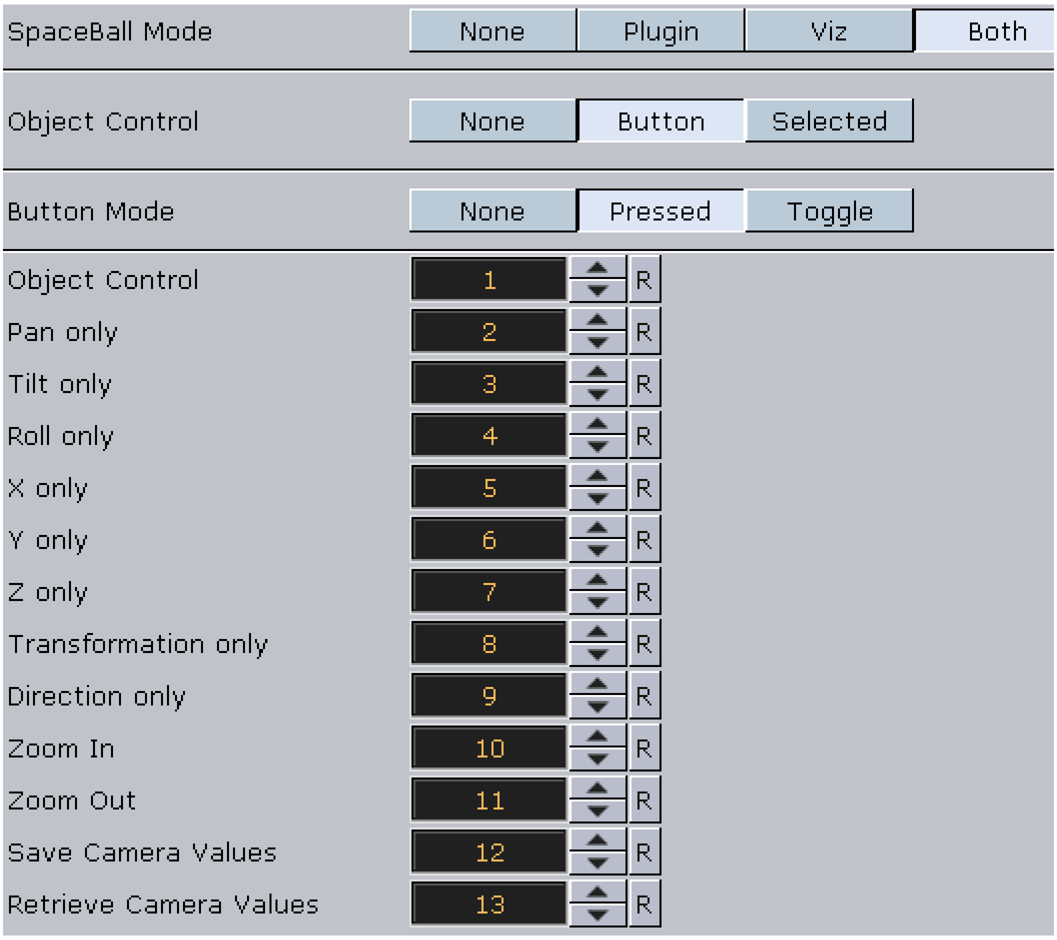
- **Key**: Sets the key mode. Alternatives are; Virtual Set, Full Screen, Overlay, and Auto Key.
- **Animation Motion Path**: Sets the default setting for all new position keyframes. When set to Smooth all handles in the scene output enables a smooth motion path. When set to Linear no handles are available; hence the motion path is not smooth. This setting corresponds with the Path Control setting in the keyframe editor. When set to manual handles are made available in the spline view that enables speed to be manually handled between keyframes. When set to Linear no handles are available.
- Animation Motion Speed: Sets the default setting for all new position keyframes.
- **Flicker Filter**: When enabled it will reduce interlaced flicker on high contrast objects. For example small lines and hard objects. Alternatives are; Off, and Level 1-3.
- **Gamma**: Sets the gamma radiation/emission for textures.
- **Camera Clipping Plane**: Sets the range of the virtual camera. Near sets the close range while far defines the far range. Only objects within this range will be rendered.
  - **Near**: Sets the Near value to clip unwanted objects from the foreground. Default value is 50.
  - **Far**: Sets the Far value to clip unwanted objects from the background. Default value is 20000.

Note: The camera range is where the Z-buffer is within. So if Z-buffer problems arise, they may be solved by editing the camera clipping plane settings.

- **Ringing Filter**: Sets the default value for the Ringing Filter. A ringing filter reduces high frequency values in the video signal created by high contrast and color changes in horizontal directions.
- **Show Merge Style Dialog**: Enables the user, when opening a scene in Viz Artist, to open old-style merged objects and expose containers within it.
  - This feature relates to scenes using old-style ordering of containers within merged objects, and solves the problem with auto-follow. When loading such scenes the dialog lets the user decide how to deal with them.
  - Users that are aware of this and decide to keep the old style can deactivate this dialog.
- **Output Region**: Shows which region of the screen Viz is licensed to render.
- **Half Edition**: Allows Viz to render half screen. Available options are; left, right, top, bottom, bottom left, bottom right and bottom top.
- **Quarter Edition**: Allows Viz to render quarter screen. Available options are; left, right, top, bottom, middle left, middle right, middle top and middle bottom.
- **Show Black After End**: Shows black after a clip has finished playing.
- **Texture Sharpen Default**: Sets the default sharpen value for textures.

10.19 Spaceball

Figure 154: Spaceball



SpaceBall Mode: None | Plugin | Viz | [✓ Both]

Object Control: None | [✓ Button] | Selected

Button Mode: None | [✓ Pressed] | Toggle

Object Control 1 R
Pan only 2 R
Tilt only 3 R
Roll only 4 R
X only 5 R
Y only 6 R
Z only 7 R
Transformation only 8 R
Direction only 9 R
Zoom In 10 R
Zoom Out 11 R
Save Camera Values 12 R
Retrieve Camera Values 13 R

The Spaceball section is used to configure a 3D navigation device.

- **SpaceBall Mode**: Sets special setups where the spaceball should only control specific plug-ins without influencing the scene (objects/camera). Alternatives are; None, Plugin, Viz, and Both.
  - **None**: No setup.
  - **Plugin**: Controls plug-ins.
  - **Viz**: Controls Viz objects.
  - **Both**: Controls both plug-ins and Viz objects.
- **Object Control**: Alternatives are; None, Button, and Selected.
  - **None**: No setup.
  - **Selected**: Modifies only the selected object.
- **Button Mode**: Alternatives are; None, Pressed, and Toggle.
  - **None**: No setup.
  - **Pressed**: When set to Pressed, this mode will trigger an action like a button in a user interface.
  - **Toggle**: When set to Toggle, this mode will set a state. For example when a button is pressed only the dominant axis will be considered in a move, whereas when the button is released all movements are applied.

The numeric fields are used to map the various buttons on the 3D navigation device. This varies by the vendor and the vendors model; hence, the button numbers need to be looked up in the user manual for the respective device.

Button options are: Object Control, Pan Only, Tilt Only, Roll Only, X Only, Y Only, Z Only, Transformation Only, Direction Only, Zoom In, Zoom Out, Save Camera Values, and Retrieve Camera Values.

10.20 User Interface

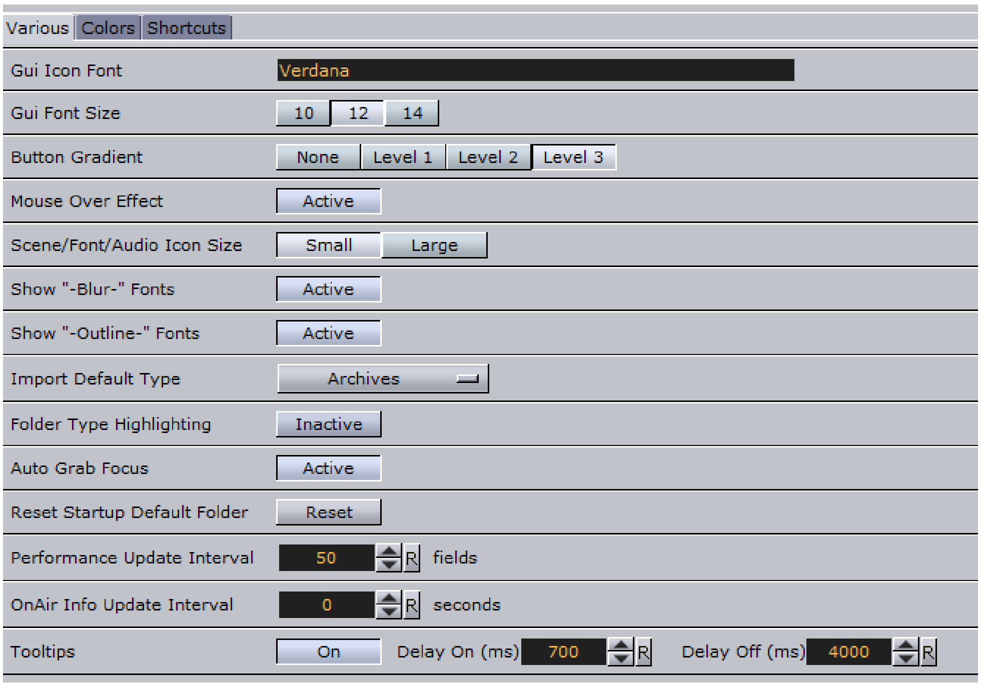
This section describes the user interface settings. Some of these settings are also available in Viz Artist.

This section contains information on the following topics:

- Various
- Colors
- Shortcuts

10.20.1 Various

Figure 155: Various user interface settings



Gui Icon Font	Verdana
Gui Font Size	10 / 12 / 14
Button Gradient	None / Level 1 / Level 2 / Level 3
Mouse Over Effect	Active
Scene/Font/Audio Icon Size	Small / Large
Show "-Blur-" Fonts	Active
Show "-Outline-" Fonts	Active
Import Default Type	Archives
Folder Type Highlighting	Inactive
Auto Grab Focus	Active
Reset Startup Default Folder	Reset
Performance Update Interval	50 R fields
OnAir Info Update Interval	0 R seconds
Tooltips	On Delay On (ms) 700 R Delay Off (ms) 4000 R

- **GUI Icon Font**: Sets the Font type for all icons of elements such as scenes, objects, materials, images, fonts, and audio clips. Complex character sets such as Arabic, Hebrew and Chinese must change the default font type in order to display proper names for the icons.
- **GUI Font Size**: Sets a global font size for the Viz GUI. Alternatives are; 10, 12, and 14 pixels.
- **Button Gradient**: Sets the gradient level of the buttons in the Viz Artist user interface.

- **Mouse Over Effect**: Mouse over effect for buttons in the Viz Artist user interface. Set to Active or Inactive.
- **Scene/font/Audio Icon size**: Sets a size preference for scene, font and audio icons. Icon size can also be switched using the context menu in the server view in Viz Artist; however, a switch in Viz Artist will not be saved as a preference for later sessions.
- **Show “Blur” fonts**: Sets the default for showing or hiding the blur fonts in the Server area. The option can then be toggled using the Server File Context Menu.
- **Show “Outline” fonts**: Sets the default for showing or hiding the outline fonts in the Server area. The option can then be toggled using the Server File Context Menu.
- **Import Default Type**: Selects the default type for imports. This will then be pre-selected in the Import Menu. Options are:
  - Fonts
  - Images
  - Objects
  - Scenes
  - Audio
  - Archives
- **Folder Type Highlighting**: When enabled this will highlight the folders that contain content matching the current Viz Artist’s Server view (for example Scene, Object, Material, Image, Font, Audio and so on).

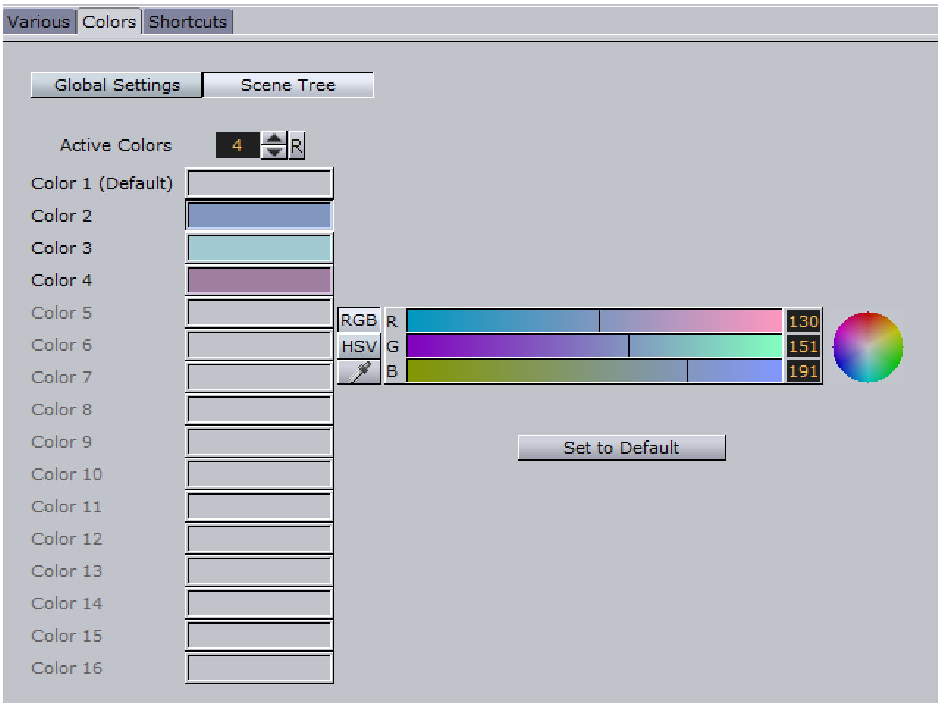
Note: This can cause some performance overhead, when switching to different types and/or with opening sub folders (but only first time, as the information is cached).

- **Auto Grab Focus**: When Inactive, a middle mouse click grabs the focus, as a left and right click do. When audio grab focus is Active, the focus is grabbed as soon as you move the mouse over a widget.
- **Reset Startup Default Folder**: Resets the startup folder. If Viz Artist is unable to start due to problems with the last saved server view, clicking the Reset button will reset the Server view to its top node.
- **Performance Update Interval**: Sets how often Viz Engine should update the Performance Bar when it is opened.
- **OnAir Update Interval**: Update interval for the OnAir Info Window. Note that a shorter interval decreases render performance. Seting to 0 means that no update occurs.
- **Tooltips**: Enables or disables the tool tip information (pop-ups) in the Viz Artist 3 user interface.
  - **Delay On (ms)**: Sets the amount of time in milliseconds before the tool tip appears. Default is 1500 ms.
  - **Delay Off (ms)**: Sets the amount of time before the tool tip disappears. Default is 4000 ms.

Note: A millisecond is one thousandth of a second.

10.20.2 Colors

Figure 156: User Interface colors - Scene Tree



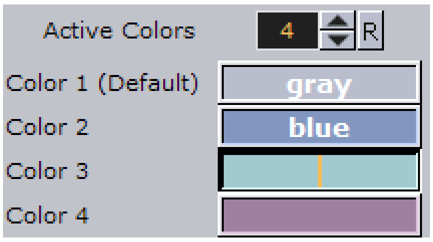
In the Scene Tree color configuration section, you can set up the tree’s color coding in such a way as to sort elements by colors, search for colors within a tree, and restrict the scene tree to certain colors, so as not to require the handling the whole tree for certain simple transactions.

For example, all text containers can be colored gray and tagged Text, while all containers that hold images can be colored green and tagged Image and so on. Four active colors are configured by default with no text descriptions.

The text is scene specific and saved with the scene on Viz Graphics Hub; hence, the colors might change if the scene is edited using another editor with a different color setting.

The color options are available in the Scene Settings pane as the Tree Color Text setting, and available for use in the Scene Tree pane.

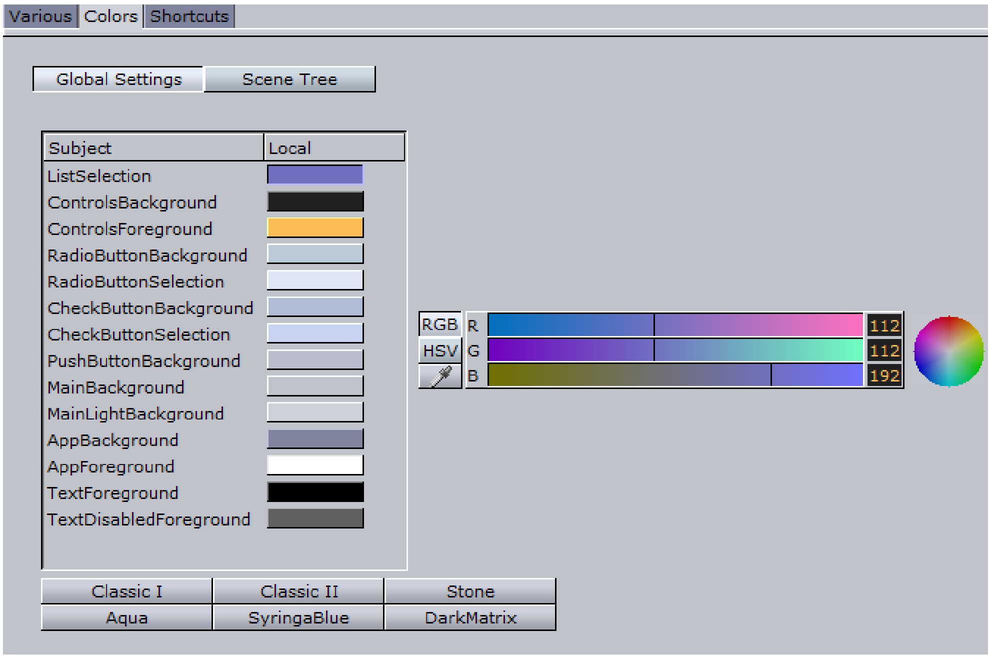
To create a new color



1. Select an unused scene tree Color container
2. *Optional*: Add a descriptive text

3. Set the color values
4. Optional: Click the color box and enter a name for it.
5. Click **Save** and restart Viz Artist.

Figure 157: User Interface colors - Global Settings



Subject	Local
ListSelection	🟪
ControlsBackground	⬛
ControlsForeground	🟧
RadioButtonBackground	⬜
RadioButtonSelection	⬜
CheckButtonBackground	⬜
CheckButtonSelection	⬜
PushButtonBackground	⬜
MainBackground	⬜
MainLightBackground	⬜
AppBackground	🟪
AppForeground	⬜
TextForeground	⬛
TextDisabledForeground	⬛

The Global Settings change the entire look of the Viz Artist user interface. You can customize each element in the left-pane list, using the color slider/picker. You can click and drag one of the current element colors and drop it onto another category as well.

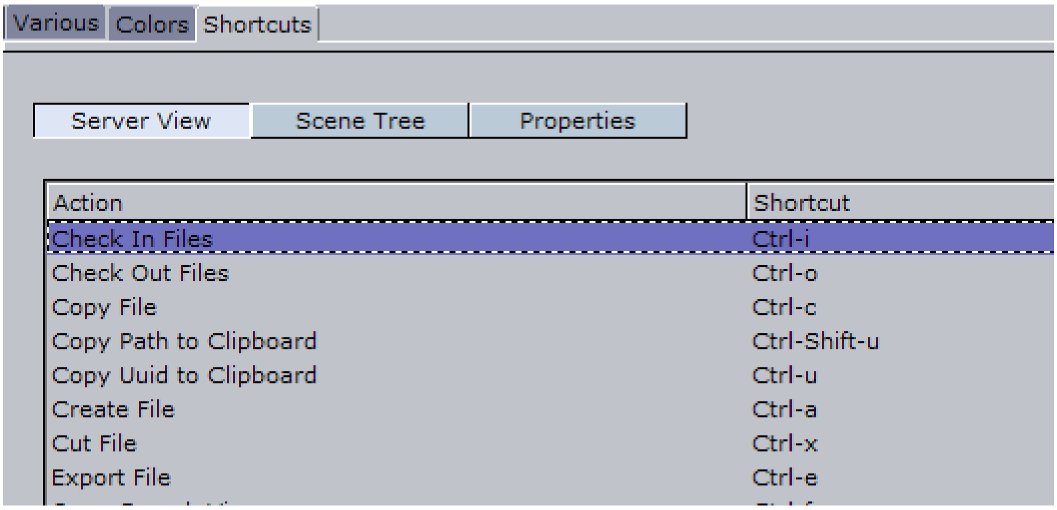
Alternatively you can choose one of the pre-defined color schemes for the user interface. **Classic 1** is the traditional Viz Artist look.

Note: All the changes made to the user interface are local on your machine. You need to restart Viz Artist for the changes to take effect.

10.20.3 Shortcuts

The shortcuts view displays all available server, scene tree and property actions and the currently assigned shortcuts. All shortcut configurations are saved to the database into the user table for personalization.

Figure 158: User interface shortcut settings

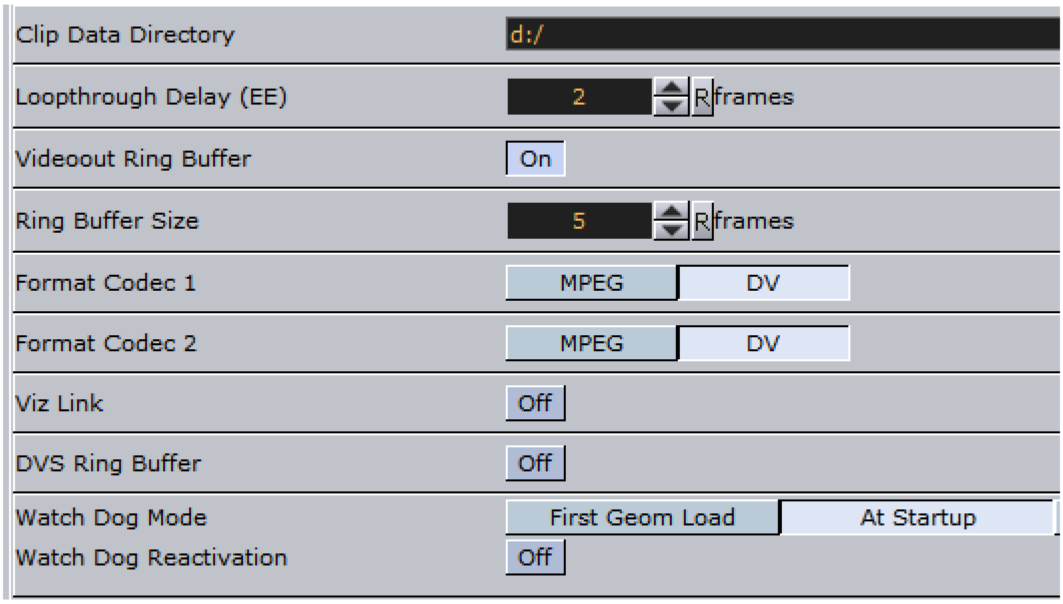


Action	Shortcut
Check In Files	Ctrl-i
Check Out Files	Ctrl-o
Copy File	Ctrl-c
Copy Path to Clipboard	Ctrl-Shift-u
Copy Uuid to Clipboard	Ctrl-u
Create File	Ctrl-a
Cut File	Ctrl-x
Export File	Ctrl-e

- **Server, Scene Tree and Properties**: Displays all server actions and their currently assigned shortcuts.
- **Shortcuts**: When clicked, displays a list of available shortcut key-combinations that can be assigned to the selected action.
- **Reset All**: Resets all shortcuts to the default setup.

10.21 Video Board

Figure 159: Video Board



Clip Data Directory	d:/
Loopthrough Delay (EE)	2 R frames
Videoout Ring Buffer	On
Ring Buffer Size	5 R frames
Format Codec 1	MPEG DV
Format Codec 2	MPEG DV
Viz Link	Off
DVS Ring Buffer	Off
Watch Dog Mode	First Geom Load At Startup
Watch Dog Reactivation	Off

This section is used to configure video input, output and clip playback related settings.

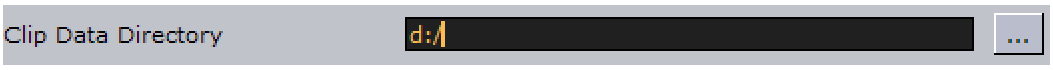
- **Clip Data Directory**: Sets the clip directory for clip transfer in a Viz Link environment. Default directory is V:\ drive.

Note: The directory has to match the directory set when installing the Mediaftp FTP service for video transfer from Viz Video Hub.

- **Loopthrough Delay (EE):** Sets delay for live video input in DVE mode. This setting is only used for Targa boards. This setting applies for all input channels. For Matrox, see (Matrox0.VideoIn1.VideoDelayDVE) ... Configuration File.
- **Videoout Ringbuffer**: Sets the render buffer for video output. Helps to prevent frame drops on the video output during execution of commands or loading of objects. When enabled the engine will render a number of graphics frames in advance and provide it to the video hardware. The number of frames rendered forwardly is defined by the Ringbuffer Size. Large buffer sizes will delay the output and increase the input to output delay for video textures.
- **Ringbuffer size**: Sets the size of the Ringbuffer in frames.
- **Format Codec 1**: Enables the codec matching the codec of the Targa board. Alternatives are MPEG and DV. This only applies to video clips.
- **Format Codec 2**: Enables the codec matching the codec of the Targa board. Alternatives are MPEG and DV. This only applies to video clips.
- **Viz Link**: Turning Viz Link on has the following effects:
  - In the VGA version, setting the clip takes immediate effect instead of the command travelling through the ring buffer.
  - In the VGA version a clip channel set to DVE will play the clip in texture mode.
  - Activates the asynchronous Command feedback when setting a clip. The first answer in this case is a `ANSWER DELAYED` message to the client and after that, either an error message or a success message is sent back to the client.
  - A clip which is currently playing in the active player will not be loaded back to back.
- **DVS Ring Buffer**: To have a separate ring buffer setting for DVS (especially for combinations with Matrox DSXLE2), turn this setting on.
- **Watch Dog Mode**: The following options are available:
  - **First Geom Load**: Video Out is activated when the first geometry is loaded.
  - **At Startup**: Video Out is activated at start up.
  - **First Geom Load/DVE**: Video Out is activated when the first geometry is loaded or an input or clip channel is set to DVE.
- **Watch Dog Reactivation**: Turn On to activate the watch dog again if scenes are unloaded from renderer.

10.22 Video Clip

Figure 160: Video clip settings

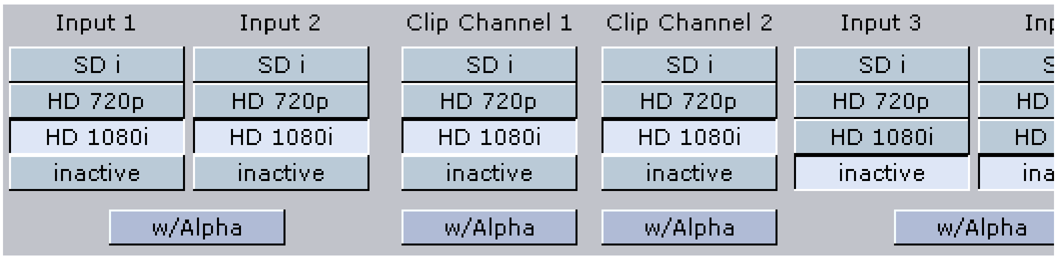


Clip Data Directory d:/

- **Clip Data Directory**: Sets the clip directory for clip transfer in a Viz Link environment. Default directory is D:\ drive.

10.23 Video Input

Figure 161: Video input settings



Input channels are enabled in this section. To be able to use a channel – no matter if it is a live input channel or a clip channel – it first has to be enabled.

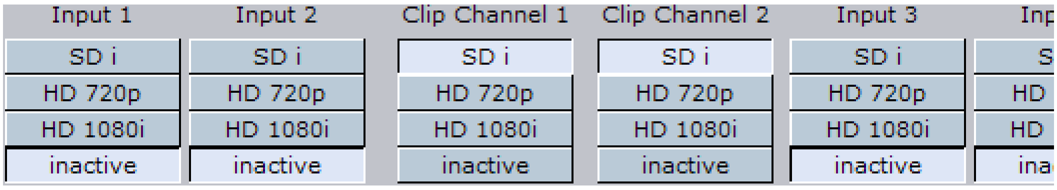
This dialog not only activates the channel, it also defines the video standard the channel should be configured to. It is possible to mix video standards – for example a Viz Engine configured to play out 1080i50 can have inputs configured to SD. In this dialog Input 1 and Input 2 refer to live input channel 1 and live input channel 2. Clip channels are for video clips. The frequency of the channels will be defined by the configurations in the Output Format section.

Clip Channel 1 and Clip Channel 2 must be configured to the same video standard.

- **Input w/Alpha**: For the Input this will combine the two (Input 1 and Input 2) channels to one channel carrying fill and key. Meaning the Y (luminance) from Input 2 in YUV will be used as the key signal.
- **Clip w/Alpha**: When enabled the clip file must either contain key information or a key clip must be provided. The key clip must have the string "_key" appended to the filename. For a clip named *i422_50M.avi* this would be *i422_50M_key.avi*.

Note: Video input format may be limited by hardware capabilities on the video board (for example Targa boards only support SD, and Matrox 6000 series boards does not support HD clips).

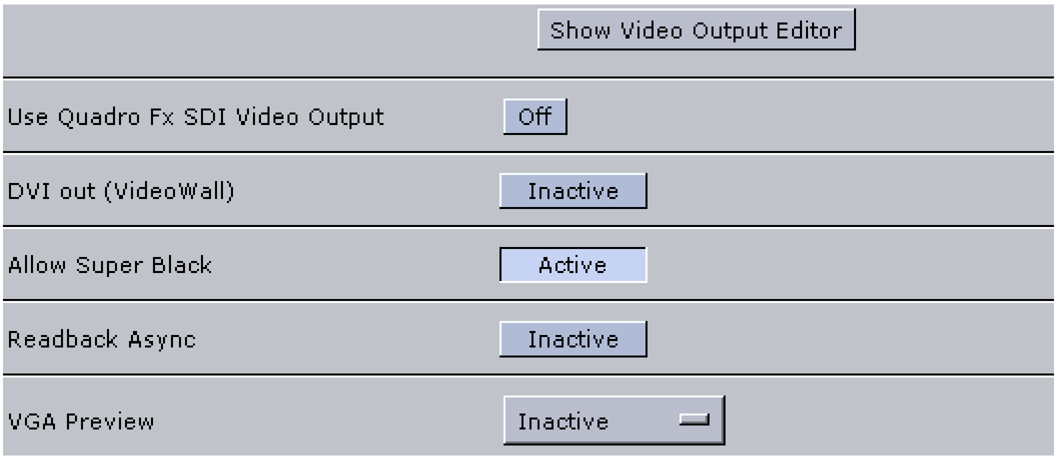
Figure 162: Video input settings without video board



When there is no onboard video board (standard-PC) the video input section no longer displays the video with alpha options. Video input for standard-PC versions enable clip playback via scene setup.

10.24 Video Output

Figure 163: Video Output



Use Quadro Fx SDI Video Output: Off

DVI out (VideoWall): Inactive

Allow Super Black: Active

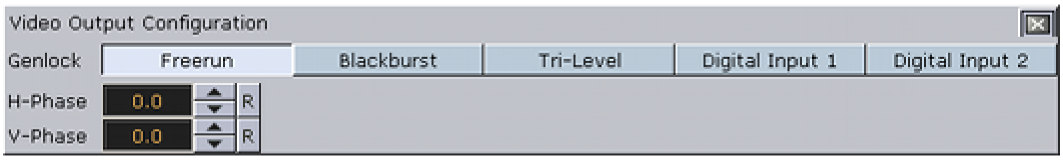
Readback Async: Inactive

VGA Preview: Inactive

This section is used to configure special settings for video output such as SPG settings and so on.

- **Show Video Output Editor (ALT+V)**: Opens the Video Output Editor.
- **Use Quadro FX 4000 SDI Video Output**: Enables NVIDIA’s SDI module to output video. This will shorten the output delay when outputting graphics and video through the video board. However, this requires that audio to be handled externally since NVIDIA’s SDI module does not support embedded audio output.
- **DVI Out (Viz Video Wall)**: Sets the main output to DVI.
- **Allow Super Black**: Controls whether the key output uses the video range (Inactive) or the full signal range (Active). If set to Active it will then determine whether or not to clip the key output video signal that is under 7.5 IRE units or over 100 IRE units. Default mode is *Inactive*. Inactive (using older drivers) is only valid in conjunction with NVIDIA SDI and driver versions older than 259.12.
- **VGA Preview**: Sets the default value for the Preview button (see Control Buttons) when Viz Engine is in On Air mode.
  - When inactive, rendering will only be done for video out signals. This will increase performance as the renderer does not have to render into an editor on screen and into pixel buffer.
  - When active, rendering will be done for both video out signals and on screen; however, this will decrease the performance.
  - Standard-PC versions do not have a video out option; hence, the button is not visible in On Air mode and VGA Preview is always active.

Figure 164: Video Output Editor



Video Output Configuration

Genlock: Freerun

H-Phase: 0.0

V-Phase: 0.0

The Video Output Editor defines the synchronization standard and the output signal phases.

- **Freerun**: Locks Viz Engine to a clock signal on the video board.
- **Blackburst and Tri-level**: Locks Viz Engine to a GenLock signal.
- **Digital Input 1 and 2**: Locks Viz Engine to the signal on Input 1 or 2.
- **H-Phase and V-Phase**: Shifts the output signal with respect to the sync signal.

To make the V- and H-phase values coincide

1. Set the **V-phase value**
   - The *V-granularity* is taken from the genlock
   - The *V-delay* is calculated from *V-phase * V-granularity*
   - The genlock is set with this *V-delay*

2. Set the **H-phase value.** Note that there is a distinction whether the H-phase is a positive or a negative value.

   a. If the **H-phase** > **0**
   - The genlock *H-delay* is set to 0
   - The *H-granularity* is taken from the video out channel
   - The *H-delay* is calculated from *H-phase * H-granularity*
   - The fill and key channels are set with this *H-delay*

   b. If the **H-phase** <=**0**
   - The fill and key channel *H-delay* is set to 0
   - The *H-granularity* is taken from the genlock
   - The *H-delay* is calculated from *- H-phase * H-granularity*
   - The genlock is set with this *H-delay*

Note: The granularity and possible min/max values are printed to the Viz console during startup.

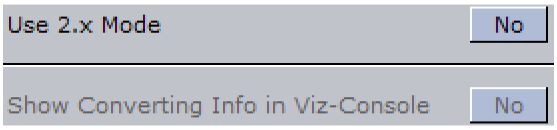
It should be taken into account that when the genlock video format is different from the fill/key video format the value of the *V-delay* matches the genlock lines and not the video output lines. The same applies to *negative H-phase* values.

See Also

- Output Format
- Video Input

10.25 Viz Artist 2.x

Figure 165: Viz Artist 2.x



Use 2.x Mode: No

Show Converting Info in Viz-Console: No

In this section, Viz Artist 3 can be switched to Viz Artist 2 mode for compatibility issues.

In Viz Artist versions prior to 3.0, assets were stored in specific subdirectories (Scene, Object, Material, Font, and Image). As Viz Artist 3 stores assets anywhere in the directory, older control applications may not find these assets. If enabling

2.x mode, the assets will be looked for in the subdirectories, and calls for scenes will be represented by green lines in the console window.

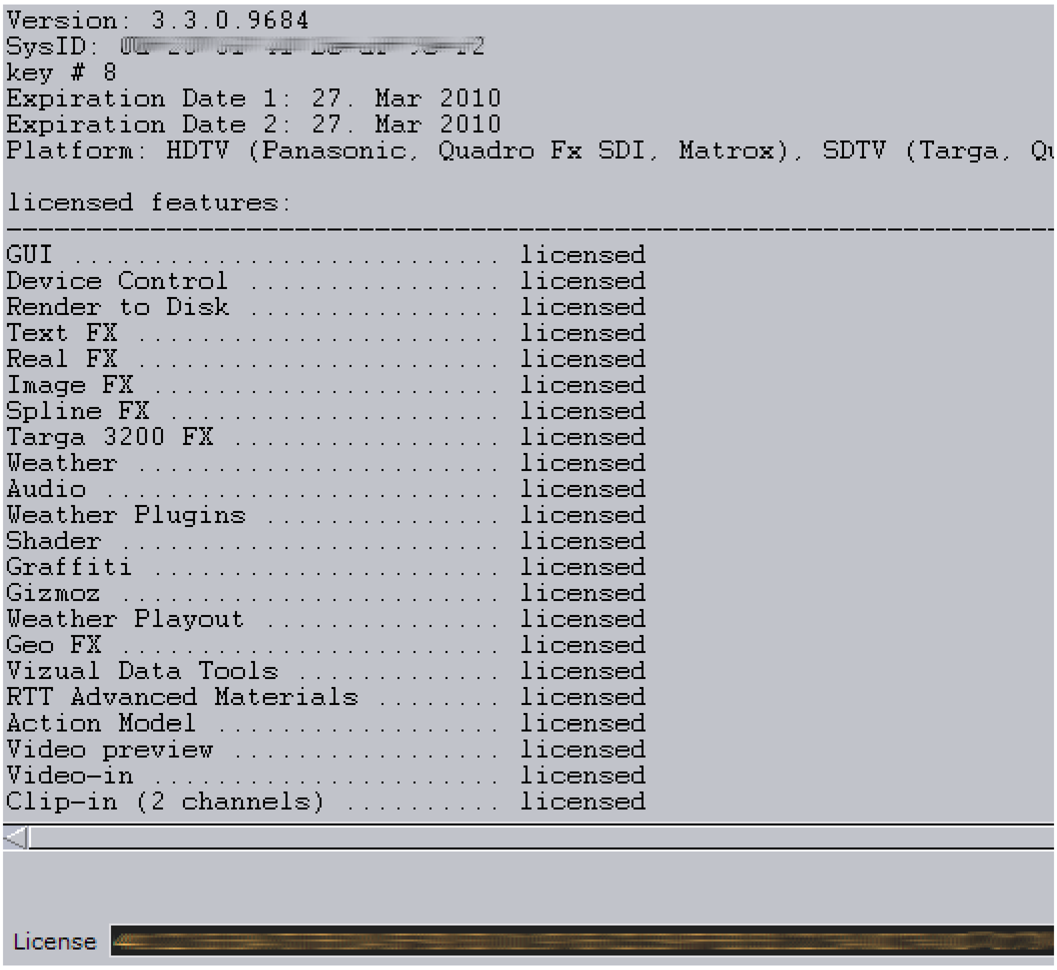
When working in 2.x mode, *Viz_2x* will automatically be added before the default messaging path. Furthermore, the paths sent by the external control applications will be converted to lower case (as the names of the data will be converted to lower case when importing). So, external control programs must not be re-written when migrating from Viz Artist 2 to 3.

- **Use 2.x mode**: Enables Viz Engine to run in 2.x mode.
- **Show converting Info in Viz-Console**: Enables the calls for scenes to be shown as green text.

Note: It is only recommended to use 2.x mode if new scenes are organized according to the old 2.x data structure.

10.26 Viz License Information

Figure 166: Viz License Information



```
Version: 3.3.0.9684
SysID: [redacted]
key # 8
Expiration Date 1: 27. Mar 2010
Expiration Date 2: 27. Mar 2010
Platform: HDTV (Panasonic, Quadro Fx SDI, Matrox), SDTV (Targa, Q

licensed features:
-------------------------------------------------------------------
GUI ............................ licensed
Device Control ................. licensed
Render to Disk ................. licensed
Text FX ........................ licensed
Real FX ........................ licensed
Image FX ....................... licensed
Spline FX ...................... licensed
Targa 3200 FX .................. licensed
Weather ........................ licensed
Audio .......................... licensed
Weather Plugins ................ licensed
Shader ......................... licensed
Graffiti ....................... licensed
Gizmoz ......................... licensed
Weather Playout ................ licensed
Geo FX ......................... licensed
Vizual Data Tools .............. licensed
RTT Advanced Materials ......... licensed
Action Model ................... licensed
Video preview .................. licensed
Video-in ....................... licensed
Clip-in (2 channels) ........... licensed
```

License [redacted]

In this section, information about the current Viz license is displayed. The section lists the various licensed features, for example the available plug-in packages.

- **License**: Enter a new license key into the License field, and click Save. Restart Viz in order for the new license key to take effect.

See Also

- Plugins

11 Geometry Plugins

Geometry plug-ins are housed in the following folders:

- Default
- Vizual Data Tools

11.1 Default

The following geom plugins are housed in the Default folder:

- 2D Line
- 2D Patch
- 2D Ribbon
- Alpha Map
- Arrow
- Circle
- Cloth
- Cog Wheel
- Cone
- Connector
- Cube
- Cylinder
- Cylinder3
- Dexter
- DisplacementMap
- Eclipse
- Fade Rectangle
- Filecard
- Graph
- Graph2d
- Icosahedron
- Image FX
- Noggi
- N Quad
- Pointer
- Polygon
- Rectangle
- Ring
- Roll
- Shape View
- Soft Clip Draw Pixels
- Sphere
- Spline Path
- Spline Strip

- Spring
- Torus
- Trio Scroll
- Wall
- Wave

11.1.1 2D Line

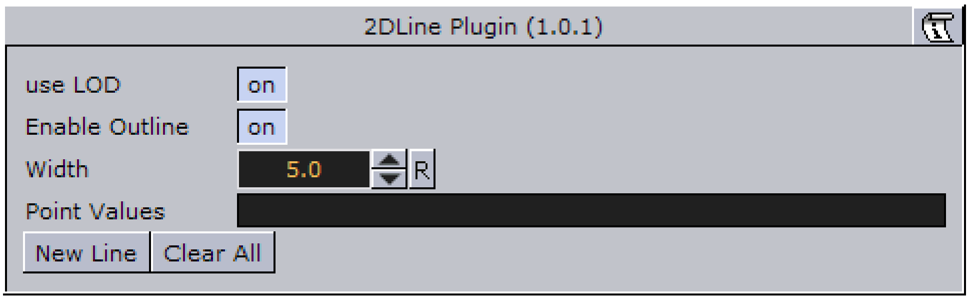
Draws a simple 2D line through given 2D coordinates, respectively point values.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\2DLine.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create 2D lines

Properties and Parameters

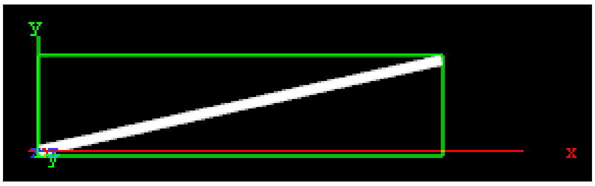


2DLine Plugin (1.0.1)

use LOD: on
Enable Outline: on
Width: 5.0
Point Values:

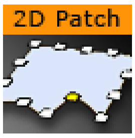
- **use LOD**: Enables level of detail. Range: On - Off. Default: ON
- **Enable Outline**: Enables drawing of an outline. Range: On - Off. Default: ON
- **Width**: Width of the drawn line. Range: 0.0 - 500.0. Default: 5.0
- **Point values**: List of x,y pairs separated by ':' character. x and y values are separated by blanks. Example: 0 0: 100 100: 200 70: 300 240: 400 280: 500 240: 600 400.
- **New Line**: Needs to be pressed before entering values to (i.e. Point values), or else the line will not be rendered visible.
- **Clear All**: Deletes all line-segments.

To create 2D lines



1. Enter the **Point Values.** For example: 0 0 : 200 45 entered in the Point Values field will create a line that starts at the point X1(0), Y1(0) and ends at point X2(200), Y2(45).
2. Click the **New Line** button to create the line.

11.1.2 2D Patch



The 2D Patch is a two dimensional planar curved grid of polygons. The grid is defined by control points that are located on the perimeter of the grid. The parameters of each one of the control points is the location X and Y, and the assigned texture coordinates U and V. The calculation of the internal grid points is done with a mix of a one-dimensional cubic spline and a two dimensional bilinear interpolation.

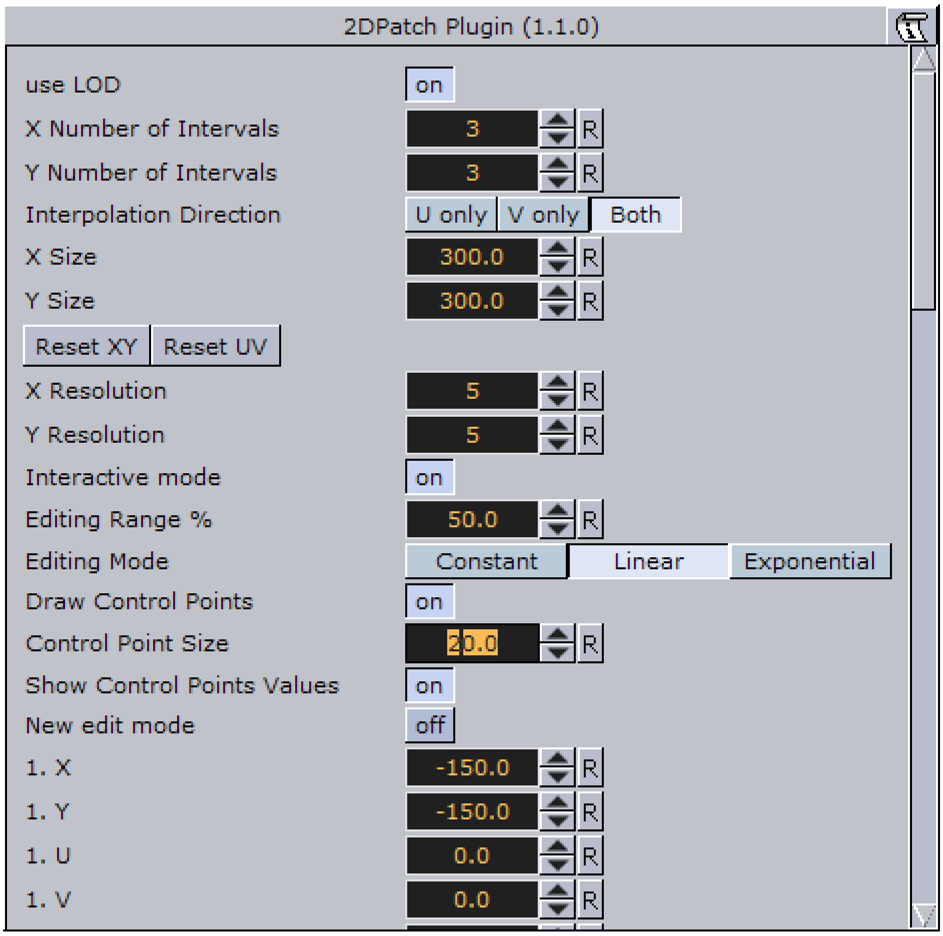
It is possible to specify the interpolation direction to be just in one direction, or in both. Depending on the shape that should be created, it is important to choose the right direction (for example it is not too difficult to create an annulus with this plug-in, after choosing the right interpolation direction along the radius).

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\2DPatch.vip`.

This section contains information on the following topics:

- Properties and Parameters
- Workflow
- Tips
- Known Limitations

Properties and Parameters



2DPatch Plugin (1.1.0)

use LOD: on
X Number of Intervals: 3
Y Number of Intervals: 3
Interpolation Direction: U only | V only | Both
X Size: 300.0
Y Size: 300.0
Reset XY | Reset UV
X Resolution: 5
Y Resolution: 5
Interactive mode: on
Editing Range %: 50.0
Editing Mode: Constant | Linear | Exponential
Draw Control Points: on
Control Point Size: 20.0
Show Control Points Values: on
New edit mode: off
1. X: -150.0
1. Y: -150.0
1. U: 0.0
1. V: 0.0

- **use LOD**: Enables/disables dynamic level of detail.
- **X and Y Number of Intervals**: Sets the number of intervals between the control points in the X and Y direction respectively. Parameter name: NX/NY.
- Interpolation Direction: Defines the direction of interpolation from the control points on the perimeter to the internal grid points. Parameter name: interDirect.
- **X Size, Y Size, Reset XY (button) and Reset UV (button)**: Gives initial values for the control points. The values of X and Y-Size will take affect just after hitting the Reset XY button. The Reset UV button is setting default values to the texture coordinates. Parameter name: SX/SY, resetXY/resetUV.
- **X Resolution and Y Resolution**: Defines the number of subdivisions to polygons between the control points in the X- and Y-direction respectively.Parameter name: MX/MY.
- **Interactive mode**: A toggle to enable or disable the interactive mode of direction manipulation with the mouse. Parameter name: interactive.
  - **Editing Range %**: Sets the amount of “effect” the movement of one point (move with Shift pressed) should have on the neighboring points. Parameter name: EditingRange.
  - **Editing Mode**: When moving a point, it is possible, by pressing Shift, to move the other points simultaneously. How the points move depends on the editing mode that is chosen. Parameter name: EditingMode.

  - **Constant (button)**: The other points will move their positions exactly equal to the one you move.
  - **Linear (button)**: The other points move in a linear way, based on the distance from the point that is being moved.
  - **Exponential (button)**: The other points move in an exponential way, based on the distance from the point it is being moved.
- **Draw Control Points**: A toggle to draw or not to draw the control points. Enables the Control Point Size parameter. Parameter name: drawControlPoints.
  - **Control Point Size**: Sets the control point size.
- **Show Control Points Values**: Enables the manipulation of the control points in the rendering window. Parameter name: controlPointSize.
- **New edit mode**: When enabled, and you press down the CTRL-key, you can see the control points and also you have the opportunity to move the control points. The “Interactive mode” and the handle “script/plugins event mode” must be enabled.
- **X, Y, U and V**: Sets position and texture coordinates for each of the control points. These values will typically be changed in the rendering window only with the mouse.

Note: There are 12 control points that each have their own X,Y,U and V parameters.

Workflow

1. Set the right number of control points, according to your estimation of how complex the patch is you are going to create. This number can be changed at any time. The plugin will redistribute the new points along the already defined patch.
2. Estimate the final length and width of the patch.
3. Move the patch to the right place in your scene. If it is going to be on some horizontal plane, it might be easier first to edit it in the default orientation, and afterwards rotate it to the right place. There is no problem to continue working on it after the rotation has been made, but because of the perspective, it might be less straightforward.
4. Turn on the Draw Control Points toggle and the Interactive toggle. While working with the cursor, you do not have to be very close to the control point that you would like to manipulate. The plugin will find by itself the nearest control point to where is your cursor is. The selected control point will get yellow color, as compared to all the rest that are white, to indicate that it has been selected for editing.
5. Note that you will be able to do the direct manipulation just if the object is the selected object in the Viz Artist container tree. Remember also, that regardless of the position of your cursor, if the patch is the selected object, and the interactive toggle is on, always one control point will become selected and will be changed while working with the cursor. Hitting the space key on the keyboard deselects the object currently selected, and then the cursor gets back its usual functionality until the patch object is selected again.
6. With the left button, according to the selection roles above, you can change the X and Y location of each of the control points.
7. Once the shape of the patch is more or less defined, it is a advisable to set the resolution parameter. Performance wise you should set it as low as possible, but it should be high enough to guarantee a smooth enough look.

8. Choose an image and drag it on the patch in the usual way. After doing it open the image texture editor and set the Mapping property to Vertex, and the Wrap property to Clamp.
9. The next step is to map more accurately the image on the patch. Set first the Texture Length according to your estimation, then choose the Texture Direction, and then play with the Texture Head Location to see the texture flowing along the patch. This is a good point to review again all the parameters you set before.
10. The last step is to define an animation if necessary. Typically with the patch the animation will be just on the Texture Head Location. To learn more about animation see Creating Animations.

CAUTION! While defining an animation on the parameters of a 2D Patch, you must make sure that the right window is open in Viz Artist. Without it, the changes that you are doing with the cursor will not take effect with regard to the animation.

Tips

1. It is sometimes useful to work with 2D Patch while in wireframe mode. If you have a texture with alpha, in order to see it correctly, you need to turn the image off with the small enable/disable button near the image icon.
2. Remember that if the 2D Patch is not selected, the interactive mode is not active. As result of this, the first click with the mouse on unselected object will be always with the normal Viz Artist functionality of the cursor, of moving, rotating and so on the object.
3. If you are done with the interactive session of defining a 2D Patch, turn the **Interactive** toggle **off**.
4. While working on a 2D Patch in the interactive mode, it is not possible to select another object with the cursor in the Viz Artist render output. The way to select another object is first to hit *Space*, and thereafter select the other object.
5. Doing animation on the 2D Patch is quite expensive performance wise. Be aware to it, and try to use as much as possible a small number of control points and polygons.
6. For the texture used for the 2D Patch, make sure that along the flow direction, you have on both texture edges a clean line with alpha equal to 0. Other wise you will get a wake of the image edges in front or behind it.
7. Do not change the number of control point along an animation. It is possible, if necessary, to change the resolution number.
8. While defining an animation, make sure to remove from it, if necessary, the toggle values of Show Control Points, Interactive Mode and so on.

Known Limitations

1. Note that when you use 2D Patch in a scene, by choosing an object in the Viz Artist rendering window, the Viz Artist user interface is not updated automatically.
2. While copying a 2D Patch, in order to make it work in the interactive mode, it is necessary to save the scene first.
3. If the texture coordinates on the control points are getting outside the [0,1] range artifacts may appear. Be aware to keep it within the [0,1] range.

11.1.3 2D Ribbon

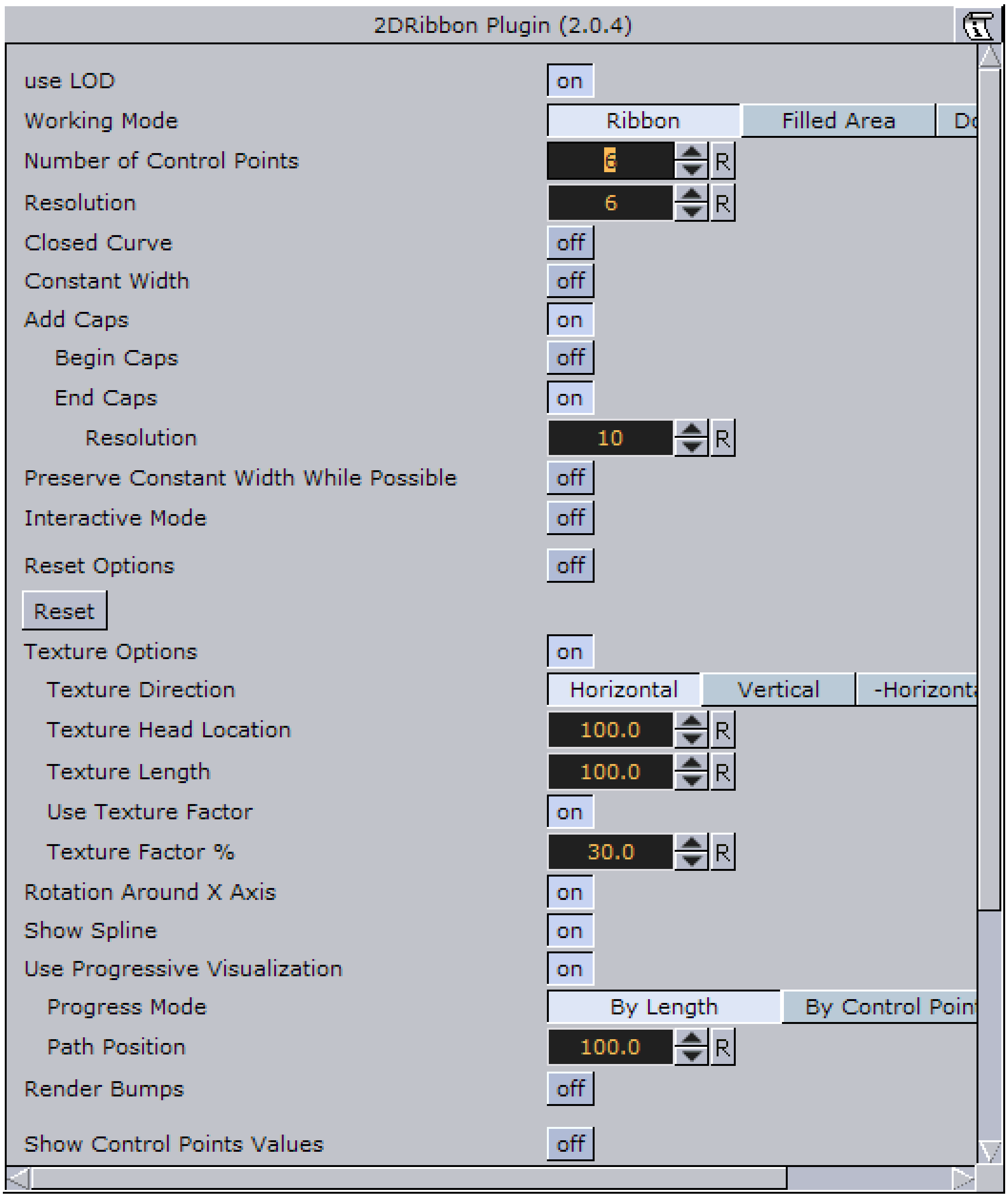
The 2D Ribbon is a curved strip. It is ideal for making customized curves to symbolize frontiers on a map or similar. The way the object is curved is easily edited in the property editor. It is possible to enable interactivity so you can drag the counterpoints of the ribbon by using the cursor.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\2DRibbon.vip`.

This section contains information on the following topics:

- Properties and Parameters
- Workflow
- Tips
- Known Limitations

Properties and Parameters



2DRibbon Plugin (2.0.4)

use LOD: on
Working Mode: Ribbon | Filled Area | Do
Number of Control Points: 6
Resolution: 6
Closed Curve: off
Constant Width: off
Add Caps: on
Begin Caps: off
End Caps: on
Resolution: 10
Preserve Constant Width While Possible: off
Interactive Mode: off
Reset Options: off
Reset
Texture Options: on
Texture Direction: Horizontal | Vertical | -Horizonta
Texture Head Location: 100.0
Texture Length: 100.0
Use Texture Factor: on
Texture Factor %: 30.0
Rotation Around X Axis: on
Show Spline: on
Use Progressive Visualization: on
Progress Mode: By Length | By Control Poin
Path Position: 100.0
Render Bumps: off
Show Control Points Values: off

- **Use LOD**: Enables/disables dynamic level of detail.
- **Working Mode**: Available modes are Ribbon, Filled Area and Double Outline. Default is the Ribbon mode.
- **Number of control points**: Sets the number of control points along the ribbon.
- **Texture Direction**: Defines which direction the texture is to be mapped along the 2D Ribbon.
- **Closed Curve**: Creates a closed a curve like a circle. The forward end touches the backward end.
- **Constant Width**: With this option enabled, the width of the ribbon is fixed. The value that defines the width is then the width of the first control point.
- **Texture Head Location**: Defines the position on the ribbon where the edge of the texture is mapped. The unit is percentage of the ribbon length.

- Texture Length: Defines the portion of the ribbon, in percentage, that the texture will be mapped onto.
- Resolution: Defines the number of subdivisions to polygons between the control points.
- Control Point Size: Sets the size of the control points, in the Viz Artist normal units.
- Reset Options: These properties are useful for giving initial values for the control points. The values of Ribbon Length and Ribbon Width will take affect after the Reset button has been pressed.
- Reset: Reset all created curves but doesn't delete the existing control points.
- Interactive Mode: A toggle to enable or disable the interactive mode of direction manipulation with the cursor.
  - Editing Range %: Sets the amount of "effect" the movement of one point (move with Shift) should have on the neighboring points.
  - Editing Mode: When moving a point, it is possible, by pressing Shift, to move the other points simultaneously. How the points move depends on the editing mode that is chosen.
  - Constant (button): The other points will move their positions exactly equal to the one you move.
  - Linear (button): The other points move in a linear way, based on the distance from the point that is being moved.
  - Exponential (button): The other points move in an exponential way, based on the distance from the point it is being moved.
- Texture Options: If this toggle is switched on some texture settings are available.
- Texture Direction: Shows the texture in different directions. For example if you have a pointer right-showing used as texture the pointer shows to the right side if the "Horizontal" option is activated. Otherwise to the left side if the "-Horizontal" option is set. Same for the "Vertical" and "-Vertical" option. The rendered pointer shows to the top or to the bottom side.
- Texture Head Location: Moves the texture either to top or bottom direction.
- Texture Length: This options stretches the texture.
- Use Texture Factor: When enabled (On) it enables the texture factor parameter.
- Texture Factor %: Sets the texture factor.
- Show Spline: Switches the display of the ribbon in the render output on and off.
- Use Progressive Visualization: Is this toggle is switched to on the activated parameters will affect the rendering.
- Progress Mode: You have two options "By Length" and by "Control Points". Slices the curve in x-direction by length or by the control point index.
- Path Position: It limits the workspace so you have fewer control points for preparing.
- Render Bumps: The curves are top-barbed which is useful for weather broadcasting.
- Avoid Internal Outline: This option is only available if the option "Working mode" is set to "Ribbon" and the "Closed Curve" option is set to "on". Also the "Expert" plug-in, located in "Container Plugins/Global" must be added to the same group which is holding the "2DRibbon" plug-in. Switch the "Draw mode" to "Outline" in the "Expert" plug-in for unlocking.
- Show Control Points Values: Enables the manipulation of the control points in the rendering window. X, Y, and W values sets the X/Y position and the width

of the curve at the different control points. These values will typically be changed through editing directly in the Viz Artist render output.

Workflow

1. Drag the 2D Ribbon from the Viz Artist built in geometries onto the scene tree.
2. Set the right number of control points, according to your estimation of how complex the ribbon you are going to create will be. This number can be changed at any time. The plugin will redistribute the new points along the already defined ribbon.
3. Estimate the final length and width of the ribbon, enter it to the right fields and hit the Reset button. You will get a straight ribbon with the right sizes.
4. Set correctly the toggle of the Constant Width according to your requirements.
5. Move the ribbon to the right place in your scene. If it is going to be on some horizontal plane, it might be easier first to edit it in the default orientation, and afterwards rotate it to the right place. There is no problem to continue working on it after the rotation has been made, but because of the perspective, it might be less straightforward.
6. Turn on the Draw Control Points toggle and the Interactive toggle. While working with the mouse, you do not have to be very close to the control point that you would like to manipulate. The plugin will find by itself the nearest control point to where is your cursor is. The selected control point will get yellow color, as compared to all the rest that are white, to indicate that it has been selected for editing.
7. Note that you will be able to do the direct manipulation just if the 2D Ribbon is the selected object in the Viz Artist container tree. Remember also, that regardless of the position of your cursor, if the ribbon is the selected object, and the interactive toggle is on, always one control point will become selected and will be changed while working with the cursor. Hitting Space deselects the object currently selected, and then the cursor gets back its usual functionality until the ribbon is selected again.
8. With the left button, according to the selection roles above, you can change the X and Y location of each of the control points. With the right button you can change the width of the ribbon. According to the Constant Width toggle the change will be just on the control point, or all over the ribbon. All three values are interpolated with cubic spline, and accordingly the affect of moving a control point is global, but with decreasing affect with the distance from it along the ribbon.
9. Once the shape of the ribbon is more or less defined, it is a advisable to set the resolution parameter. Performance wise you should set it as low as possible, but it should be high enough to guarantee a smooth enough look in the corners of the spline, and also a smooth enough mapping of the image/ texture on the ribbon.
10. Choose an image and drag it on the ribbon object in the usual way. After doing it open the texture editor and set the Mapping property to Vertex, and the Wrap property to Clamp.
11. The next step is to map the image more accurately on the ribbon. Set first the Texture Length according to your estimation, then choose the Texture Direction, and then play with the Texture Head Location to see the texture flowing along the ribbon. This is a good point to review again all the parameters you set before.
12. The last step is to define an animation if necessary. Typically with the ribbon the animation will be just on the Texture Head Location.

CAUTION! While defining an animation on the parameters of a 2D Ribbon, you must make sure that the right window is open in Viz Artist. Without it the changes that you are doing with the cursor will not take effect with regard to the animation.

Tips

1. It is sometimes useful to work with 2D Ribbon while in wireframe mode. If you have a texture with alpha, in order to see it correctly you need to turn the image off with the small enable/disable button near the image icon.
2. Remember that if the 2D Ribbon is not selected, the interactive mode is not active. As result of this, the first click on unselected object will be always with the normal Viz Artist functionality of moving, rotating and so on the object.
3. If you are done with the interactive session of defining a 2D Ribbon, turn the Interactive toggle off.
4. While working on a 2D Ribbon in the interactive mode, it is not possible to select another object with the cursor in the Viz Artist render output. The way to select another object is first to hit Space and thereafter select the other object.
5. Doing animation on the 2D Ribbon is quite expensive performance wise. Be aware to it, and try to keep the number of control points and polygons as low as possible.
6. For the texture used for the 2D Ribbon, make sure that along the flow direction, you have on both texture edges a clean line with alpha equal to 0. Other wise you will get a wake of the image edges in front or behind it.
7. Do not change the number of control point along an animation. It is possible, if necessary, to change the resolution number.
8. While defining an animation, make sure to remove from it, if necessary, the toggle values of Show Control Points, Interactive Mode and so on.

Known Limitations

1. Note that when you have in the scene a 2D Ribbon, by choosing an object in the Viz Artist render output, the Viz Artist user interface is not updated automatically.
2. While copying a 2D Ribbon, in order to make it work in the interactive mode, it is necessary to save the scene first.
3. If the texture coordinates on the control point are getting outside the [0,1] range artifacts may appear. Be aware to keep it within the [0,1] range.

See Also

- 2D Follow
- Creating Animations

11.1.4 Alpha Map

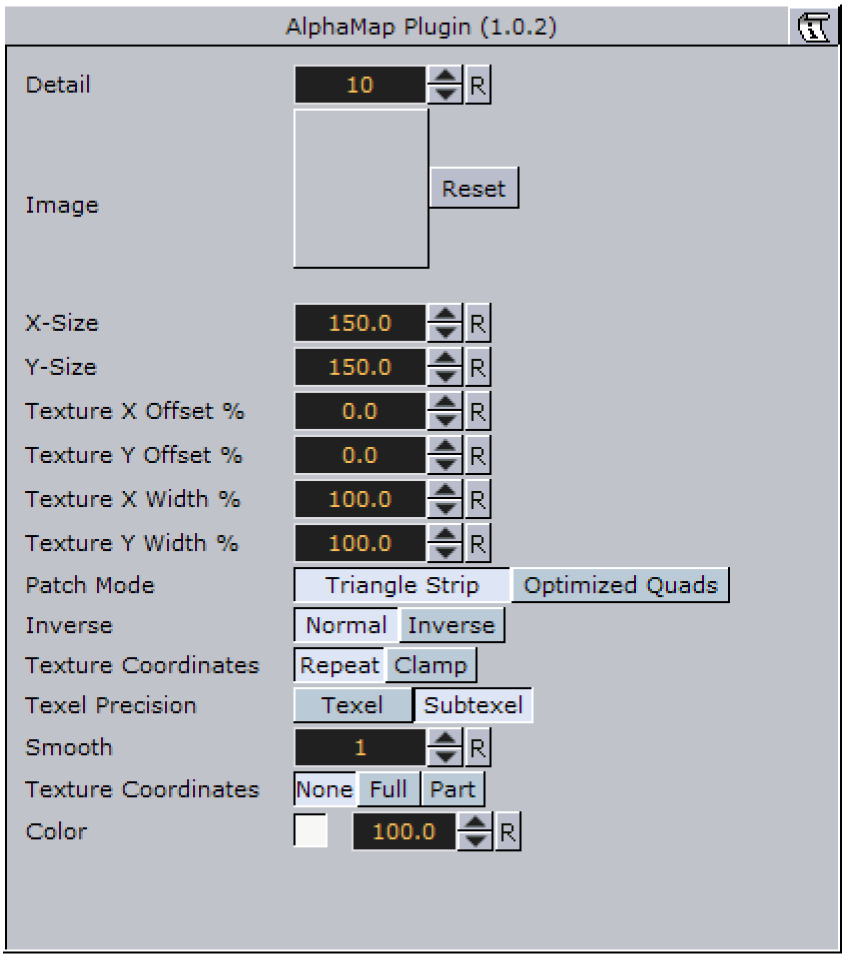
This plug-in enables you to create alpha maps by means of a grayscale image which cannot use an alpha image as an alpha channel directly. The intensity value of the grayscale image is translated to an opacity (alpha) value. An image can have another image as alpha channel, but a video source cannot. So if you want a video source to have an overlying alpha mask, you can use the alpha map.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\AlphaMap.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To add an alpha map

Properties and Parameters


AlphaMap Plugin (1.0.2)

Detail 10

Image

X-Size 150.0

Y-Size 150.0

Texture X Offset % 0.0

Texture Y Offset % 0.0

Texture X Width % 100.0

Texture Y Width % 100.0

Patch Mode Triangle Strip | Optimized Quads

Inverse Normal | Inverse

Texture Coordinates Repeat | Clamp

Texel Precision Texel | Subtexel

Smooth 1

Texture Coordinates None | Full | Part

Color 100.0

- **Detail**: Sets the detail level of the map, the lower the value the higher the level of detail.
- **Image**: Drag the image you want to use creating the map, onto the drop zone here. The image must be a valid alpha image.
- **X-size**: Sets the size of the map along the X-axis.
- **Y-size**: Sets the size of the map along the Y-axis.
- **Texture X Offset %**: Sets the offset of the texture in the X-axis.
- **Texture Y Offset %**: Sets the offset of the texture in the Y-axis.
- **Texture X Width %**: Stretches or compresses the texture in the X-axis.
- **Texture Y Width %**: Stretches or compresses the texture in the Y-axis.
- **Patch mode**:
  - **Triangle Strip**: Uses the same size of triangle polygon on the whole surface.
  - **Optimized Quads**: Uses bigger triangle polygons where the surface of the displacement map is flat, thereby reducing the total number of polygons and improving performance.
- **Inverse**: Mirrors all coordinates.
- **Texture Coordinates Repeat/Clamp**: Sets if the texture is the be repeated or clamped when it is to small to fit onto the alpha maps rectangle. Clamp stretches the texture to make it fit, Repeat starts drawing the texture over again.
- **Texel Precision**: Use “Subtexel” for better texture image quality.

    - Texel no subpixel/subtexel correction are made.
    - With Subtexel there is a limited number of pixels available on the screen, if a line does not run through a real pixel, it must be moved to the nearest one, this introduces a positional error. If subtexel is selected Viz Artist will break up pixels into smaller sub pixels in memory so that the line can be drawn to the nearest sub pixel.
- **Smooth**: Is used to create a smooth look of the map, without reducing the polygon details of the map. In some cases you need the number of polygons you have to obtain the correct lightning.
- **Texture Coordinates**: Decides if texture coordinates are to be created, and to which level, either none, full or partially. You need this enabled to some level if a texture is applied using a vertex mapping. Texture mapping is resource consuming, so if you do not need it keep it off.
- **Color**: Sets the color of the alpha map rectangle.

To add an alpha map

1. Drag and drop the Alpha Map plug-in onto the tree.
2. Add the alpha image onto the alpha map editor's Image placeholder and drag the video source icon onto the alpha map's container.

11.1.5 Arrow

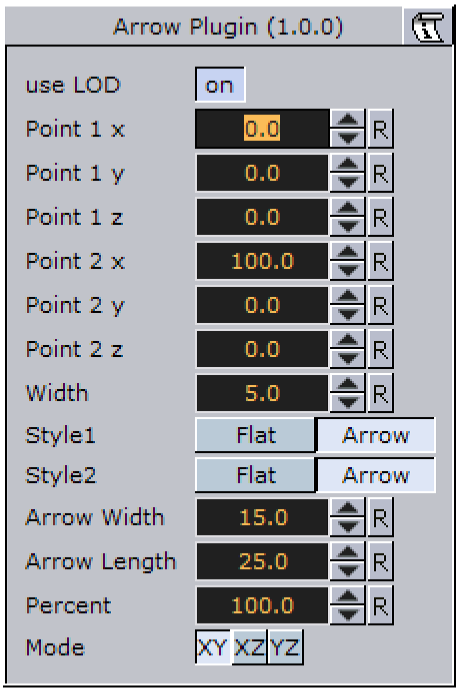
The Arrow built in geometry allows you to create and customize an arrow. The arrow is a 2D object. By clicking on the arrow icon in a container, the arrow editor opens. The default arrow has two heads. At the tip of each arrowhead the reference points are placed. The left point is 1 and the right is 2.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Arrow.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create an arrow

Properties and Parameters

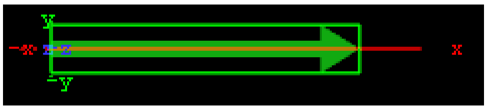


Arrow Plugin (1.0.0)

use LOD: on
Point 1 x: 0.0
Point 1 y: 0.0
Point 1 z: 0.0
Point 2 x: 100.0
Point 2 y: 0.0
Point 2 z: 0.0
Width: 5.0
Style1: Flat | Arrow
Style2: Flat | Arrow
Arrow Width: 15.0
Arrow Length: 25.0
Percent: 100.0
Mode: XY | XZ | YZ

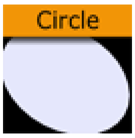
- **Use LOD**: Enables/disables dynamic level of detail.
- **Point 1 X**: Allows you to alter point 1s X offset.
- **Point 1 Y**: Allows you to alter point 1s Y offset.
- **Point 1 Z**: Allows you to alter point 1s Z offset.
- **Point 2 X**: Allows you to alter point 2s X offset.
- **Point 2 Y**: Allows you to alter point 2s Y offset.
- **Point 2 Z**: Allows you to alter point 2s Z offset.
- **Width**: Sets the width of the arrow shaft.
- **Style 1**: Allows you to select if the arrows end at point 1 should have a head or not.
- **Style 2**: Allows you to select if the arrows end at point 2 should have a head or not.
- **Arrow Width**: Sets the width of the arrow heads.
- **Arrow Length**: Sets the length of the arrow head.
- **Percent**: Scales the arrow in percent of the size defined by X-, Y- and Z-values.
- **Mode**: Defines if the arrow lies in an XY, XZ or YZ plane.

To create an arrow



1. Create a group and add the Arrow plug-in to it.
2. Add a material and/or an image to the same container as the arrow plug-in to add color and/or texture to it.

11.1.6 Circle



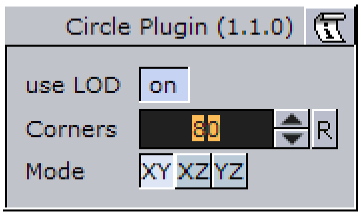
With this plugin you can create a circle with different corner levels.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Circle.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a circle

Properties and Parameters



Circle Plugin (1.1.0)

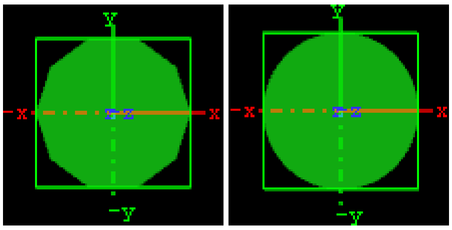
use LOD: on

Corners: 80

Mode: XY XZ YZ

- **Use LOD**: Enables/disables dynamic level of detail.
- **Corners**: Allows you to decide the number of corner the circle is to be constructed of. The circle is set together by tiled triangles and the higher the number of corners is set, the more triangles are used to construct the circle. The higher the value is set the more rendering performance will be used.
- **Mode**: Changes the space perspective. Possible values are “XY” (front – view), “XZ” (bottom – view), “YZ” (left – view)

To create a circle



1. Create a group and add the Circle plug-in to it.
2. Add a material and/or an image to the same container as the circle plug-in to add color and/or texture to it.
3. Set the number of corners.

11.1.7 Cloth

Cloth provides a simulation of an elastic vertex system. The vertex system can be configured with many different parameters. Most of the parameters depend on each other, and some must be set in conjunction with others. The stress of the system will be mentioned below, and the rule is that the greater the stress is the greater the chance is that the system will run amok (unstable). With low stress factors the system is completely stable.

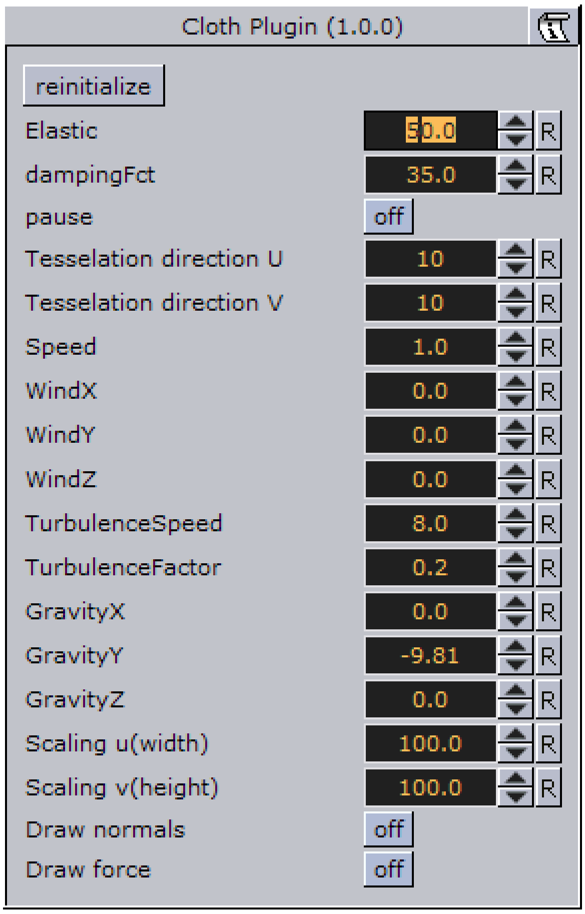
Adding the Cloth plug-in to a container will also add the Expert plug-in.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Cloth.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a cloth effect

Properties and Parameters



Cloth Plugin (1.0.0)

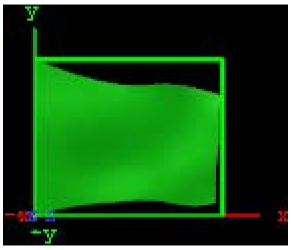
reinitialize

Elastic	50.0
dampingFct	35.0
pause	off
Tesselation direction U	10
Tesselation direction V	10
Speed	1.0
WindX	0.0
WindY	0.0
WindZ	0.0
TurbulenceSpeed	8.0
TurbulenceFactor	0.2
GravityX	0.0
GravityY	-9.81
GravityZ	0.0
Scaling u(width)	100.0
Scaling v(height)	100.0
Draw normals	off
Draw force	off

- **reinitialize**: It is rendering the object from beginning.
- **Elastic**: Changes the elastic properties of the system. If you set it to 0.1 you will have a system which will have properties like latex, and at 1.0 more like cloth. When this factor increases the stress of the system increases.
- **dampingFct**: Specifies how much speed each vertex should loose. The lower this factor is the less speed the system will loose. The stress of the system decreases when this factor increases.
- **pause**: Pauses all calculations and freezes the system's animation.
- **Tesselation direction U and V**: Sets the tesselation in the two directions on the systems surface. The calculation time needed will increase with more vertices. The stress of the system also increases slightly when the number of vertices increase.
- **Speed**: This factor is directly connected to how many calculations needed per frame. The less calculation done the slower the system will move. You can also increase the number of calculations and the system will move faster. This factor does not affect the stability of the system.
- **WindX, Y and Z**: Three factors combined gives the speed and direction of the wind affecting the system. If the absolute value of this factor increase then the stress on the system will also increase.
- **TurbulenceSpeed**: Three factors combined gives the speed and direction of the wind affecting the system. If the absolute value of this factor increase then the stress on the system will also increase.
- **TurbulenceFactor**: This factor multiplies up the wind speed.

- **GravityX, Y and Z**: A vector which defines gravity affecting all vertices of the system. When this factor increases the stress of the system increases.
- **Scaling u(width)**: Changes the width of the rectangle which is including the animated cloth without increasing the number of triangles.
- **Scaling v(height)**: Changes the height of the animated rectangle which is including the animated cloth without increasing the number of triangles.
- **Draw normals**: Shows the normals for each triangle.
- **Draw force**: Shows the appealing wind in color.

To create a cloth effect



1. Add the **Cloth** plug-in to a container.
2. Add a **material** to the same container.
3. Set **WindX** to 2.0.
4. Click the **reinitialize** button to see the result.

See Also

- Expert

11.1.8 Cog Wheel

Use this plug-in to create a cog wheel. It has a range of with that has a range of Properties and Parameters that can be set to adjust the look and feel of the cog wheel.

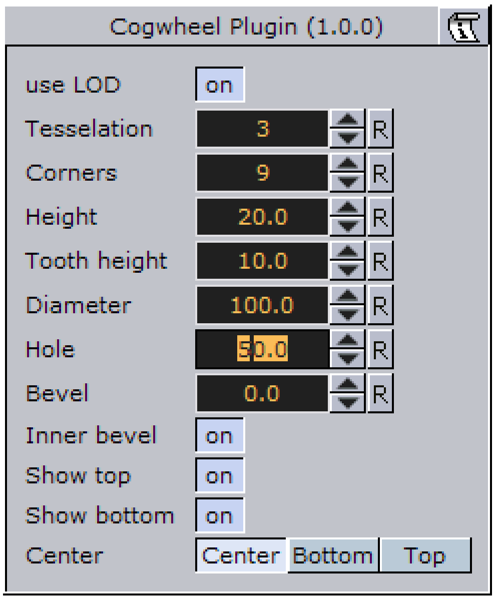
To avoid decreasing the system performance, set the tessellation to a lower value since it does not make much difference in quality.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Cogwheel.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a cogwheel

Properties and Parameters

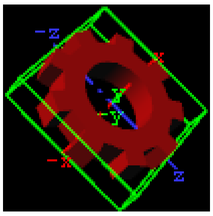


Cogwheel Plugin (1.0.0)

use LOD	on
Tesselation	3
Corners	9
Height	20.0
Tooth height	10.0
Diameter	100.0
Hole	50.0
Bevel	0.0
Inner bevel	on
Show top	on
Show bottom	on
Center	Center Bottom Top

- **Use LOD** enables/disables dynamic level of detail.
- **Tessellation** sets the level of detail.
- **Corners** sets the number of corners the cogwheel is to be constructed of. The cog wheel gets one tooth for each corner, so the number of teeth changes correspondingly.
- **Height** sets the height or width of the cog wheel.
- **Tooth height** sets the height of the cog wheel teeth.
- **Diameter** sets the diameter of the cog wheel.
- **Hole** creates and sets the size of a hole in the cog wheel.
- **Bevel** sets the degree of bevel at the cog wheel.
- **Inner Bevel** enables or disables bevel in the cog wheel hole.
- **Show Top** enables or disables visualization of the top.
- **Show Bottom** enables or disables visualization of bottom.
- **Center** allows you to select where the geometrical centre should be placed on the cog wheel, either **Center**, **Bottom** or **Top**.

To create a cogwheel



1. Add the **Cog Wheel** plug-in to a container.
2. Add a **material** to the same container.
3. Set **Hole** parameter to 50.0.
4. Set the **transformation** parameters for the container to:
   - Set **rotation** X: 90.0, Y:45.0 and Z: -45.0.

11.1.9 Cone

This plug-in creates a cone geometry.

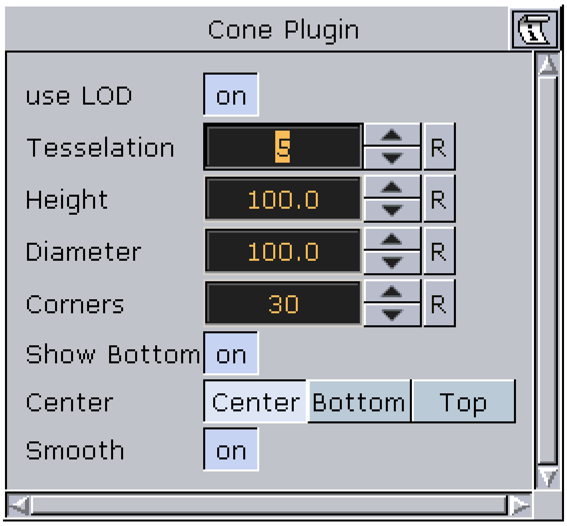
Set the tesselation to a reasonable value since it does not make a great difference in quality.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Cone.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a cone

Properties and Parameters



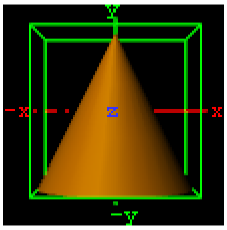
Cone Plugin

use LOD: on
Tesselation: 5
Height: 100.0
Diameter: 100.0
Corners: 30
Show Bottom: on
Center: Center | Bottom | Top
Smooth: on

- **Use LOD**: Enables/disables dynamic level of detail.
- **Tessellation**: Sets the degree of detail. Parameter name: Tesselation.
- **Height**: Sets the height of the cone. Parameter name: height.
- **Diameter**: Sets the diameter of the cone. Parameter name: diameter.
- **Corners**: Allows you to decide the number of corners the cone is to be constructed of. Parameter name: corners.
- **Show Bottom**: Enables or disables visualization of the cone bottom. Parameter name: show_bottom.
- **Center**: Allows you to select where the geometrical center should be placed on the cone, either Center, Bottom or Top. Parameter name: center.
- **Smooth**: Enables a smoothing of the cone edges. Parameter name: smooth.
- **Rounded Tip**: Enables the user to adjust the roundness of the cone's tip. Parameter name: roundedTip.

- **Rounded Tip Height**: Sets the parameter for the roundness of the tip. The angle of the cone's side does not change, the rounded tip is created by “taking away” from the pointed tip. Therefore the cone will not have the full height.

To create a cone



1. Add the **Cone** plug-in to a container.
2. Add a **material** to the same container.

11.1.10 Connector

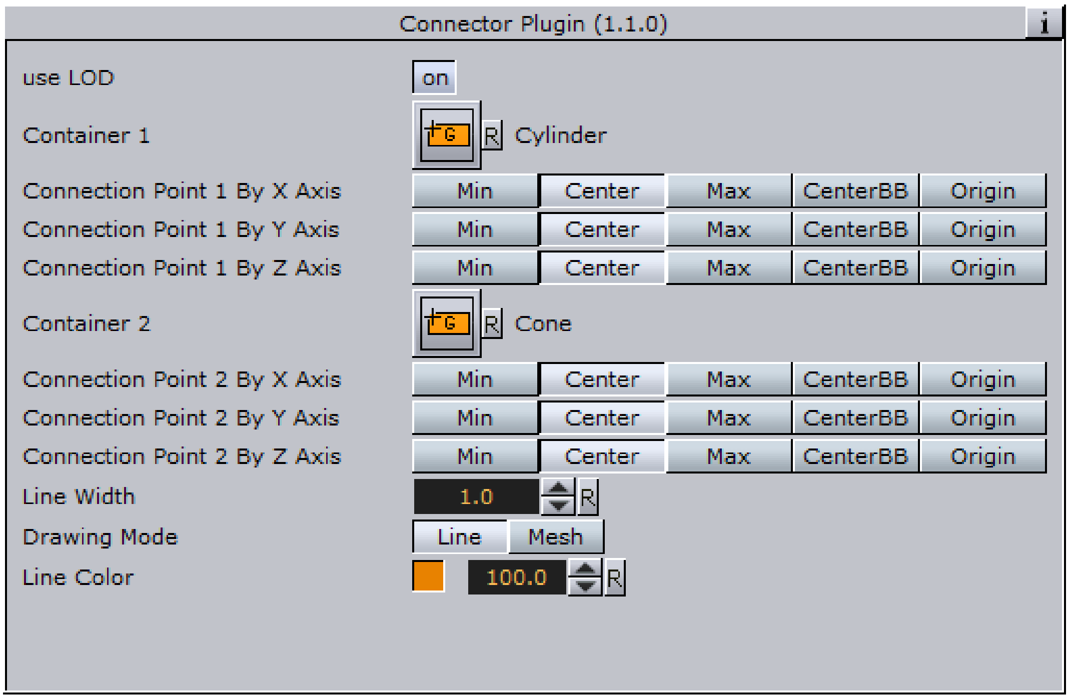
This plug-in is useful to connect two objects with a line in a desired color and width.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Connector.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To connect two objects

Properties and Parameters

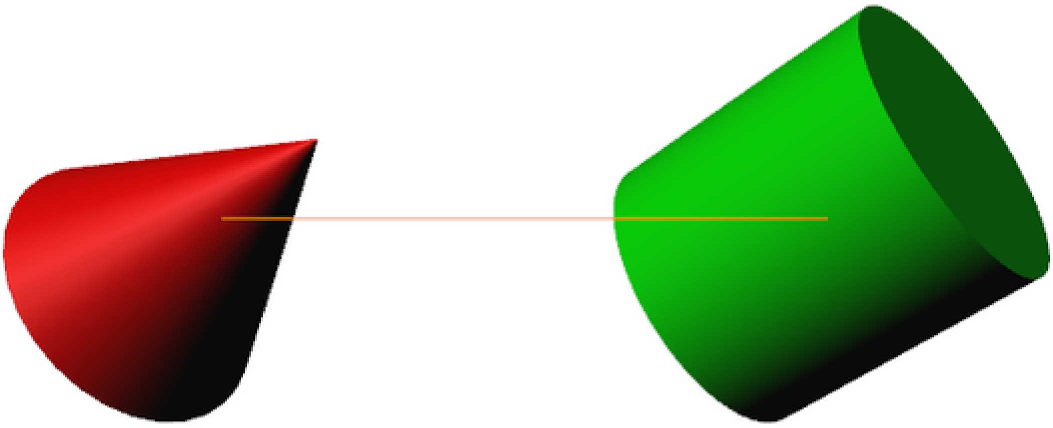


Connector Plugin (1.1.0)

use LOD	on
Container 1	R Cylinder
Connection Point 1 By X Axis	Min Center Max CenterBB Origin
Connection Point 1 By Y Axis	Min Center Max CenterBB Origin
Connection Point 1 By Z Axis	Min Center Max CenterBB Origin
Container 2	R Cone
Connection Point 2 By X Axis	Min Center Max CenterBB Origin
Connection Point 2 By Y Axis	Min Center Max CenterBB Origin
Connection Point 2 By Z Axis	Min Center Max CenterBB Origin
Line Width	1.0 R
Drawing Mode	Line Mesh
Line Color	100.0 R

- **Use LOD**: Enables/disables dynamic level of detail.
- **Container 1 and 2**: Placeholder for container 1 and 2. Drag the container onto the drop zone. Click R to remove the container.
- **Connection Point 1 and 2 By X, Y and Z Axis**: is the X, Y and Z-value of container 1 and 2 where the line should start. Can be set to Min, Center, Max, CenterBB or Origin.
- **Line Width**: Sets the width of the connecting line.
- **Drawing Mode**: Avilable options are Line and Mesh.
- **Line Color**: Defines the color of the line.

To connect two objects



1. Create a new group and add the **Connector** plug-in to it
2. Create two new group containers and add a geometry object (e.g. Cylinder and Cone) to each group

3. Open each container’s transformation editor and move the objects a part
4. Add a material and/or an image to the geometry objects
5. Open the connector plug-in editor and drag and drop the two containers to the Container 1 and Container 2 placeholders, respectively
6. Adjust the color line
7. Open the Connector plugin editor and play with the settings
8. Add more objects to Container 2 and animate it

11.1.11 Cube

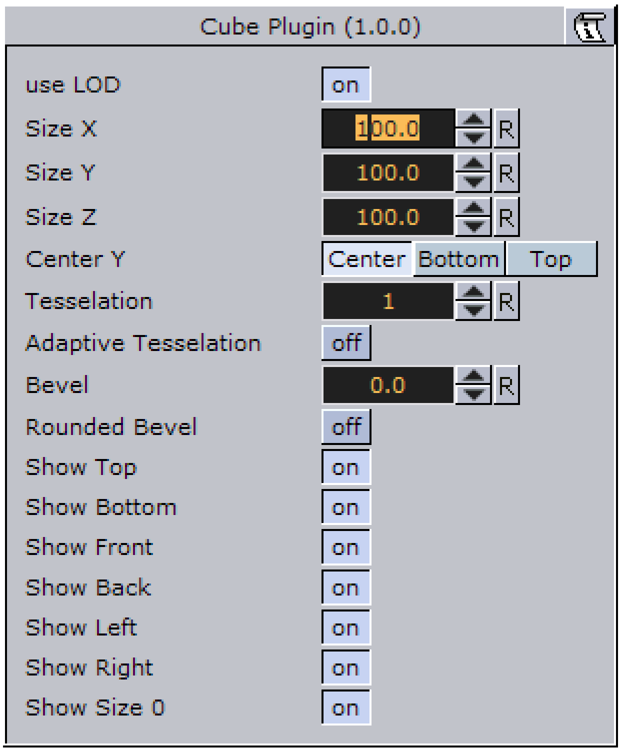
Use this plug-in to create cubes with particular widths, heights, depths and other attributes.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugin\Cube.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a cube

Properties and Parameters



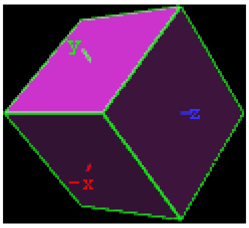
Cube Plugin (1.0.0)

use LOD	on
Size X	100.0
Size Y	100.0
Size Z	100.0
Center Y	Center Bottom Top
Tesselation	1
Adaptive Tesselation	off
Bevel	0.0
Rounded Bevel	off
Show Top	on
Show Bottom	on
Show Front	on
Show Back	on
Show Left	on
Show Right	on
Show Size 0	on

- **Use LOD**: Enables/disables dynamic level of detail.

- **Size X**: Sets the size of the cube in the X-axis. Parameter name: size_X.
- **Size Y**: Sets the size of the cube in the Y-axis. Parameter name: size_Y.
- **Size Z**: Sets the size of the cube in the Z-axis. Parameter name: size_Z.
- **Center Y**: Sets the position of the cube center along the Y-axis. You can choose between Center, Bottom or Top. Parameter name: center_Y.
- **Tessellation**: Sets the degree of detail. Parameter name: tesselation.
- **Adaptive Tesselation**: Adjusts tesselation on each axis separately, depending on the extension of the cube on that axis. The reference value is an extension of 100. For a face of size 200 the tesselation-parameter will be doubled, for a face of size 50 the tesselation is halved. The automatically adapted tesselation still never exceeds its maximum value of 100. Parameter name: adaptiveTesselation.
- **Bevel**: Adds a bevel of the given size to corners and edges of the cube. The size of the bevel reduces the size of the cube's axis-aligned faces which: in case 'Adaptive tesselation' is enabled: will cause them to become less tesselated accordingly. Tesselation of the bevel itself is affected by 'adaptive tesselation' too. Parameter name: bevel.
- **Rounded Bevel**: If enabled the beveled edges and corners will appear 'rounded'. How much the round bevel is tesselated will be affected by 'adaptive tesselation. Parameter name: roundedBevel.
- **Show Top, Bottom, Front, Back, Left, Right**: Shows or hides the cube's face and adjacent corners and edges. Parameter names: Top, Bottom, Front, Back, Left and Right.
- **Show Size 0**: Enables/disables display of the cube at axis value zero (0). If one of the axis values are set to 0 the cube will remain visible if the setting is enabled (On). Disabling (Off) it will make the cube become invisible as long as one of the axis values are 0. A situation where this parameter is useful is when creating bar chart animations. In such animations it would then be possible to hide the bar at value 0 by disabling this setting.

To create a cube



1. Create a new group container and add the Cube plug-in to it.
2. Add a material and/or an image to the same container.
3. Open the cube editor and set Rotation Y and Z values to 45.0.

11.1.12 Cylinder

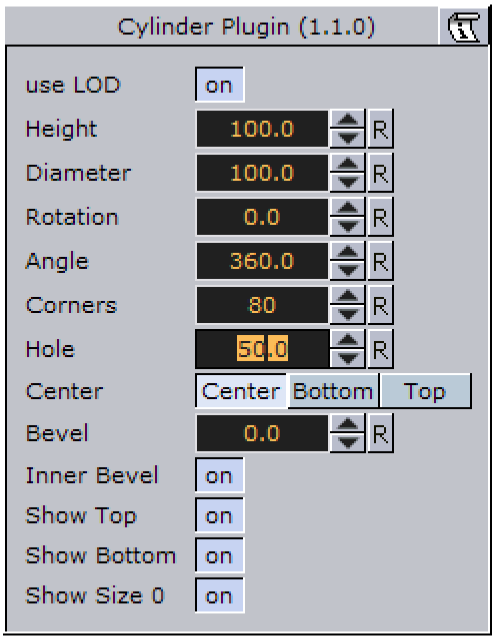
Use this plug-in to create cylinders with different heights, widths and depths and other attributes.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Cylinder.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a cylinder

Properties and Parameters

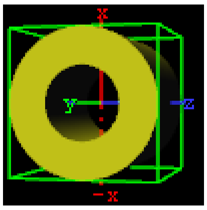


Cylinder Plugin (1.1.0)

use LOD: on
Height: 100.0
Diameter: 100.0
Rotation: 0.0
Angle: 360.0
Corners: 80
Hole: 50.0
Center: Center Bottom Top
Bevel: 0.0
Inner Bevel: on
Show Top: on
Show Bottom: on
Show Size 0: on

- **Use LOD**: Enables/disables dynamic level of detail.
- **Height**: Sets the height of the cylinder.
- **Diameter**: Sets the diameter of the cylinder
- **Rotation**: Sets the rotation around X-axis.
- **Angle**: Sets the opening angle from 0-360°. As the value decreases below 360° an increasing angle will open in the cylinder.
- **Corners**: Sets the number of corners the cylinder is to be made up from.
- **Hole**: Creates a hole in the cylinder, making it into a tube.
- **Center**: Sets the position for the center. You can choose between Center, Bottom or Top.
- **Bevel**: Sets the size of bevel at the cylinder.
- **Inner Bevel**: Enables/disables bevel in the cylinder hole if bevel is set at the bevel parameter.
- **Show Top**: Turns off/on visualization of cylinder top.
- **Show Bottom**: Turns off/on visualization of cylinder bottom.
- **Show size 0**: Enables/disables display of the cylinder at height value 0. If you set height to zero, the cylinder is visible by default. If you disable this option, the cylinder becomes invisible. If you make a pie chart visualization you will have use for this option.

To create a cylinder



1. Create a new group container and add the Cylinder or Cylinder3 plug-in to it.
2. Open the Cylinder editor and set Hole to 50.0.
3. Add a material and/or an image to the same container.
4. Open the cylinder editor and set Rotation Y to 75.0 and Z to 90.0.

11.1.13 Cylinder3

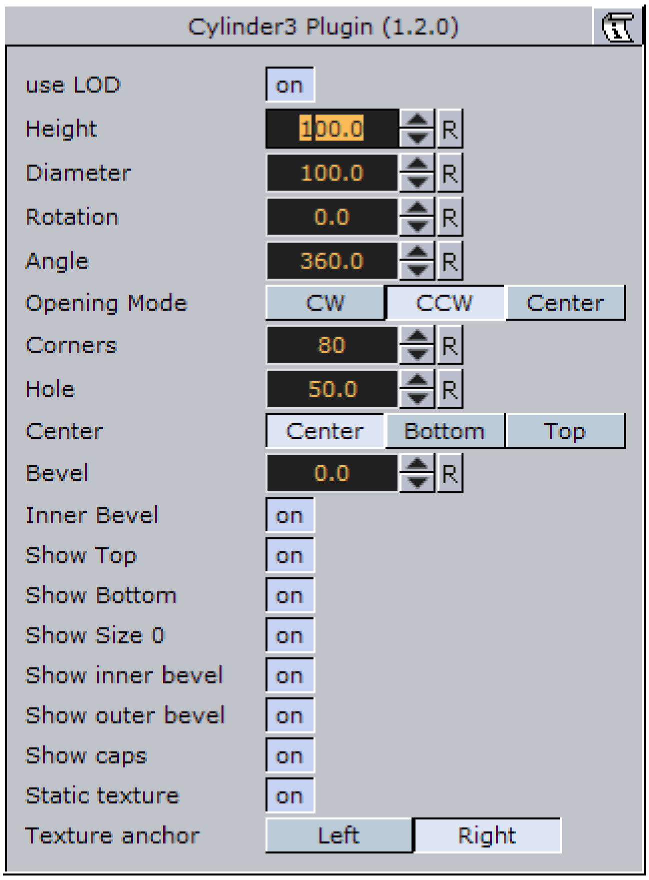
Use this plug-in to create cylinders. In relation to the ordinary Cylinder plug-in you have more options and settings available with the Cylinder 3 plug-in.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Cylinder3.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a cylinder

Properties and Parameters



Cylinder3 Plugin (1.2.0)

use LOD	on
Height	100.0 R
Diameter	100.0 R
Rotation	0.0 R
Angle	360.0 R
Opening Mode	CW CCW Center
Corners	80 R
Hole	50.0 R
Center	Center Bottom Top
Bevel	0.0 R
Inner Bevel	on
Show Top	on
Show Bottom	on
Show Size 0	on
Show inner bevel	on
Show outer bevel	on
Show caps	on
Static texture	on
Texture anchor	Left Right

- **Use LOD**: Enables/disables dynamic level of detail.
- **Height**: Sets the height of the cylinder.
- **Diameter**: Sets the diameter of the cylinder
- **Rotation**: Sets the rotation around X-axis.
- **Angle**: Sets the opening angle from 0-360°. As the value decreases below 360° an increasing angle will open in the cylinder.
- **Opening Mode**: Describes in which direction the cylinder will be opened if you change the “Angel” parameter.
- **Corners**: Sets the number of corners the cylinder is to be made up from.
- **Hole**: Creates a hole in the cylinder, making it into a tube.
- **Center**: Sets the position for the center. You can choose between Center, Bottom or Top.
- **Bevel**: Sets the size of bevel at the cylinder.
- **Inner Bevel**: Enables/disables bevel in the cylinder hole if bevel is set at the bevel parameter.
- **Show Top**: Turns off/on visualization of cylinder top.
- **Show Bottom**: Turns off/on visualization of cylinder bottom.
- **Show size 0**: Enables/disables display of the cylinder at height value 0. If you set height to zero, the cylinder is visible by default. If you disable this option, the cylinder becomes invisible. If you make a pie chart visualization you will have use for this option.
- **Show inner bevel**: If this toggle is off, the inner bevel, respectively the inner cylinder is not rendered.

- **Show outer bevel**: If this toggle is off, the outer bevel, respectively the outer cylinder is not rendered.
- **Show caps**: This setting only affects the geometry object if the angle is lesser than 360°.
- **Static texture**: If you change the texture mapping of the texture to “Vertex”: mode this option will bend or not bend the texture around the object.
- **Texture anchor**: It changes the texture orientation in x-direction to left or right. You can see the result if the angle is lesser than 360°.

11.1.14 Dexter

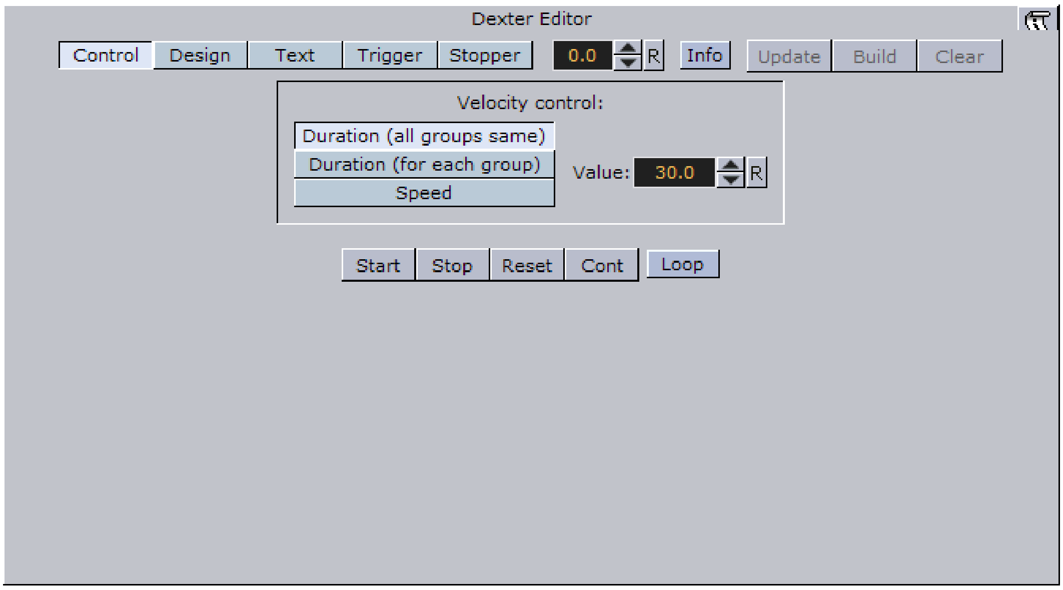
Dexter is a macro-based text-ticker tool that generates static horizontal text crawl and vertical text rolls. The Dexter editor is organized in five sections, Control, Design, Text, Trigger and Stopper. When adding the plug-in, two sub-containers are also added; Animation (visible) and Templates (hidden). In addition it has a set of scripting possibilities.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Dexter.vip`.

This section contains information on the following topics:

- Control
- Design
- Text
- Trigger
- Stopper
- Control Buttons
- Parameters
- Script Programming - Parameters
- Script Programming - Tokens
- Script Programming - Syntax
- Script Programming - Templates
- Script Programming - Newline Logic
- Script Programming - Triggers

Control



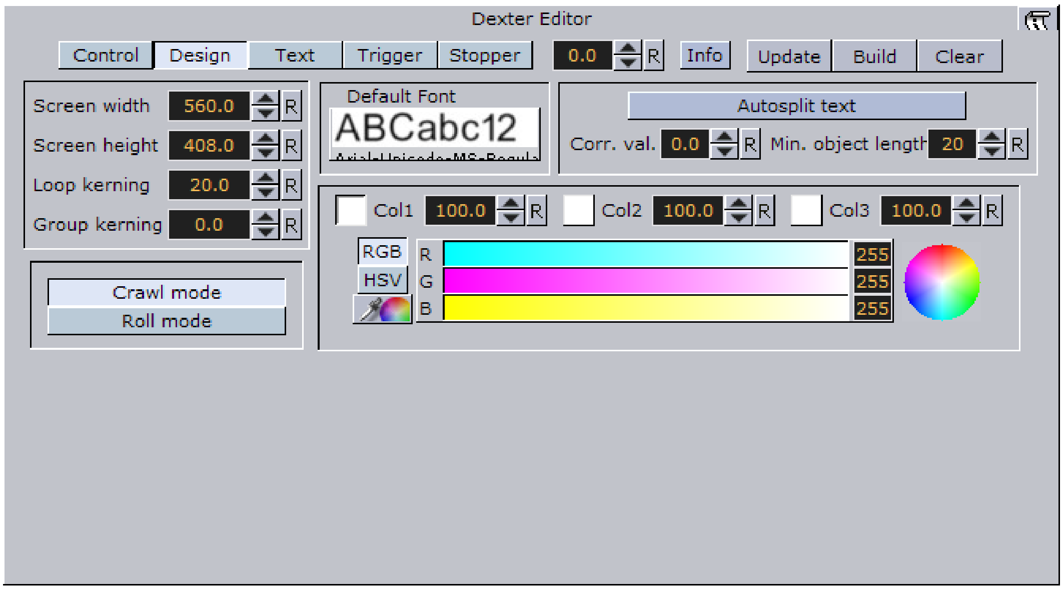
The control section allows you to start, stop, continue and reset the scroll/roll. In addition it contains buttons that allows you to select the velocity mode and speed of the ticker.

- **Start** starts the animation.
- **Stop** stops the animation.
- **Reset** resets the animation.
- **Cont** continues the animation.
- **Loop** loops the scroller/crawl.

The velocity control allows you to define how the speed of the crawl/roll is to be calculated.

- **Duration (all groups same)** makes all group use the same amount of time to crawl/scroll from start to end. The duration will be measured at the group containing the longest text string.
- **Duration (for each group)** makes all groups use exactly the same time. As a consequence, longer text strings will scroll faster then shorter ones.
- **Speed** makes all groups scroll at the same speed, as a consequence long message will take longer time to scroll than shorter messages. The speed unit is pixels per frame divided by 10.

Design



0.0

Screen width 560.0

Screen height 408.0

Loop kerning 20.0

Group kerning 0.0

Corr. val. 0.0

Min. object length 20

Col1 100.0

Col2 100.0

Col3 100.0

R 255

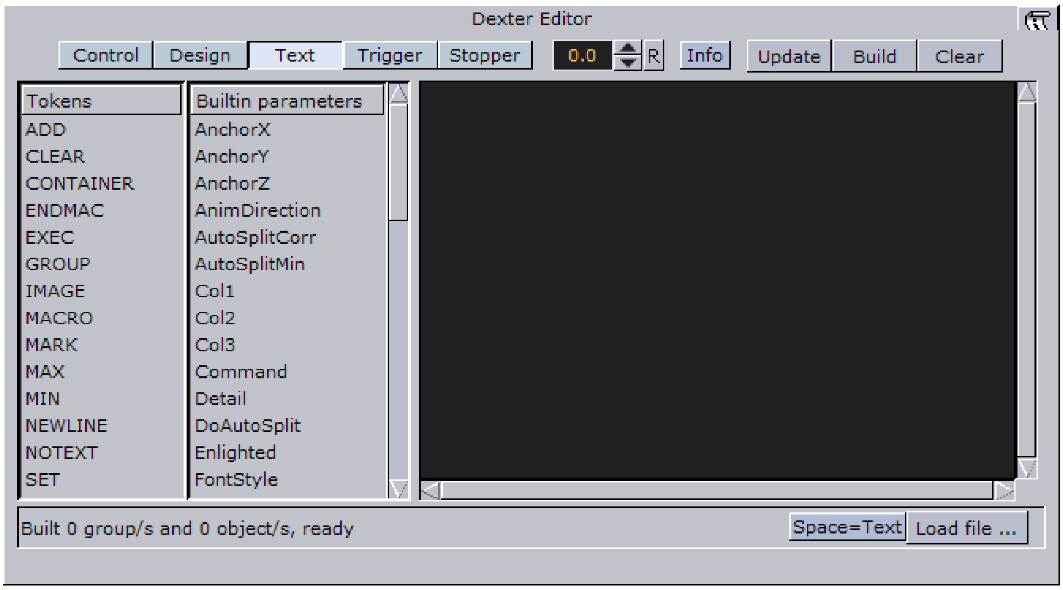
G 255

B 255

In the design section of the dexter editor you can set a range of parameters to how the text will appear.

- **Screen width**: Sets the width of the screen used for placing the text objects and culling. Possible values range from 20 to 4000.
- **Screen height**: Sets the height of the screen used for placing the text objects and culling.
- **Loop kerning**: Defines the spacing between two loops. Possible values range from 0 to 1000.
- **Group kerning**: Sets the vertical distance between group elements.
- **Crawl mode**: Creates a normal horizontal scrolling text.
- **Roll mode**: Creates a vertical scroll.
- **Default Font**: Shows the default font. To load a font, drag a it from the server view and onto the font drop zone. This must be set to get any text unless you specify a font in the macro and all text are executed with macros.
- **Autosplit text**: If you have a long text string written in one line, without the autosplit feature, this is built as a big text object. This is not good for culling and performance. If you activate autosplit, long text objects will be split up into smaller ones which is better for performance.
- **Corr. val.**: Every time the autosplit feature splits a text object, the correction value is added to the gap between the two new objects. Although it reads the length of a space character from the font to make a proper placing of the objects there is a slight difference in length between a split and a non-split text. Therefore the correction value is inserted to correct text that does not look good because of incorrect placing when the objects are split.
- **Min. object length**: Sets the minimum object length for an autosplit object.
- **Col1**: Allows you to set the default color 1. To set it, either drag a material from the server view onto the col1 icon, or use the color editor below.
- **Col2**: Allows you to set the default color 2. To set it, either drag a material from the server view onto the col1 icon, or use the color editor below.
- **Col3**: Allows you to set the default color 3. To set it, either drag a material from the server view onto the col1 icon, or use the color editor below.

Text

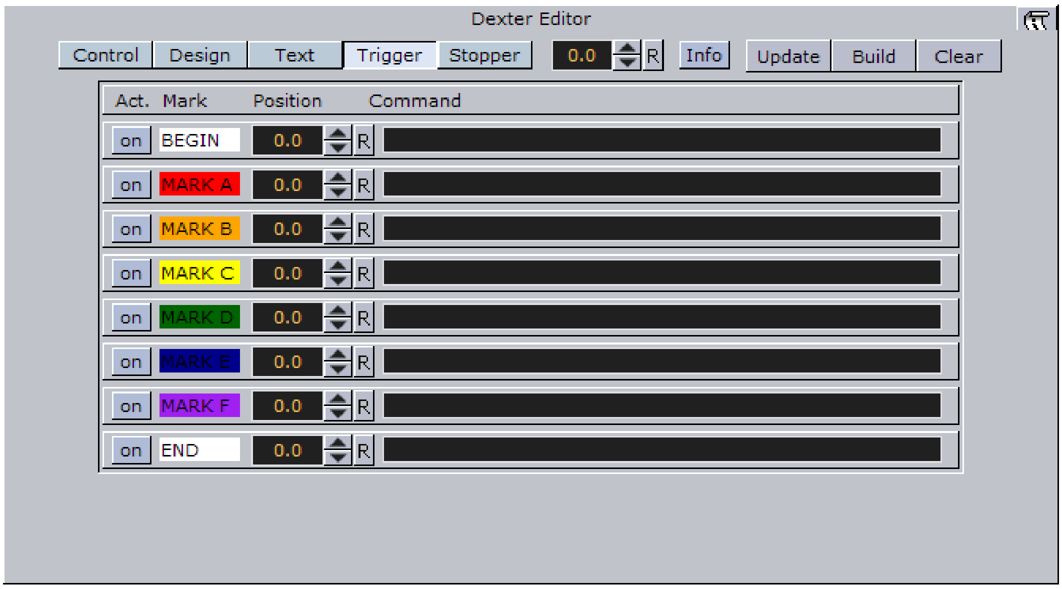


Built 0 group/s and 0 object/s, ready

The text editor contains the text area where you enter the script commands. In addition it contains a list of all tokens and parameters. By clicking on a token or parameter, you automatically insert it into the script text at cursor position.

- **Space=Text**: Forces the parser to interpret spaces at the beginning of a line/ after a token has ended as legal text (in its default state those spaces are abandoned).
- **Load File**: Allows you to browse for a script file. The default extension for the file is .edx. The script file must contain macro script with standard syntax. When the file loads, the contents is written onto the text field.

Trigger



Act.	Mark	Position	Command
on	BEGIN	0.0 R	
on	MARK A	0.0 R	
on	MARK B	0.0 R	
on	MARK C	0.0 R	
on	MARK D	0.0 R	
on	MARK E	0.0 R	
on	MARK F	0.0 R	
on	END	0.0 R	

- **Act.**: If this toggle is on, the marker will be evaluated.

- **Mark**: The name of the marker for orientation.
- **Position**: Here you can set the marker position where an event should occur.
- **Command**: You can type in here desired viz commandos.

The concept of marks and triggers allows you to send commands to Viz when the animation reaches a certain position. To make it work you need to define two elements:

First of all you need to mark the trigger positions in the text crawl/roll. This is done using the token <MARK>.

Animation starts<MARK A>here.

You have six marks, named A, B, C, D, E and F, that you can put wherever you want in the text. If you enable **Info** you will see a representation of the mark on screen.

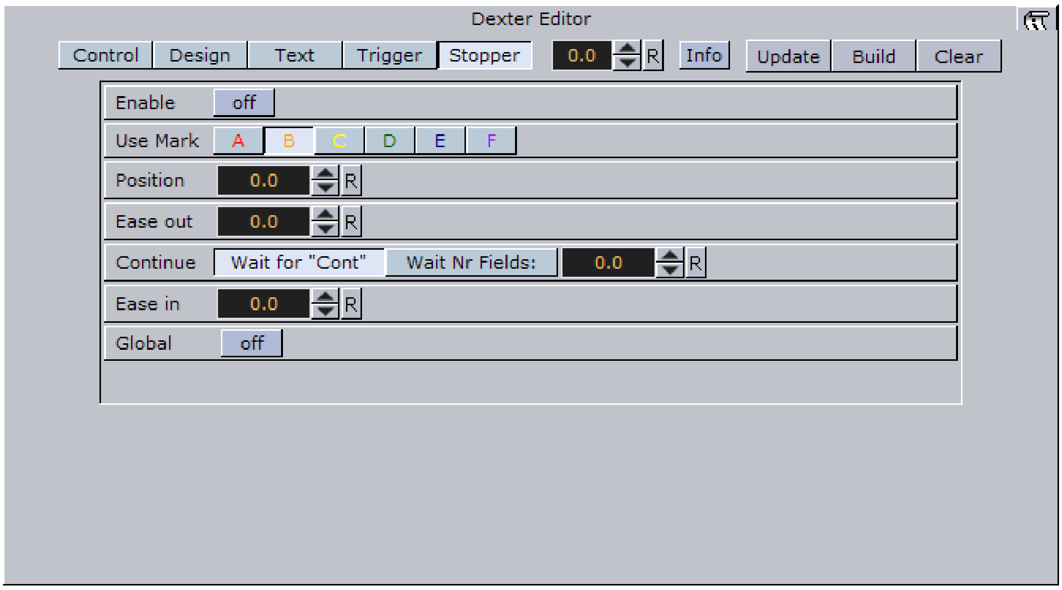
In addition to the six user-definable marks, there are two marks that are created automatically, named BEGIN and END. They are located at the beginning and end of the crawl/roll. Please keep in mind that the END mark is located before all the containers created for the loop feature and that marks are not looped.

After setting marks and building the crawl/roll, you must specify trigger actions. All eight marks are listed in the trigger section of the editor. The **on**/**off** switch left to the mark name is used to set that mark active/inactive. The position value right to the mark name sets the position where the triggering should occur (value 0 is at the end of the dexters bounding box. When animation is running and the mark in the text reaches this position, the specified command is triggered. If you switch on **Info** you will see a line representing the trigger position. Simply switch it on and watch the marks and lines interacting to see how this works. Right to the trigger position you will find a text field where you can enter the command which should be sent to Viz Artist when a mark triggers. You may use every command available via external control, for example you may change values in Viz Artist, start animations, execute plugins, and so on.

First you may issue two or more commands when a mark reaches its trigger position. Simply split them using a ; character (semi-colon). Also you may use the DEXTER location to send pseudo-commands to Dexter.

Currently two such pseudo-commands are implemented: The first one is DEXTER*STOP, which stops Dexter's animation, the second one is DEXTER*PAUSE X, where Dexter will stop the animation for X seconds (of course this is a floating point value, so DEXTER*PAUSE 0.5 is also possible) and continue with the same speed as before. Please note that these two pseudo-commands are not supported by Viz Artist command interface, they are only available using Dexter's trigger feature. Dexter supports the THIS_SCENE pseudo location as well; you may want to use this to refer to objects/animations in the local scene.

Stopper



Enable off

Use Mark A B C D E F

Position 0.0 R

Ease out 0.0 R

Continue Wait for "Cont" Wait Nr Fields: 0.0 R

Ease in 0.0 R

Global off

- **Enable**: If this toggle is on, the markers will be evaluated.
- **Use Mark**: If this toggle is switched to "on" the given positions will be used.
- **Position**: Sets the position of the "Ease-Out" and "Ease-In" range.
- **Ease Out**: The text object is leaving this position during the movement.
- **Continue**: This parameter has two options. "Wait for Cont" and "Wait Nr Fields". In first case the text object waits until an "continue" viz commando is being sent. In second case the text object waits for the number of fields. Consider that two fields comply one frame.
- **Ease In**: The text object slows down and waits for a continue signal or until the number of fields are reached.
- **Global**: Stops global or local to a certain group.
- **Info**: It shows the delimited borders defined by the mark settings.
- **Update**: Applies the changes on the already existing containers.
- **Build**: Rebuilds all objects.
- **Clear**: Clears the text objects.

The Stopper is an extension to the triggering ability of Dexter. It allows you to create ease in/out animations at defined points along the crawl/roll.First you need to create a set of marks using the <MARK> token to define the stop points in your text. Then open the Stopper page in the Dexter editor and check the **Enable** button.

Select which marks to use. If you have defined your stop points using the <MARK C> token you need to select C here. If you enable **Info** you will see a representation of the animation curve in the editor. Using the position, ease in and ease out slider you can move and alter that animation like you need. (Please keep in mind that the ease out part of the animation is executed before the ease in part: although usually it is the other way round. But here we do not want to start an animation smoothly and then stop it, we want to stop it first and then restart it again).

Using the continue switch you can select what Dexter should do after the animation has stopped: Either wait a certain number of fields or wait until the

Continue button in the Control section is pressed. If you have more than one group in crawl mode you may want to check the Global as well. If it is checked, all groups will be halted as soon as a marker in any group reaches the stopper position. If it is unchecked only the group that contains this marker is stopped.

Control Buttons

These buttons are found top right of the dexter editor and are not active in control mode.

- **Update**: Applies the changes on the already existing containers.
- **Build**: Rebuilds all objects.
- **Clear**: Clears the text objects.

Parameters

Table 22: Parameters

Name	Type	Function
text	string	This is the script dexter uses for building the text.
screenw	float	Screen width (used for placing the text objects and culling) 20/4000.
screenh	float	Screen height (used for placing the text objects and culling) 20/4000.
time	float	Animation length (in seconds) or animation speed 1/500.
file	string	File name (default extension .*).
loopmode	bool	Loop on or off.
loopkern	float	Defines the spacing between two loops (global) 0/1000.
groupkern	float	Defines the spacing between two groups (global) 0/1000.
showarea	bool	View the culling area on or off.
doautosplit	bool	If on, it splits text objects into pieces.
autosplitcorr	float	Correction value for autosplit feature -1e6/ +1e6.
autosplitmin	int	Minimum number of characters for split text objects 0/100000.
timeline	float	Value for preview control (works if animation is stopped) 0/1000.
deffont	string	Default font (e.g. Peak/AvantGarde/ AvantGarde-Book).
col1	unsigned long	Default color value.
col2	unsigned long	Color value #2.
col3	unsigned long	Color value #3.
durmode	int	Selects animation mode (duration/speed) 0/2.
roll	bool	Switches between vertical (0) and horizontal (1) animation.
trigonbegin	bool	Turns on/off trigger BEGIN (see 2.6).
trigona	bool	Turns on/off trigger A.
trigonb	bool	Turns on/off trigger B.
trigonc	bool	Turns on/off trigger C.

Table 22: Parameters

Name	Type	Function
trigond	bool	Turns on/off trigger D.
trigone	bool	Turns on/off trigger E.
trigonf	bool	Turns on/off trigger F.
trigonend	bool	Turns on/off trigger END.
trigposbegin	float	Position of trigger BEGIN -100000/100000.
trigposa	float	Position of trigger A -100000/100000.
trigposb	float	Position of trigger B -100000/100000.
trigposc	float	Position of trigger C -100000/100000.
trigposd	float	Position of trigger D -100000/100000.
trigpose	float	Position of trigger E -100000/100000.
trigposf	float	Position of trigger F -100000/100000.
trigposend	float	Position of trigger END -100000/100000.
trigcombegin	string	Command for trigger BEGIN.
trigcoma	string	Command for trigger A.
trigcomb	string	Command for trigger B.
trigcomc	string	Command for trigger C.
trigcomd	string	Command for trigger D.
trigcome	string	Command for trigger E.
trigcomf	string	Command for trigger F.
trigcomend	string	Command for trigger END.
status	bool	While animation is running, this parameter is set to 1 (read-only).

Boolean values can either be 1 (on) or 0 (off). Color values are 4 byte unsigned integers, byte 0 = red, 1 = green, 2 = blue, 3 = alpha.

Script Programming – Parameters

Table 23: Parameters

Name	Default value	Function
AnchorX, AnchorY, AnchorZ	0	X, Y and Z position of a new text object.
AnimDirection	Default	Animation direction, set to anything but "Default" to reverse animation
AutoSplitCorr	0	Correction value for autosplit feature
AutoSplitMin	20	Minimum number of characters for split text objects
Col1, Col2, Col3	255/255/255/255	Holds the color value from the gui/external control (parameter Col1)
Command	–	Sends a command to Viz.
Detail	Auto	Sets detail/fontstyle of text objects (see above).
DoAutoSplit	0	Enables/disables autosplit feature.
Enlighted	0	Set to 1 if you want enlighted text.
FontStyle	–	Sets the font used to create text objects (e.g. vizrt/AvantGarde/AvantGarde-Book).
GroupKerning	–	Sets horizontal spacing between groups (like Jump).
Jump	0	Sets horizontal spacing between groups.
LastContainer	–	Holds the container number of the last created container.
LastX, LastY	0	If a new text object is created, the last X and Yposition will be stored here.
MarkOffsX, MarkOffsY	0	X and Y offset for marks (see Script Programming – Tokens).
Material	255/255/255/255	The material used for creating text objects.
ObjectKerning	5	Sets spacing between two text objects.
RollAlign	C	Align groups in roll mode. values are R/r (right), L/l (left) and C/c (center).
RollBorder	0	Sets left/right border for groups (if aligned left or right in roll mode).
Scaling	20	Sets scaling in percent (20 means a scaling of 0.2).
ShadowDirection	320	Direction of shadow.

Table 23: Parameters

Name	Default value	Function
ShadowDistance	10	Distance of shadow
ShadowMaterial	204/204/204/204	Material of shadow.
ShadowZOffset	-1	Z offset of shadow.
SoftShadowLevel	1	Level of soft shadow (1-4).
Step	15	Sets the amount AnchorX is increased on a newline (see newline logic).
Template	-	Sets template to be used for creating text objects (see Script Programming - Templates).
TextKerning	0	Adjusts text kerning (in text object).
UseShadow	0	Set to 1 to enable shadow.
UseSoftShadow	0	Set to 1 to enable soft shadow.
WordSpacing	0	Adjusts word spacing (in text object).
XScaling, YScaling, ZScaling	20	Scaling value for X, Y and Z axes.

Color values are described in the form r/g/b/a with decimal numbers (base 10). Col1 is also used as the default color value (Material is initialized with this value). Material can also hold an existing material in the viz material pool, e.g. “Vizrt/Artdeco/artdeco.0” (see Col1, Col2, Col3).

Detail can have one of the following values: 1, 2, 3, 4, 5, 6 or T. 1 through 6 are detail levels, T means texture font. Every other value (like the default “Auto” will be interpreted as detail level auto) (see Detail).

Script Programming - Tokens

Table 24: Tokens

Name	Syntax	Function
ADD	<ADD Parameter Value>	Adds Value to Parameter (see example below).
CLEAR	<CLEAR Parameter>	Clears Parameter (sets it to "").
CONTAINER	<CONTAINER Path>	Inserts a container into the text (see example below).
ENDMAC	<ENDMAC>	Ends macro definition (nested macros are not allowed).
EXEC	<EXEC Name>	Executes the macro Name.
GROUP	<GROUP>	Begins a new group of containers.
IMAGE	<IMAGE name>	Inserts an image in the text.
MACRO	<MACRO Name>	Defines the macro Name.
MARK	<MARK Type>	Sets a mark (see Script Programming - Triggers).
MAX	<MAX Parameter Value>	If Value is higher than Parameter then Parameter = Value.
MIN	<MIN Parameter Value>	Works like MAX but takes the smaller value.
NEWLINE	<NEWLINE>	See Script Programming - Newline Logic.
NOTEXT	<NOTEXT>	See Script Programming - Newline Logic.
SET	<SET Parameter Value>	Sets Parameter to Value.

The ADD token adds either a string or a float to one. If either Parameter or Value are strings (see Table 24: Tokens), the two strings can be combined to one. If both are floats, an arithmetic addition will be done.

For example:

```
<SET Test Vi><ADD Test zrt>
```

results in “Vizrt”, but:

```
<SET Test 5><ADD Test 6.3>
```

results in Test holding the value 11.3.

The CONTAINER token is used to define a container path for inserting a container into the text. A container path may either look like “1/3/2/3/1” (the same system as used in the GUI), “$Containername” or “#Containernumber”.

```
<CONTAINER 1/3/2/4>
<CONTAINER $$Sphere> (or: <CONTAINER “$Sphere”>)
<CONTAINER #307>
```

The MARK token sets a mark that can be A, B, C, D, E or F.

The GROUP token begins a new group of containers. Instead of writing <GROUP> you may leave one line empty as this has the same effect. Use groups for making lines in roll mode.

Script Programming - Syntax

The syntax consists of tokens and text. <tokens> are surrounded by <>, e.g. <ENDMAC>. Spaces are used to separate tokens and arguments. Text is everything that is not bracketed. Tokens can be everywhere in the text, e.g. Viz is <SET Material 234/34/58/114>great is valid. You do not need to write every token in a new line. Nested tokens are not allowed. A macro cannot execute itself.

In addition to the tokens, there is the escape character $:

- $Parameter inserts the value of Parameter in the text
- $_ inserts a space character in the text (for use in tokens)
- $< inserts a < character
- $> inserts a > character
- $$ inserts a $ character

The $ character can be placed in tokens and in the text, e.g. $<Dexter$> says:

```
<SET Text Hello><ADD Text $_World>$Text
```

Inside tokens you may use " characters to insert text.

```
<SET Text “This is a text”>
```

All $ characters and spaces inside two " characters are ignored for parsing/ executing. Two " characters are interpreted as one:

```
<SET Name ""Harry"">
```

Script Programming - Templates

You can define templates for the creation of text objects. Under the Dexter container you will find a container named “Templates”. Under this container you can create a set of template containers. These can hold animations, plug-ins, key, alpha, etc. Give every template container a unique name. To access it in the script, you need to set the Template parameter to the name of the template you want to use:

```
<SET Template AlphaKey1>
```

After you set the Template parameter, Dexter will use this template container as basis for every text object. To switch this behavior off set Template to its default value (which is ""):

```
<CLEAR Template>
```

CAUTION! Certain parameters (like material) are overwritten by Dexter after the container has been created from the template.

Script Programming – Newline Logic

The token <NEWLINE> describes a new text object, which is separated from the previous one. Every newline moves AnchorX (see AnchorX, AnchorY, AnchorZ) a bit further, depending on the value of Step. This token will be inserted internally to prevent you from typing <NEWLINE> over and over again.

The following text

```
One
Two
```

is converted to

```
One
<NEWLINE>
Two
```

No newlines:

- From the beginning of the script until the first occurrence of text
- After a line that doesn't contain text (if the line is empty a new group will be created)
- After the tokens MACRO, ENDMAC and NOTEXT, and the last newline before those tokens will be deleted

For example:

Table 25:

Lines	Explanation
	No new lines because no text occurred.
<SET Camera Bobby>	No text, no new line.
Return of the	Here is the first new line.
Killer Tomatoes II	No new line because of token MACRO in the next line. From here on there are no new lines until the end of the MACRO.
<MACRO SetDir>	
<SET Director Jimmy>	
<ENDMAC>	
<EXEC SetDir>	No new line because there is no text in the line.
Director:	New line.
$Director	New line.
<SET FontStyle Arial>	No new line because there is no text in the line.

Table 25:

Lines	Explanation
Camera:	New line.
$Camera	
<NOTEXT>	The <NOTEXT> token will not add new lines before the next text element occurs.
x	New line.
y	New line.
<SET Material $Col2>	No new line because there is no text in the line.
z	New line.
1<NEWLINE>2	New line for 1 and 2.

The macro SetDir in the example above can be used as follows without creating a newline:

```
Hello<EXEC SetDir>World
```

The same rules applies to automatically inserted GROUP tokens.

Script Programming - Triggers

The concept of marks and triggers allows you to send commands to Viz when the animation reaches a certain position. Therefore you need to define two elements:

- First is setting a mark in the text. You have six different marks to place wherever you think they are useful, together with the automatically created BEGIN and END mark this gives you a total of eight marks.
- The second elements are the triggers, where you have as well eight different triggers. While marks are set with a token in the text, triggers can be defined in the trigger section of the dexter editor or using external control. You can define a position and a command for every trigger. When the corresponding mark reaches the trigger position, the command is issued to Viz.

Note: Only marks and triggers of the same type work together! If for example mark B reaches the position of trigger D, nothing will happen.

Marks are set using the MARK token, for example: Animation starts exactly`<MARK D>`here!

User defined marks are numbered A through F and are color coded in the trigger editor and as graphical symbols when show area is turned on. In addition to the user defined marks there are the automatically created marks BEGIN and END, which are inserted at the beginning and ending of the crawl/roll. The commands being issued do not have to be numbered, this is optional. If you do not give them a number, -1 is added before the command is sent to viz. you can as well define more than one command, separate them by a ; character (semi-colon).

You may invoke two internal commands using the trigger function with the `DEXTER*` location: stop and pause.

```
DEXTER*STOP

DEXTER*PAUSE X
```

Note: X is the time in seconds.

11.1.15 DisplacementMap

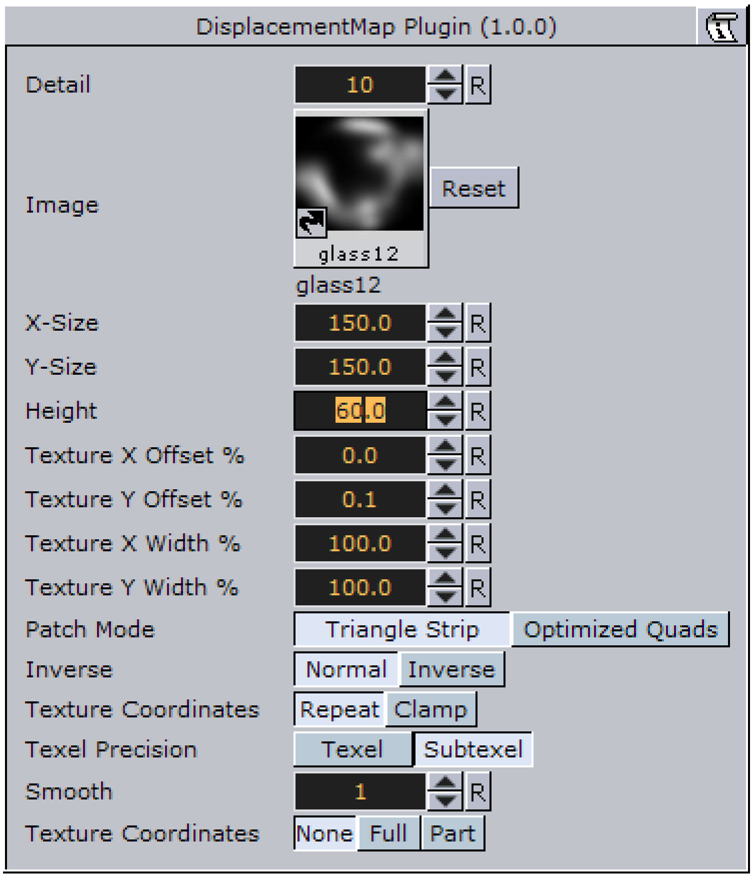
This plugin enables you to create topographical (height) maps by means of a grayscale image. The intensity value of the gray scale image is translated to a height value.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\DisplacementMap.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a displacement map

Properties and Parameters



DisplacementMap Plugin (1.0.0)

Detail: 10

Image: glass12 (Reset)

glass12

X-Size: 150.0

Y-Size: 150.0

Height: 60.0

Texture X Offset %: 0.0

Texture Y Offset %: 0.1

Texture X Width %: 100.0

Texture Y Width %: 100.0

Patch Mode: Triangle Strip | Optimized Quads

Inverse: Normal | Inverse

Texture Coordinates: Repeat | Clamp

Texel Precision: Texel | Subtexel

Smooth: 1

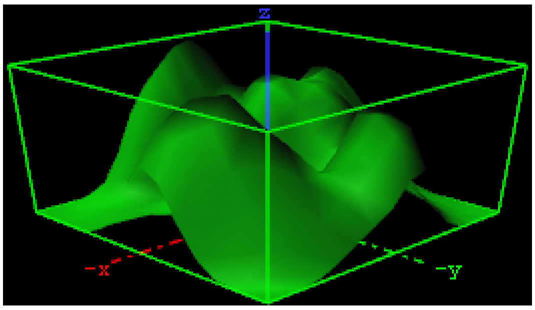
Texture Coordinates: None | Full | Part

- **Detail**: Sets the detail level of the map, the lower the value the higher the level of detail.
- **Image**: Drag the image you want to use creating the map, onto the placeholder. The image must be a valid luminance or alpha image.
- **X-size**: Sets the size of the map along the X-axis.
- **Y-size**: Sets the size of the map along the Y-axes.
- **Height**: Sets the height of the map.
- **Texture X Offset** %: Sets the offset of the texture in the X-axis.
- **Texture Y Offset** %: Sets the offset of the texture in the Y-axis.
- **Texture X Width** %: Stretches or compresses the texture in the X-axis.
- **Texture Y Width** %: Stretches or compresses the texture in the Y-axis.
- **Patch Mode**
  - **Triangle Strip**: Uses the same size of triangle polygon on the whole surface.
  - **Optimized Quads**: Uses bigger triangle polygons where the surface of the displacement map is flat, thereby reducing the total number of polygons and improving performance.
- **Inverse**: Mirrors all topographic coordinates through the zero level. This will make a mountain top to a crater.
- **Texture Coordinates**: Repeat/Clamp sets if the texture is the be repeated or clamped if it is too small to fit onto the displacement maps rectangle. Clamp stretches the texture to make it fit, Repeat starts drawing the texture over again when it reaches the end.
- **Texel Precision**

  - **Texel**: No subpixel/subtexel correction is made.
  - **Subtexel**: There is a limited number of pixels available on the screen, if a line does not run through a real pixel, it must be moved to the nearest one, this introduces a positional error. If subtexel is selected Viz Artist will break up pixels into smaller sub pixels in memory so that the line can be drawn to the nearest sub pixel.
- **Smooth**: Use this parameter to smooth the look of the map, without reducing the polygon details of the map. In some cases you need the number of polygons you have to obtain the correct lightning.
- **Texture Coordinates**: Decides if texture coordinates are to be created, and to which level, either **None**, **Full** or **Partially**. You need this enabled to some level, if a texture is applied using a vertex mapping. Texture mapping is resource consuming, so if you do not need it, keep it off.

Note: Please be aware not to increase to number of polygons in detail not too much, as this is a common mistake.

To create a displacement map



1. Create a new group and add the displacement plug-in to it.
2. Add material to the group container.
3. Open the displacement plug-in editor, and drag and drop a grey-scaled image onto the image placeholder.
4. Open the group container's transformation editor and adjust the Rotation values.

Note: An RGB or similar image does not work, and you will get an information in the log field. The image must be a valid luminance or alpha image.

11.1.16 Eclipse

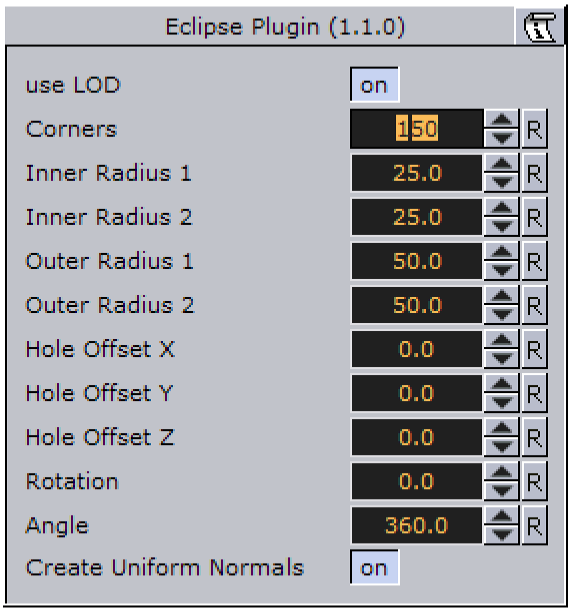
This plugin enables you to create an eclipse shape.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Eclipse.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create an eclipse

Properties and Parameters

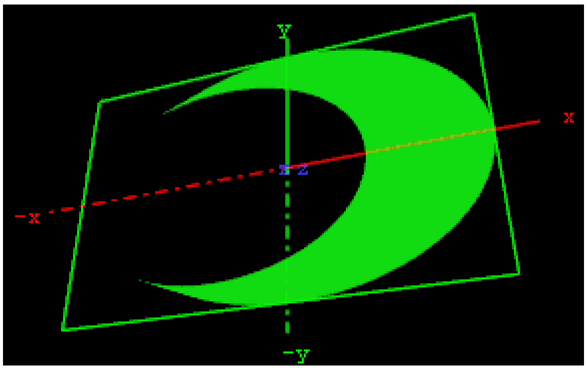


Eclipse Plugin (1.1.0)

use LOD	on
Corners	150
Inner Radius 1	25.0
Inner Radius 2	25.0
Outer Radius 1	50.0
Outer Radius 2	50.0
Hole Offset X	0.0
Hole Offset Y	0.0
Hole Offset Z	0.0
Rotation	0.0
Angle	360.0
Create Uniform Normals	on

- **Use LOD**: LOD stands for “Level of Detail”. In depending of the camera <-> object distance a high or lower detailed object will be rendered with marginal quality casualties if this toggle is switched on.
- **Corners**: It changes the number of triangles the object has. If you want a good quality increase the value. A value of 40 is a reasonably value.
- **Inner Radius 1 and 2**: Changes the inner radius in X-Position (1) and the inner radius in Y-Position (2).
- **Outer Radius 1 and 2**: Changes the outer radius in X-Position (1) and the outer radius in Y-Position (2).
- **Hole Offset X, Y and Z**: Moves the inner hole to X, Y, Z-position. With this parameter you can shape great objects like a volcano.
- **Rotation**: Rotates the eclipse. The result is visible if you change the angle less than 360°.
- **Angle**: Change this to create a view like a cake respectively a divided circle.
- **Create Uniform Normals**: It adapts the varying normals to the majority of normals in same direction.

To create an eclipse



1. Create a new group and add the eclipse plug-in to it.
2. Add material to the group container.
3. Open the eclipse plug-in editor, and adjust the following values:
   - Set Inner Radius 1 and 2 to 40.0
   - Set Hole Offset X to 20.0
4. Open the group container's transformation editor and adjust the Rotation values.

11.1.17 Fade Rectangle

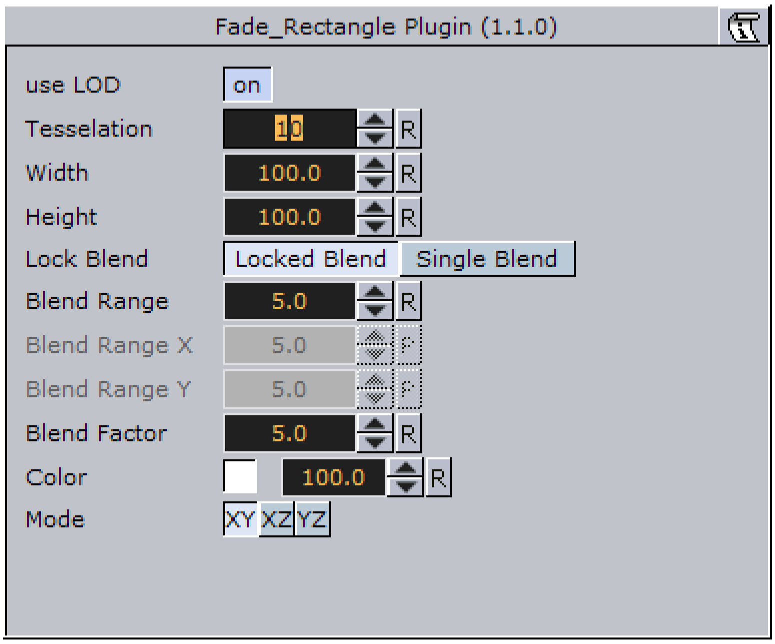
The Fade Rectangle's four sides can be set to fade in a way specified by the user. Both the degree of fading an the area influenced by the fading can be customized.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Fade_Rectangle.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a faded rectangle

Properties and Parameters

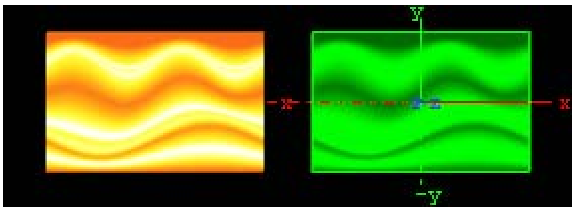


Fade_Rectangle Plugin (1.1.0)

use LOD	on
Tesselation	10
Width	100.0
Height	100.0
Lock Blend	Locked Blend Single Blend
Blend Range	5.0
Blend Range X	5.0
Blend Range Y	5.0
Blend Factor	5.0
Color	100.0
Mode	XY XZ YZ

- **Use LOD**: Enables/disables dynamic level of detail.
- **Tessellation**: Sets the degree of detail.
- **Width**: Sets the width of the fade rectangle.
- **Height**: Sets the height of the fade rectangle.
- **Lock Blend**: Allows you to decide whether to adjust the blending looked for both the X- and Y-axis or single (each axis separately).
- **Blend Range**: Allows you to set the blend range if looked mode is selected.
- **Blend Range X and Y**: Sets the blend range of the X and/or Y axis if single blend is selected.
- **Blend Factor**: Sets the alpha ramp of the fade rectangle. To see the effect set the alpha value to 0 and adjust the blend factor.
- **Color**: Displays the color of the rectangle. Set the color either in the editor below or drag a material from the server view onto the color icon.

To create a faded rectangle



1. Create a new group and add the fade rectangle plug-in onto it.
2. Add an image to this group.
3. Open the fade rectangle editor and change the desired color through the color parameter and adjust the width and the height.

11.1.18 Filecard

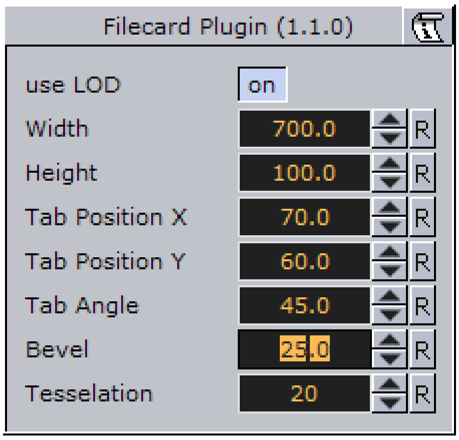
Use this plug-in to create filecards with different widths and heights and other attributes.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\FileCard.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a filecard

Properties and Parameters

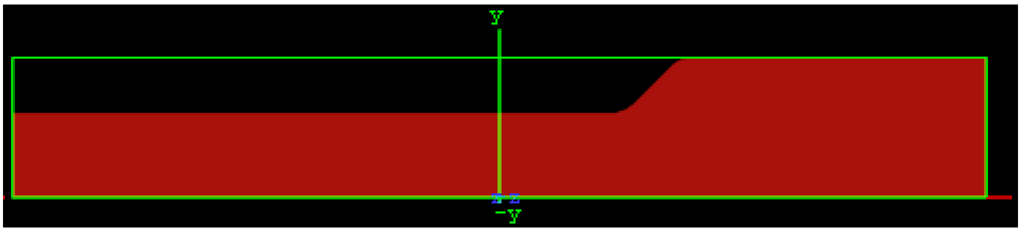


Filecard Plugin (1.1.0)

use LOD: on
Width: 700.0
Height: 100.0
Tab Position X: 70.0
Tab Position Y: 60.0
Tab Angle: 45.0
Bevel: 25.0
Tesselation: 20

- **use LOD**: Enables/disables dynamic level of detail.
- **Width**: Changes the width of the filecard.
- **Height**: Changes the height of the filecard.
- **Tab Position X and Y**: Changes the tab position in x-direction and y-direction.
- **Tab Angle**: Change this value to get a smaller curve.
- **Bevel**: Increase this value to get a soft edge.
- **Tesselation**: This value changes the number of triangles of the rendered filecard to increase the visual quality. Please consider that the number of triangles will affect the performance of the system.

To create a filecard



1. Create a new group and add the filecard plug-in to it.
2. Add a material and/or a texture to the group.
3. Open the filecard editor and set the following parameters:
   - Set Width to 700.0
   - Set Tab Position X to 70.0, and Tab Position Y to 60.0
   - Set Tab Angle to 45.0
   - Set Bevel to 25.0

11.1.19 Graph

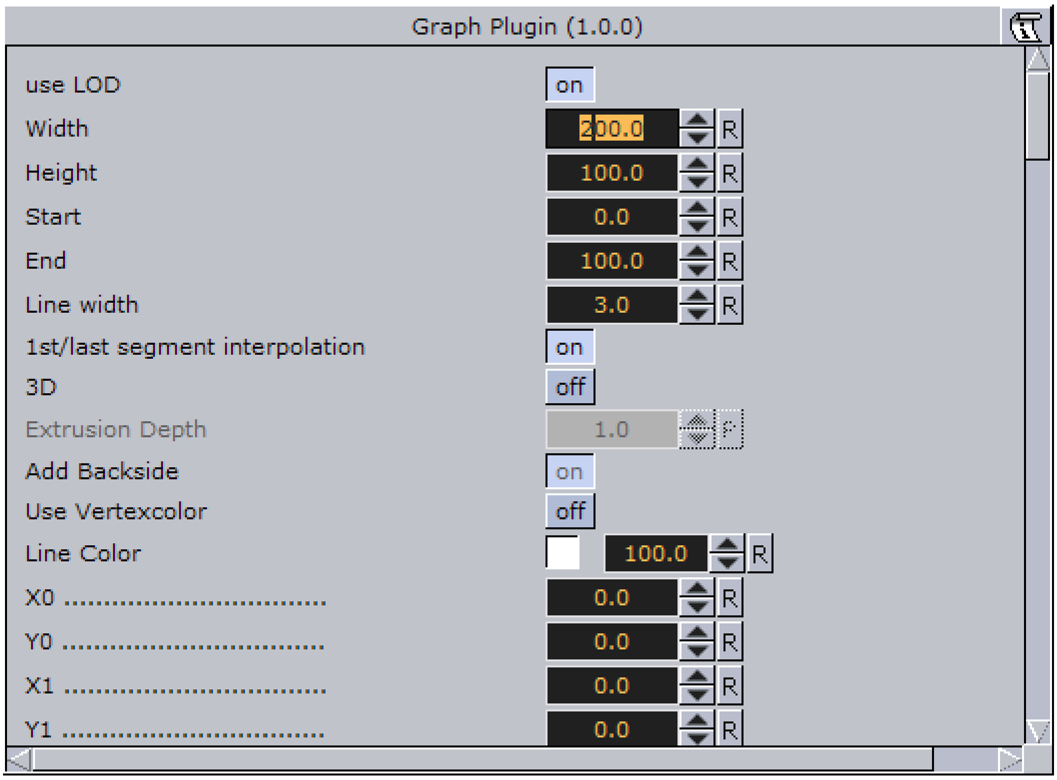
The Graph object allows you to create a 2D or 3D graph with up to 50 values.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Graph.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a graph

Properties and Parameters

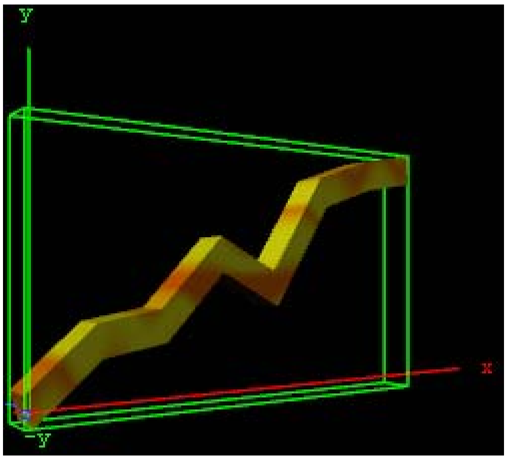


Graph Plugin (1.0.0)

use LOD on
Width 200.0 R
Height 100.0 R
Start 0.0 R
End 100.0 R
Line width 3.0 R
1st/last segment interpolation on
3D off
Extrusion Depth 1.0 R
Add Backside on
Use Vertexcolor off
Line Color 100.0 R
X0 .... 0.0 R
Y0 .... 0.0 R
X1 .... 0.0 R
Y1 .... 0.0 R

- **Use LOD**: Enables/disables dynamic level of detail.
- **Width**: Sets the width of the graph.
- **Height**: Sets the height of the graph.
- **Start**: Sets the starting point of the graph.
- **End**: Sets the ending point of the graph.
- **Line Width**: Defines the line width.
- **1st/last segment interpolation**: Switches the interpolation of the first and last segment on or off.
- **3D**: Enables/disables 3D visualization of the graph.
- **Extrusion depth**: Sets the graph’s extrusion depth.
- **Add backside**: Enables/disables visualization of the graph backside.
- **Use vertex color**: Enables a base color for the graph. Set the values in the color editor or drag a material onto the small square below the parameter.
- **Line color**: Enables the use of a vertex color.
- **X0/Y0 to X49/Y49**: Enables up to 50 different value points of the graph by giving coordinates on the X- and Y-axis.

To create a graph



1. Create a new group and add the graph plug-in to it.
2. Add a material and/or a texture to the group, or open the graph editor and enable Use Vertexcolor and set the Line Color.
3. Open the graph editor and set the Start and End points.
4. Enable 3D and set Extrusion Depth to 30.0.
5. Add points for the X- and Y-directions (i.e. X0/Y0, X1/Y1 and so on).

11.1.20 Graph2d

Graph 2D

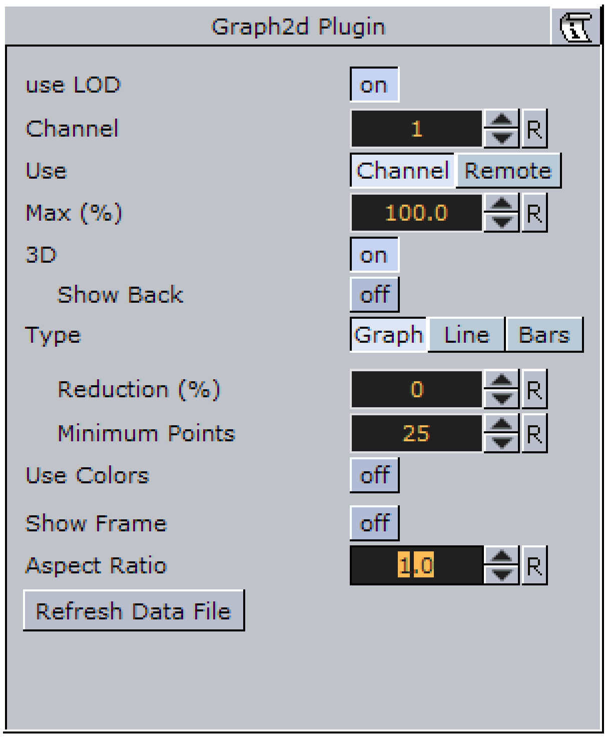
Use this plug-in to create 2D and 3D graphs, lines and bars from imported files.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Graph2D.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a 2D graph
- To create a channel file for the 2D graph

Properties and Parameters



Graph2d Plugin

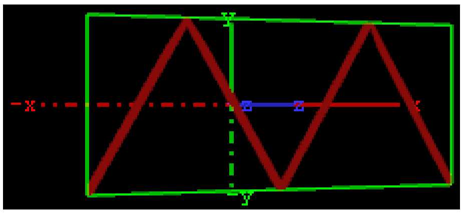
use LOD	on
Channel	1 R
Use	Channel Remote
Max (%)	100.0 R
3D	on
Show Back	off
Type	Graph Line Bars
Reduction (%)	0 R
Minimum Points	25 R
Use Colors	off
Show Frame	off
Aspect Ratio	1.0 R

Refresh Data File

- **use LOD**: LOD stands for “Level of Detail”. In depending of the camera <-> object distance a high or lower detailed object will be rendered with marginal quality casualties if this toggle is switched on.
- **Channel**: Affects the filename to search for. The file is holding the vertex and other data to define the entire graph.
- **Use**: Load the file through a channel file or remote access.
- **Max (%)**: It sets the rage for rendering. For example if you set the range to 50% the graph will be cut 50% in width.
- **3D**: Enables 3D rendering of the graph, and the Show Back option.
  - **Show Back**: Enables the object to show the back in case the object will be rotated.
- **Type**: Following types are available: Graph, Line and Bars. Take a change if you want another visual depiction. **Graph** enables the Reduction and Minimum Points settings. **Line** enables the Constant Width, Line Width (%), Reduction (%) and Minimum Points settings. Bars enables the Bars Width (%) and Bar Animation settings.
  - **Reduction (%)**: The points will be reduced by the Douglas-Peucker algorithm. The Douglas-Peucker algorithm is an algorithm for reducing the number of points in a curve that is approximated by a series of points.
  - **Minimum Points**: Changes the number of points for rendering the graph.
  - **Constant Width**: Sets a constant width.
  - **Line Width (%)**: It changes the width of the line.
  - **Reduction (%)**: The points will be reduced by the Douglas-Peucker algorithm. The Douglas-Peucker algorithm is an algorithm for reducing the number of points in a curve that is approximated by a series of points.
  - **Minimum Points**: It changes the number of points for rendering the graph.
  - **Bar Width (%)**: It changes the width of all bars.

  - **Bar Animation**: One by One: Each bar will be separately rendered. All together: All bars will be rendered at the same time.
- **Use Colors**: Sets the color for positive and negative values. This setting is only available for graphs and bars (not line).
- **Show Frame**: Shows the frame around the graph. Useful for orientation.
- **Aspect Ratio**: Stretch the graph evenly in x-direction.
- **Refresh Data File (button)**: Reload the file which holds the data value for the x- and y-points.

To create a 2D graph



1. Create a new group and add the Graph2D plug-in to it.
2. Add a material and/or a texture to the group.
3. Change the Channel parameter to the existing file (e.g. 2 for CHANNEL2).
4. Refresh the data file if the file has changed its content.

To create a channel file for the 2D graph

1. Create a text file (e.g. “CHANNEL2”) with no file extension which is holding the following data: 5 0 0 10 20 20 0 30 20 40 0.
   - The first number (5) holds the number of points available in the file.
   - The other definitions are the X and Y points -> (0/0), (10/20), (20/0) etc.
2. Create the following directory <**Viz directory**>**data****plugin****graph2D**. and place the channel file in it.

IMPORTANT! The channel file must always be named “CHANNEL<Channel number>”.

11.1.21 Icosahedron

An Icosahedron is a Platonic solid composed of twenty faces that span twelve vertices, each face of which is an equilateral triangle. An icosahedron can be considered a rough approximation for a sphere.

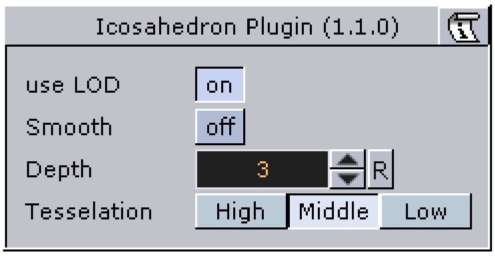
The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Icosahedron.vip`.

This section contains information on the following topics:

- Properties and Parameters

- To create an icosahedron

Properties and Parameters

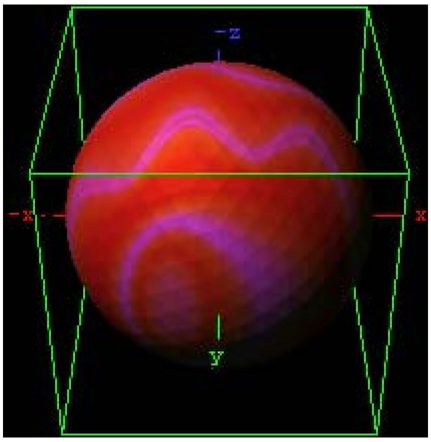


Icosahedron Plugin (1.1.0)

use LOD on
Smooth off
Depth 3 R
Tesselation High Middle Low

- **Use LOD**: Enables/disables dynamic level of detail.
- **Smooth**: Enables smoothing of edges.
- **Depth**: Sets the number of planes at the icosahedron. The more planes you set the more it looks like a sphere.
- **Tessellation**: Allows you to choose level of detail setting between **Low**, **Middle** and **High**.

To create an icosahedron



1. Create a new group and add the Icosahedron plug-in to it.
2. Add a material and/or a texture to the group.
3. Open the Icosahedron editor and disable the Smooth option, set Depth to 3 and Tesselation to Middle.

11.1.22 Image FX

Image FX is a tool that enables you to create a wide variety of transitions between one or more images. The plugin uses many different effect models that can be customized through parameters. Some properties use advanced mathematical formulas to create the effects, and not all of them can be explained meaningfully in normal words. Because of this, not all properties can be explained completely here, you will have to “play” with some properties to see the effects they create in a given situation (what some properties are set to will have impact on the effect that other properties give).

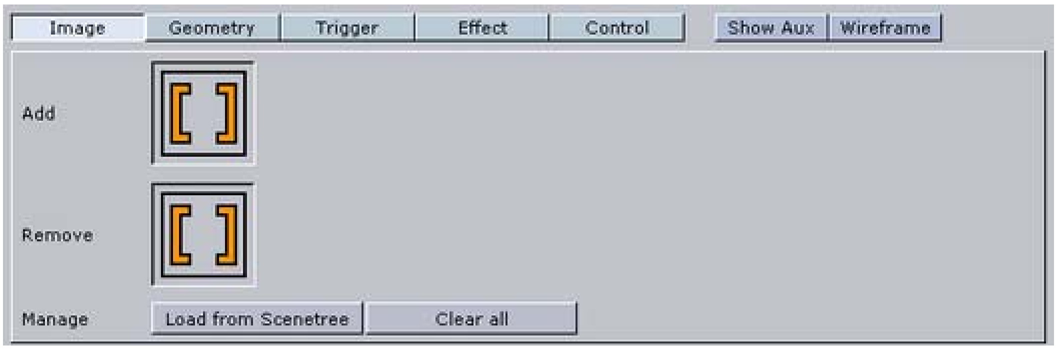
The images are split into a number of user-defined sections and every section is controlled independently based on progress maps. Any number of images of any Viz Artist supported format can be controlled using ImageFx. All images will be resized (if needed) to the first image size. The image transition order can be modified on the fly and images may be added to the sequence using external commands.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ImageFx.vip`.

This section also contains information on the following topics:

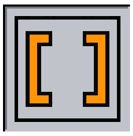
- Properties and Parameters
  - Image
  - Geometry
  - Trigger
  - Effect
  - Control
  - To add, load and remove images

Properties and Parameters



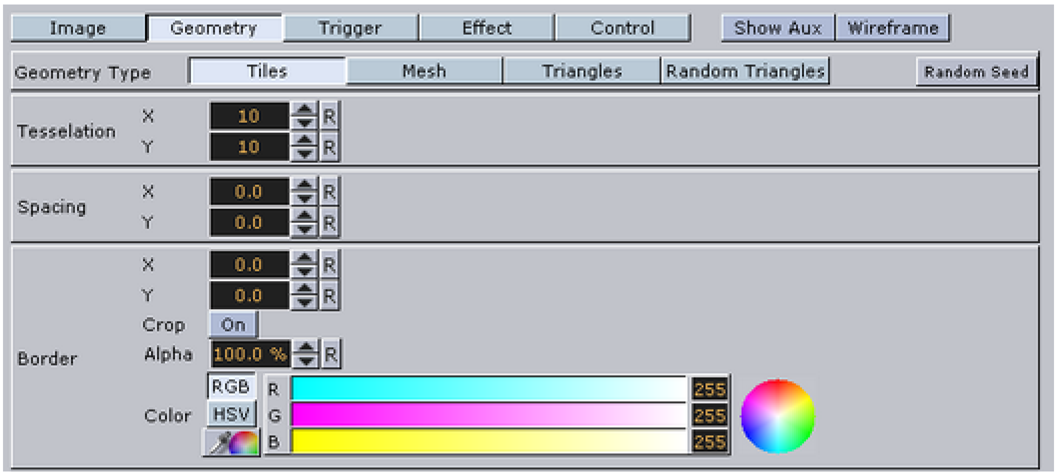
At the top of the editor you will find a set of buttons to switch the properties.

Image



- **Add and Remove**: Images dropped onto the Add/Remove drop zone will be added/removed as sub-containers of the plug-in container in the scene tree.
- **Load from Scenetree**: Loads images that are added directly as sub-containers to the ImageFX plug-in container.
- **Clear all**: Removes all sub-containers to the ImageFX plug-in container.

Geometry



Geometry Type: Tiles

Tesselation X 10 R
Tesselation Y 10 R

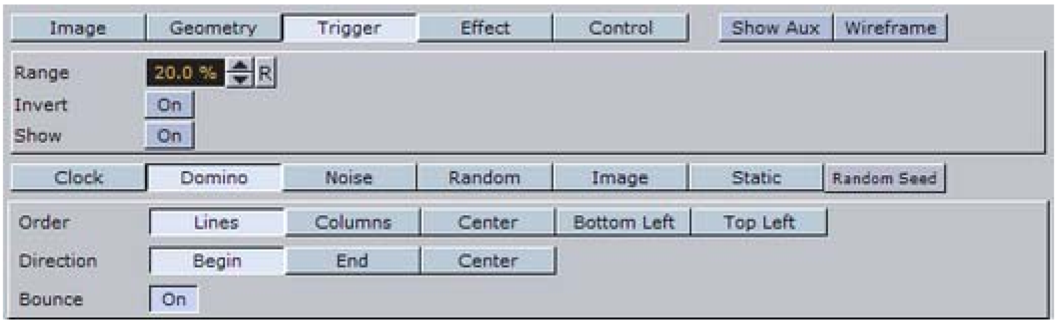
Spacing X 0.0 R
Spacing Y 0.0 R

Border X 0.0 R
Border Y 0.0 R
Crop On
Alpha 100.0 % R
Color R 255
Color G 255
Color B 255

The image is split into a number of stand-alone sections that can be modified independently.

- **Geometry Type**: Select here which kind of geometry you want the images to be split up in.
  - **Tiles**: GL quads, the number of quads are X tessellation times Y tessellation.
  - **Mesh**: (vertices) GL quad strip, the number of vertices are (X tessellation +1) times (Y tessellation +1). Modification can apply to every vertex in the mesh.
  - **Triangles**: GL triangles, the number of triangles are X tessellation times Y tessellation times 2. Modification can apply to one triangle (3 vertices).
  - **Random Triangles**: GL triangles, triangles are generated using random points. Number of triangles is a minimum of X tessellation times the Y tessellation. Modification can apply to one triangle (3 vertices).
  - **Random Seed**: Is relevant if you have chosen random triangles. It specifies a seed for the random number generator. Even though Viz Artist use random numbers, the layout of a specific random seed will always look the same. Press the wireframe button and click the new random seed button and you will see the effect.
- **Tesselation**: Sets the degree of detail (enable wireframe to view the effect).
- **Spacing**: Sets a spacing between the geometrical parts in the image.
- **Border**: If spacing has some value, this parameter will draw a border with the width defined here between single sections.
- **Crop**: Enables/disables cropping of the border.
- **Alpha**: Sets an alpha value for the border.
- **Color**: Sets a color for the border.

Trigger



Range 20.0 %
Invert On
Show On
Order Lines
Direction Begin
Bounce On

In the trigger section you can select between different progress maps that defines how every section of the image is to be triggered by the global progress. Each section of the image receives its base value, the global progress is a value that changes as you animate by using one of the options in the control menu. Since every section gets a different base value, it will be triggered differently by the global progress.

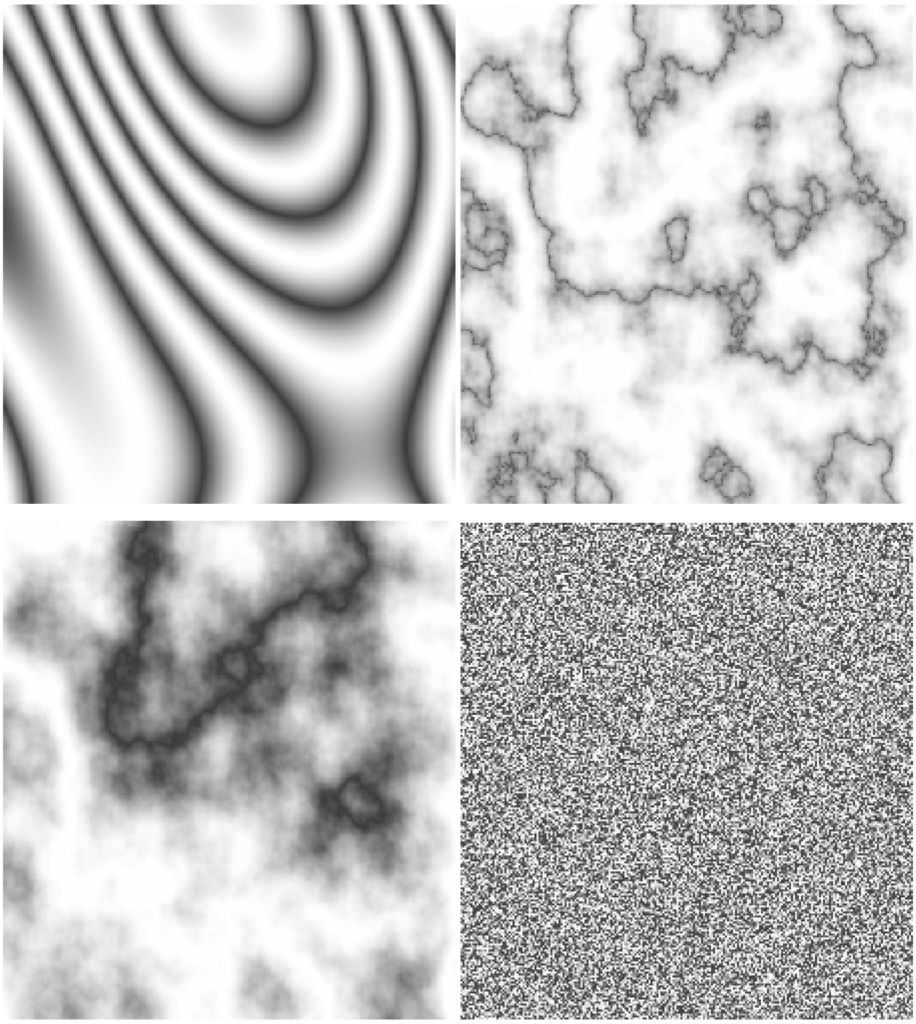
Example: In a domino effect which starts at the bottom left the section at the bottom left will receive a value of 100 and the section at the top right will receive a value of 1. As the progress moves section with high values will be influences, fast then sections with low values. If we continue with the above example with a fade effect, the bottom left section will fade out when progress is 20 but the top right section will only fade out when progress is 100.

The options for progress maps are:

- **Clock**: Sets the values for the sections that are computed based on a clock hand movement. It has the following properties:
  - **Center**: Sets the center for the clock hand movement.
  - **Start Angle**: Sets the starting angle for the clock hand.
  - **Whirl**: Applies a whirl effect to the movement.
  - **Direction**: Sets the direction for the clock hand movement.
- **Domino**: Sets the values for the sections that are computed based on a domino movement. It has the following options:
  - **Order**: Sets the order in which the tiles will be moved. Choose between **Lines** (Line by line movement), **Columns** (Column by column movement, **Center** (From center and outwards movement), **Bottom Left** (Bottom left tiles first and then diagonal movement towards top right corner) or **Top Left** (Top left tiles first and then diagonal movement towards bottom right).
  - **Direction**: Sets the starting point of the movement.
- **Noise**: Sets the values for the sections that are computed based on user-defined noise map created using noise algorithms (perlin noise).
  - The settings create a noise texture and the image transformations are based on this image.
  - When you edit the noise properties it is a good idea to enable the *show* button. This will let you see the image that are created and you will see changes you do instantly.
  - **Wave**, **Cloud**, **Marble** and **Noise** are predefined noise parameter settings. It is normally a good idea to start with one of these and thereafter edit until you get the desired effect.

The four images below show how the four noise parameters look like when the show option is enabled:

Figure 167: Wave, marble, cloud and noise.



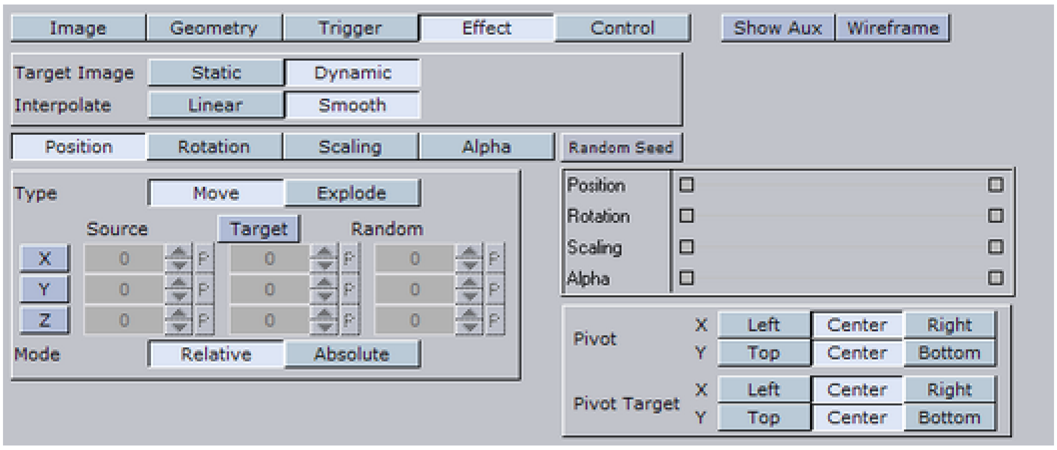
- **Turbulence 1** to **3**: Sets the noise level. The settings must be *played* with to see what kind of effect you get as they influence each other. So what effect one gives depends on the state of the other.
- **Frequency**: Defines something that can be described as the "wave length" of the noise function. A large number gives a small wave length which will create a lot of changes to the image, while a small number will create a smoother image.
- **Exponent**: Controls the ratio between the white areas and the black areas without changing the shape significantly. Exponent zero give a full white picture, and as the value will grow, the black area will become larger and larger.
- **Scale** X/Y: Are quite similar to the usual texture scale, but they are working with reverse logic. In the usual texture scaling, increasing the scale is like zooming into the image, but in Image FX it works like zooming out. Also the units are different, but beside this, they are quite similar.

- **Random**: The values for the sections are computed randomly.
- **Image**: The values for the sections are computed based on pixel values of an image you drag onto the drop zone. What the function does is to create links

between the animated image tiles and the pixels of the image and then use the luminance level of each pixel to set the base value for the tiles. By default 100% luminance sets high base value and 0% sets a low base value. This option allows you to generate any progress map you want by using other tools.

- **Static**: All sections receive the same value. The complete image will move out and the other will move in.
- **Random Seed**: Is relevant if you have chosen random order. It specifies a seed for the random number generator. Even though Viz Artist use random numbers, the animation for a specific random seed will always look the same.

Effect



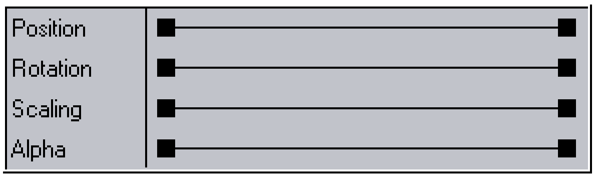
The trigger section defined the order in which the tiles would be influenced by the global process. The effect section allows you define what kind of effect/transition that will be used.

- **Target Image**
  - **Static**: The target image is not effected at all the by the effect. A typical example would be a reveal effect where target image is “hidden” behind main image and a transition effect on the main image will reveal the target image.
  - **Dynamic**: The target image is controlled by the effect. A typical example would be main image flying out of the frame and at the same time target image flying in. If a dynamic target image is used, its effect can be either being the invert of main image or can be defined separately.
- **Interpolate**
  - **Linear**: A linear interpolation will be used for the transformations.
  - **Smooth**: A smooth interpolation will be used for the transformations.
- **Position** > **Move**: You can specify a position for the source image, for the target image and a random position.
  - For the source image, which is the image that is currently displayed, you specify a source position. This is the position that the image tiles will move to when the animation runs.
  - If no target position is specified, the target image will animate from the source position to the initial position of the source image.
  - If you specify a target value the target image will move from that position and onto the initial position of the source image.
  - Switch on **X**, **Y** and **Z** to alter the values.

  - **Absolute**: All sections will move to the same end position.
  - **Relative**: The sections will maintain their original relationship.
- **Position** > **Explode**: Final positions for the sections are calculated based on a user-controlled explode algorithm. The tiles of the image are treated as particles in an explosion and they are moved in one to three axis depending on what you specify. All tiles are thrown out of a emitter that can be defined to have any opening angle, so the particles can be sent within a narrow angle, like a canon fire, or in all direction like a explosion. The parameters are:
  - **Duration**: Sets the duration for the movement of the tiles/particles.
  - **Opening Angle**: Sets the opening angle for the emitter. Remember to choose more than one axis to get any clear effect.
  - **Angle Rotate X**: Rotates the emitter hole on the X-axis.
  - **Angle Rotate Y**: Rotates the emitter hole on the Y-axis.
  - **Force** sets the degree of force that throws the particles out in space.
  - **Force Spread %**: Randomizes the initial impels of the tiles. Spread 0 % means all elements have the same impulse, 100% means a high degree of randomness.
  - **Gravity**: Sets a gravity force for the environment.
  - **Use Axis**: Select which axes or axis to use.
- **Rotation**: The sections are rotated.
- **Scaling**: The sections are scaled.
- **Alpha**: The alpha value of the sections are changed from **Alpha Start** to **Alpha End**.

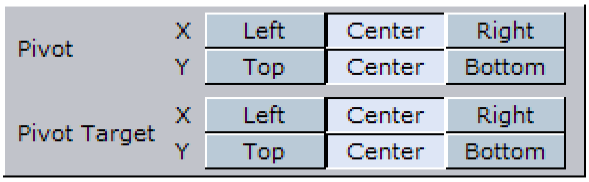
To all options (except alpha) a certain degree of user-controlled randomness can be added.

- **Random Seed**: Is relevant if you have a random property. It specifies a seed for the random number generator. Even though Viz Artist use random numbers, the animation for a specific random seed will always look the same.



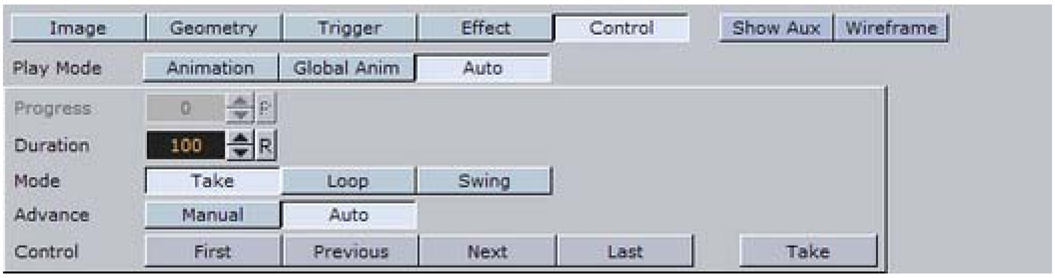
The final effect of transformation can be a combination of effects where the timing between the different effects can be tuned using the mini-stage. Move the start and end keyframes to time the effects.

Example: A position explode effect can be combined with a rotation effect so the sections will rotate as they explode. An alpha effect can be added to fade out the sections at the end.



The pivot properties have relevance when you use rotation on the tiles. The settings decide where the rotation center should be on the tiles. To see the effect, create a low tesselation transition with tiles and a 180 degree X or Y rotation. Change the pivot properties for Y pivot if you have an X rotation or the X pivot if you have an Y rotation and see the effect.

Control



The global progress value that is used to animate the sections can be controlled in different ways:

- **Auto**: When using auto mode ImageFx will start the effect as soon as the "Take" button is invoked. Useful for external control in order to run a sequence of images.
- **Animation**: The global progress value is animated and the effect follows the stage progress. In this mode you can combine the effect animation with other animations.
- **Global Animation**: When using a sequence of images the complete progress will animate all images.

To add, load and remove images

1. Add the images you want to create transitions between by dragging them from the server view and dropping them onto the **Add** drop zone.
   - The added images will be visible in the scene tree as sub-containers of the ImageFX plug-in container.
2. To add images directly as sub-containers to the plug-in container you must click the **Load from Scenetree** button to make the plug-in recognize the added images.
3. Remove images by dragging them from the server view and onto the **Remove** drop zone, or simply remove them directly from the scene tree.
4. To remove all, click the **Clear all** button.

11.1.23 Noggi

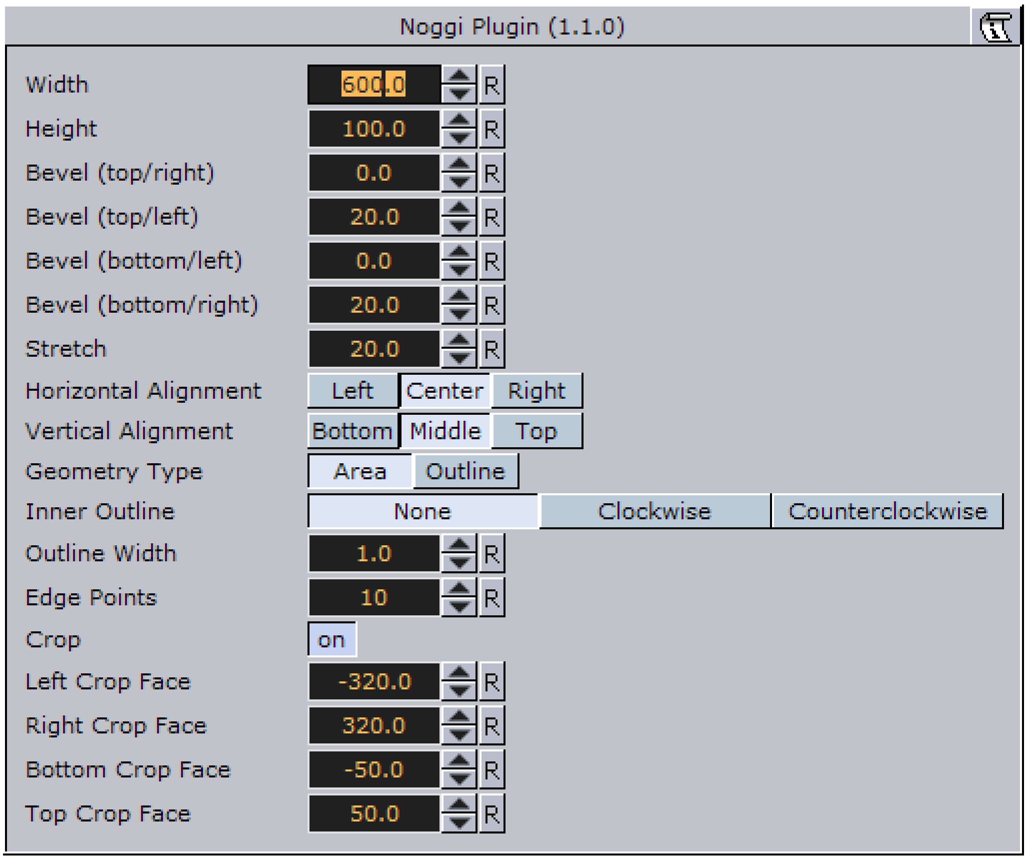
Use this plug-in to create geometry objects of beveled rectangles with some attributes.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Noggi.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a noggi

Properties and Parameters



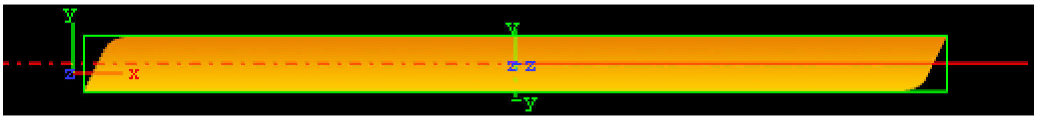
Noggi Plugin (1.1.0)

Width	600.0
Height	100.0
Bevel (top/right)	0.0
Bevel (top/left)	20.0
Bevel (bottom/left)	0.0
Bevel (bottom/right)	20.0
Stretch	20.0
Horizontal Alignment	Left / Center / Right
Vertical Alignment	Bottom / Middle / Top
Geometry Type	Area / Outline
Inner Outline	None / Clockwise / Counterclockwise
Outline Width	1.0
Edge Points	10
Crop	on
Left Crop Face	-320.0
Right Crop Face	320.0
Bottom Crop Face	-50.0
Top Crop Face	50.0

- **Width**: Changes the width in X-direction.
- **Height**: Changes the height in Y-direction.
- **Bevel (top/right, top/left, bottom/left and bottom/right)**: Changes the bevel on each corner.
- **Stretch**: The result is an object like a parallelogram. Stretch is in percent (= tan(alpha)). If Geometry Type is Outline.

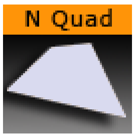
- **Horizontal Alignment**: Changes the object alignment to left, center or right. This change will affect the y-rotation.
- **Vertical Alignment**: Changes the object alignment to bottom, middle or top. This change will affect the x-rotation.
- **Geometry Type**: Renders the object as area- or with outline look.
- **Inner Outline**: Changes the vertex sequence in outline mode. Inner Outline is important for face orientation when extruding the geometry.
- **Outline Width**: Changes the thickness of the rendered outline.
- **Edge Points**: Change the number of triangles to render an beveled corner. A lower value means a
- **Crop**: Enables/disables cropping of the object. Cropping and vertex texture are only available if Geometry Type is Area.
  - **Crop Face (Left, Right, Bottom and Top)**: Crops the object on each given side.

To create a noggi



1. Create a new group and add the Noggi plug-in to it.
2. Add a material and/or a texture to the group.
3. Open the transformation editor and set Position X to -150.0.
4. Open the Noggi editor and set the following parameters:
   - Set Width and Height to 600.0 and 100.0.
   - Set Bevel top/right and bottom/left to 0.0.
   - Set Bevel top/left and bottom/right to 20.0.

11.1.24 N Quad



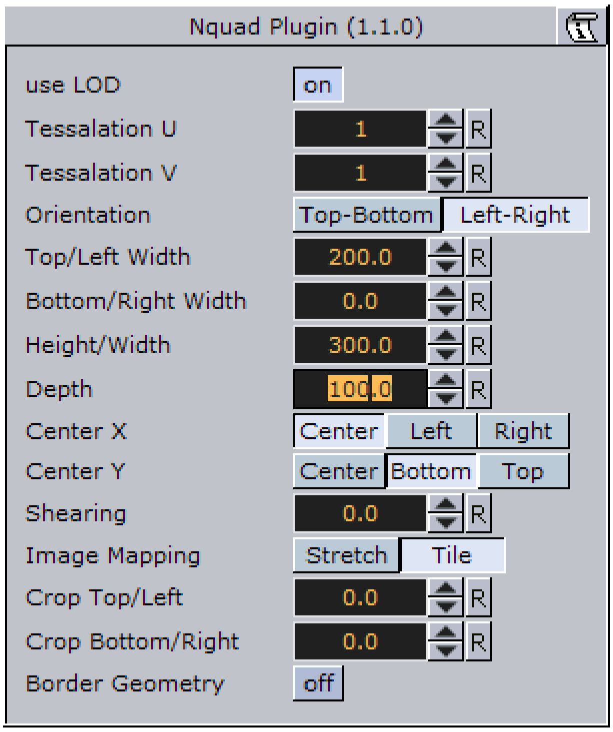
Use this plug-in to create a rectangle with different attributes.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Nquad.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create an N Quad

Properties and Parameters



Nquad Plugin (1.1.0)

use LOD: on

Tessalation U: 1

Tessalation V: 1

Orientation: Top-Bottom | Left-Right

Top/Left Width: 200.0

Bottom/Right Width: 0.0

Height/Width: 300.0

Depth: 100.0

Center X: Center | Left | Right

Center Y: Center | Bottom | Top

Shearing: 0.0

Image Mapping: Stretch | Tile

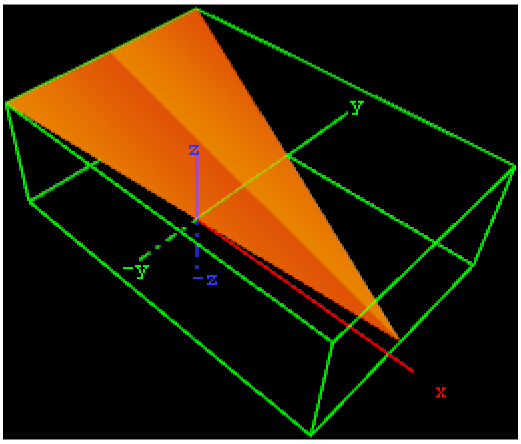
Crop Top/Left: 0.0

Crop Bottom/Right: 0.0

Border Geometry: off

- **Use LOD**: Enables/disables dynamic level of detail.
- **Tessellation U**: Sets the degree of detail.
- **Tessellation V**: Sets the degree of detail.
- **Orientation**: Switches the orientation from **Top-Bottom** to **Left-Right**.
- **Top/Left Width**: Sets the width.
- **Bottom/Right Width**: Sets the width.
- **Height/Width**: Sets the height.
- **Depth**: Sets the depth.
- **Center X**: Switches the center in X between **Center**, **Top** and **Bottom**.
- **Center Y**: Switches the center in Y between **Center**, **Left** and **Right**.
- **Shearing**: Sets the shearing value, so the bottom is shifted.
- **Image Mapping**: May be set to **Stretch** or **Tile**.
- **Crop Top/Left**: Crops the top.
- **Crop Bottom/Right**: Crops the bottom.
- **Border Geometry**: Enables/disables the top and bottom border width parameters. When enabled, only the border outline is visible.
- **Top Border Width**: Sets the width of the top border.
- **Bottom Border Width**: Sets the width of the bottom border.

To create an N Quad



1. Create a new group and add the N Quad plug-in to it.
2. Add a material and/or a texture to the group.
3. Open the transformation editor and set Rotation X and Z to -45.0.
4. Open the N Quad editor and set the following parameters:
    - Set Orientation to Left-Right.
    - Set Top/Left Width to 200.0.
    - Set Bottom/Right Width to 0.0.
    - Set Height/Width to 300.0.
    - Set Depth to 100.0.
    - Set Center X and Y to Center and Bottom, respectively.

11.1.25 Pointer

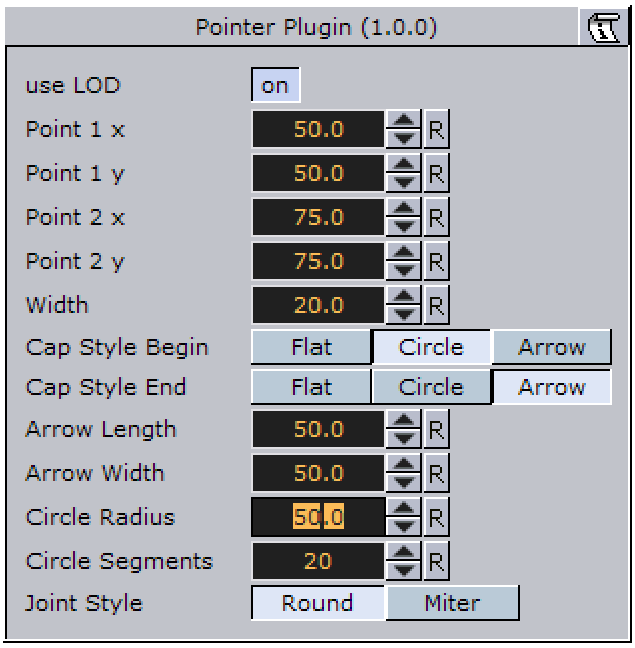
The pointer is similar to the Arrow. The main difference is that the pointer has a joint link, so it looks more like a *bend* arrow. The pointer has two reference points which controls the pointer's position. The first point is in the joint link and the second point is at the arrow's head.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Pointer.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a pointer

Properties and Parameters

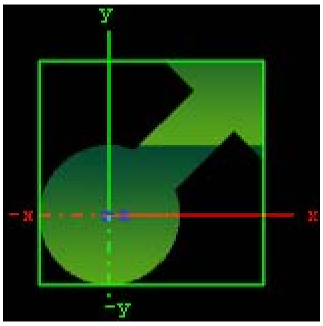


Pointer Plugin (1.0.0)

use LOD	on
Point 1 x	50.0
Point 1 y	50.0
Point 2 x	75.0
Point 2 y	75.0
Width	20.0
Cap Style Begin	Flat [✓ Circle] Arrow
Cap Style End	Flat Circle [✓ Arrow]
Arrow Length	50.0
Arrow Width	50.0
Circle Radius	50.0
Circle Segments	20
Joint Style	[✓ Round] Miter

- **Use LOD**: Enables/disables dynamic level of detail.
- **Point 1 X**: Sets the position of point 1 on the X-axis.
- **Point 1 Y**: Sets the position of point 1 on the Y-axis.
- **Point 2 X**: Sets the position of point 2 on the X-axis.
- **Point 2 Y**: Sets the position of point 2 on the Y-axis.
- **Width**: Sets the width of the pointer shaft.
- **Cap Style Begin**: Allows you to select the style on the pointers beginning. The options are **Flat**, **Circle** or **Arrow**.
- **Cap Style End**: Allows you to select the style on the pointers end. The options are **Flat**, **Circle** or **Arrow**.
- **Arrow Length**: Sets the length of the arrow head.
- **Arrow Width**: Sets the width of the arrow head.
- **Circle Radius**: Sets the radius of the circle, if cap style is selected.
- **Circle Segments**: Sets the resolution of the outer side of the pointer angle. Switch to wireframe to see the effect as you change the value.
- **Joint Style**: Sets the style of the pointer joint, either **Round** or **Miter**.

To create a pointer



1. Create a new group and add the Pointer plug-in to it.
2. Add a material and/or a texture to the group.
3. Open the Pointer editor and set the following parameters:
    - Set Point 1 X and Y to 50.0.
    - Set Point 2 X and Y to 75.0.
    - Set Width to 20.0.
    - Set Cap Style Begin and End to Circle and Arrow, respectively.
    - Set Arrow Length, Width and Radius to 50.0.

11.1.26 Polygon

The Polygon built-in object enables you to create a user defined polygon. The polygon is a 2D object. The extrusion can be applied to it to make it into a 3D object. If you open the property editor for the polygon plug-in, a drawing pad will be shown. Note that polygons that overlap each other will act as masks.

Note: Polygon is not a plug-in with its own file as it is part of the internal Viz core.

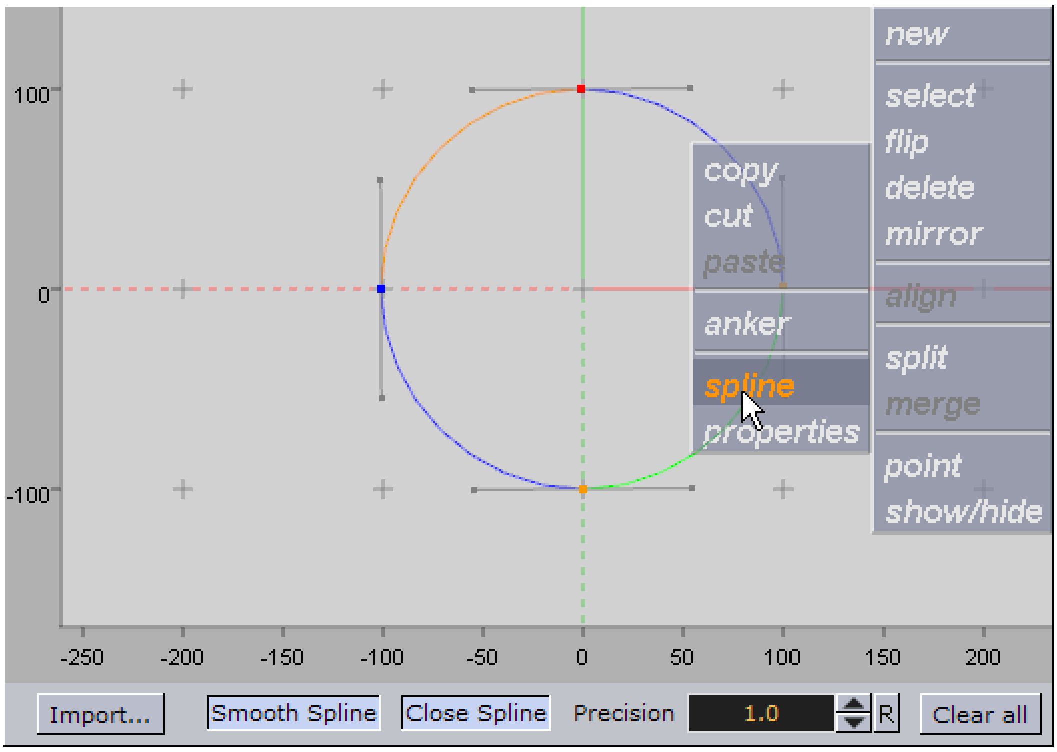
This section contains information on the following topics:

- Properties and Parameters
- To draw a spline
- To draw a new spline
- To undo
- To move and scale a polygon
- To alter the shape of the polygon
- To select existing elements for editing
- To mirror the spline sections
- To flip spline sections
- To rotate a spline
- To split a spline
- To merge two splines

See Also

- Polygon Plug-in Editor Shortcuts

Properties and Parameters



At the bottom of the drawing pad you will find a main menu, offering you the following options:

- **Import...**: Allows you to import an Adobe Illustrator file to be used as spline.
- **Smooth Spline**: When enabled the spline will be calculated as curve connecting the intersection points. If switched off the intersection points will be connected by straight lines.
- **Close Spline**: Closes the spline to one curve from the first to the last point. If switched off the first and last point of the spline will be connected by a straight line to form the border of the polygon.
- **Precision**: Sets the tessellation. To view the effect of changes, switch the render output to wireframe.
- **Clear all**: Deletes all splines.

For information on how to use the drawing pad’s context menu, see the different procedures.

To draw a spline

An outline of the spline created on the drawing pad is displayed as a polygon in the render output. As the polygon needs a closed border in the render output the spline will automatically be closed. This is done without taking the shape of the spline into consideration.

1. Move the cursor on the drawing pad to the position you want to start from and do a left-click.
   - This will set the first point of your spline.
2. Move the cursor to the place where the second intersection point should be placed and do another left-click.
3. Repeat step 2 until your spline is defined by enough intersection points in order to create the desired polygon shape.
   - For ease of use all intersection points are numbered.
   - If you added an intersection point by accident press the *Delete* button to remove it.

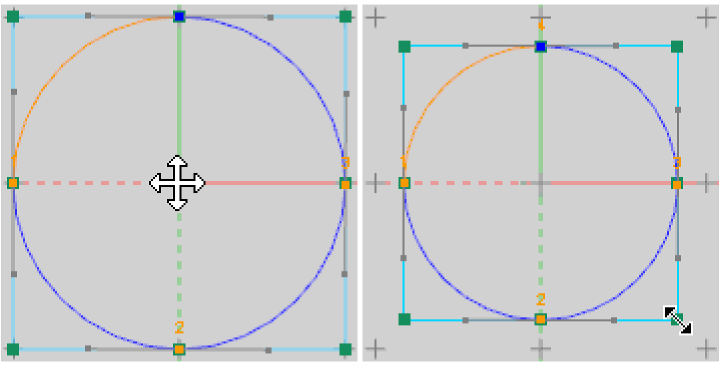
To draw a new spline

- Right-click and select the point on the drawing pad where the new spline should start, and from the appearing context menu select **spline** and then **new**.

To undo

- Press **DELETE**, or press and hold the **CTRL** button while turning the mouse wheel backwards (one step at a time) to undo the previous action.

To move and scale a polygon



1. Press the **SPACE** bar, or right-click and select a spline, and from the appearing context menu select **properties**, **editor** and **bounding box**.
2. Place the cursor in the center of the polygon, left-click and hold the mouse button while dragging the polygon.
3. Drag the handles of the bounding box to scale the polygon.

To alter the shape of the polygon

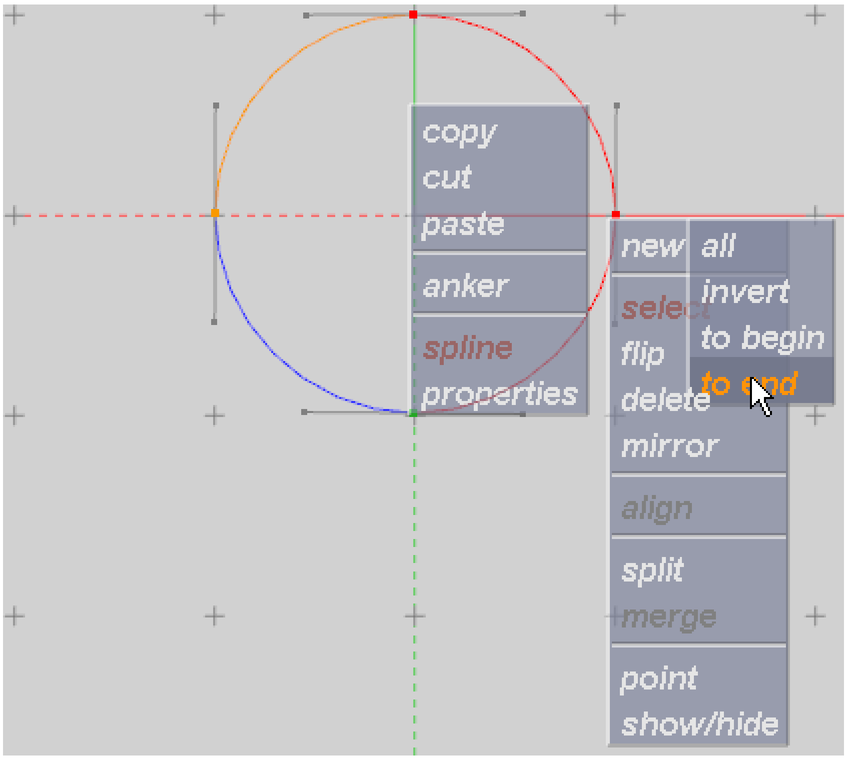
To change the shape of the polygon you have to alter the spline. You are provided with a set of options to do so:

1. Move an intersection point by clicking and holding the left mouse button while moving the cursor around. Release the button to set the new position of the point.
   - While hovering the intersection point the color of the intersection point will become green and the status information displayed.

2. Add an intersection point by hovering above the spline and left-clicking the mouse button, or hold the mouse button to move the intersection point to another position.
3. Remove an intersection point by left-clicking and holding an intersection point while pressing the **Delete** button.
4. Edit handles (i.e. the ones on the right and left of the intersection point) by left-clicking and holding the mouse button to move the handles around. Release the button to set the new position of the handle.

- Set sharp edges on selected or all intersection points by right-clicking the spline and selecting **spline**, **point** and **sharp edge**. Reset changes by selecting the **reset handles** option.

To select existing elements for editing



1. Press and hold the **CTRL** button and the **left mouse button** while drawing a rectangle that covers the intersection points to be selected, or
2. Right-click and select an intersection point, and from the appearing context menu, select **spline**, **select** and then one of the following options:
   - **all**: Selects all spline sections of the polygon.
   - **to begin**: Selects all spline sections from the point selected to the beginning of the polygon.
   - **to end**: Selects all spline sections from the point selected to the end of the polygon.
   - **invert**: Inverts the selection already made. If all is selected, invert will change it to none selected, and conversely. If *to begin* was selected, it will transform it to *to end*, and conversely.

To mirror the spline sections

- Right-click the drawing pad, and choose **spline**, **mirror** and one of the following options:
  - **all**: Mirrors whole spline along X- and Y-axis.
  - **all x**: Mirrors whole spline along X-axis.
  - **all y**: Mirrors whole spline along Y-axis.
  - **selected**: Mirrors only selected points along X- and Y-axis.
  - **selected x**: Mirrors only selected points along X-axis.
  - **selected y**: Mirrors only selected points along Y-axis.

To flip spline sections

What the flip function does is to switch the X or Y values between the points in an order where the first point switch value with the last point, the second point switch value with the last but one, and so on.

- Right-click and select **spline**, **flip** and one of the following options:
  - **all x**: Flips whole spline over X-axis.
  - **all y**: Flips whole spline over Y-axis.
  - **selected x**: Flips selected points over X-axis.
  - **selected y**: Flips selected points over Y-axis.

To rotate a spline

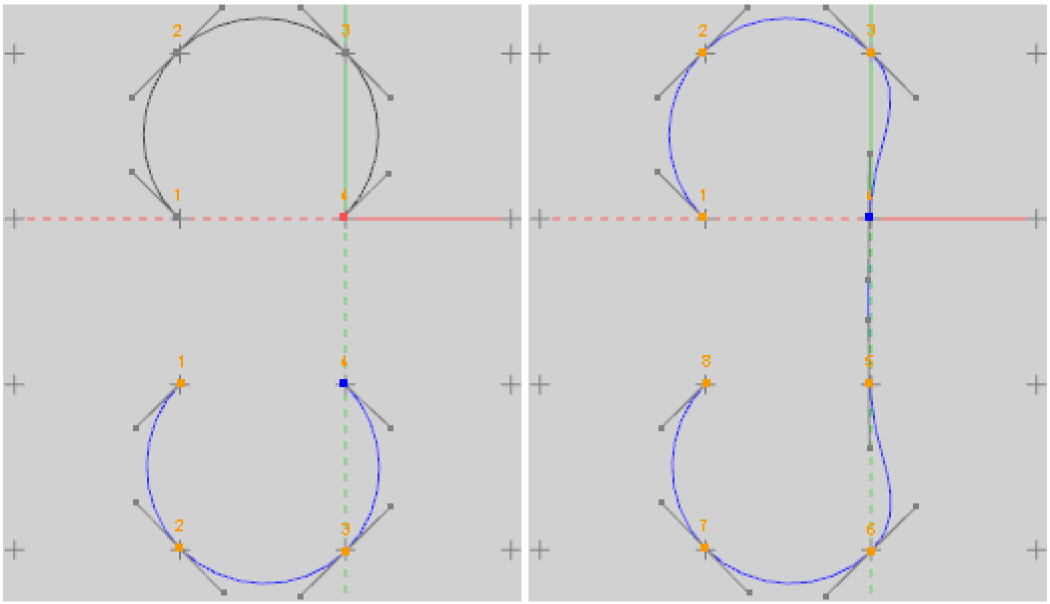
1. Select and right-click the intersection point to rotate around, and from the appearing context menu select **anchor** and then **set.**
2. Press the **SPACE** bar, or right-click and select a spline, and from the appearing context menu select **properties**, **editor** and **bounding box**.
3. Move the cursor next to a vertex of the bounding box and the cursor will be shown as rotating arrow.
4. Press the left button on the mouse and move the cursor to rotate the spline.

To split a spline

Splitting a spline section can be done in three ways:

1. Right-click and select the section to split and from the appearing context menu select **spline** and then **split**, or
2. Select the section and press **S**, or
3. Press **SHIFT+S** (showing a pair of scissors) and left click the mouse on the section to be split. Right-click the mouse to revert to normal editing mode.
   - After you have split the section the numbering of the points will change.

To merge two splines



1. Right-click and select a start/end point to be connected, and from the appearing context menu select **spline** and then **merge**.
2. Double-click the start/end point on the other spline to be merged.
   - The cursor will become a chain.
3. Click on the start/end point on the first spline to connect them.
   - After you have merged the splines the numbering of the points changes.

11.1.27 Rectangle

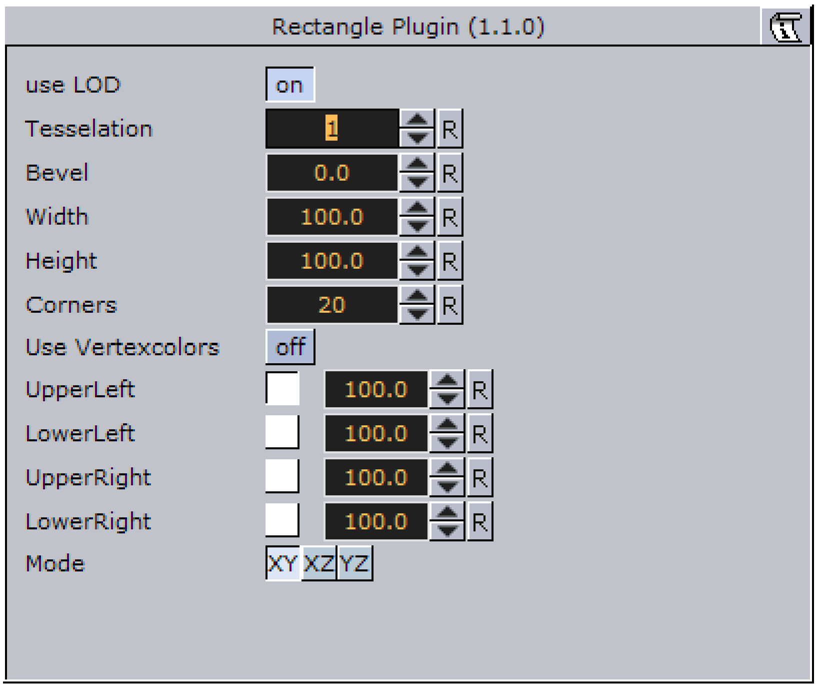
Use this plug-in to create a simply 2D rectangle with some attributes.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Rectangle.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a rectangle

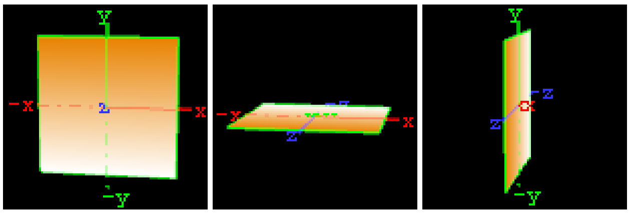
Properties and Parameters



Rectangle Plugin (1.1.0)

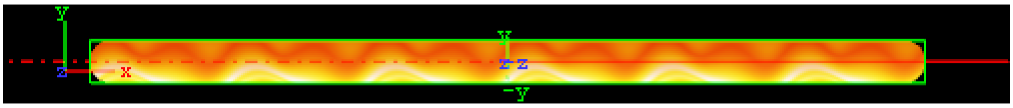
use LOD	on
Tesselation	1
Bevel	0.0
Width	100.0
Height	100.0
Corners	20
Use Vertexcolors	off
UpperLeft	100.0
LowerLeft	100.0
UpperRight	100.0
LowerRight	100.0
Mode	XY XZ YZ

- **Use LOD**: Enables/disables dynamic level of detail.
- **Tessellation**: Sets the degree of detail.
- **Bevel**: Enables and sets the degree of bevel at the rectangle edges. Bevel may have an effect on the height of the rectangle. For example if Height is set to 100.0 and Bevel is set to 51.0, the effective height is 102.0.
- **Width**: Sets the width of the rectangle.
- **Height**: Sets the height of the rectangle.
- **Corners**: Sets the number of corners the rectangle is build up from, it become visible, when you set a bevel on the rectangle.
- **Use Vertexcolors**: Enables you to set the four color parameters below. If this option is set, the rectangle does not respond to container color anymore.
- **Upper Left**: Sets the color and alpha of the upper left section of the rectangle. Press the colored button to enable color editing on that section and change the color in the color editor below. You can also drag a material from the server view and onto the color icon.
- **Lower left**: Sets the color and alpha of the lower left section of the rectangle. Press the colored button to enable color editing on that section and change the color in the color editor below. You can also drag a material from the server view and onto the color icon.
- **Upper Right**: Sets the color and alpha of the upper right section of the rectangle. Press the colored button to enable color editing on that section and change the color in the color editor below. You can also drag a material from the server view and onto the color icon.
- **Lower Right**: Sets the color and alpha of the lower right section of the rectangle. Press the colored button to enable color editing on that section and change the color in the color editor below. You can also drag a material from the server view and onto the color icon.



- **Mode**: Sets the orientation of the plane to XY, XZ or YZ.

To create a rectangle



1. Create a new group and add the Rectangle plug-in to it.
2. Add a material and/or a texture to the group, and/or open the Rectangle plug-in editor and enable and set the vertex colors.
3. Open the transformation editor and set Position Y to -150.0.
4. Open the Rectangle editor and set the following parameters:
   - Set Bevel to 15.0.
   - Set Width to 600.0.
   - Set Height to 30.0.

11.1.28 Ring

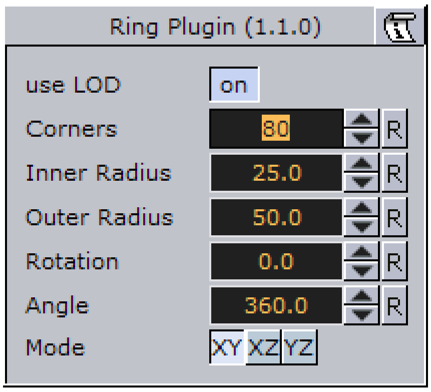
Use this plug-in to simply create an open or closed ring with some attributes.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Ring.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a ring

Properties and Parameters

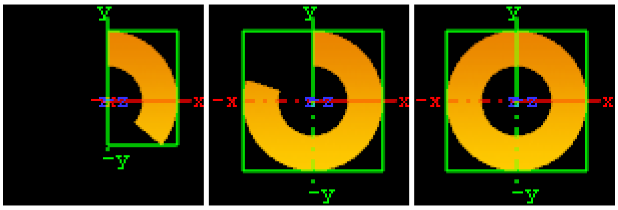


Ring Plugin (1.1.0)

use LOD on
Corners 80
Inner Radius 25.0
Outer Radius 50.0
Rotation 0.0
Angle 360.0
Mode XY XZ YZ

- **Use LOD**: Enables/disables dynamic level of detail.
- **Corners**: Sets the number of corners on the ring.
- **Inner Radius**: Sets the size of the inner radius.
- **Outer Radius**: Sets the size of the outer radius.
- **Rotation**: Rotates the ring like a “turning wheel”. This is typically used in combination with an open angle, to place the angle at the desired point.
- **Angle**: Defines an open angle on the ring.
- **Mode**: Changes the object view coordinate. Available options are XY, XZ and YZ.

To create a ring



1. Create a new group and add the Ring plug-in to it.
2. Add a material and/or a texture to the group.
3. Open the Ring plug-in editor and **animate** the Angle.
   - Set Angle to 0.0 (60/50 fps).
   - Set Angle to 360.0 (120/100 fps).

11.1.29 Roll

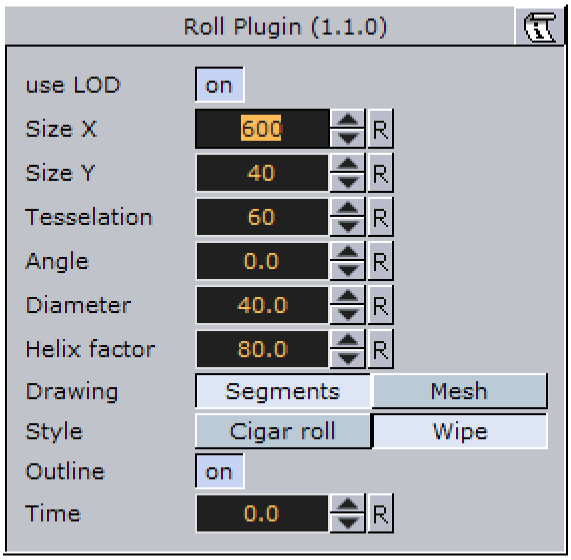
The roll allows you to create a rectangle that rolls up in different ways. This is typically used for different kinds of unveiling. To make the roll look realistic, the expert plugin must be added to the container to enable visualization of backface and two-sided lightening.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\RFxSmoke.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a roll

Properties and Parameters



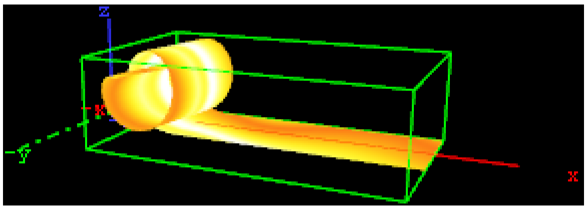
Roll Plugin (1.1.0)

use LOD: on
Size X: 600
Size Y: 40
Tesselation: 60
Angle: 0.0
Diameter: 40.0
Helix factor: 80.0
Drawing: Segments | Mesh
Style: Cigar roll | Wipe
Outline: on
Time: 0.0

- **Use LOD**: Enables/disables dynamic level of detail.
- **Size X**: Sets the size of the rectangle on the X-axis.
- **Size Y**: Sets the size of the rectangle on the Y-axis.
- **Tessellation**: Sets the level of detail.
- **Angle**: Sets the angle of the rolling up.
- **Diameter**: Sets the diameter of the roll.
- **Helix factor**: Defines how tight the roll is to be rolled up. With a high helix factor, there will be much space between the layers of the roll.
- **Drawing**: Selects how the roll rectangle is to be drawn up. If you switch to wireframe mode and toggle between the two settings you will see the difference:
  - **Segments**: Draws the rectangle up by using many long lines stretching from edge to edge. This mode does not look as good as **Mesh** but it demands less performance of the render engine.
  - **Mesh**: Draws the rectangle using many small equally sized triangles. The number and size depends on the level of tessellation selected. This drawing mode can be heavy to render if the tessellation is set high. **Mesh** creates a better lightning of the roll rectangle.
- **Style** sets the style of the roll:
  - **Cigar roll** makes the rectangle roll up in a normal manner.
  - **Wipe** rolls the carpet up without creating a roll.
- **Outline** enables an outline view for the roll. You must in addition add the expert plugin and enable the outline setting there.

- **Time** sets the time for the roll sequence. Animate this value to create an animation of the roll object. To learn more about animation see Creating Animations.

To create a roll



1. Create a new group and add the Roll plug-in to it.
2. Add the Expert plug-in to the group.
3. Open the Expert editor and enable (On) the Back Face property.
4. Add a material and/or a texture to the group.
5. Open the Roll editor and set the following parameters:
   - Set Size X to 150.0.
   - Set Size Y to 20.0.
   - Set Angle to 10.0.
6. Adjust the Time parameter to see the roll effect.

11.1.30 Shape View

Use this plug-in to use shape files.

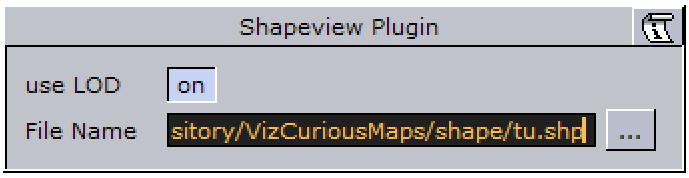
Mandatory files:
- * .shp: Shape format; the feature geometry itself.
- * .shx: Shape index format; a positional index of the feature geometry to allow seeking forwards and backwards quickly.
- * .dbf: Attribute format; columnar attributes for each shape, in dBase III format.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\shapeview.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To use a shape file

Properties and Parameters



Shapeview Plugin

use LOD: on

File Name: sitory/VizCuriousMaps/shape/tu.shp

- **Use LOD**: Enables/disables dynamic level of detail.
- **File Name**: Defines a path to the shape file (*.shp), containing the border definitions. Shape files are bought from vendors specialized in Geographic Information System (GIS) and holds the actual shape data; polygons, splines, and others. Note that shape files must be stored in individual folders.

To use a shape file

1. Create a new group and add the ShapeView plug-in to it.
   - The Expert plug-in is automatically added to the group.
2. Open the ShapeView editor and load the shape file.
3. Add a material and/or a texture to the group.

See Also

- 3D Border Manager (Viz World plug-in)

11.1.31 Soft Clip Draw Pixels

Plugin for playing video clips by drawing pixels.

Choose a video file to play. Playback can be achieved by either animating over the frame numbers or by turning on live playback. While in live playback mode, the frames will be displayed synchronized to viz rendering, or according to the Frames Per Second (FPS) setting, depending on the play mode that is chosen.

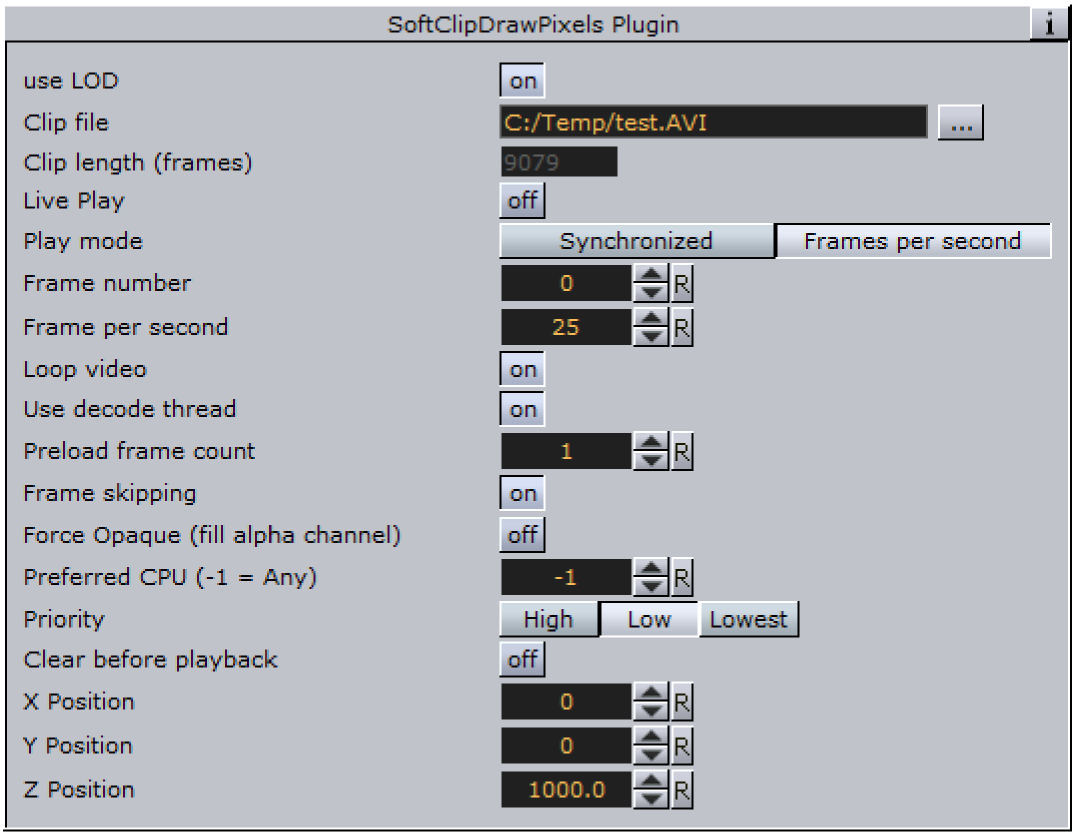
On machines with multiple CPUs turning on the decode thread will help balancing the CPU load over the CPUs. If you use live play, the Preload frame count can help reducing spike-loads and avoid dropped frames.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\SoftclipDrawPixels.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



SoftClipDrawPixels Plugin

use LOD	on
Clip file	C:/Temp/test.AVI ...
Clip length (frames)	9079
Live Play	off
Play mode	Synchronized | Frames per second
Frame number	0
Frame per second	25
Loop video	on
Use decode thread	on
Preload frame count	1
Frame skipping	on
Force Opaque (fill alpha channel)	off
Preferred CPU (-1 = Any)	-1
Priority	High | Low | Lowest
Clear before playback	off
X Position	0
Y Position	0
Z Position	1000.0

- **use LOD**: Enables/disables dynamic level of detail.
- **Clip file**: Loads the AVI clip file.
- **Live Play**: The clip is being automatically played.
- **Play mode**: Plays the frames synchronized to viz rendering or according to the given Frames per second.
- **Frame per second**: Renders the number of frames per second.
- **Loop video**: Plays the video in an endless loop.
- **Use decode thread**: Useful if the machine owns more than one CPU core. It increases the rendering performance if this parameter is set.
- **Preload frame count**: Increases the performance through loading frames in RAM before they are currently needed. Recommended setting is 1.
- **Frame skipping**: Skips frames if the machine hasn't enough performance to render all frames in the given time range instead of slowing down the video playback.
- **Force Opaque (fill alpha channel)**: Ignores the transparency.
- **Preferred CPU (0 = Any)**: Use this option if you want choose manually a core for rendering and processing. Choose 0 for automatically selection.

- **Priority**: Allows you to set the priority if the entire achievement is needed because of clip rendering lapse. Setting priority to Low should not reduce the overall render performance.
- **X, Y and Z Position**: Changes the X, Y and Z coordinates for clip translation.

11.1.32 Sphere

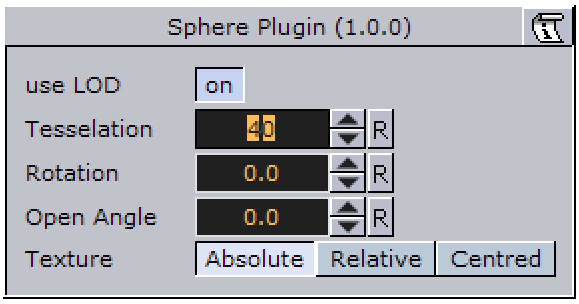
Use this plug-in to create a simple sphere with some attributes.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Sphere.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



Sphere Plugin (1.0.0)

use LOD: on
Tesselation: 40
Rotation: 0.0
Open Angle: 0.0
Texture: Absolute | Relative | Centred

- **Use LOD**: Enables/disables dynamic level of detail.
- **Tessellation**: Sets the degree of detail for the sphere.
- **Rotation**: Allows you to rotate the sphere around its Y-axis.
- **Open Angle**: Allows you to create an open angle in the sphere.
- **Texture**: Allows you to select between three different positions of the texture mapping coordinates when a vertex texture is mapped onto the sphere. To visualize the effect of the different settings, add a plain texture onto the sphere, like a chessboard image and set the texture mapping to vertex in the texture editor. Create an opening in the sphere and change the sphere texture parameters. Increase and decrease the opening angle with the different settings enabled and the effect should be visible.
  - **Absolute**: Compresses and expands the texture mapping coordinates as the opening of the sphere changes.
  - **Relative**: Cuts/adds texture coordinates as the opening changes. The cut/add is done on the one side of the opening angle. The coordinates are anchored at the other side.
  - **Centered**: Does the same as the relative mapping, but cuts/adds coordinates from both sides. The texture is anchored in a centered position on the sphere. The center of the sphere is 180° from the point where the opening angle starts.

11.1.33 Spline Path

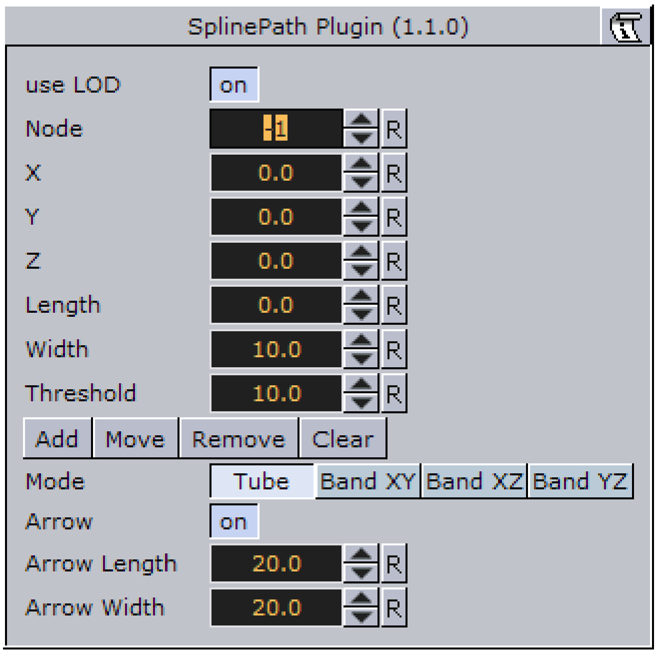
Use this plug-in if you want create complex paths with an arrow look.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\SplinePath.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a spline path

Properties and Parameters



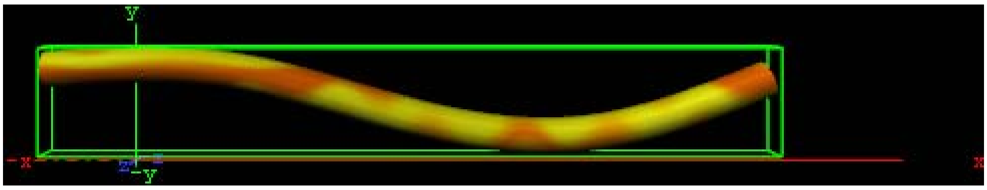
SplinePath Plugin (1.1.0)

use LOD: on
Node: -1
X: 0.0
Y: 0.0
Z: 0.0
Length: 0.0
Width: 10.0
Threshold: 10.0
Add | Move | Remove | Clear
Mode: Tube | Band XY | Band XZ | Band YZ
Arrow: on
Arrow Length: 20.0
Arrow Width: 20.0

- **use LOD**: Enables/disables dynamic level of detail.
- **Node**: Define the node index you want. For example node 3 or node 25.
- X, Y and Z: It sets the 3D point to define a node.
- **Length**: Stretches the entire path in X-direction.
- **Width**: Stretches the entire path in Y-direction.
- **Threshold**: Increase this value if you do not need a high path accuracy but more performance.
- **Add, Move, Remove and Clear**: Add adds a new node with an specific index. Move, moves an existing node to another X, Y, Z-point. Remove removes an existing node. Clear clears all nodes.
- **Mode**: The following options are available:
  - **Tube**: Renders the entire object.
  - **Band XY**: Renders only the front side.

  - **Band XZ**: Renders only the top side.
  - **Band YZ**: Renders only the left side.
- **Arrow**: Creates an array look.
- **Arrow Length**: This option works only if the Arrow toggle is switched to “on”. It stretches the arrow.
- **Arrow Width**: This option works only if the “Arrow” toggle is switched to “on”. It changes the arrow width.

To create a spline path



1. Create a new group and add the Spline Path plug-in to it.
2. Add the Expert plug-in to the group.
3. Open the Expert editor and enable (On) the Back Face property.
4. Add a material and/or a texture to the group.
5. Open the Spline Path editor and set the following parameters:
   - Set Node to 0, X to 50.0, Y to 50.0, and click Add.
   - Set Node to 1, X to 150.0, Y to 25.0, and click Add.
   - Set Node to 2, X to 250.0, Y to 10.0, and click Add.
   - Set Node to 3, X to 350.0, Y to 45.0, and click Add.

11.1.34 Spline Strip

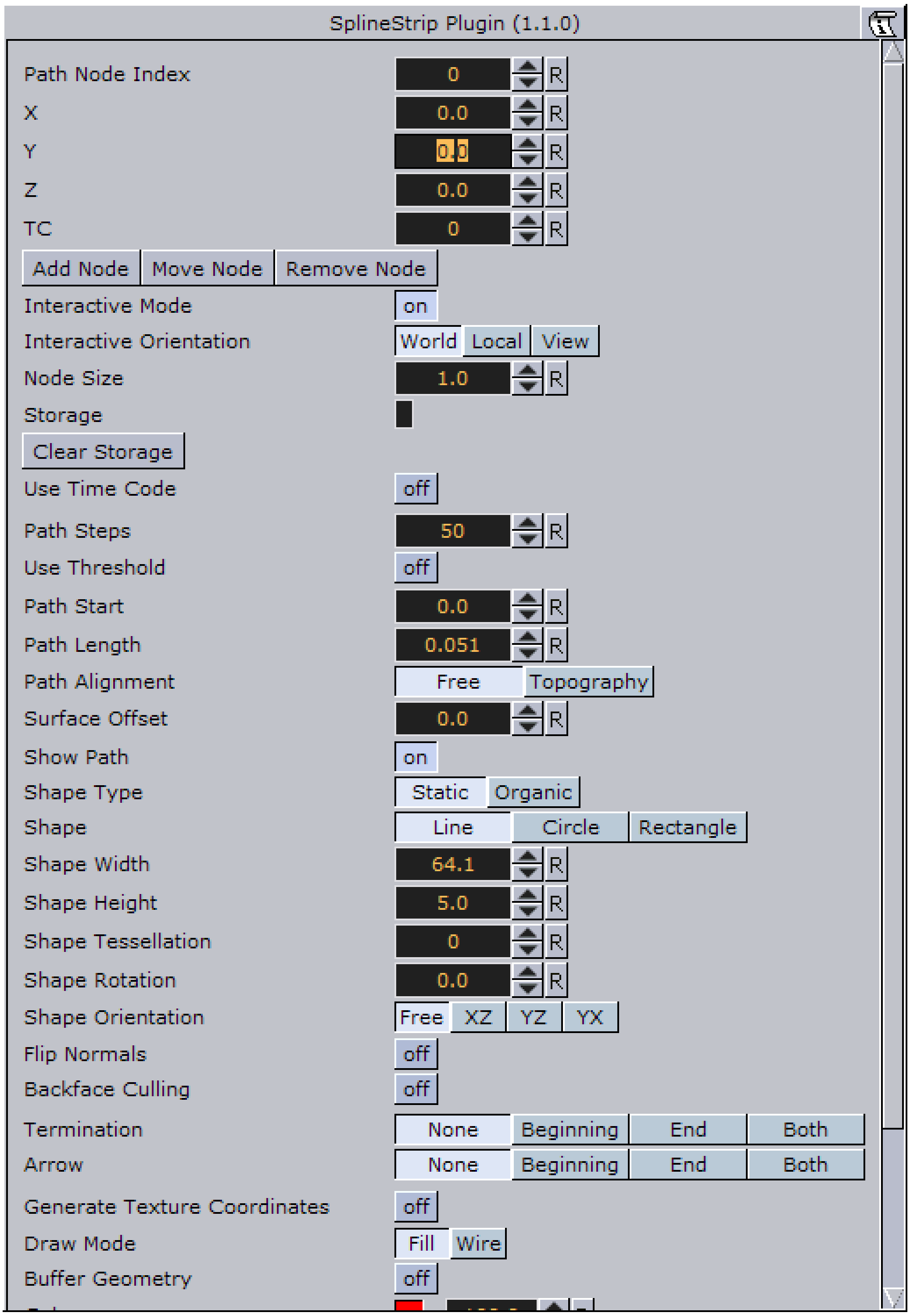
The Spline Strip plug-in is used for creating 3D interactive loft extrusions which are able to be animated in their length and surface. You can use modifiable lines, circles or rectangles for the extrusion. Arrows and terrain alignment are also supported.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\SplineStrip.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a simple spline strip

Properties and Parameters



SplineStrip Plugin (1.1.0)

Path Node Index	0
X	0.0
Y	0.0
Z	0.0
TC	0
Add Node / Move Node / Remove Node	
Interactive Mode	on
Interactive Orientation	World / Local / View
Node Size	1.0
Storage	
Clear Storage	
Use Time Code	off
Path Steps	50
Use Threshold	off
Path Start	0.0
Path Length	0.051
Path Alignment	Free / Topography
Surface Offset	0.0
Show Path	on
Shape Type	Static / Organic
Shape	Line / Circle / Rectangle
Shape Width	64.1
Shape Height	5.0
Shape Tessellation	0
Shape Rotation	0.0
Shape Orientation	Free / XZ / YZ / YX
Flip Normals	off
Backface Culling	off
Termination	None / Beginning / End / Both
Arrow	None / Beginning / End / Both
Generate Texture Coordinates	off
Draw Mode	Fill / Wire
Buffer Geometry	off

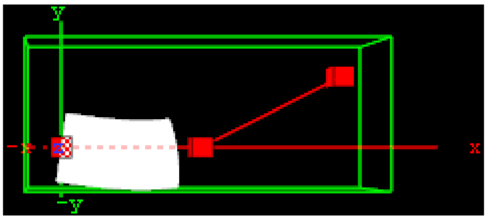
- **Path Node Index**: Defines where Path Nodes get added, moved or removed. The first node within a path owns the index 0. If you add nodes at the end, then the Path Node Index is increased automatically to provide a simple path generation.
- **X, Y and Z**: Sets the Path Node's X, Y and Z-position.
- **TC**: Sets the Path Node's TC-position.

- **Add Node**: Adds a path node at the current Path Node Index with position X, Y, Z and TC.
- **Move Node**: Moves the current path node to your desired X, Y, Z and TC position.
- **Remove Node**: Removes a path node at the current Path Node Index.
- **Interactive Mode**: If you have finished the process of setting path nodes then you are able to edit it simpler by mouse. Just drag and drop the displayed nodes to a new position. Left Mouse Button: move node in XY. Middle Mouse Button: move node in XZ.
- **Interactive Orientation**: World, Local, View. Move nodes interactive with world-, local-, or viewpoint- coordinates.
- **Node Size**: Sets the interactive node's size.
- **Storage**: Is used for storing your defined path nodes in a #-separated string. It is possible to edit nodes directly in the Storage Text field.
- **Clear Storage**: Clears the spline strip object and its storage.
- **Use Time Code**: Activates Time Code functionality.
- **TC Current**: Contains the current Time Code of an external Time Code source.
- **TC Start**: Sets the starting point of your Time Code animation.
- **TC Reference**: If you are using more than one spline strip then they probably won't start all at the same Time Code position. But you can change that by a simple time shift with the Help of TC Reference.
- **Path Steps**: Defines SplineStrip tessellation.
- **Use Threshold**: You can optimize your spline strip with that option. Path Steps are added at necessary positions only. So the number of Path Steps increases with a decreasing Threshold.
- **Threshold**: Sets the Threshold for Path Steps generation.
- **Path Start**: Sets the starting point of the path geometry.
- **Path Length**: It defines the length of our spline strip object. Feel free to animate that value!
- **Path Alignment**: Fits the spline strip to Terrain plugins in viz. (e.g. use Terrain for drawing paths over a mountain). Available options are; Free and Topography.
- **Surface Offset**: Shifts Path on Y axis.
- **Use Surface Normal**: If activated then the surface offset is computed via terrain normals.
- **Show Path**: Shows the Path which you have defined at the beginning. This option is quite helpful when you are editing Path Nodes.
- **Shape Type**: If you activate the Organic Mode, your spline strip will start to wobble in dependency of the adjusted Organic Amplitude, Organic Speed, and Organic Period's Length. Available options are; Static and Organic.
- **Shape**: Defines the shape used for the loft extrusion. Available options are; Line, Circle and Rectangle.
- **Shape Width**: Sets the shape's width.
- **Shape Height**: Sets the shape's height.
- **Shape Tessellation**: Defines how many steps are used to draw the desired shape.
- **Shape Rotation**: You are able to move the spline strip shape up to 360 degrees.
- **Shape Orientation**: When set to Free, shapes are rotated in dependency of the adjusted Shape Rotation value. When set to XZ, YZ or YX, shapes are auto

rotated to obtain a constant SplineStrip width in the desired aspect. When set to Terrain, shapes are rotated to fit the terrain beneath them.

- **Translate Vertices To Terrain**: Moves every single vertex to fit the terrain. This mode only makes sense if you are using Line extrusion.
- **Flip Normals**: Flips spline strip normals.
- **Backface Culling**: Switch off drawing spline strip's backside.
- **Organic Amplitude**: Adjusts the animated diameter reduction from 0% to 100%.
- **Organic Speed**: Changes speed of organic mode animation. Also negative values are allowed.
- **Organic Period's Length**: Sets the diameter reduction function's period length.
- **Termination**: Closes our spline strip object at the Beginning, End or at Both sides. Available options are; None, Beginning, End and Both.
- **Arrow**: Puts an Arrow to the Beginning, End or Both sides. Available options are; None, Beginning, End, Both
- **Object Width**: Changes arrow width.
- **Object Height**: Changes arrow height.
- **Object Length**: Changes arrow length.
- **Object Tessellation**: Changes arrow tessellation.
- **Generate Texture Coordinates**: Activate this function if you are using vertex maps on your spline strip.
- **Stretch Texture to Path Length**: If activated then the map is stretched to the actual path length.
- **Draw Mode**: Switch between fill- and wire-frame mode. Available options are; Fill and Wire.
- **Buffer Geometry**: The whole spline strip data is buffered to memory. If you use spline strips with a huge number of vertices then Buffer Geometry is able to improve performance. Activate this mode only if you do not use any kind of animation in your current spline strip.
- **Color**: Defines path's and interactive-mode-node's color. Color is also used if Wire is activated and no maps are applied onto your spline strip object.
- **Child Distance**: Defines distance between path end and child container.
- **Rotate Child**: Activates child container rotation.

To create a simple spline strip



1. Create a new group and add the Spline Strip plug-in to it.
2. Open the Spline Strip editor.
3. Use all default values, and click Add Node.
4. Change X to 100.0, and click Add Node.
5. Change X to 200.0, Y to 50.0, and click Add Node.
6. Set Shape Width and Shape Height to 50.0.
7. Go to Shape and change it to Circle.

8. Set Shape Tessellation to 20.
9. Play with the Path Length parameter. Optionally you can click and try Interactive Mode.
10. Change Node Size to for example 15.0.
11. Drag and drop the small Cubes.

11.1.35 Spring

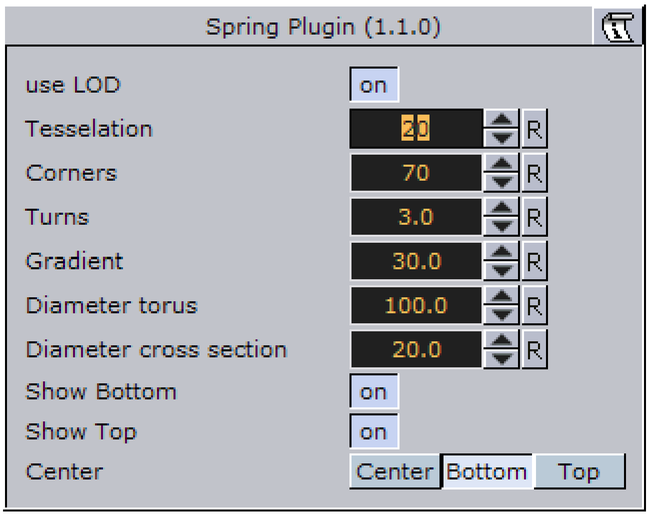
The spring object creates a spiral spring. Be aware this object can easily be very heavy to render if a high level of tessellation and corners are selected.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Spring.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a spring

Properties and Parameters

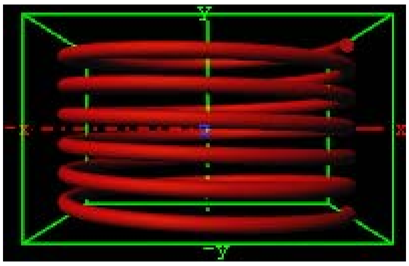


Spring Plugin (1.1.0)

use LOD	on
Tesselation	20
Corners	70
Turns	3.0
Gradient	30.0
Diameter torus	100.0
Diameter cross section	20.0
Show Bottom	on
Show Top	on
Center	Center [✓ Bottom] Top

- **Use LOD**: Enables/disables dynamic level of detail.
- **Tessellation**: Sets the degree of detail for the spring
- **Corners**: Sets the number of corners the spring should have.
- **Gradient**: Sets the size of the spring gradient.
- **Diameter torus**: Sets the diameter of the springs.
- **Diameter cross section**: Sets the diameter of the spring cord.
- **Show Bottom**: Shows/hides bottom of the spring.
- **Show Top**: Shows/hides Top.
- **Center**: Sets the location of the spring center, either **Center**, **Bottom** or **Top**.

To create a spring



1. Create a new group and add the Spring plug-in to it.
2. Open the Spring editor and set the following parameters:
   - Set Turns to 6.0.
   - Set Gradient to 20.0.
   - Set Diameter torus to 200.0.
   - Set Diameter cross section to 100.0.
   - Set Center to Center.
3. Add a material and/or a texture to it.

11.1.36 Torus

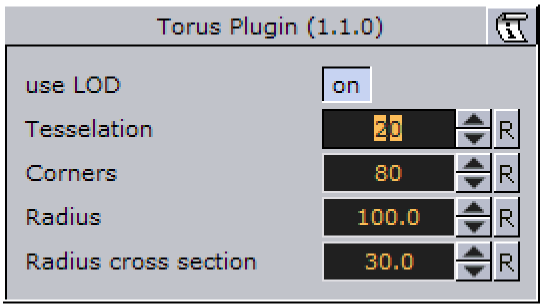
Use this plug-in to create a torus with different attributes.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Torus.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a torus

Properties and Parameters



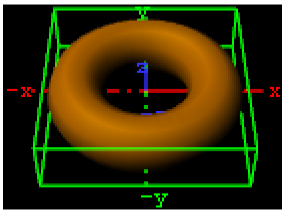
Torus Plugin (1.1.0)

use LOD	on
Tesselation	20
Corners	80
Radius	100.0
Radius cross section	30.0

- **Use LOD**: Enables/disables dynamic level of detail.
- **Tessellation**: Sets the degree of detail for the torus.
- **Corners**: Sets the number of corners on the torus.

- **Radius**: Sets the radius of the torus from its center in the middle of the hole.
- **Radius cross section**: Sets the radius of the torus cross section or “tube”.

To create a torus



1. Create a new group and add the Torus plug-in to it.
2. Open the Torus editor and set the following parameters:
   - Set Radius to 50.0.
   - Set Radius cross section 20.0.
3. Add a material and/or a texture to it.

11.1.37 Trio Scroll

The Trio Scroll plug-in is essentially a geometry plug-in for positioning and scrolling a fixed set of elements in a specified direction, often used for credit lists or crawls.

Scrolling templates can be built in Viz Trio using the Create New Scroll feature. The scroll scene that is automatically created by Viz Trio can be edited in Viz Artist, but it is also possible to build them manually.

Some of the supported features are:

- Scrolls can be created by receiving element data from Viz Trio's XML format.
- Fine-tuned control of spacing between individual elements.
- Easepoints (ease in and out) on particular elements.

Scrolls are usually built in Viz Trio, but it is possible to add elements in Viz Artist by adding merged objects under the `base_elements` group and clicking the **Initialize** button.

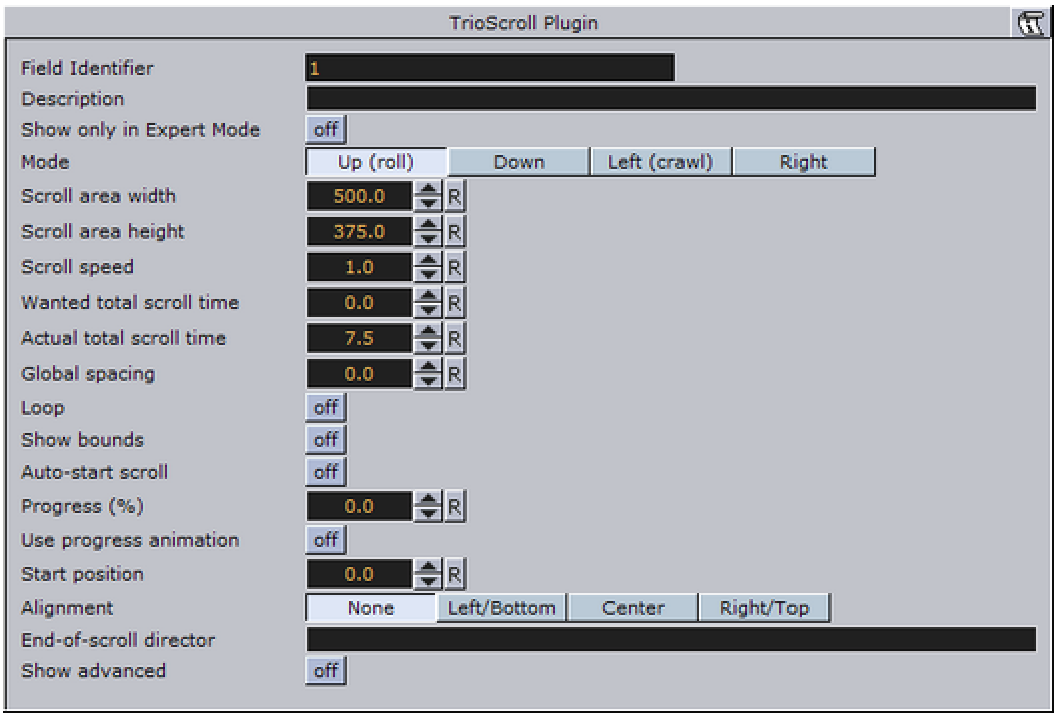
Adding the Trio Scroll plug-in to the scene tree will automatically add the Control Object plug-in (if it is not present).

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\TrioScroll.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



TrioScroll Plugin

Field Identifier	1
Description	
Show only in Expert Mode	off
Mode	Up (roll) | Down | Left (crawl) | Right
Scroll area width	500.0
Scroll area height	375.0
Scroll speed	1.0
Wanted total scroll time	0.0
Actual total scroll time	7.5
Global spacing	0.0
Loop	off
Show bounds	off
Auto-start scroll	off
Progress (%)	0.0
Use progress animation	off
Start position	0.0
Alignment	None | Left/Bottom | Center | Right/Top
End-of-scroll director	
Show advanced	off

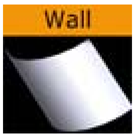
- **Field Identifier**: When making templates for Viz Trio, this should be a numeric value, Range: 1-n. For Viz Content Pilot and Viz Ticker, this can also be a descriptive text string. This value is used by the control clients to identify the editable element in the scene. In Viz Trio the numbers from the control plug-ins are used to create the tab-order between the editable elements.
- **Description**: A description of the tab field or editable object. This will be used as a description for the elements when used in the control clients.
- **Show only in Expert Mode**: If Viz Trio is connected, changes are possible if this toggle is switched to off.
- **Mode**: Select the direction the scroller should move across the screen.
- **Scroll area width**: Select the area in width the scroller should use.
- **Scroll area height**: Select the area in height the scroller should use.
- **Scroll speed**: Set the scroller speed.
- **Wanted total scroll time**: Sets the scroll time.
- **Actual total scroll time**: Sets the scroll time.
- **Global spacing**: Sets spacing between the pages.
- **Loop**: Enables the Loop property Turn on or off (default) looping. Turning off looping will not stop the playlist from being played till its end.
- **Show bounds**: Enables and displays the scroll’s bounding-box on the preview and program renderer.
- **Auto-start scroll**: Enables the Auto-scroll on take property.
- **Start position**: Sets the start position for the scroll. This setting will affect the Actual total scroll time.
- **Alignment**: Sets the alignment of the scroll. The position is relative to the position of the scene design. Available options are: None, Left/Bottom, Center and Right/Top.

- **End-of-scroll director**: Triggers a Continue on a specified end director. The director is triggered when the scroll leaves the scroll area. It is only supported in normal mode (real time), not in post rendering or NLE mode.

See Also

- *Create New Scroll* in the *Viz Trio User's Guide*.
- Control Object
- Trio Scroll Element

11.1.38 Wall



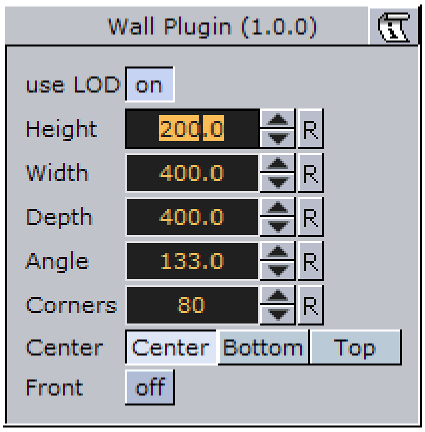
The wall object is a curved wall. The size and curving of the wall can be customized.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Wall.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a wall

Properties and Parameters



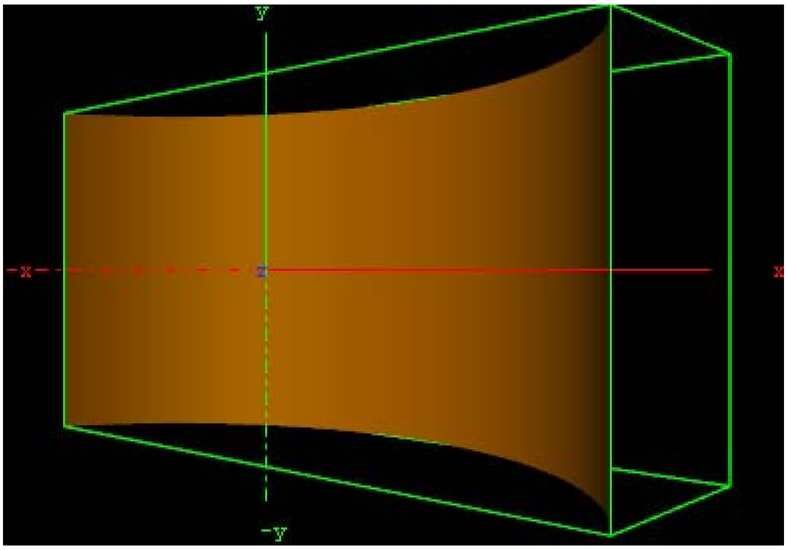
Wall Plugin (1.0.0)

use LOD: on
Height: 200.0
Width: 400.0
Depth: 400.0
Angle: 133.0
Corners: 80
Center: [✓ Center] Bottom Top
Front: off

- **Use LOD**: Enables/disables dynamic level of detail.
- **Height**: Sets the height of the wall.
- **Width**: Sets the width of the wall.
- **Depth**: Sets the depth of the wall curving. The higher this value is set, the more curved the wall will appear.
- **Angle**: Sets the angle of the walls curving. Use this parameter together with **Depth**: To achieve the wanted curving of the wall.
- **Corners**: Sets the number of internal corners the wall should be built up from.

- **Center**: Sets the center axis of the object. Choose between **Center**, **Bottom** or **Top**.
- **Front**: Lets you switch between front and back.

To create a wall



1. Create a new group and add the Wall plug-in to it.
2. Open the Wall editor and set the following parameters:
   - Set Height to 200.0.
   - Set Width to 400.0.
   - Set Depth to 400.0.
   - Set Center to Center.
3. Add a material and/or a texture to it.
4. Open the Transformation Editor and set Rotation Y to -45.0.

11.1.39 Wave

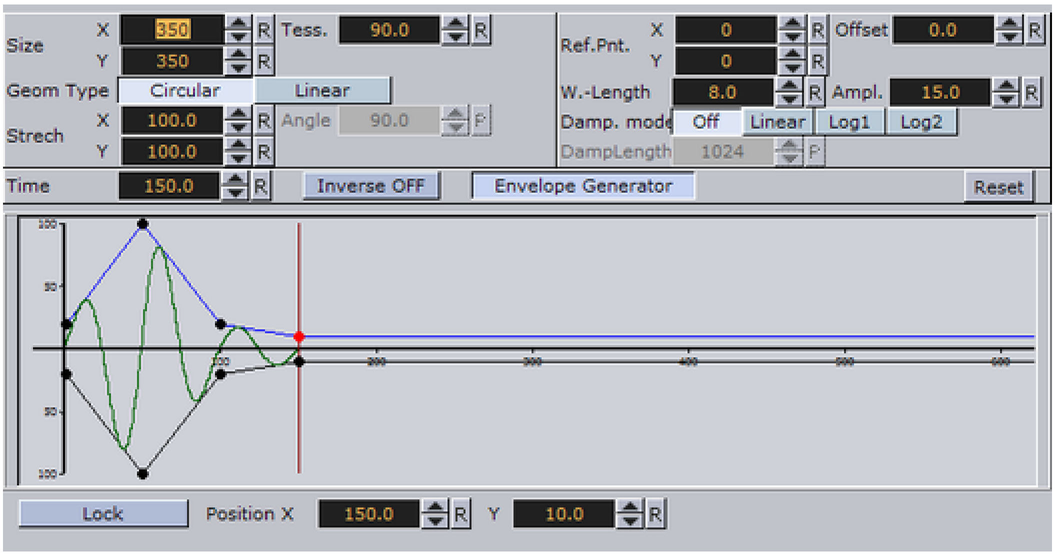
The wave allows you to create a wave sequence on a flat surface. The properties of the sequence are visualized in a graphical display in the wave editor.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Wave.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a wave

Properties and Parameters



Size X 350

Size Y 350

Tess. 90.0

Geom Type: Circular / Linear

Strech X 100.0

Strech Y 100.0

Angle 90.0

Ref.Pnt. X 0

Ref.Pnt. Y 0

Offset 0.0

W.-Length 8.0

Ampl. 15.0

Damp. mode: Off / Linear / Log1 / Log2

DampLength 1024

Time 150.0

Inverse OFF

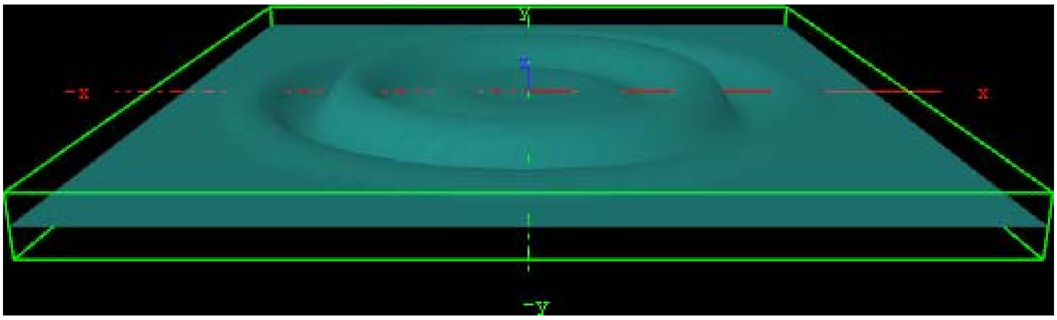
Envelope Generator

Position X 150.0

Y 10.0

- **Size X** and **Y**: Sets the size of the rectangle where the waves appear.
- **Geom Type**: Allows you to select between **Circular** and **Linear** waves.
- **Stretch**: Allows you to customize the form of the circular waves. By altering the stretch **X** and **Y** values, you can for instance obtain an oval or a compressed form of the waves.
- **Time**: Is the timeline for a wave sequence. You see a thin red vertical line moving in the graphical display as you alter the time value. You can also alter the value by dragging the red line.
- **Tess.**: Sets the tessellation / level of detail.
- **Angle**: Sets the angle of the waves when a linear mode is chosen.
- **Ref.Pnt X** and **Y**: Sets the position of the waves starting point.
- **W. Length**: Sets the length of the waves.
- **Damp. mode**: Enables damping of the waves. The amplitude of the wave is reduced by the damp length calculated from the center of the wave.
- **Damp. Length**: Sets the length the distance that is used to damp the waves in the **Damp. mode** function.
- **Offset**: Sets an offset value of the **Ref.Pnt.**
- **Ampl**: Amplifies the waves, resulting in both bigger and higher waves.
- **Inverse OFF/ON**: Inverts the wave curves.
- **Envelope Generator**: Enable this option if you want to draw the wave manually. When disabled the wave is built by a sinus wave function.
- **Position X** and **Y**: Allows you to edit the position values of the wave control points. Select a point by clicking on it. Its color changes to red to show that it is selected. Now you can drag it around.
- **Lock**: Allows you to decide whether the points controlling the crest of the waves should be edited in a locked manner, with the points controlling the troughs of the waves or conversely, if they are to be edited separately.
- **Spline Curve**: The control points of the spline curve, that shows the wave sequence, can be edited by using the cursor. Click on a point and drag it to the desired position.

To create a wave



1. Create a new group and add the Wave plug-in to it.
2. Open the Transformation Editor and set Rotation X to -75.0.
3. Add a material and/or a texture to it.
4. Open the Wave editor and set the following parameters:
   - Set Size X and Y to 350.0.
   - Set W.-Length to 8.0.
5. Click the Lock button to unlock the spline curve editor.
6. Play with the spline curve handles, Time, Position X and Y values to see how the wave behaves with different settings.

11.2 Vizual Data Tools

The following geometry plugins are housed in the Vizual Data Tools folder:

- Overview
- Area Chart
- Bar Chart
- Line Chart
- Pie Chart
- Scatter Chart
- Stock Chart

See Also

- Data Fit
- Data Import
- Data Label
- Data Storage
- Tutorial on www.vizrt.com under the Training section

11.2.1 Overview

Viz Vizual Data Tools (VDT) were designed for Viz Artist 3 to display any kind of statistic data in the best fitting way. All plugins use the same transport channel: Viz Artist 3 Shared Memory. Shared Memory is a map holding user-defined variables indexed by a string - this is also known as a key-value pair.
There are 3 types of Shared Memory Maps - the difference between them is their distribution location:

- Scene
- Local - (local engine, data sharing between scenes in different layers)
- Distributed - Shared Memory (share data between engines connected to the same GH)

The key-value pairs of Shared Memory can be accessed via UDP (external control applications), Viz commands, Viz plugin API and Viz Scripting language (refer to the Script documentation, section Data Sharing for more details). Shared memory contains functionality which allows you to register for value changes. VDT is using this functionality which means that if a value in the shared memory map is changed the chart which is registered to the certain shared memory key immediately redraws itself.

For example there could be a control PC running a Viz Content Pilot template that sends stock prices to a Viz Engine for displaying the data as a stock chart using SDI output and to another one using HD video output. The data has to be sent only once. If you run a 3D stereoscopic render system with two engines you can also just send the data to one engine. Another way this could be used would be to display data on each renderer in a different chart type.

The VDT plugin suite consists of Geometry Plugins and Container Plugins. The geometry part is responsible for drawing the chart itself while the container plugins are used for chart labeling, data scaling, and so on.

To learn more about the use of Vizual Data Tools you can download a tutorial on www.vizrt.com under the Training section.

See Also

- Container Plugins
- Geometry Plugins
- Scripting
- Scripting Data Types and Procedures

11.2.2 Area Chart

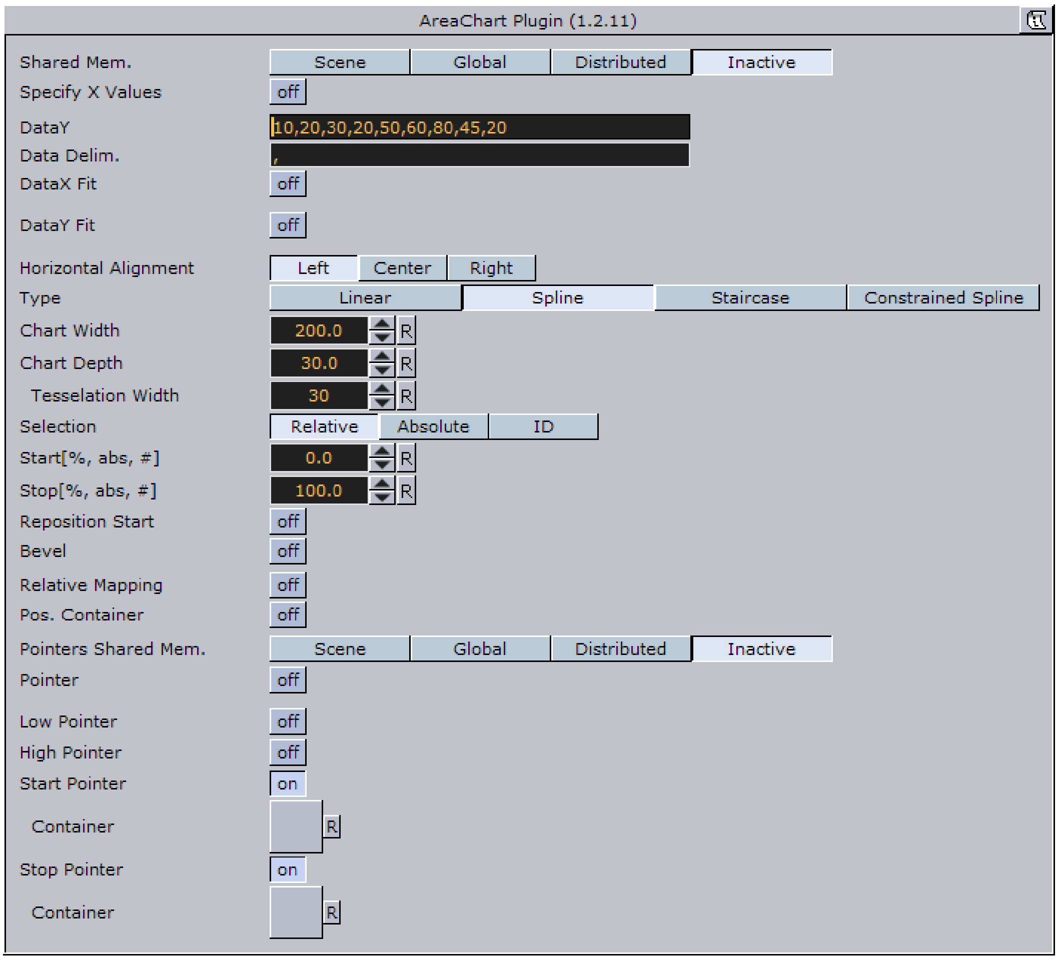
This plug-in draws an area chart, filled with data out of a Shared Memory Map. You can use delimited strings or arrays for data transfer via Scene-, Global- or Distributed-Shared Memory.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\AreaChart.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a scene using area chart

Properties and Parameters



AreaChart Plugin (1.2.11)

Shared Mem. Scene Global Distributed [✓ Inactive]
Specify X Values off
DataY 10,20,30,20,50,60,80,45,20
Data Delim. ,
DataX Fit off
DataY Fit off
Horizontal Alignment [✓ Left] Center Right
Type Linear [✓ Spline] Staircase Constrained Spline
Chart Width 200.0
Chart Depth 30.0
Tesselation Width 30
Selection [✓ Relative] Absolute ID
Start[%, abs, #] 0.0
Stop[%, abs, #] 100.0
Reposition Start off
Bevel off
Relative Mapping off
Pos. Container off
Pointers Shared Mem. Scene Global Distributed [✓ Inactive]
Pointer off
Low Pointer off
High Pointer off
Start Pointer on
Container
Stop Pointer on
Container

- **Shared Mem.**: Changes between Scene-, Global- and Distributed-Shared Memory. Use Inactive memory to not forward any values via Shared memory.
- **Specify X Values**: Enables DataX input.
  - **Key DataX**: Shared Mem. key name for X values.
  - **DataX**: Parameter for X values if Shared memory is set to Inactive.
- **Key DataY**: Shared Mem. key name for Y values.
- **DataY**: Parameter for Y values if Shared memory is set to Inactive.
- **Transfer Mode**: Sets string- or array-based data transfer. Not available when set shared memory is set to Inactive.
- **Data Delim.**: Defines the value separator sign(s).
- **DataX, -Y Fit**: Enables data normalization.
  - **DataX, -Y Scale**: Scales input by the selected factor.
  - **DataX, -Y Offset**: Adds an offset to the incoming data.
  - **DataX, -Y Auto Scale**: Enables automatic data normalization.
  - **DataX, Y Fit Size**: **Total** scales the whole chart to the defined borders. **Current** scales the current chart segment to the set borders.
  - **DataX, -Y Detect Limits**: Detects minimum and maximum of all values and scales them to adjusted Start and Stop. This option is used to upscale the interesting part of the chart - especially if there are only little changes between the data values.
  - **DataX, -Y Threshold**: Adds a definable offset to the detected limit.

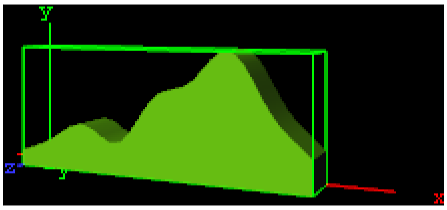
  - **DataX, -Y Start**: Lower Auto Scale edge.
  - **DataX, -Y Stop**: Upper Auto Scale edge.
- **Horizontal Alignment**: Sets horizontal orientation to left, center or right.
- **Type**: Creates different graph looks. Available options are Normal, Spline, Staircase and Constrained Spline.
  - **Normal**: Enables direct connections between entered values.
  - **Spline**: Interpolates extra values to chamfer the graph.
  - **Staircase**: Creates a staircase look.
  - **Constrained Spline**: Same as Spline mode but with a different algorithm to prevent overshooting.
- **Stair Width**: Defines width for a single stair or for all stairs if Specify X Values is activated.
- **Chart Width**: Adjusts the chart width.
- **Chart Depth**: Adjusts extrusion.
  - **Tessellation Width**: Sets tessellation in Spline mode.
- **Selection**: Specifies Start- and Stop-type.
  - **Relative**: 0%: 100%.
  - **Absolute**: depends on your specified data.
  - **ID**: Value ID starting from 0.
- **Start[%, abs, #]**: Graph's starting point.
- **Stop[%, abs, #]**: Graph's stopping point.
- **Reposition Start**: Translates the whole chart always to the same starting point independent of varying Start and Stop positions.
- **Bevel**: Activates Bevel mode.
  - **Bevel Bottom**: Chamfer the chart's bottom.
  - **Bevel Size**: Adjusts bevel's size.
  - **Bevel Steps**: Sets roundness via the number of segmentation steps.

Note: A chamfer is a beveled edge connecting two surfaces.

- **RelativeMapping**: Stretches texture to fit graph width.
- **Pos. Container**: Translates every child container to a bar's top.
  - **Container Offset**: Adds a certain offset to each container.
  - **Container Offset Z**: Adds a Z Axis offset to each container.
  - **Center Container**: Centers each translated container.
- **Pointers Shared Mem.**: Select the map where the pointer values should be distributed to.
- **Key Data Pointers**: Define Shared Memory Keyname.
- **Pointer**: Activates the Pointer.
  - **Container**: Define Container which should be used as pointer.
  - **Selection**: Choose navigation type.
  - **Position[%, abs, #]**: Sets pointer position in dependency of the selected navigation type.
  - **Offset**: Define pointer offset from the chart.
  - **Normal Offset**: Rotate offset to the direction of the current surface normal.
  - **Rotate**: Rotate the pointer to the direction of the current surface normal.
  - **Offset Z**: Add an additional Z offset to the pointer container.
  - **Center Container**: Center Container - useful for varying pointer sizes.
- **Low Pointer**: Activates the low pointer.

    - **Container**: Choose a container for the pointer.
- **High Pointer**: Activates the high pointer.
    - **Container**: Choose a container for the pointer.
- **Start Pointer**: Activates the Start Pointer.
    - **Container**: Choose a container for the pointer.
- **Stop Pointer**: Activates the stop pointer.
    - **Container**: Choose a container for the pointer.

To create a scene using area chart



1. Create a new group and add the Area Chart plug-in to it.
2. Open the Transformation Editor and set Position X and Y to -100.0 and Rotation Y to -25.0.
3. Add a material and/or a texture to it.
4. Open the Area Chart editor and set the following parameters:
    - Set the following DataY values: 10,20,30,20,50,60,80,45,20.
    - Set Type to Spline.
    - Set Width to 200.0.
    - Set Depth to 30.0.

To create a scene using area chart and data storage

1. Create a new group and add the Area Chart plug-in to it.
2. Add the Data Storage plugin to the same container as Area Chart.
    - The plugin will automatically jump to your Area Chart container.
3. Select your Area Chart plugin, and set the SharedMem. parameter to Scene.
4. Rename the Key Data Y parameter to `MyDataY`.
5. Open the Data Storage editor and enter the same key name for **Key Data1** as you just did in the Area Chart plugin to `MyDataY`, and set the SharedMem. parameter to Scene.
6. Go to the Data1 parameter in the Data Storage plugin and type the following text sequence into the text field. 10,40,50,20,80,90,60,50.
    - While you are typing you can see the graph already growing. We are using “,” as a delimiter sign which is the decimal sign in some countries. But you can use any delimiter you want. Just go to the chart plugin, AreaChart in our case, and set Data Delim. to the desired sign.

See Also

- Control Chart
- Data Fit
- Data Import
- Data Label
- Data Storage

11.2.3 Bar Chart

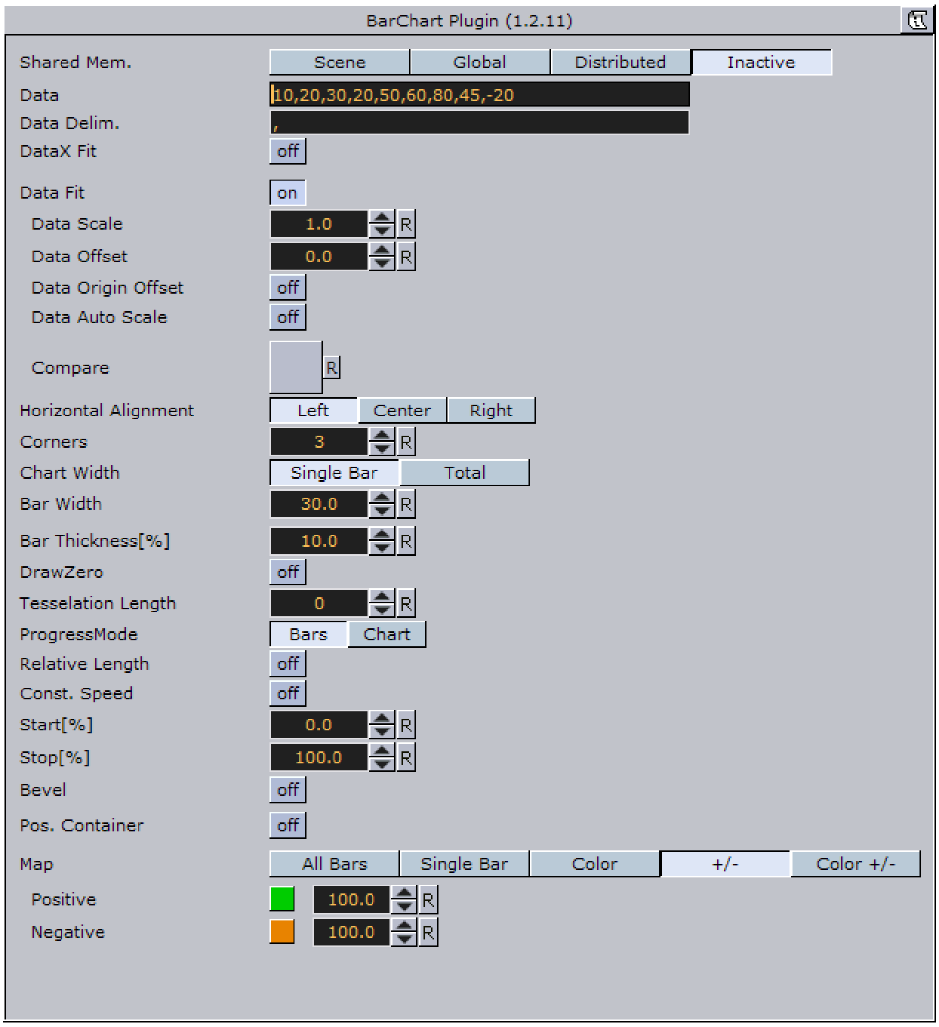
This plug-in draws a bar chart, filled with data out of a Shared Memory Map. You can use delimited strings or arrays for data transfer via Scene-, Global- or Distributed-Shared Memory.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\BarChart.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a bar chart

Properties and Parameters



BarChart Plugin (1.2.11)

Shared Mem. Scene Global Distributed [✓ Inactive]
Data 10,20,30,20,50,60,80,45,-20
Data Delim. ,
DataX Fit off
Data Fit on
Data Scale 1.0 R
Data Offset 0.0 R
Data Origin Offset off
Data Auto Scale off
Compare R
Horizontal Alignment [✓ Left] Center Right
Corners 3 R
Chart Width [✓ Single Bar] Total
Bar Width 30.0 R
Bar Thickness[%] 10.0 R
DrawZero off
Tesselation Length 0 R
ProgressMode [✓ Bars] Chart
Relative Length off
Const. Speed off
Start[%] 0.0 R
Stop[%] 100.0 R
Bevel off
Pos. Container off
Map All Bars Single Bar Color [✓ +/-] Color +/-
Positive 🟩 100.0 R
Negative 🟧 100.0 R

- **Shared Mem.**: Changes between Scene-, Global- and Distributed-Shared Memory. Use Inactive memory to not forward any values via Shared memory.
- **Data**: Shared Mem. key name.
- **Data**: Input parameter for data.
- **Transfer Mode**: Sets string- or array-based data transfer.
- **Data Delim.**: Defines the value separator sign(s).
- **Data Fit**: Enables data normalization.
  - **Data Scale**: Scales input by the selected factor.
  - **Data Offset**: Adds an offset to the incoming data.
  - **Data Auto Scale**: Enables automatic data normalization.
  - **Data Fit Size**: Scales complete chart in dependency of the defined Start and Stop settings (total) or scales the Current chart segment to the set borders.
  - **Data Detect Limits**: Detects minimum and maximum of all values and scales them to adjusted Start and Stop.
  - **Data Threshold**: Adds a definable offset to the detected limit.
  - **Data Start**: Lower Auto Scale edge.
  - **Data Stop**: Upper Auto Scale edge.

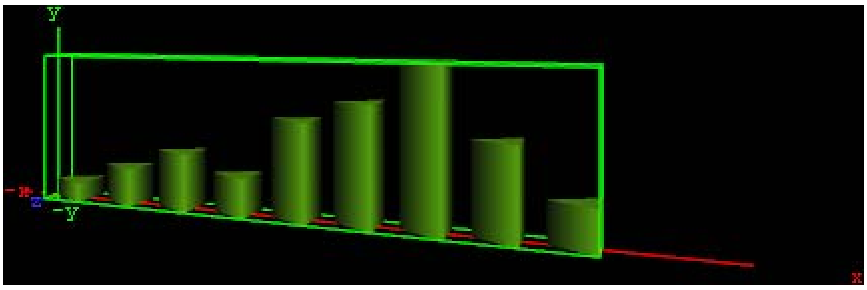
- **Compare**: Specify another BarChart for comparison. You have to set this parameter also in the other chart(s) so that each chart can react on changes of the other(s).
- **Horizontal alignment**: Sets horizontal orientation to left, center or right.
- **Corners**: Sets the bar's corners.
- **Chart Width**: Defines width for a single bar or for all bars.
- **Bar Width**: Value of the previous parameter.
- **Max Width**: This setting can be used to allow bars to grow only to a certain width.
- **Bar Gap RatioMax. Width**: This setting can be used to allow bars to grow only to a certain width.
- **Depth**: Defines if the bar depth should equal its width or lets you set a custom depth.
- **Bar Depth**: Adjusts bar depth in custom depth mode.
- **Bar Thickness[%]**: Adjusts the gap between the single bars, 0% means no gap.
- **Draw Zero**: If the bar's value is equivalent to 0 than there will be drawn a very flat bar instead of nothing.
- **Tesselation Length**: Tesselates bar length.
- **Progress Mode**:
  - **Bars**: Each bar grows to its value
  - **Chart**: The whole chart grows horizontally and the incoming bars already have their final height.
- **Relative Length**: If activated, each bar will have it's own 100% length (e.g. 7 bars equal 700%).
- **Const. Speed**: Sets the same animation duration for each bar.
- **Start[%]**: Graph's starting point
- **Stop[%]**: Graph's stop point
- **Bevel**: Activates Bevel mode.
  - **Bevel Top**: Chamfer bar's top.
  - **Bevel Bottom**: Chamfer bar's bottom.
  - **Bevel Size[%]**: Adjusts bevel's size from 0 to 100.
  - **Bevel Steps**: Sets roundness via the number of segmentation steps.

Note: A chamfer is a beveled edge connecting two surfaces.

- **Pos. Container**: Translates every child container to a bar's top.
  - **Container Offset**: Adds a certain offset to each container.
  - **Container Offset X**: Adds a X Axis offset to each container.
  - **Container Offset Y**: Adds a Y Axis offset to each container.
  - **Container Offset Z**: Adds a Z Axis offset to each container.
  - **Center Container**: Centers each translated container.
  - **Size Compensation**: This option can be used to reposition containers if the can change their size. E.g. Text geometries used as chart labels. The labels will vary in their width which requires a repositioning if a horizontal BarChart (rotated 90 deg) is used.
  - **None**: Container size compensation inactive.
  - **Width**: Compensates with changes.
  - **Height**: Compensates height changes.

- **Map**: Calculates texture coordinates for all bars, a single bar or defines vertex colors instead. Colors can be set individually for each bar or value dependent (+/- color). In the Color +/- mode you can override value dependent colors.
  - **Color ID**: Moves between the available vertex colors.
  - **Color**: Choose color for the current ID.
  - **Positive**: Choose color for positive values.
  - **Negative**: Choose color for negative values.
  - **Color Override**: Activates or deactivates overriding of a positive or negative color with a defined alternative color.

To create a bar chart



1. Create a new group and add the Bar Chart plug-in to it.
2. Open the Transformation Editor and set Position X and Y to -100.0 and Rotation Y to -25.0.
3. Add a material and/or a texture to it.
4. Open the Bar Chart editor and set the following parameters:
   - Set the following DataY values: 10,20,30,20,50,60,80,45,20.
   - Set Corners to 3.
   - Set Bar Width to 20.0.
   - Set Bar Gap Ratio[%] to 10.0.

See Also

- Control Chart
- Data Fit
- Data Import
- Data Label
- Data Storage
- Presenter

11.2.4 Line Chart

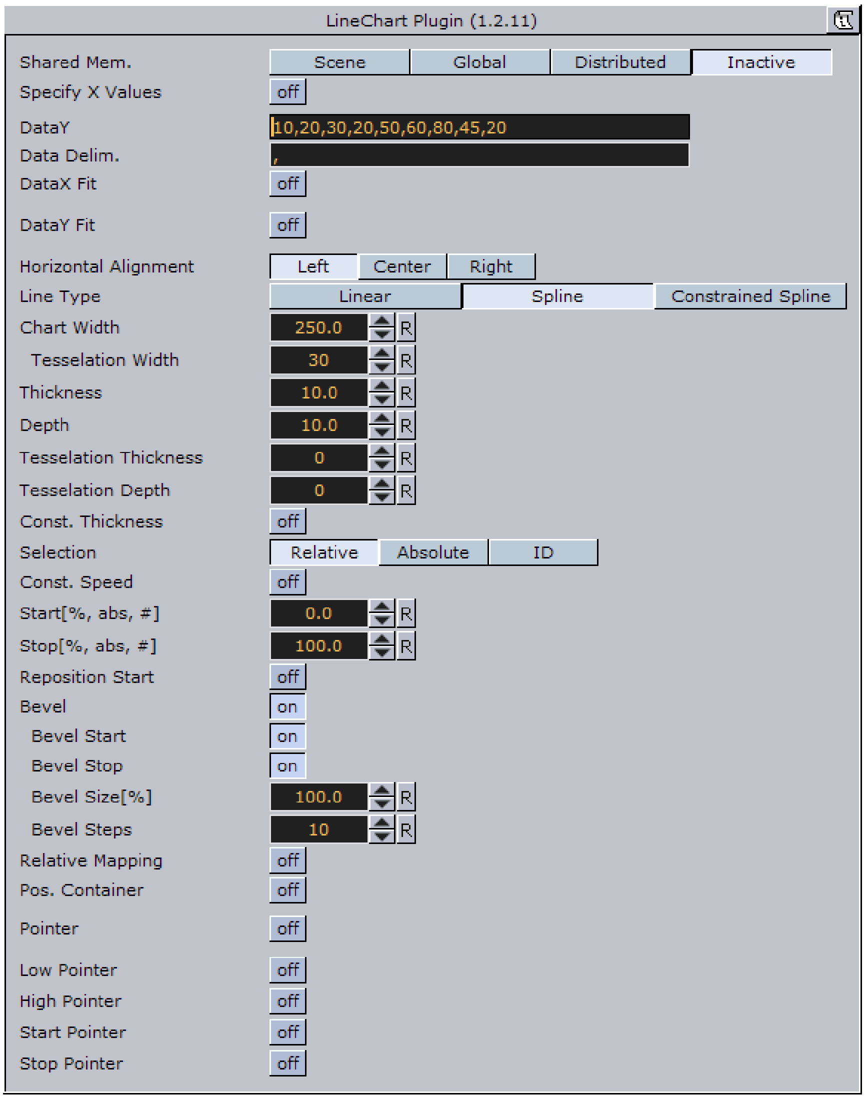
This plugin draws a line chart, filled with data out of a Shared Memory Map. You can use delimited strings or arrays for data transfer via Scene-, Global- or Distributed-Shared Memory.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\LineChart.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a line chart

Properties and Parameters



LineChart Plugin (1.2.11)

Shared Mem.	Scene, Global, Distributed, [✓ Inactive]
Specify X Values	off
DataY	10,20,30,20,50,60,80,45,20
Data Delim.	,
DataX Fit	off
DataY Fit	off
Horizontal Alignment	[✓ Left], Center, Right
Line Type	Linear, [✓ Spline], Constrained Spline
Chart Width	250.0
Tesselation Width	30
Thickness	10.0
Depth	10.0
Tesselation Thickness	0
Tesselation Depth	0
Const. Thickness	off
Selection	[✓ Relative], Absolute, ID
Const. Speed	off
Start[%, abs, #]	0.0
Stop[%, abs, #]	100.0
Reposition Start	off
Bevel	on
Bevel Start	on
Bevel Stop	on
Bevel Size[%]	100.0
Bevel Steps	10
Relative Mapping	off
Pos. Container	off
Pointer	off
Low Pointer	off
High Pointer	off
Start Pointer	off
Stop Pointer	off

- **Shared Mem.**: Changes between Scene-, Global- and Distributed-Shared Memory. Use Inactive memory to not forward any values via Shared memory.
- **Specify X Values**: Enables DataX input.
  - **DataX**: Shared Mem. key name for X values. DataX is the default input parameter for X values.
  - **Key DataX**: Shared Mem. key name for X values.
- **DataY**: Shared Mem. key name for Y values. DataY is the default input parameter for Y values.
- **Key DataY**: Shared Mem. key name for Y values.

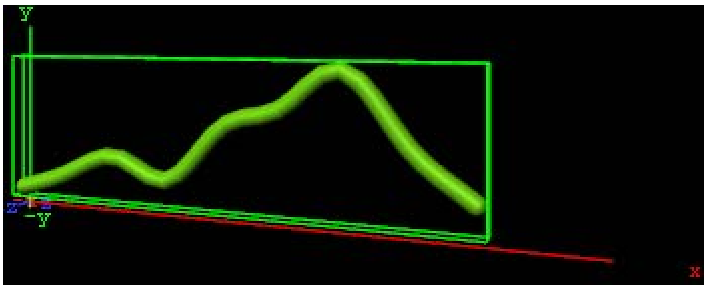
- **Transfer Mode**: Sets string- or array-based data transfer.
- **Data Delim.**: Defines the value separator sign(s).
- **DataX, -Y Fit**: Enables data normalization.
  - **DataX, -Y Scale**: Scales input by the selected factor.
  - **DataX, -Y Offset**: Adds an offset to the incoming data.
  - **DataX, -Y Auto Scale**: Enables automatic data normalization.
  - **DataX, -Y Fit Size**: **Total**: Scales the whole chart to the defined borders. **Current**: Scales the current chart segment to the set borders.
  - **DataX, -Y Detect Limits**: Detects minimum and maximum of all values and scales them to adjusted Start and Stop. This option is used to upscale the interesting part of a chart - especially if there are only little changes between the data values.
  - **DataX, -Y Threshold**: Adds a definable offset to the detected limit.
  - **DataX, -Y Start**: Lower Auto Scale edge.
  - **DataX, -Y Stop**: Upper Auto Scale edge.
- **Horizontal Alignment**: Sets horizontal orientation to left, center or right.
- **Line Type**:
  - **Normal**: direct connections between entered values.
  - **Spline**: interpolates extra values to chamfer the graph.
  - **Constrained Spline**: Same as Spline mode but with a different algorithm to prevent overshooting.
- **Chart Width**: Adjust line width (0 for 2D mode).
  - **Tesselation Width**: Sets tesselation length in spline modes.
- **Thickness**: Adjusts line thickness (0 for 2D mode).
- **Depth**: Adjust line depth (0 for 2D mode).
- **Tesselation Thickness**: Tesselates line thickness.
- **Tesselation Depth**: Tesselates line depth.
- **Const. Thickness**: Builds the geometry with a constant width.
  - **Miter**: Applies a miter cut-off to avoid huge peaks for small data point angles.
- **Selection**: Specifies Start- and Stop-type. Relative: 0%: 100%. Absolute: depends on your specified data. ID: Value ID starting from 0.
- **Start[%, abs, #]**: Graph's starting point.
- **Stop[%, abs, #]**: Graph's stop point.
- **Reposition Start**: Translates the whole chart always to the same starting point.
- **Bevel**: Activates Bevel mode.
  - **Bevel Start**: Chamfer start.
  - **Bevel Stop**: Chamfer stop.
  - **Bevel Size[%]**: Adjusts bevel's size from 0 to 100.
  - **Bevel Steps**: Sets roundness via the number of segmentation steps.

Note: A chamfer is a beveled edge connecting two surfaces.

- **RelativeMapping**: Stretches texture to fit graph length.
- **Pos. Container**: Translates every child container to a bar's top.
  - **Container Offset**: Adds a certain offset to each container.
  - **Value Dependent Pos.**: Translates the container above or below the line in dependency on the data values.
  - **Container Offset Z**: Adds a Z axis offset to each container.

    - **Center Container**: Centers each translated container.
- **Pointer**: You can use a container as a pointer and move it along the chart's shape.
    - **Container**: This container represents your pointer.
    - **Selection**: Specifies the position-type. Relative: 0%: 100%. Absolute: depends on your specified data. ID: Value ID starting from 0.
    - **Position[%, abs, #]**: The pointer is translated by this parameter.
    - **Offset**: Adds an offset to the container's position.
    - **Normal Offset**: Rotates the offset so that it is perpendicular to the chart's surface.
    - **Rotate**: Rotates the container in dependency on the chart's shape.
    - **Offset Z**: Adds a Z axis offset to the pointer.
    - **Center Container**: Use the container's center for translation and rotation.
- **Low Pointer**: Activates the low pointer.
    - **Container**: Choose a container for the pointer.
- **High Pointer**: Activates the high pointer.
    - **Container**: Choose a container for the pointer.
- **Start Pointer**: Activates the Start Pointer.
    - **Container**: Choose a container for the pointer.
- **Stop Pointer**: Activates the stop pointer.
    - **Container**: Choose a container for the pointer.

To create a line chart



1. Create a new group and add the Line Chart plug-in to it.
2. Open the Transformation Editor and set Position X and Y to -100.0 and Rotation Y to -25.0.
3. Add a material and/or a texture to it.
4. Open the Line Chart editor and set the following parameters:
    - Set the following DataY values: 10,20,30,20,50,60,80,45,20.
    - Set Line Type to Spline.
    - Set Length to 250.0.
    - Set Depth to 10.0.
    - Enable Bevel, Bevel Start, Bevel Stop.
    - Set Bevel Size[%] to 100.0.
    - Set Bevel Steps to 10.0.

See Also

- Control Chart
- Data Fit

- Data Import
- Data Label
- Data Storage

11.2.5 Pie Chart

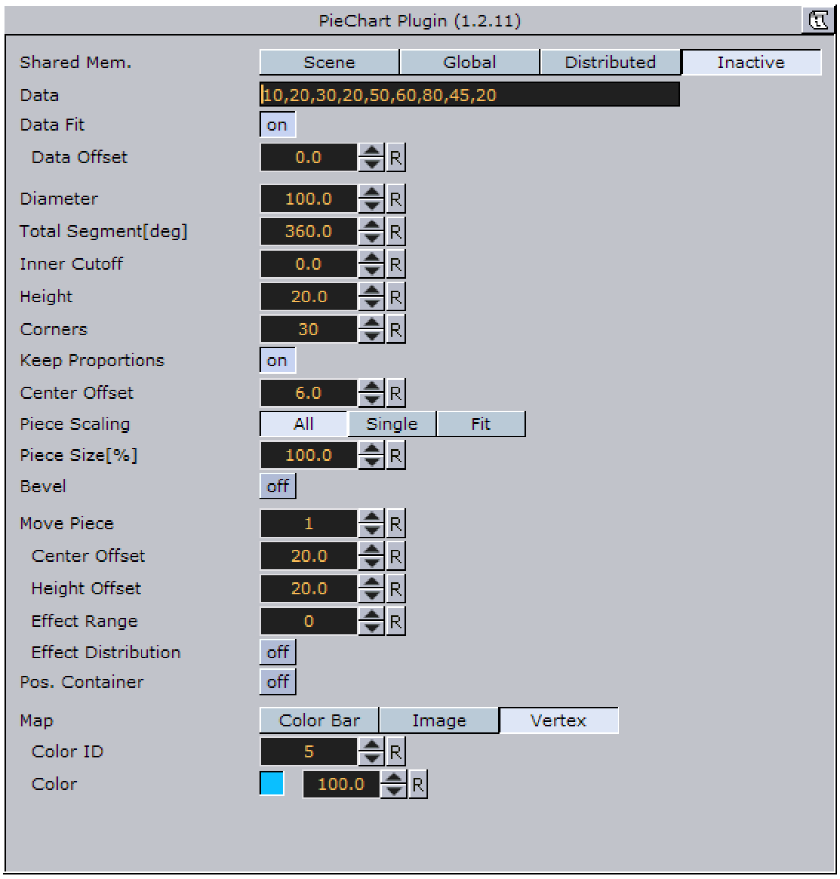
This plugin draws a pie chart, filled with data out of a Shared Memory Map. You can use delimited strings or arrays for data transfer via Scene-, Global- or Distributed-Shared Memory.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\PieChart.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a pie chart

Properties and Parameters



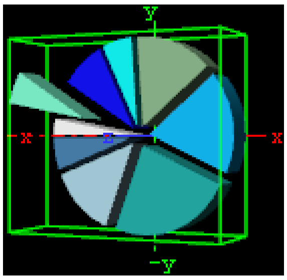
PieChart Plugin (1.2.11)

Shared Mem.	Scene / Global / Distributed / Inactive
Data	10,20,30,20,50,60,80,45,20
Data Fit	on
Data Offset	0.0
Diameter	100.0
Total Segment[deg]	360.0
Inner Cutoff	0.0
Height	20.0
Corners	30
Keep Proportions	on
Center Offset	6.0
Piece Scaling	All / Single / Fit
Piece Size[%]	100.0
Bevel	off
Move Piece	1
Center Offset	20.0
Height Offset	20.0
Effect Range	0
Effect Distribution	off
Pos. Container	off
Map	Color Bar / Image / Vertex
Color ID	5
Color	🟦 100.0

- **Shared Mem.**: Changes between Scene-, Global- and Distributed-Shared Memory. Use Inactive memory to not forward any values via Shared memory.
- **Data**: Shared Mem. key name. Data is the default input parameter.
- **Transfer Mode**: Sets string- or array-based data transfer.
- **Data Delim.**: Defines the value separator sign(s).
- **Data Fit**: Enables data normalization.
  - **Data Offset**: Adds an offset to the incoming data.
- **Diameter**: Sets pie's size.
- **Total Segment[deg]**: Set's the pie's maximum angle in degrees (default: 100% are equal to 360 degrees).
- **Inner Cutoff**: Cuts off a round piece from the cake's center.
- **Height**: Adjust height.
- **Corners**: Defines the minimum amount of corners at the pie's edge. In dependency on the number of pieces an algorithm adds corners to give the cake always the same appearance.
- **Keep Proportions**: If enabled, the cake would look like a real one. Otherwise each piece's radius is recalculated.
- **Const. Offset**: Sets a constant gap width.
- **Center Offset**: Every pieces' distance from the pie's center.
- **Piece Scaling**: All: All pieces equal 100%. Single: Every piece's size is described by it's own 100%(e.g. 3 pieces mean 300%). Fit: All pieces equal 100%, but they are scaled relatively to fit their new area with the correct values.
- **Piece Size[%]**: Parameter for the previous option.
- **Bevel**: Activates Bevel mode.
  - **Bevel Top**: Chamfer cake's top.
  - **Bevel Bottom**: Chamfer pie's bottom.
  - **Bevel Size[%]**: Adjusts bevel's size from 0 to 100.
  - **Bevel Steps**: Sets roundness via the number of segmentation steps.
- **Move Piece**: Defines the piece for the next parameter operation. Starts with 0, -1 means nothing selected.
  - **Center Offset**: Additional center shift for a certain piece.
  - **Height Offset**: Additional height shift for a certain piece.
  - **Effect Range**: Sets the range of pieces to move.
  - **Effect Distribution**: Sets the distribution of the effect range.
- **Pos. Container**: Translates every child container to a piece of cake.
  - **Container Offset**: Adds a certain center offset to each container.
  - **Center Container**: Centers each translated container.
  - **PosX, Y, Z**: Activates container translation on the particular axis.
  - **Container Pos. Z[%]**: Sets the relative position on the z axis for each child container.
- **Map**:
  - **Color Bar**: Sets a single color (V texture coordinate) for a certain piece(e.g. 3 pieces: V = 0.0, V = 0.5 and V = 1.0).
  - **Vertex**: Use vertex colors for each piece starting with ID 0. **Color ID** moves between the available vertex colors. **Color** lets you choose color for the current ID.

Note: A chamfer is a beveled edge connecting two surfaces.

To create a pie chart



1. Create a new group and add the Pie Chart plug-in to it.
2. Open the Transformation Editor and set Position X and Y to -100.0 and Rotation Y to -25.0.
3. Add a material and/or a texture to it.
4. Open the Pie Chart editor and set the following parameters:
   - Set the following DataY values: 10,20,30,20,50,60,80,45,20.
   - Enable Keep Proportions.
   - Set Center Offset to 6.0.
   - Set Move Piece to 1.
   - Set Center and Height Offset to 20.0.
   - Enable Vertex and set different colors for all Color IDs.

See Also

- Control Pie
- Data Fit
- Data Import
- Data Label
- Data Storage
- Presenter

11.2.6 Scatter Chart

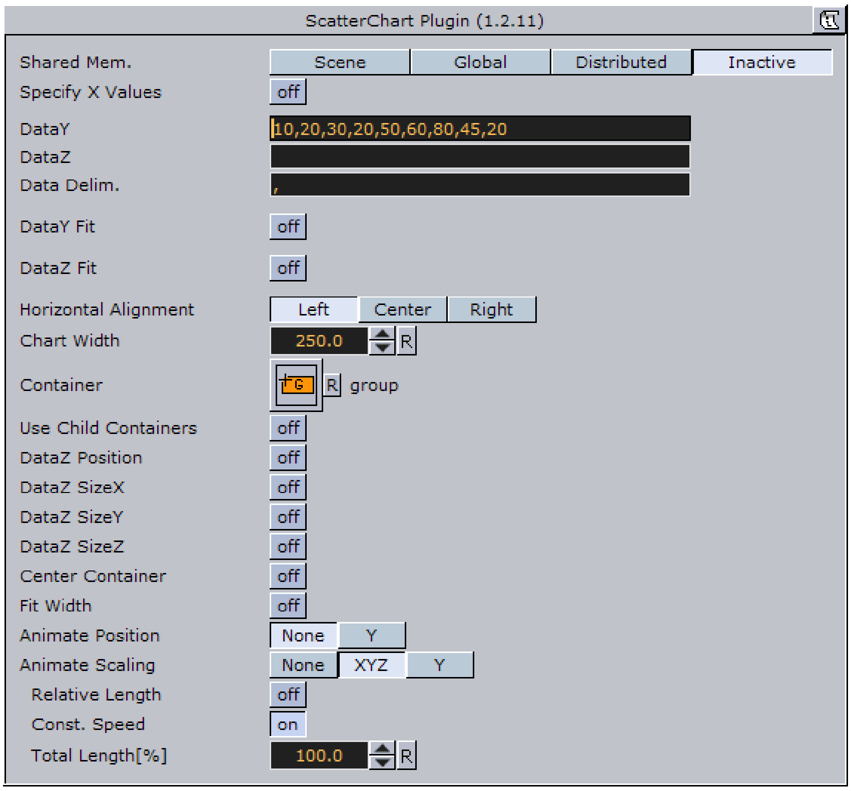
This plug-in draws a scatter chart, filled with data out of a Shared Memory Map. You can use delimited strings or arrays for data transfer via Scene-, Global- or Distributed-Shared Memory.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ScatterChart.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a scatter chart

Properties and Parameters



ScatterChart Plugin (1.2.11)

Shared Mem. Scene Global Distributed Inactive

Specify X Values off

DataY 10,20,30,20,50,60,80,45,20

DataZ

Data Delim. ,

DataY Fit off

DataZ Fit off

Horizontal Alignment Left Center Right

Chart Width 250.0 R

Container G R group

Use Child Containers off

DataZ Position off

DataZ SizeX off

DataZ SizeY off

DataZ SizeZ off

Center Container off

Fit Width off

Animate Position None Y

Animate Scaling None XYZ Y

Relative Length off

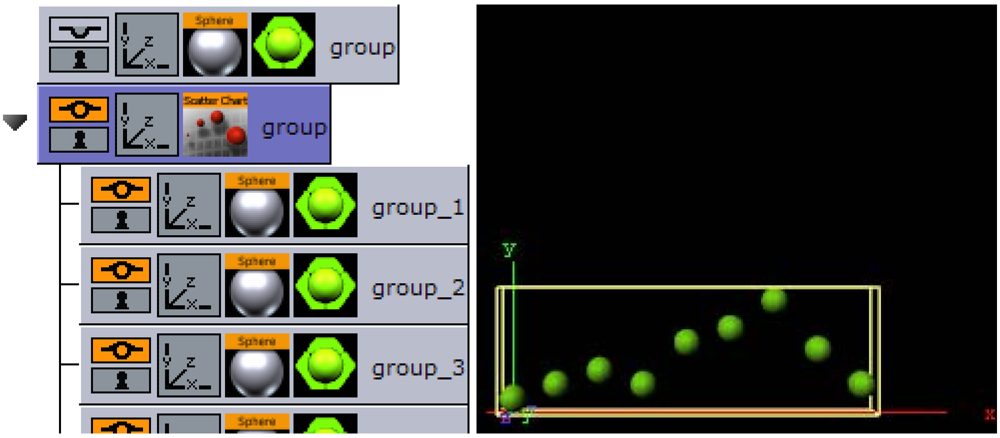
Const. Speed on

Total Length[%] 100.0 R

- **Shared Mem.**: Changes between Scene-, Global- and Distributed-Shared Memory. Use Inactive memory to not forward any values via Shared memory.
- **Key DataX**: Shared Mem. key name for X values. DataX is the default input parameter for X values.
- **Key DataY**: Shared Mem. key name for Y values. DataY is the default input parameter for Y values.
- **Key DataZ**: Shared Mem. key name for Z values. DataZ is the default input parameter for Z values.
- **Transfer Mode**: Sets string- or array-based data transfer.
- **Data Delim.**: Defines the value separator sign(s).
  - **DataX, -Y, -Z Fit**: Enables data normalization.
  - **DataX, -Y, -Z Scale**: Scales input by the selected factor.
  - **DataX, -Y, -Z Offset**: Adds an offset to the incoming data.
  - **DataX, -Y, -Z Auto Scale**: Enables automatic data normalization.
  - **DataX, -Y, -Z Detect Limits**: Detects minimum and maximum of all values and scales them to adjusted Start and Stop.
  - **DataX, -Y, -Z Threshold**: Adds a definable offset to the detected limit.
  - **DataX, -Y, -Z Start**: Lower Auto Scale edge.
  - **DataX, -Y, -Z Stop**: Upper Auto Scale edge.
- **Horizontal Alignment**: Sets horizontal orientation to left, center or right
- **Chart Width**: Adjusts the chart width.
- **Container**: Drag drop the container which will be used for the scatter nodes.
- **DataZ Position**: Use DataZ for Z-axis positioning.
- **DataZ SizeX**: Z Values influence X scaling.
- **DataZ SizeY**: Z Values influence Y scaling.

- **DataZ SizeZ**: Z Values influence Z scaling.
- **Center Container**: Centers each translated container.
- **Fit Width**: Considers the width of the scatter node containers which means that the containers do not reach outside of the specified range.
- **Animate Position**: Sets the translation type: None: no animation, Y: animates the container along the Y axis (suggestion: Scale Y + DataLabel for BarChart alike labeling).
- **Animate Scaling**: Sets the scaling type: None: no scaling at all, XYZ: linear scaling in all directions, Y: scales in y direction only (->BarChart)
  - **Relative Length**: If activated, each scatter node will have its own 100% size (e.g. 7 nodes equal 700%).
  - **Const. Speed**: Sets the same animation duration for each bar.
  - **Total Length[%]**: Sets the accumulated size of all scatter nodes in percent.

To create a scatter chart



1. Create a group and add the Sphere plug-in to it.
2. Open the transformation editor and set the Scaling (locked) to 0.2.
3. Add a material and/or a texture to the sphere.
4. Set the group with the sphere to hidden.
5. Create a new group and add the Scatter Chart plug-in to it.
6. Open the Transformation Editor and set Position X and Y to -100.0.
7. Open the Scatter Chart editor and set the following parameters:
   - Set the following DataY values: 10,20,30,20,50,60,80,45,20.
   - Set Width to 250.0.
   - Drag and drop the container with the Sphere to the Container placeholder.

See Also

- Control Chart
- Data Fit
- Data Import
- Data Label
- Data Storage

11.2.7 Stock Chart

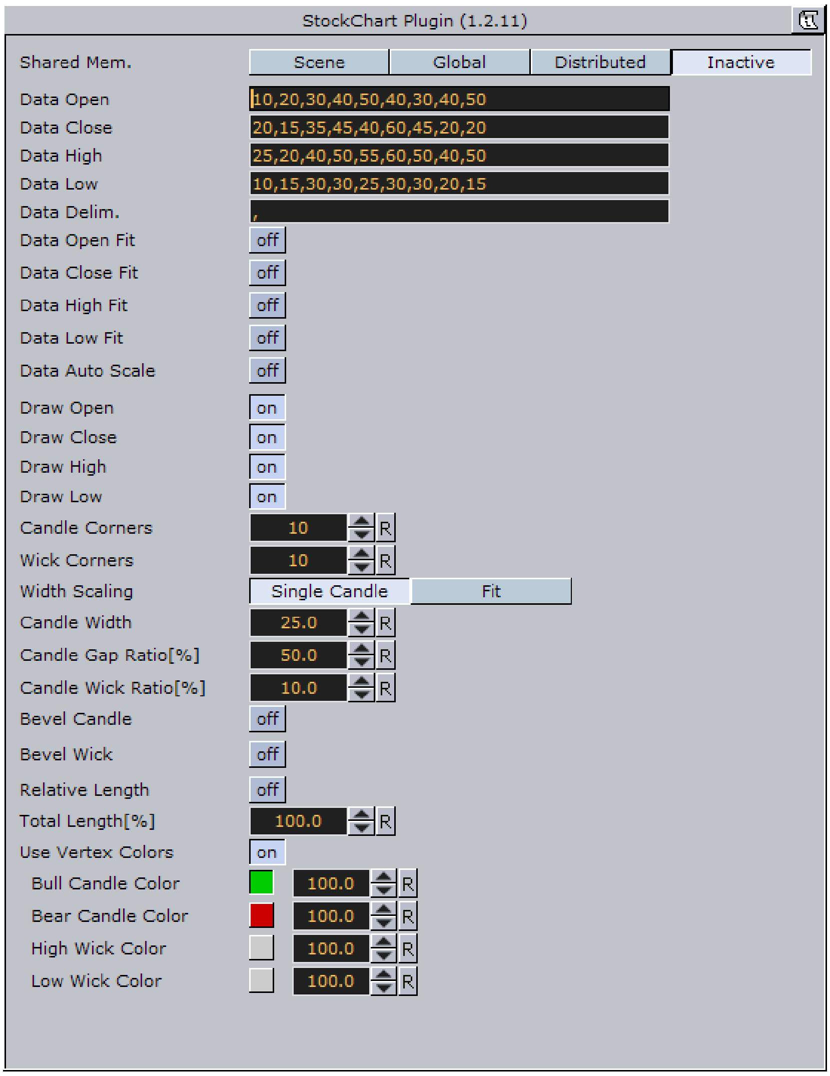
This plug-in draws a stock chart, filled with data out of a Shared Memory Map. You can use delimited strings or arrays for data transfer via Scene-, Global- or Distributed-Shared Memory.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\StockChart.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a stock chart

Properties and Parameters



StockChart Plugin (1.2.11)

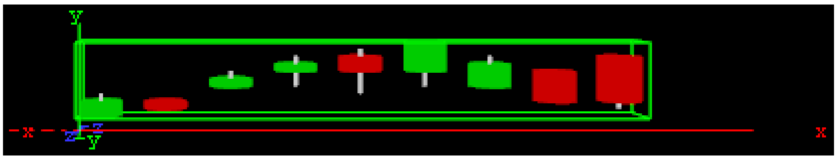
Shared Mem.	Scene / Global / Distributed / [✓ Inactive]
Data Open	10,20,30,40,50,40,30,40,50
Data Close	20,15,35,45,40,60,45,20,20
Data High	25,20,40,50,55,60,50,40,50
Data Low	10,15,30,30,25,30,30,20,15
Data Delim.	,
Data Open Fit	off
Data Close Fit	off
Data High Fit	off
Data Low Fit	off
Data Auto Scale	off
Draw Open	on
Draw Close	on
Draw High	on
Draw Low	on
Candle Corners	10
Wick Corners	10
Width Scaling	[✓ Single Candle] / Fit
Candle Width	25.0
Candle Gap Ratio[%]	50.0
Candle Wick Ratio[%]	10.0
Bevel Candle	off
Bevel Wick	off
Relative Length	off
Total Length[%]	100.0
Use Vertex Colors	on
Bull Candle Color	🟩 100.0
Bear Candle Color	🟥 100.0
High Wick Color	⬜ 100.0
Low Wick Color	⬜ 100.0

- **Shared Mem.**: Changes between Scene-, Global- and Distributed-Shared Memory. Use Inactive memory to not forward any values via Shared memory.
- Key Open: Shared Mem. key name for open values. DataOpen is the default input parameter for open values.
- **Key Close**: Shared Mem. key name for close values. DataClose is the default input parameter for close values.
- **Key High**: Shared Mem. key name for high values. DataHigh is the default input parameter for high values.
- **Key Low**: Shared Mem. key name for low values. DataLow is the default input parameter for low values.
- **Transfer Mode**: Sets string- or array-based data transfer.
- **Data Delim.**: Defines the value separator sign(s).
- **Data Open, -Close, -High, -Low Fit**: Enables data normalization.
  - **Data Open, -Close, -High, -Low Scale**: Scales input by the selected factor.
  - **Data Open, -Close, -High, -Low Offset**: Adds an offset to the incoming data.
  - **Data Auto Scale**: Enables automatic data normalization.

- **Data Detect Limits**: Detects minimum and maximum of all values and scales them to adjusted Start and Stop.
- **Data Start**: Lower Auto Scale edge.
- **Data Stop**: Upper Auto Scale edge.
- **Draw Open**: Draws open data.
- **Draw Close**: Draws close data.
- **Draw High**: Draws high data.
- **Draw Low**: Draws low data.
- **Candle Corners**: Sets candle segments.
- **Wick Corners**: Sets wick corners.
- **Width Scaling**: Single Candle: Adjust size for a single candle. Fit: Adjust all candles to fit a certain width.
- **Candle Width**: Value for the previous parameter.
- **Candle Gap Ratio[%]**: 100% means that the gaps have the same size as the candles.
- **Candle Wick Ratio[%]**: 100% means that the wicks have the same size as the candles.
- **Bevel Candle**: Activates Bevel mode for the candle geometry.
  - **Bevel Candle Top**: Chamfer candle's top.
  - **Bevel Candle Bottom**: Chamfer candle's bottom.
  - **Bevel Candle Size[%]**: Adjusts bevel's size from 0 to 100.
  - **Bevel Candle Steps**: Sets roundness via the number of segmentation steps.
- **Bevel Wick**: Activates Bevel mode for the wick geometry.
  - **Bevel Wick Top**: Chamfer wick's top.
  - **Bevel Wick Bottom**: Chamfer wick's bottom.
  - **Bevel Wick Size[%]**: Adjusts bevel's size from 0 to 100.
  - **Bevel Wick Steps**: Sets roundness via the number of segmentation steps.
- **Relative Length**: on: all accumulated candle lengths equal 100%. off: every candle has its own 100% (e.g. 3 candles: 300%).
- **Total Length**: Value for the previous parameter.
- **Use Vertex Colors**: Activates the following parameters.
  - **Bull Candle Color**: Defines the color for all candles where the close value is higher than the open value.
  - **Bear Candle Color**: Defines the color for all candles where the close value is lower than the open value.
  - **High Wick Color**: Sets high wick's color.
  - **Low Wick Color**: Sets low wick's color.

Note: A chamfer is a beveled edge connecting two surfaces.

To create a stock chart



1. Create a new group and add the Stock Chart plug-in to it.

2. Open the Transformation Editor and set Position X and Y to -100.0.
3. Open the Stock Chart editor and set the following parameters:
    - Set the Data Open parameter to the following values: 10,20,30,40,50,40,30,40,50.
    - Set the Data Close parameter to the following values: 20,15,35,45,40,60,45,20,20.
    - Set the Data High parameter to the following values: 25,20,40,50,55,60,50,40,50.
    - Set the Data High parameter to the following values: 10,15,30,30,25,30,30,20,15.
    - Set Candle and Wick Corners to 10.
    - Set Candle Width to 25.0.

See Also

- Control Chart
- Data Fit
- Data Import
- Data Label
- Data Storage

12 Container Plugins

Container plugs are housed in the following folders:

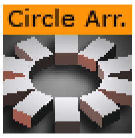
- Arrange
- Container
- Container FX
- Control
- Default
- Global
- MPS
- Presenter
- Script Plugins
- Spline FX
- Text
- Text FX
- Texture
- Time
- Tools
- Transformation
- Vizual Data Tools

12.1 Arrange

The following container plugins are housed in the Arrange folder:

- Circle Arrange
- Grid Arrange
- Time Displacement

12.1.1 Circle Arrange



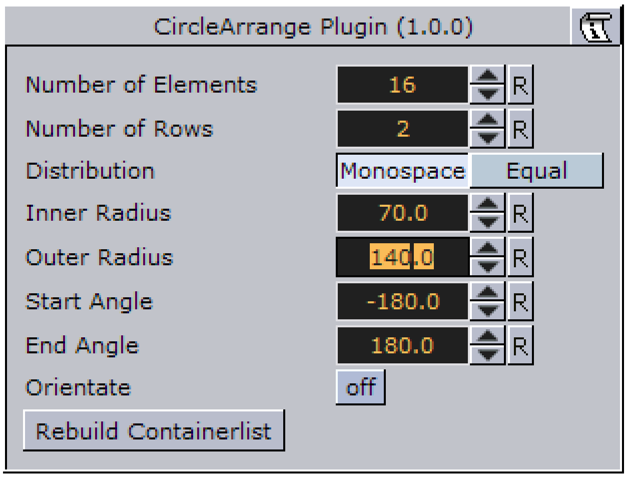
Circle Arrange is a tool for arranging a set of containers into a circular structure.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\CircleArrange.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To arrange containers in a circle

Properties and Parameters



CircleArrange Plugin (1.0.0)

Number of Elements 16

Number of Rows 2

Distribution Monospace Equal

Inner Radius 70.0

Outer Radius 140.0

Start Angle -180.0

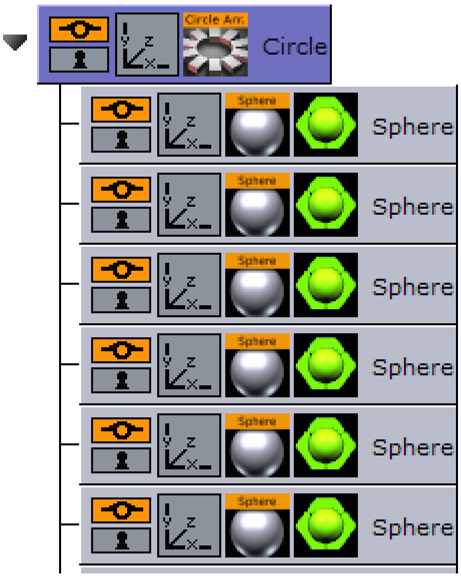
End Angle 180.0

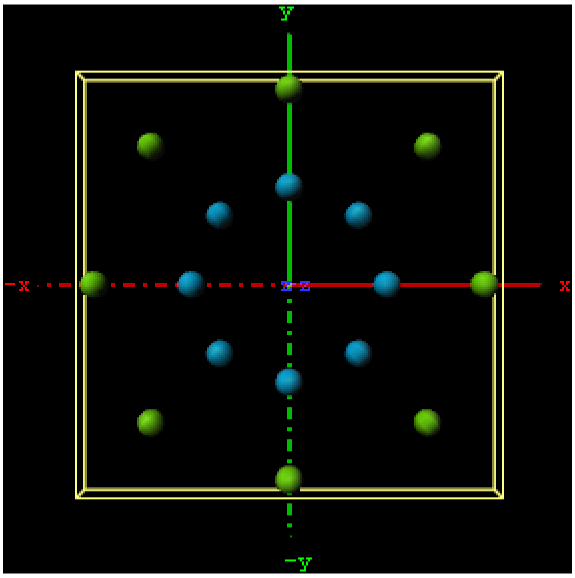
Orientate off

Rebuild Containerlist

- **Number of Elements**: Allows you to define how many elements the structure is to be constructed by.
- **Number of Rows**: Sets the number of rows/rings for the circular structure
- **Distribution**: Allows you to define how the containers are to distributed in the circular shape.
  - **Monospace**: Sets the same number of elements in each circle row. The spacing between objects in the outer rows will then be bigger than in the inner rows.
  - **Equal**: Distributes the objects between the rows so the distance between the objects in the different rows are equal. The number of containers will then be larger in the outer rows.
- **Inner Radius**: Sets the radius from the center to the first container ring.
- **Outer Radius**: Sets the radius from the center to the outer ring.
- **Start Angle**: Setting this above −180° will create an open section in the circle structure. The opening is created in a clockwise direction.
- **End Angle**: Setting this below 180 will create an open section in the circle structure. The opening is created in a counterclockwise direction.
- **Orientate**: If this option is enabled, the containers will orientate themselves with their bottom towards the center of the circular structure.
- **Rebuild Containerlist**: Use this button to rebuild the structure after having added or removed containers form the group.

To arrange containers in a circle





1. Create a group container, and name it **Circle.**
2. Add a **Sphere** as a sub container to the Circle container.
3. Add a material and/or a texture to the Sphere container.
4. Open the Transformation Editor for the Sphere container, and set **Scaling** (locked) to **0.2**.
5. *Optional*: Animate the Sphere object.
6. Make 15 copies of the sub container, totalling the number of Sphere containers to 16. All copies should be placed as sub containers of the Circle container and have the same position as the original.
7. Add the Circle Arrange container plug-in to the Circle container.
8. Open the Circle Arrange editor and do the following:
    - Set **Number of Elements** to 16.
    - Set **Number of Rows** to 2.
9. Click the **Rebuild Containerlist** button.
    - The plug-in will position all sub containers in a circular structure.

The order of the containers in the group decides their placing in the circular structure. If more than one row is selected, the first containers end up in the outer circle and the last containers in the inner circle.

See Also

- Grid Arrange

12.1.2 Grid Arrange

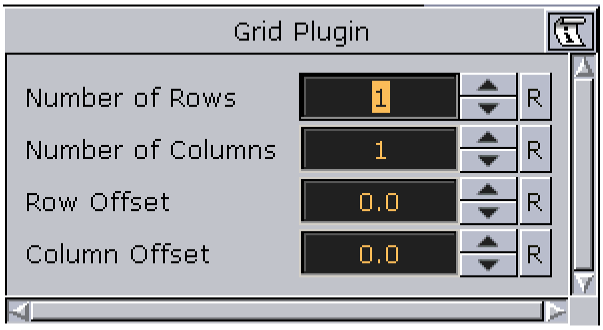
The Grid function is an effective tool for arranging a set of containers into a grid or tabular structure.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\GridArrange.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To arrange containers in a grid

Properties and Parameters



Grid Plugin

Number of Rows 1

Number of Columns 1

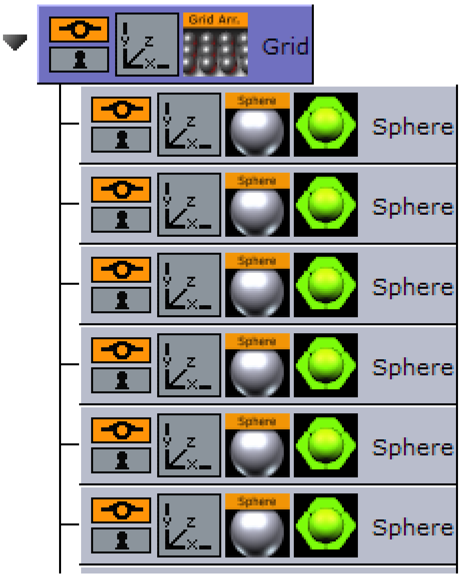
Row Offset 0.0

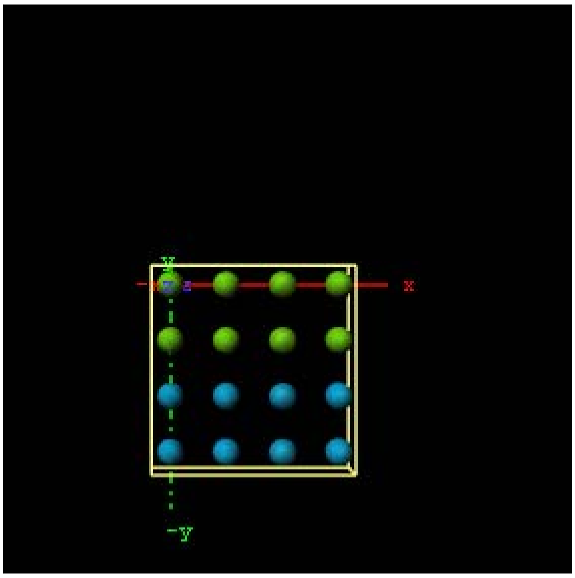
Column Offset 0.0

- **Number of Rows**: Sets the number of rows the containers are to be distributed over.
- **Number of Columns**: Sets the number of columns the containers are to be distributed over.
- **Row Offset**: Sets the distance between the rows.
- **Column Offset**: Sets the distance between the columns.

Note: The product of number of rows and number of columns should normally be equal number of containers. If it is higher, all rows will not be equal, if it is lower, some containers will not be shown, but they will still be rendered.

To arrange containers in a grid





1. Create a group container, and name it **Grid**.
2. Add a **Sphere** as a sub container to the Grid container.
3. Add a material and/or a texture to the Sphere container.
4. Open the Transformation Editor for the Sphere container, and set **Scaling** (locked) to **0.2**.
5. *Optional*: Animate the Sphere object.
6. Make 15 copies of the sub container, totalling the number of Sphere containers to 16. All copies should be placed as sub containers of the Grid container and have the same position as the original.
7. Add the Grid Arrange container plug-in to the Grid container.
8. Open the Grid Arrange editor and do the following:
    - Set **Number of Rows** to **4**.
    - Set **Number of Columns** to **4**.
    - Set **Row Offset** to **40.0**.
    - Set **Column Offset** to **40.0**.
    - The plug-in will position all sub containers in a grid structure.

The order of the containers in the group decides their placing in the grid structure. If more than one row is selected, the first containers end up in the first row(s) and the last containers in the last row(s).

See Also

- Circle Arrange

12.1.3 Time Displacement

Time Displacement allows you to use a grayscaled alpha image for setting animation time offsets for all the objects in a group container. This is typically useful if you want to create some kind of randomized animation effect or that should follow a certain pattern.

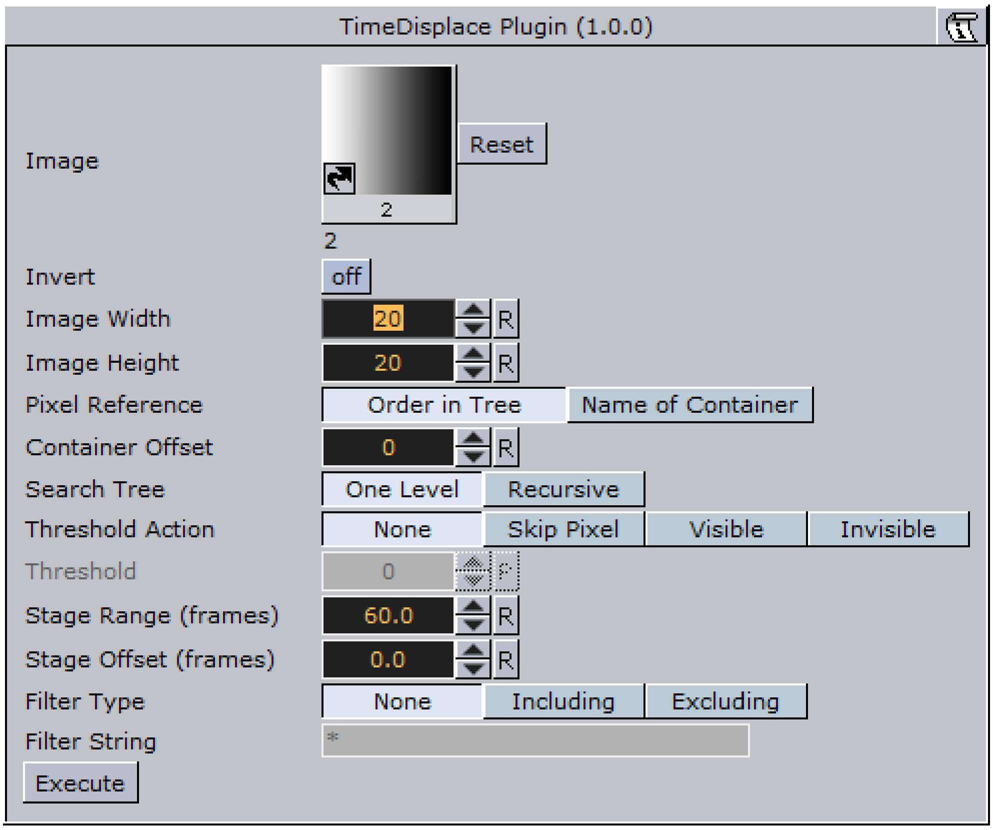
What the function does is to create links between the animated containers and the pixels of the grayscale image and then use the alpha level of each pixel to set the offset of the containers. The range from 100% alpha to 0% alpha on the pixels sets the offset level. By default 100% alpha sets no offset and 0% sets maximum offset.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\TimeDisplacement.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To set animation time offsets using a grayscaled alpha image

Properties and Parameters



TimeDisplace Plugin (1.0.0)

Image: 2 (Reset)

Invert: off

Image Width: 20

Image Height: 20

Pixel Reference: Order in Tree | Name of Container

Container Offset: 0

Search Tree: One Level | Recursive

Threshold Action: None | Skip Pixel | Visible | Invisible

Threshold: 0

Stage Range (frames): 60.0

Stage Offset (frames): 0.0

Filter Type: None | Including | Excluding

Filter String: *

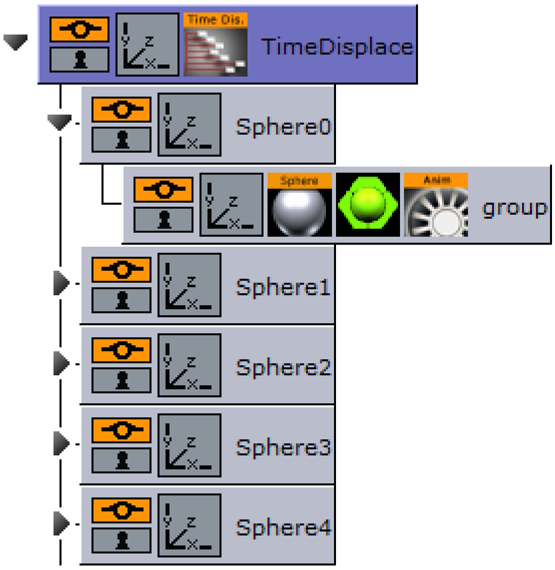
Execute

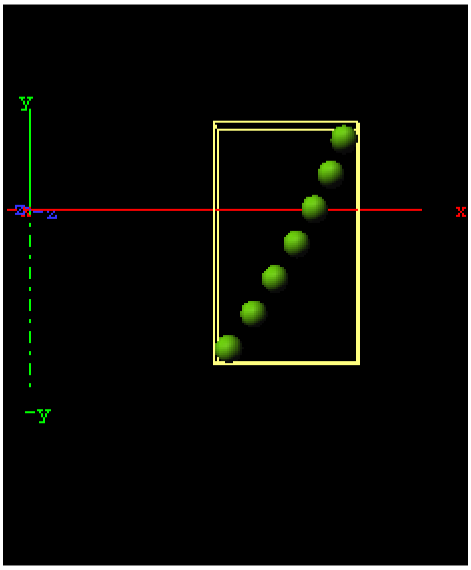
- **Image**: Drag the alpha image you want to use for creating the containers offsets onto the drop zone here.
- **Invert**: Inverts the link between the image pixels and offsets. 0% gives no offset and 100% gives maximum.
- **Image Width**: Sets the height of the alpha image.
- **Image Height**: Sets the height of the alpha image. Image Width*Image Height should normally be the same as the number of containers that are to be used in the function.
- **Pixel Reference**: Defines the way the pixels should be associated to the animated containers. You can choose between:
  - **Order in Tree**: The container order in the transformation tree decides how the pixels should be associated.
  - **Name of Container**: The container name sets the order in which the containers will be associated with the pixels.
- **Search Tree**: Sets the search level of the function, when it decides which sub containers to include in the function:
  - **One Level**: Only the first level of sub containers is being included in the function.
  - **Recursive**: All levels of sub containers are being included.
- **Threshold Action**: Is executed if the pixel value is lower than the threshold value.
  - **None**: No action is performed.
  - **Skip Pixel**: The pixel is skipped in the offset process.
  - **Visible** – Makes objects visible when the pixel value is lower than the threshold value.

- **Invisible** - Makes objects invisible when the pixel value is lower than the threshold value.

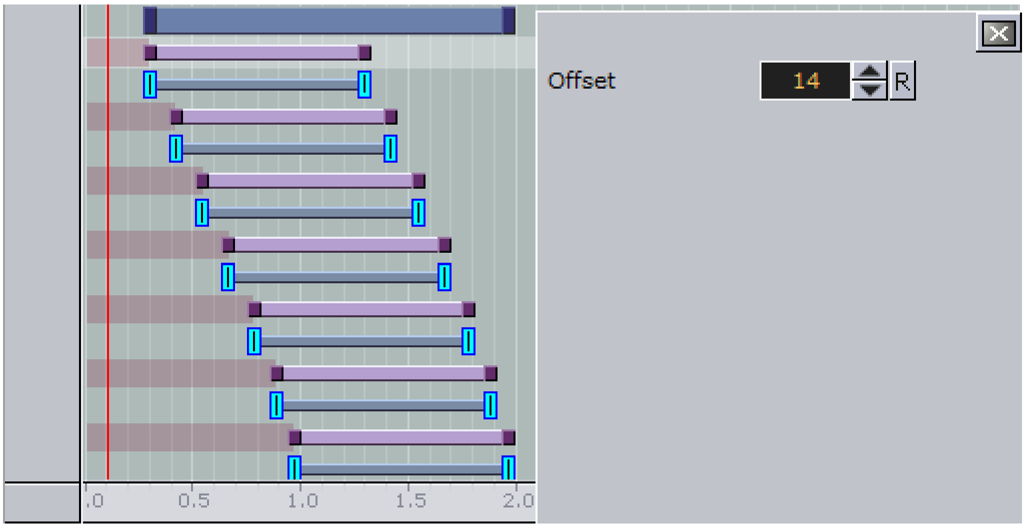
- **Threshold**: Sets the Threshold value where one of the above selected actions will be performed.
- **Stage Range (frames)**: Sets the range of the offset distribution. A high value will result in bigger differences in offset between containers that are associated with different pixel alpha values.
- **Stage Offset (frames)**: Adds a static offset value to all containers.
- **Filter Type** - Filters the containers in or out. Available options are; **None**, **Including** and **Excluding**.
- **Filter String** - Sets the name of the containers that are to be included in the filter. Use of asterisks is supported.
- **Execute**: Starts the operation

To set animation time offsets using a grayscaled alpha image





1. Add a group to the scene tree and name it TimeDisplace, and the TimeDisplace plug-in to it.
2. Create a sub container of the TimeDisplace container and name it Sphere.
3. Create a new group container as a sub container of Sphere.
4. Add a Sphere geometry plug-in and material to the group container.
5. Animate the group container from Position X -200.0 to 200.0.
6. Make 6 copies of the Sphere container under the TimeDisplace container.
7. Add the Grid Arrange plug-in to the TimeDisplace container.
8. Open the Grid Arrange editor and set Number of Rows to 6 and Row Offset to 30.0.
9. *Optional*: Remove the Grid Arrange plug-in.
10. Add the Corena plug-in and rename the containers to Sphere, starting at index 0. This will allow you to filter out specific containers using the TimeDisplace filter.
11. *Optional*: Remove the Corena plug-in.
12. Open the TimeDisplace editor and add an alpha image to the Image drop zone.
13. Click Execute.



Offset 14

14. Click the Stage to see the offsets being added to the containers.
15. Run the animation to see the time offset in action!

See Also

- Image Editor
- Grid Arrange
- Corena

12.2 Container

The following container plugins are housed in the Container folder:

- BoundingBox
- Cobra
- Coco
- Colin
- Cora
- Corena
- Toggle

12.2.1 BoundingBox

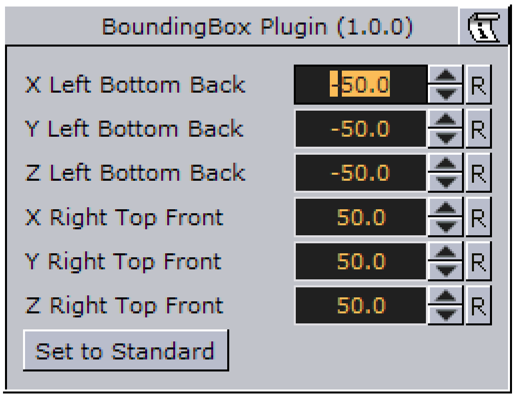
Use this plug-in to redefine the standard Bounding Box which is bounding all geometry objects.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\BoundingBox.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



BoundingBox Plugin (1.0.0)

X Left Bottom Back: -50.0

Y Left Bottom Back: -50.0

Z Left Bottom Back: -50.0

X Right Top Front: 50.0

Y Right Top Front: 50.0

Z Right Top Front: 50.0

- **X Left Bottom Back**: It changes the dimension in X-direction starting from left.
- **Left Bottom Back**: It changes the dimension in Y-direction starting from bottom.
- **Left Bottom Back**: It changes the dimension in Z-direction starting from flipside.
- **Right Top Front**: It changes the dimension in X-direction starting from right.
- **Right Top Front**: It changes the dimension in Y-direction starting from top.
- **Right Top Front**: It changes the dimension in Z-direction starting from front.
- **Set to Standard**: Initialize minimum values to hold the entire geometry object.

12.2.2 Cobra

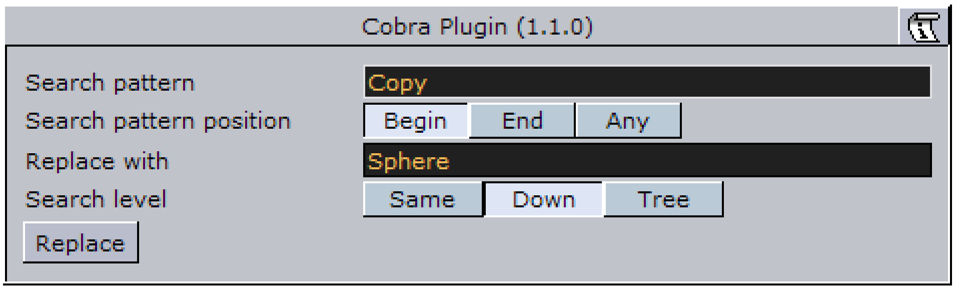
The Cobra function is an easy to use function replace (parts) of container names. Enter a search pattern and a string the search should be replaced with and execute.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Cobra.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To rename container(s)

Properties and Parameters



Cobra Plugin (1.1.0)

Search pattern: Copy

Search pattern position: Begin | End | Any

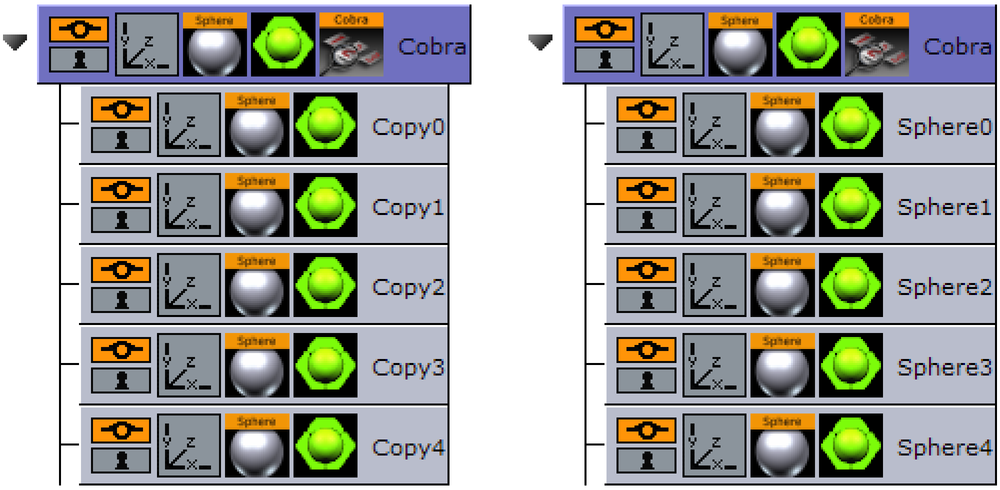
Replace with: Sphere

Search level: Same | Down | Tree

Replace

- **Search pattern**: Sets the string you want to replace.
- **Search pattern position**: Defines the position within the string. It can be set to Begin, End and Any.
- **Replace with**: Sets the string you want to use instead of the search pattern.
- **Search level**
  - **Same**: Only container on the same hierarchy level will be affected.
  - **Down**: Only sub-container of the current container will be affected.
  - **Tree**: Searches the whole scene tree.
- **Replace**: Starts the replace process.

To rename container(s)



1. Add the Cobra plug-in to a container that for example has sub containers that should be renamed.
2. Open the Cobra editor.
3. Enter the Search pattern string (e.g. Copy), and set the Search patter position to Begin.
4. Enter the Replace with string (e.g. Sphere), and set the Search level to Down.
5. Click the **Replace** button.

See Also

- Coco
- Corena

12.2.3 Coco

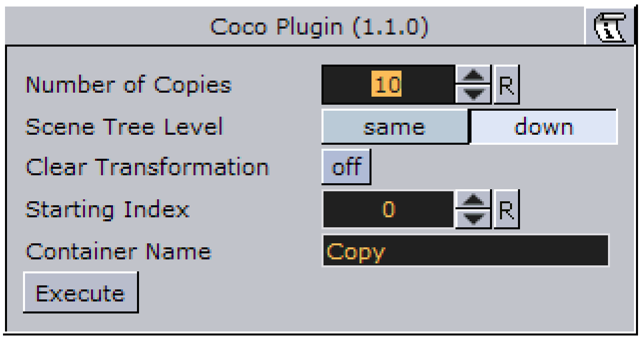
The Coco plug-in allows you to create copies of a container with great ease. This function is typically used together with the Arrange plug-ins.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Coco.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create copies of containers

Properties and Parameters



Coco Plugin (1.1.0)

Number of Copies: 10

Scene Tree Level: same | down

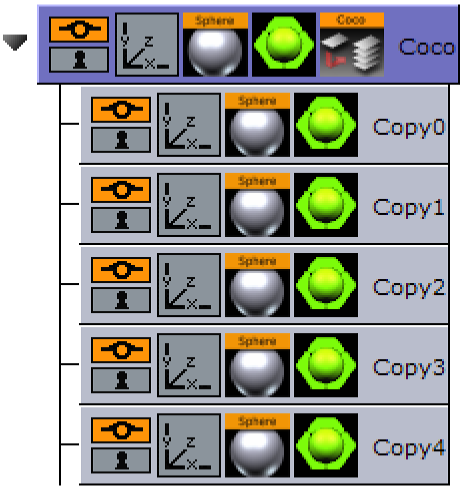
Clear Transformation: off

Starting Index: 0

Container Name: Copy

- **Number of Copies**: Sets the number of copies to be created when the function is executed.
- **Scene Tree Level**: Allows you to select the hierarchic position of the copies.
  - **same**: Sets the copies on the same level as the source container.
  - **down**: Sets the copies as sub containers to the source container.
- **Clear Transformation**: Sets all transformation values on the new copies, like scaling, position and rotation, back to zero. All copies will then be stacked at a initial position with all transformation values set to zero.
- **Starting Index**: Allows you to alter the starting point of the index numbering of the copies. By default the numbering starts on 1.
- **Container Name**: Allows you to change the name the containers will be given by default.
- **Execute**: After having set the required parameters, click this button to execute the duplicating process.

To create copies of containers



1. Add the Coco plug-in to the container you want to copy.
2. Open the Coco editor and enter the Number of Copies (e.g. 10)
3. Set Scene Tree Level to **down**.
4. Click the **Execute** button.

See Also

- Arrange
- Cobra

12.2.4 Colin

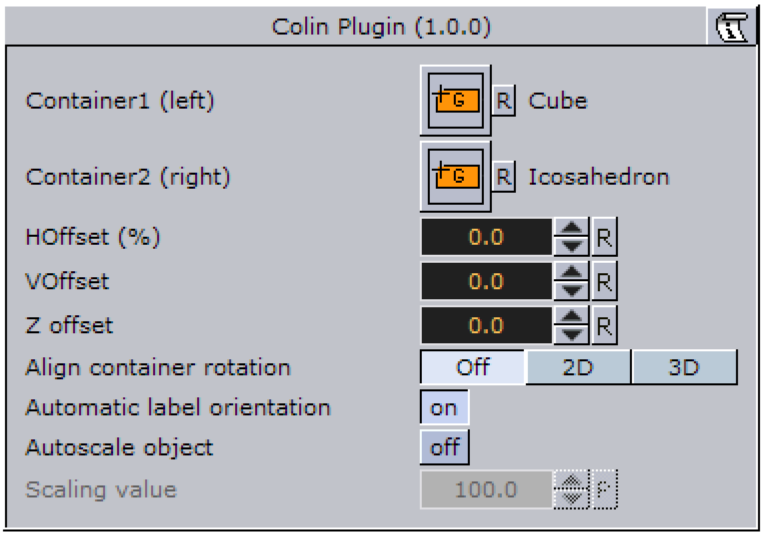
The Colin plug-in allows you to align a container in the scene using two other containers as reference. This makes it easy to maintain a good and controlled symmetry in the scene.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Colin.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To align containers

Properties and Parameters



Colin Plugin (1.0.0)

Container1 (left) — Cube

Container2 (right) — Icosahedron

HOffset (%) 0.0

VOffset 0.0

Z offset 0.0

Align container rotation: Off | 2D | 3D

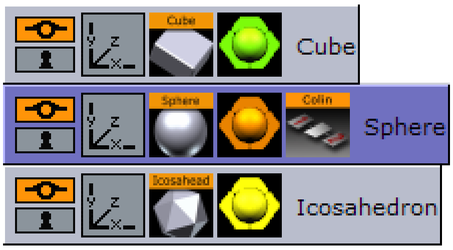
Automatic label orientation: on

Autoscale object: off

Scaling value 100.0

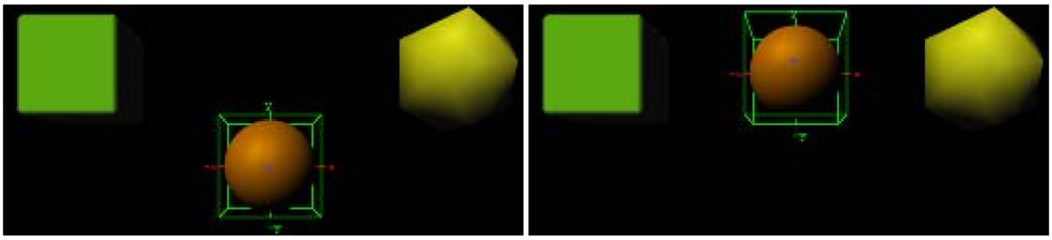
- **Container 1**: Is the drop zone for the first referential container.
- **Container 2**: Is the drop zone for the second referential container.
- **HOffset %**: Sets the horizontal offset between the referential containers in percent. Horizontal means here an invisible line between the referential containers regardless of the actual boning of the line.
- **VOffset**: Sets the vertical offset from the horizontal line.
- **ZOffset**: Sets the Z offset from the line between the referential containers.
- **Align container rotation**
  - **2D**: When enabled, this option will position and rotate the container in the X- and Y-axis to keep itself aligned on the invisible line between the referential containers as they move.
  - **3D**: When enabled, this option will position and rotate the container in the X-, Y- and Z-axis to keep itself aligned on the invisible line between the referential containers as they move.
- **Automatic label orientation**: This parameter only works if the **Align container rotation** parameter is switched to **2D** or **3D.** It adapts automatically the rotation angles.
- **Autoscale object**: Unlocks the **Scaling** value.
- **Scaling**: Lets you enter the value the container should be scaled to.

To align containers

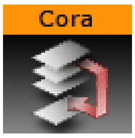


1. Add the Cube, Sphere and Icosahedron geometries to the scene tree.
2. Open the transformation editor for Cube and Icosahedron and set them to Position X -200.0 and 200.0, respectively, and Position Y 100.0.

3. Add the Colin plug-in to the Sphere container.
4. Open the Colin editor and drag and drop the Cube and Icosahedron containers onto the Container1 (left) and Container2 (right), respectively.
   - The container is then aligned between the two referential containers.



12.2.5 Cora



The Cora plug-in allows you to sort a set of sub containers based on multiple criteria. It is a useful tool when your scene starts getting complex and it is difficult to maintain the overview. To make sorts by different criteria it can help you reorganize the scene tree structure. It works on all containers under the container it is added to.

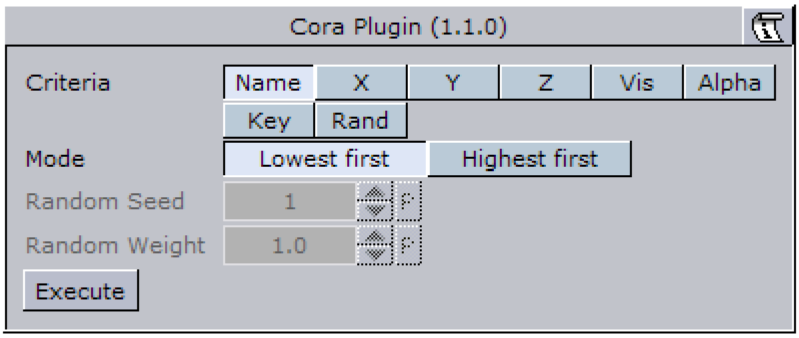
The plug-in is often used together with the Time Displacement plug-in which enables many creative possibilities.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Cora.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To organize the scene tree

Properties and Parameters



Cora Plugin (1.1.0)

Criteria: Name | X | Y | Z | Vis | Alpha | Key | Rand

Mode: Lowest first | Highest first

Random Seed: 1

Random Weight: 1.0

Execute

- **Criteria**: Selects the wanted sort criteria.
  - **Name**: Performs a case sensitive sort. In normal sort order, capitalized letters are listed first. Be aware, use letter case consistently if you want to have a correct result when you perform a sort by name.

  - **X**: Sorts by X-values.
  - **Y**: Sorts by Y-values.
  - **Z**: Sorts by Z-values.
  - **Vis**: Sorts all the visible containers first.
  - **Alpha**: Sorts by the alpha value among those containers who have an alpha function attached. Alpha values set on the material of a container are ignored.
  - **Key**: Sorts containers by key function. Among those with key signal attached, the key alpha value is used for the further sorting.
  - **Rand**: Sorts by random.
- **Mode**: Sets the order of the sorting. Choose between setting the **Highest** or the **Lowest** values.
- **Random Seed**: Is relevant if you have chosen random as your sort criteria. It specifies a seed for the random number generator. Even though Viz Artist use random numbers, the animation for a specific random seed will always look the same.
- **Random Weight**: Defines how random the randomize function will sort. Random 1.0 means random, random 0.0 does nothing and random 0.5 gives you a 50% chance if a container is resorted or stays in place.
- **Execute**: Starts the operation.

To organize the scene tree

- Add the Cora plug-in to the container that holds the sub containers you want to sort, select the sort criteria and press the Execute button.

See Also

- Time Displacement

12.2.6 Corena

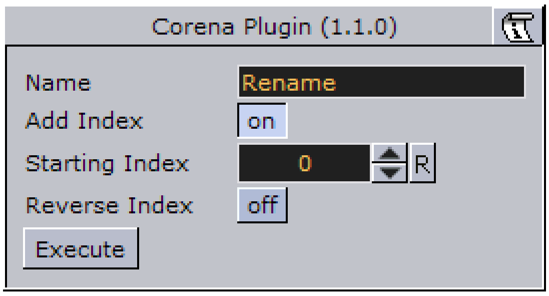
The Corena plug-in renames all sub containers of the container you apply it to. The plug-in is often used together with the Time Displacement plug-in which enables many creative possibilities.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Cora.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To rename container(s)

Properties and Parameters

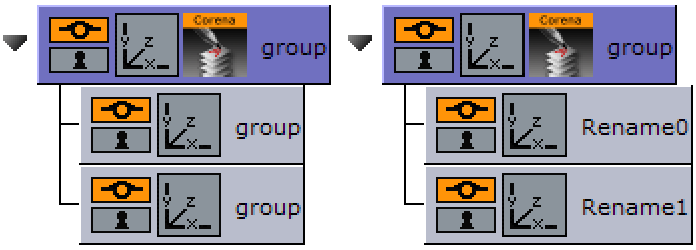


Corena Plugin (1.1.0)

Name: Rename
Add Index: on
Starting Index: 0
Reverse Index: off

- **Name** sets the name you want to rename the sub containers to. You can define the number format of the new names by entering a hash in addition to the name for each digit you want in the numbering. If you enter ###newname, the containers will be named: 001newname, 002newname, and so on.
- **Add Index** enables/disables creation of a numbering index.
- **Starting Index** sets the number you want the numbering index to start from.
- **Reverse Index** if enabled, creates the index in a reversed order.
- **Execute** start the operation.

To rename container(s)



1. Add the Corena plug-in to the container that holds the containers you want to rename.
2. Enter the new name, set the indexing parameters and click **Execute**.

See Also

- Cobra

12.2.7 Toggle

Toggle is used to toggle between two objects. The plug-in is most commonly used with Transition Logic scenes.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Toggle.vip`.

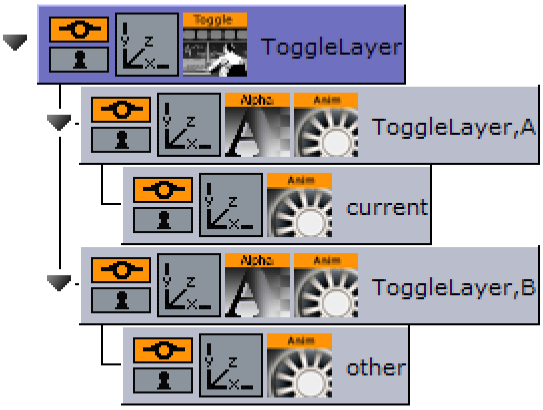
This section contains information on the following topics:

- Description and Methodological Considerations
- Properties and Parameters

Description and Methodological Considerations

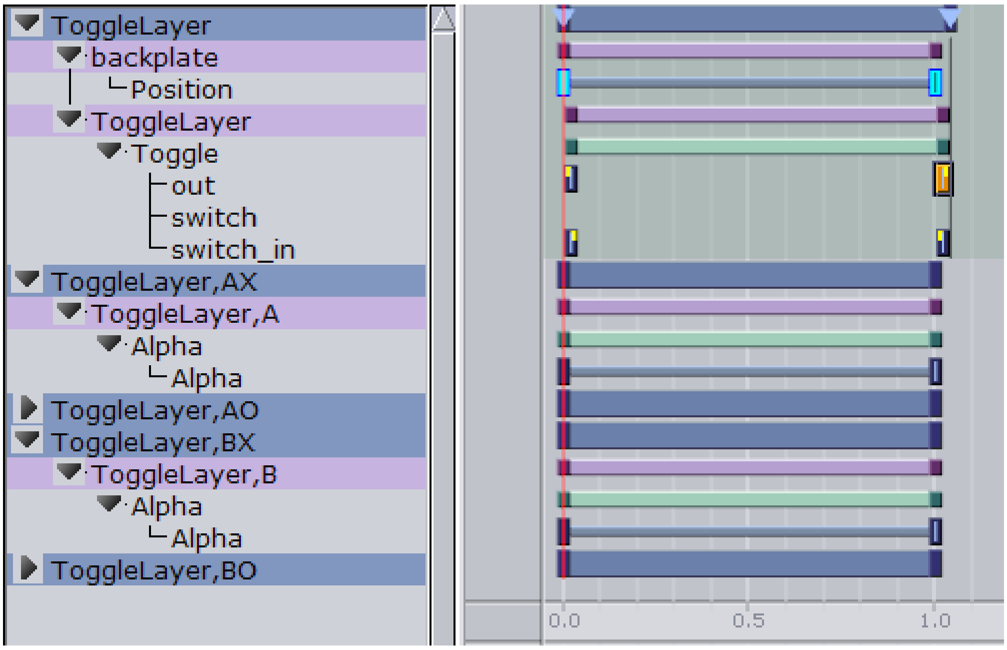
Toggle plug-in is one of the main building blocks of Transition Logic, and it is in general a relatively simple plug-in. When the Toggle plug-in is added to a container it will create two child containers that act as placeholder containers for objects that can toggle on and off screen. In addition it will also create some utility directors in the stage.

Figure 168: Toggle placeholder containers



The two placeholder containers correspond to an **A**-side and a **B**-side. Each of the placeholders will again have a child container; one named **current** the other named **other**. The *current* is typically the visible container, while the *other* is not visible. At the time of a switch, these containers will switch names. This means that at any given time it is not possible to know if the *current* parent is the *A* side or the *B* side. It is the responsibility of the Toggle plug-in to keep track of this.

Figure 169: Toggle directors and keyframes



ToggleLayer
backplate
Position
ToggleLayer
Toggle
out
switch
switch_in
ToggleLayer,AX
ToggleLayer,A
Alpha
Alpha
ToggleLayer,AO
ToggleLayer,BX
ToggleLayer,B
Alpha
Alpha
ToggleLayer,BO

The Toggle plug-in will create two directors each containing a default 1 second cross fade animation. These animations are placed in the **AX** and **BX** directors. *AX* can be interpreted as the *A*-side cross fade. The animations can be modified to show any kind of transition, but it is important that the *AX* and *BX* are the same.

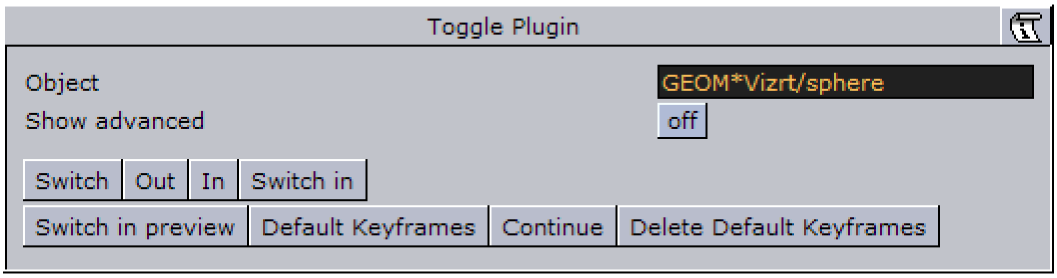
In addition to the cross fade directors the Toggle plug-in will also create placeholder directors for the *object* animations. Merged objects that are loaded into these placeholder containers during playout may contain animations embedded in the geometry object. The object animations are automatically added as a sub-director to the *AO* or *BO* director depending on if the object is being loaded on the *A*-side or *B*-side.

CAUTION! It is NOT recommended to manually adjust the AO and BO directors.

There are two different scenarios of when the utility directors will be triggered:

1. When there is a state change.
    - When there is a state change the Toggle Layer director will be animated from the current state to the desired next state. In this scenario the utility directors will be triggered according to the toggle keyframes in the stage. This means that it is the position of these keyframes that will determine the timing of when the utility directors are triggered.
2. No state change, only changing the current object.
    - When there is no state change there will be no movement of the Toggle Layer director. In this case the Toggle switch action will be triggered, which in effect will animate the utility directors at the same time; animating the current object out and the next object in at the same time.

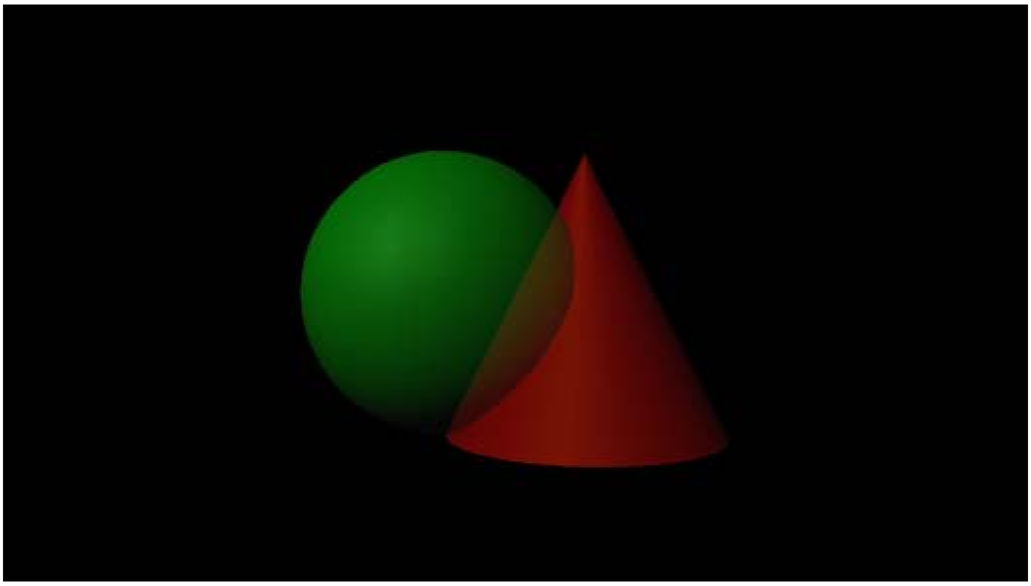
Note: Placeholder containers and directors are only generated if none exists from before.



Toggle Plugin

Object GEOM*Vizrt/sphere

Show advanced off



The plug-in has a built-in support for object loading. Enter the path to an object (e.g. GEOM*myfolder/myobject) into the **Object** field, and the plug-in will load this object on the *other* container. The object will by default be invisible and can be loaded with data, before activating the Switch action to show the object.

An object loaded this way will be cached, so that requesting the same object again will not trigger loading again. The object cache can also be preloaded by specifying all the geometries that should be preloaded before initializing the Toggle plug-in. Preloading of the objects needed by the Toggle plug-in is done when a playlist is initialized. For this reason the *Preload Object* text field should normally not be manually modified in Viz Artist.

Before adding the plug-in to a container in a scene, please make sure that:

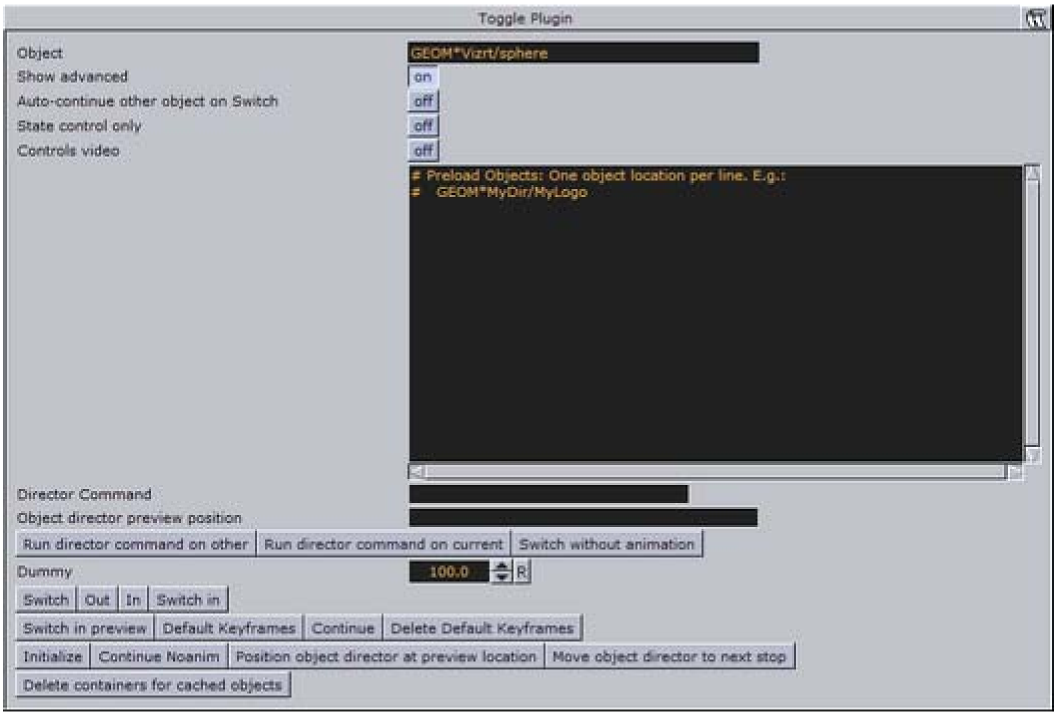
1. You give the container a unique and sensible name. This name will become the Toggle Layer name. Also, as the Toggle plug-in will automatically generate directors, which will be named based on the container name and its parents names, it is worth considering if the location of the container that will host the Toggle plug-in is in a proper position in the scene tree.
2. The container the Toggle plug-in will be added to must not have any child containers.
3. A director with the same name as the container (Toggle Layer name) is created.
4. The director has a minimum of two stop points and that these stop points are named correctly. The names must correspond with the Toggle Layer states.

IMPORTANT! One of the Transition Logic requirements is to have a stop point named “O”. Best practice is to always name the first stop point “O” (frame 0).

After the Toggle plug-in has been added to the container, the plug-in icon can be clicked to see its properties. Clicking the Default Keyframes button will create the most commonly used keyframes for triggering the toggling between the current and the other objects. Adjusting the timing can easily be done by moving the keyframes in the stage.

Properties and Parameters

Figure 170: Toggle editor



Toggle Plugin

Object: GEOM*Vizrt/sphere
Show advanced: on
Auto-continue other object on Switch: off
State control only: off
Controls video: off

```
# Preload Objects: One object location per line. E.g.:
#   GEOM*MyDir/MyLogo
```

Director Command:
Object director preview position:
Dummy: 100.0

Run director command on other
Run director command on current
Switch without animation
R
Switch
Out
In
Switch in
Switch in preview
Default Keyframes
Continue
Delete Default Keyframes
Initialize
Continue Noanim
Position object director at preview location
Move object director to next stop
Delete containers for cached objects

- **Object**: Displays the name of the currently loaded object (front scene GEOM). This field changes depending on the object loaded in the layer the toggle plug-in resides.
- **Show advanced**: Displays the advanced parameters.
  - **Auto-continue other object on Switch**: With this option enabled, toggle will issue a continue action to the other placeholder when a switch action is triggered.
  - **State control only**: If this option is enabled all commands sent to the Toggle will be ignored. This mean that no object will be loaded and only state changes on the layer director will be run.
  - **Controls video**: Controls the toggling of video channel 3 and 4. This applies when using the Control Video plug-in.
  - **Preload objects**: Specifies the objects that are to be pre-loaded on initialization.
  - **Director commands**: This field can be filled with certain director command parameters like for instance: SHOW 0.5, SHOW 50f, SHOW $pilot1. The

parameters are then formed into a valid set-director-position command, and applied to either current or other placeholder object director.

  - **Run director command on current** and **Run director command on other**: These buttons will form and apply a director command based on the director_commands field as described above. Note that these options will be deprecated in future versions.
  - **Switch without animation**: Will switch the visible container without running any of the (A|B)X directors or starting the object animation. The AX and BX directors will be set to the correct in/out positions.
  - **Dummy**: Is animated to create an animation that spans all the other toggle keyframes. This will in some cases solve problems with toggle keyframes not being triggered properly.
- **Switch**: Animates out the current placeholder, then switches the other placeholder to current and animates it in.
- **Out**: Animates out the current placeholder.
- **In**: Animates in the current placeholder.
- **Switch in**: Switches the other placeholder to current and animates it in.
- **Switch in preview**: Hides the current placeholder without animating it, then switches the other placeholder to current. Afterwards it shows the new current placeholder by jumping to a director state suitable for generating a still preview.
- **Default Keyframes**: Inserts default Toggle keyframes to invoke the toggles utility directors.
- **Continue**: Sends a continue action to the current object animation director.
- **Delete default keyframes**: Removes the default keyframes that were generated by the Default keyframes button.
- **Initialize**: Clears the cache, and reloads pre-loaded geometries.
- **Position object director at preview location**: When invoked, position the object director at given preview location in position field.
- **Move object director to next stop**: Moves the stage position to the next director stop point on a loaded object.
- **Delete containers for cached object**: Deletes all the containers used as the Toggles object cache.

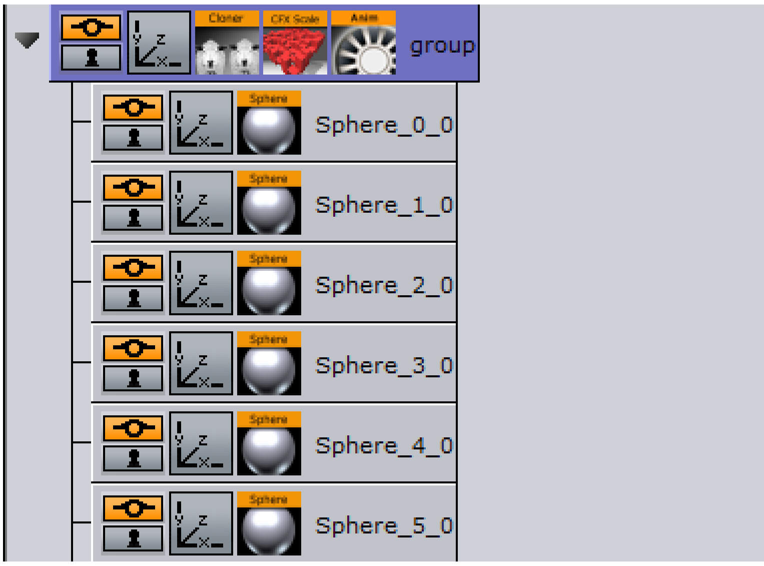
See Also

- Transition Logic

12.3 Container FX

Container FX is much like Text FX, except that instead of text, a set of containers are added under the plugin. Instead of animating the containers, Container FX animates the containers below (aka the container-Set). Those containers can be circles and so on.

Like Text FX plugins use containers, a Container FX plugin will use a set of containers under the plugin and change their position like Text FX does. It is recommended to be familiar with Text FX before designing with Container FX to understand the idea. You can also use the Cloner plugin to create the necessary containers under the plugin.

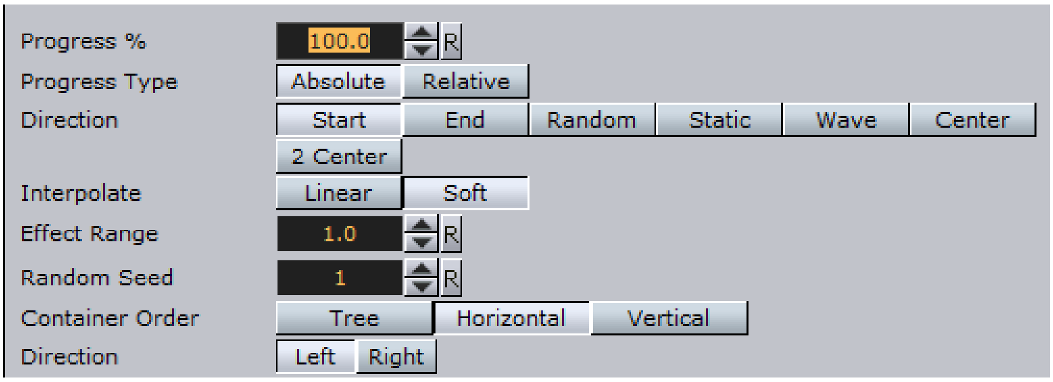


In the above example the containers under the group container with the Sphere names is the container-set, and will be animated here.

The following container effects plugins are housed in the Container FX folder:

- Common Container FX Properties
- Container FX 2D Follow
- Container FX Alpha
- Container FX Arrange
- Container FX Color
- Container FX Explode
- Container FX Jitter Alpha
- Container FX Jitter Color
- Container FX Jitter Position
- Container FX Jitter Scale
- Container FX Plus Plus
- Container FX Rotate
- Container FX Scale

12.3.1 Common Container FX Properties



Progress % 100.0 R
Progress Type: Absolute | Relative
Direction: Start | End | Random | Static | Wave | Center | 2 Center
Interpolate: Linear | Soft
Effect Range 1.0 R
Random Seed 1 R
Container Order: Tree | Horizontal | Vertical
Direction: Left | Right

The following properties and parameters are common to most of the Container FX plugins:

- **Progress %**: 0% progress is the beginning of the effect, 100% the end. Animate this value from 0% to 100% to see the effect or from 100% to 0% to animate the effect backwards.
- **Progress Type**
  - **Absolute**: 100% progress will animate all container's set, regardless of how many containers it has.
  - **Relative**: 100% progress animates 10 containers. This is needed to adjust the timing of several containers with different sizes. The effect speed should be for example 5 containers per second, so the animation must be from 0% to 100% in two seconds. This will work for with 10 containers or less. If you want to use more containers, animate the progress value over 100% (10% for each container).

- **Direction**: Sets the direction of the effect sequence, you can choose between the following options:
  - **Left**: Starts with the first container in the containers-set.
  - **Right**: Starts with the last container in the containers-set.
  - **Random**: Uses a random order.
  - **Static**: All containers are processed at the same time.
  - **Wave**: Starts with the first container, animates the effect from 0: 100% and then down again to 0%.
  - **Center**: Starts the effect from the center of the containers-set.
  - **2 Center**: Starts the effect at the same time from the beginning and the end of the containers-set. They meet at the center.

- **Interpolate**: Choose between a soft or a linear interpolation of the transition from container to container.
- **Effect Range**: Defines how many containers are processed at the same time. If for example the Effect Range is set to 4, and you manually increase the progress value, you will see that when the fifth container starts to be processed, the first is finished, when the sixth starts, the second is finished, and so on.
- **Random Seed**: Specifies a seed for the random number generator when a random direction is chosen. Even though Viz Artist uses random numbers, theanimation for a specific random seed will always look the same. This is typically useful if you combine two different container effects.

- **Container Order**: Sets the container order for the effect. Available options are As Is, Horizontal or Vertical. Horizontal and Vertical enables the containers - set Direction options.
- **Direction**: Sets the horizontal and vertical container-set direction (see Container Order). When the containers-set order is set to Horizontal, the container-set direction must be set to Left to Right or Right to Left. When the container order is set to Vertical, the containers-set direction must be set to Up to Down or Down to Up.

12.3.2 Container FX 2D Follow

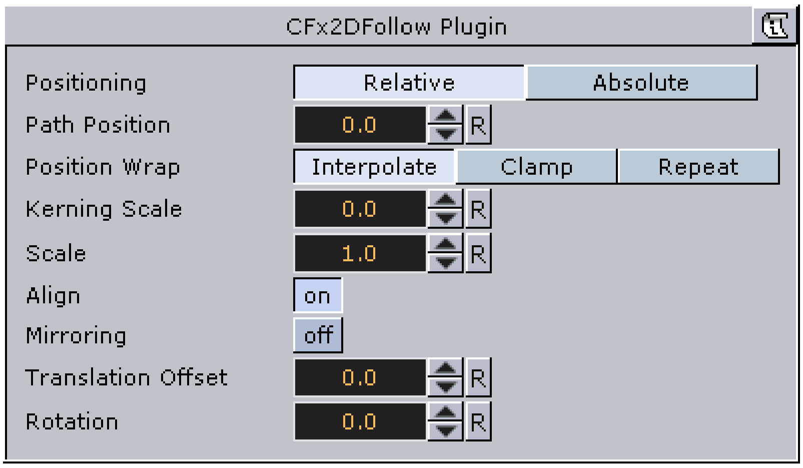
The Container FX 2D Follow plug-in allows you to create an animation of the Containers following a path (usually 2DRibbon).

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\CFx2DFollow.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



CFx2DFollow Plugin

Positioning	Relative / Absolute
Path Position	0.0
Position Wrap	Interpolate / Clamp / Repeat
Kerning Scale	0.0
Scale	1.0
Align	on
Mirroring	off
Translation Offset	0.0
Rotation	0.0

- **Positioning**: *Absolute* positioning evenly distributes characters over the length of the spline. *Relative* means that the text spacing of the string is used to position the text on the spline.
- **Path Position**: Moves the characters on the shape.
- **Position Wrap**: *Interpolate* means that text continues in the direction set by the last section of the spline; *Clamp* will stop the string on the spline end-point; *Repeat* will cause the string to move to the other spline end-point as soon as it moves over the end-point.
- **Kerning Scale**: Scales the kerning of the text on the spline.

- **Align**: Rotates the characters by Z to align the X-Axis with the tangent of the spline at the characters position.
- **Translation offset**: Moves the text from the spline in the XY plane.
- **Rotation**: Rotates the characters using the spline as a rotation axis.

12.3.3 Container FX Alpha

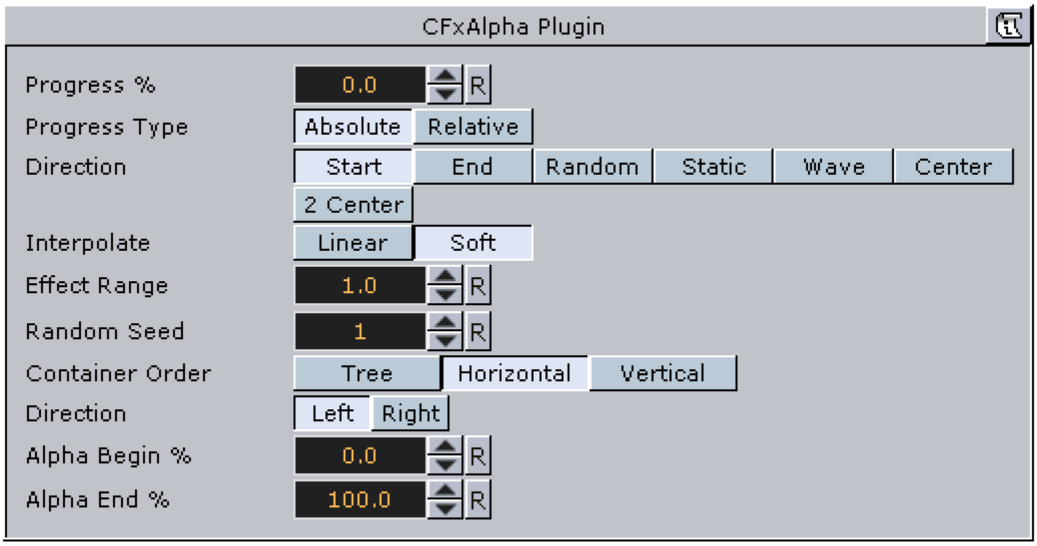
The Container FX Alpha plugin is a container-set effect that creates a fade in effect for the containers in the set. The effect sequence can be set to go many different ways. A fade in and out effect can also be achieved.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\CFxAlpha.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



CFxAlpha Plugin

Progress % 0.0 R
Progress Type Absolute Relative
Direction Start End Random Static Wave Center 2 Center
Interpolate Linear Soft
Effect Range 1.0 R
Random Seed 1 R
Container Order Tree Horizontal Vertical
Direction Left Right
Alpha Begin % 0.0 R
Alpha End % 100.0 R

- **Alpha Begin %**: Sets the alpha level of the containers at 0% effect.
- **Alpha End %**: Sets the alpha level of the containers at 100% effect.

See Also

- Common Container FX Properties

12.3.4 Container FX Arrange

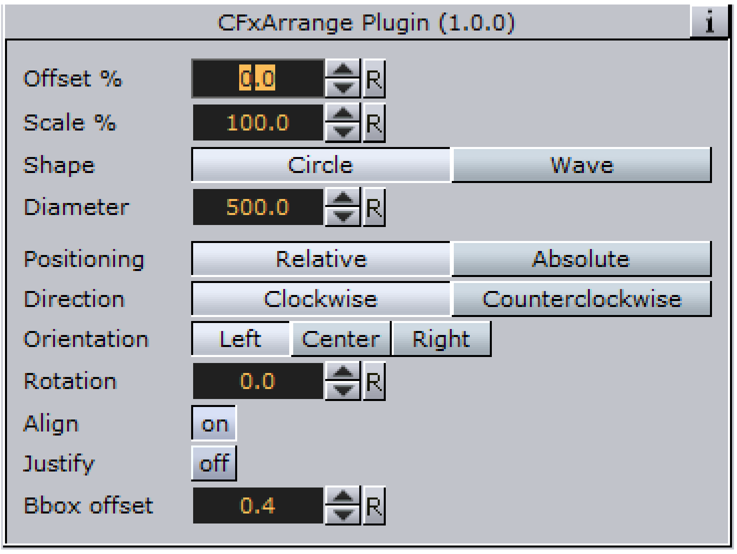
The Container FX Arrange plugin allows you to arrange containers in either a circular or a wave shape. The containers can be animated on the selected shape by animating the offset value.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\CFxArrange.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



CFxArrange Plugin (1.0.0)

Offset % 0.0
Scale % 100.0
Shape Circle | Wave
Diameter 500.0
Positioning Relative | Absolute
Direction Clockwise | Counterclockwise
Orientation Left | Center | Right
Rotation 0.0
Align on
Justify off
Bbox offset 0.4

- **Offset %**: Moves the containers on the shape. 100% means one full rotation of the containers-set on the circle.
- **Scale %**: Sets the container-set on the shape. The parameter does not scale the containers, but the center of the bouding box of the containers.
- **Shape**: Changes the shape of the container layout. The options are Circle or Wave.
- **Diameter**: Sets the diameter of the circular shape.
- **Positioning**: Defines the position of the container-set on the circular shape. Relative means that the spacing of the container-set is maintained. Absolute means that the container-set is evenly distributed out on the circle.
- **Direction**: Sets the direction of the container-set on the circle to either Clockwise or Counterclockwise.
- **Align**: Aligns the container along the curve.

12.3.5 Container FX Color

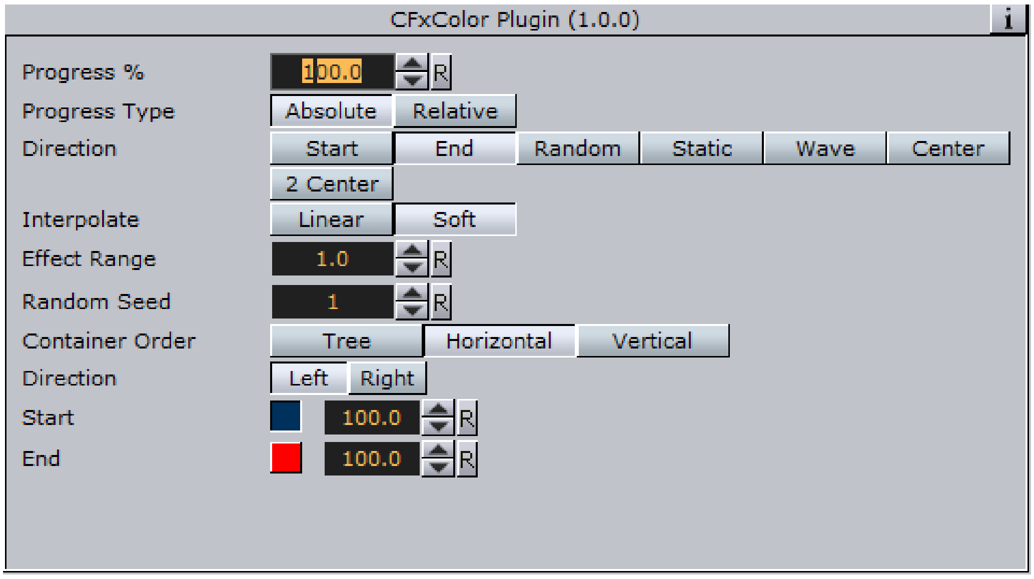
Adds a color effect to the containers-set.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\CfxColor.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



CFxColor Plugin (1.0.0)

Progress % 100.0 R

Progress Type: Absolute | Relative

Direction: Start | End | Random | Static | Wave | Center | 2 Center

Interpolate: Linear | Soft

Effect Range 1.0 R

Random Seed 1 R

Container Order: Tree | Horizontal | Vertical

Direction: Left | Right

Start 🟦 100.0 R

End 🟥 100.0 R

- **Color Start**: Sets the initial color before applying the effect.
- **Color End**: Sets the color the effect should apply.

See Also

- Common Container FX Properties

12.3.6 Container FX Explode

The Container FX Explode plug-in creates an explosion like function where the containers get thrown away from their initial position. The speed, direction and spread of the moving containers can be altered with parameters.

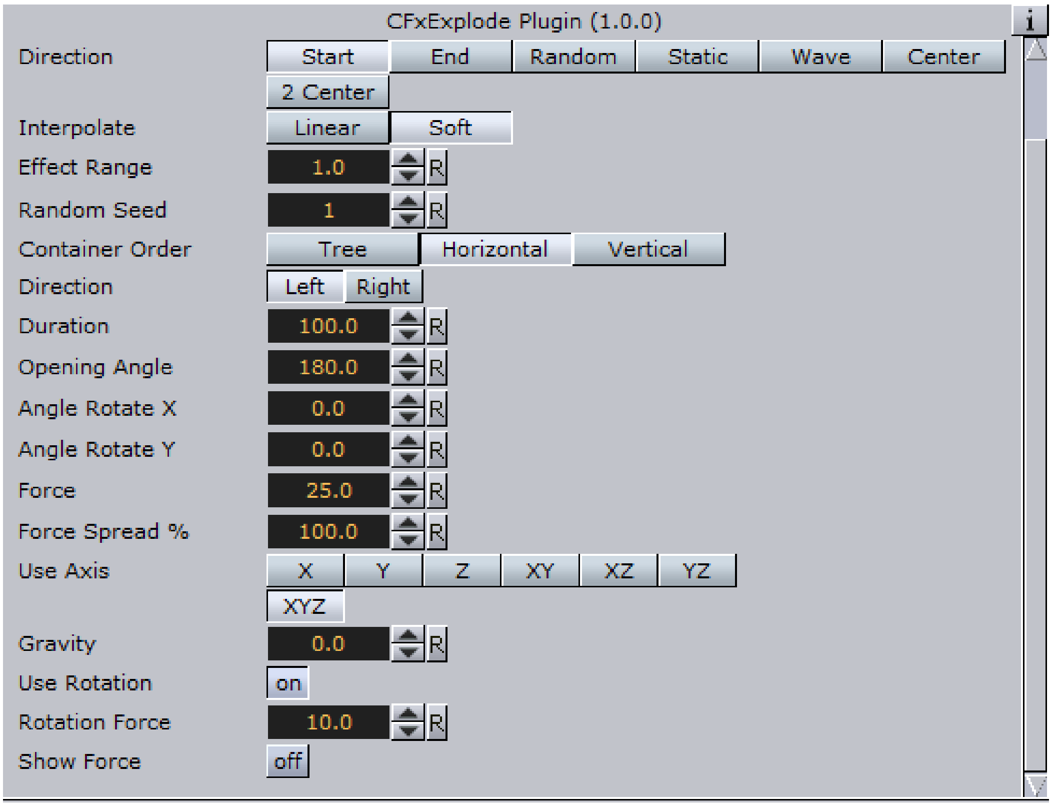
Note: Works only if the containers-set is set to texture

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\CFxExplode.vip`

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



CFxExplode Plugin (1.0.0)

Direction: Start | End | Random | Static | Wave | Center | 2 Center

Interpolate: Linear | Soft

Effect Range: 1.0

Random Seed: 1

Container Order: Tree | Horizontal | Vertical

Direction: Left | Right

Duration: 100.0

Opening Angle: 180.0

Angle Rotate X: 0.0

Angle Rotate Y: 0.0

Force: 25.0

Force Spread %: 100.0

Use Axis: X | Y | Z | XY | XZ | YZ | XYZ

Gravity: 0.0

Use Rotation: on

Rotation Force: 10.0

Show Force: off

- **Duration**: Defines the duration of the progress for each of the containers. Simply increase the value if you want it last longer. You could achieve the same by making the gravity stronger and animating the progress slower, but it is easier to increase the duration instead if your explode effect is too short.
- **Opening Angle**: Sets the angle for the spread of the containers. 0 will send them straight up, 360 will spread them in a circular shape.
- **Angle Rotate X**: Rotates the opening angle around the X-axis.
- **Angle Rotate Y**: Rotates the opening angle around the Y-axis.
- **Force**: Sets the force that throws away the containers. A high force will make them go far away, conversely a low force will create only a small motion of the containers.
- **Force Spread %**: Sets a variation of the force among the containers.
- **Use Axis**: Allows you to select on which axis or combination of axes the containers are to spread along.
- **Gravity**: Sets a gravity force that influences the path of the containers to end up going downwards. The higher the value is set, the faster each container will divert from its initial path and start going downwards.

- **Use Rotation**: With this option enabled, the container will rotate as they are being thrown away from their initial position.
- **Rotation Force**: Sets the degree of rotation as the containers are being thrown away.
- **Show Force**: With this option enabled, lines showing the containers path and speed will be visible in the render output.

See Also

- Common Container FX Properties

12.3.7 Container FX Jitter Alpha

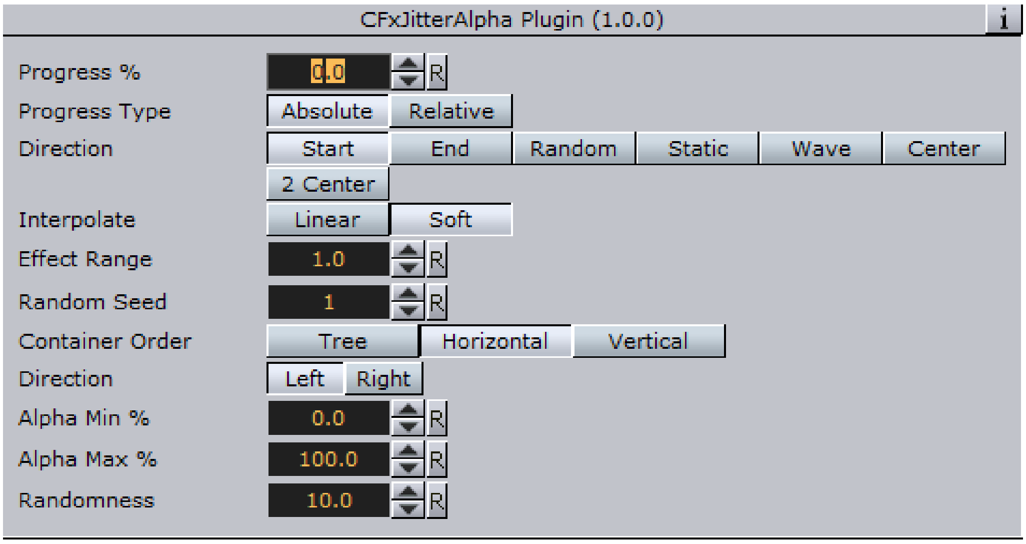
The Container FX Jitter Alpha plugin creates a jittering motion of the containers by randomly changing the alpha value of each container. The degree of change and the start sequence of the jittering can be altered. To use the plugin, add it onto a container with a font. To create an animation, animate the progress. Other values can of course be animated as well.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\CFxJitterAlpha.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



CFxJitterAlpha Plugin (1.0.0)

Progress %: 0.0
Progress Type: Absolute | Relative
Direction: Start | End | Random | Static | Wave | Center | 2 Center
Interpolate: Linear | Soft
Effect Range: 1.0
Random Seed: 1
Container Order: Tree | Horizontal | Vertical
Direction: Left | Right
Alpha Min %: 0.0
Alpha Max %: 100.0
Randomness: 10.0

- **Effect What**: Defines where the effect should have an effect. Available options are background (BG), foreground (FG) or both.
- **Alpha Begin**: Sets the alpha level of the container at 0% effect.

- **Alpha End**: Sets the alpha level of the container at 100% effect.
- **Randomness**: Sets the intensity of the jittering alpha changes.

See Also

- Common Container FX Properties

12.3.8 Container FX Jitter Color

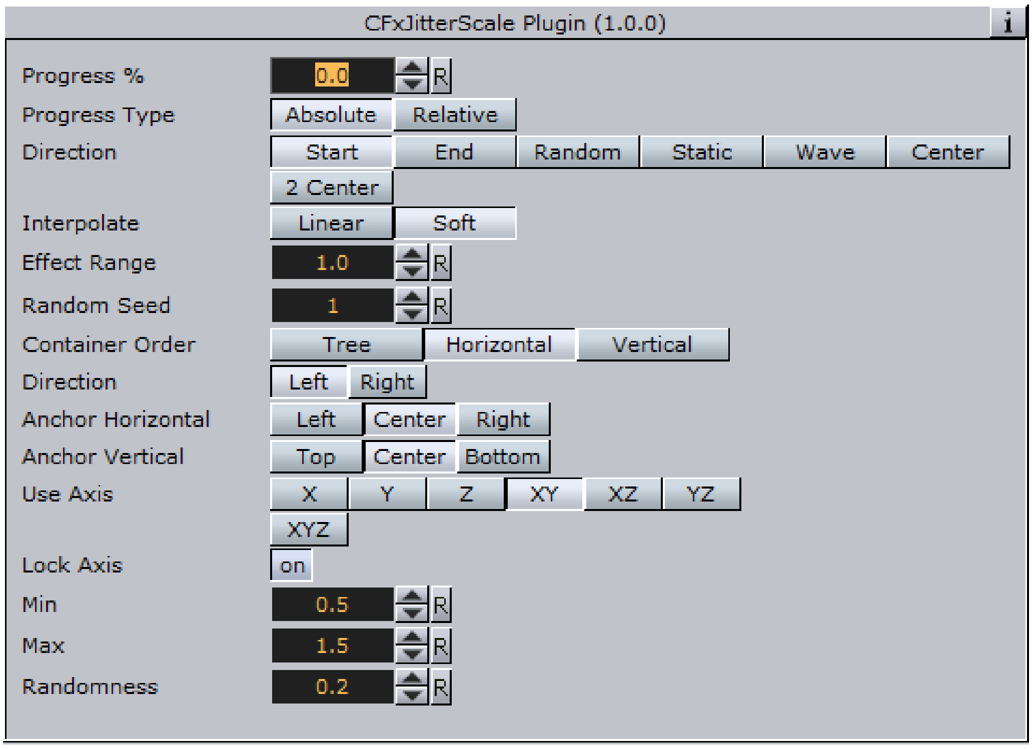
The Container FX Jitter Color animates a jittering effect on the color of the container.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\CFxJitterColor.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



CFxJitterScale Plugin (1.0.0)

Progress %	0.0
Progress Type	Absolute / Relative
Direction	Start / End / Random / Static / Wave / Center / 2 Center
Interpolate	Linear / Soft
Effect Range	1.0
Random Seed	1
Container Order	Tree / Horizontal / Vertical
Direction	Left / Right
Anchor Horizontal	Left / Center / Right
Anchor Vertical	Top / Center / Bottom
Use Axis	X / Y / Z / XY / XZ / YZ / XYZ
Lock Axis	on
Min	0.5
Max	1.5
Randomness	0.2

- **Anchor Horizontal**: Sets the anchor point for the containers on the horizontal plane.
- **Anchor vertical**: Sets the anchor point for the containers on the vertical plane.

- **Use Axis**: Defines on which axis or axes the containers scale to create the jittering effect.
- **Lock Axis**: With this option enabled, all the axes get the same Min, Max and Randomness settings. If you disable randomness, these parameters will be visible for each of the axes and must be set individually.
- **Min**: Sets the minimum scaling for the containers.
- **Max**: Sets the maximum scaling for the containers.
- **Randomness**: Sets the intensity of the jittering movement.

See Also

- Common Container FX Properties

12.3.9 Container FX Jitter Position

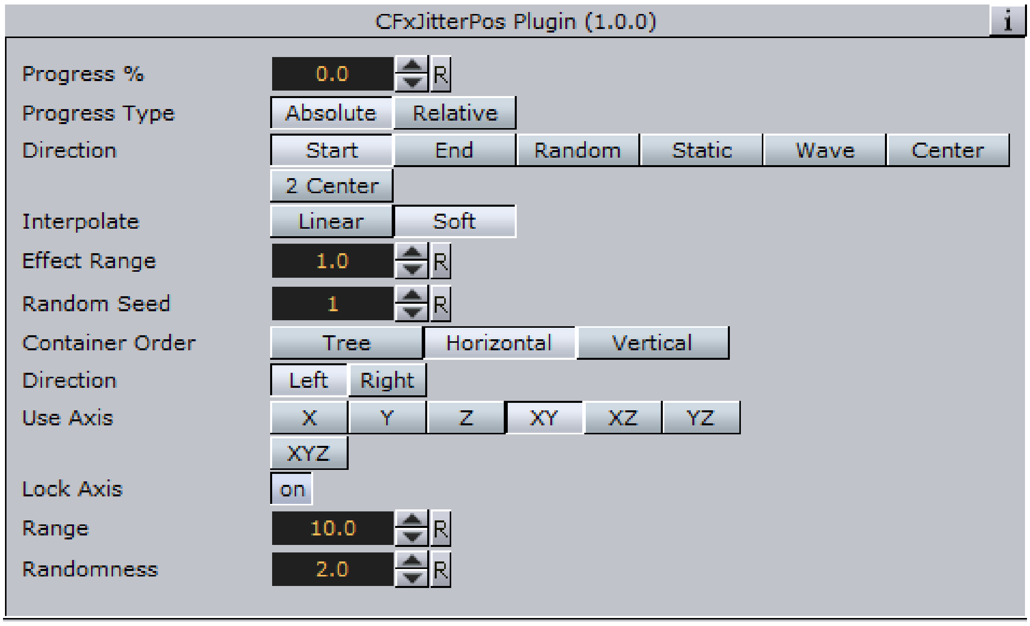
The Container FX Jitter Position plug-in creates a jittering motion of the containers by randomly changing the position of each container. The degree of position change and the starting sequence of the jittering can be altered. To create an animation, animate the progress. Other values can of course be animated as well.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\CfxJitter.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



CFxJitterPos Plugin (1.0.0)

Progress % 0.0

Progress Type: Absolute | Relative

Direction: Start | End | Random | Static | Wave | Center | 2 Center

Interpolate: Linear | Soft

Effect Range 1.0

Random Seed 1

Container Order: Tree | Horizontal | Vertical

Direction: Left | Right

Use Axis: X | Y | Z | XY | XZ | YZ | XYZ

Lock Axis on

Range 10.0

Randomness 2.0

- **Use Axis**: Defines on which axis or axes the containers moves to create the jittering effect.
- **Lock Axis**: With this option enabled Range and Randomness will be the same for all axes. If you disable it, Range and Randomness for each of the axes will be displayed and you can set them individually.
- **Range**: Sets the range of the jittering movement.
- **Randomness**: Sets the intensity of the jittering movement.

See Also

- Common Container FX Properties

12.3.10 Container FX Jitter Scale

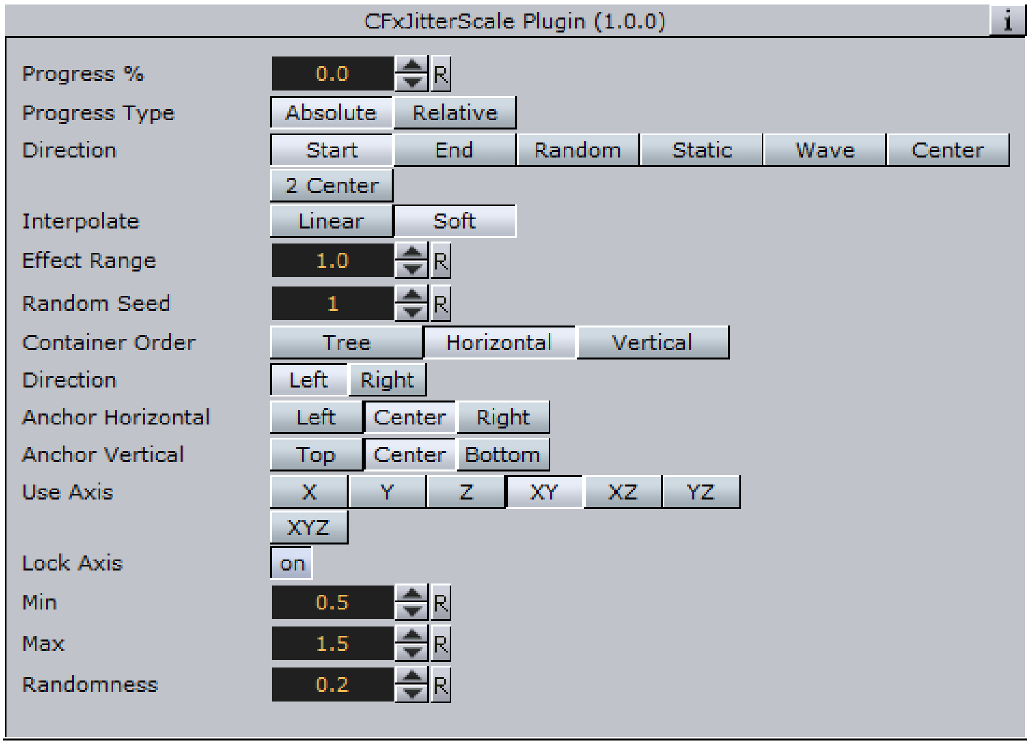
The Container FX Jitter Scale animates a jittering effect on the scale of the container.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\CFxJitterScale.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



CFxJitterScale Plugin (1.0.0)

Progress %: 0.0
Progress Type: Absolute | Relative
Direction: Start | End | Random | Static | Wave | Center | 2 Center
Interpolate: Linear | Soft
Effect Range: 1.0
Random Seed: 1
Container Order: Tree | Horizontal | Vertical
Direction: Left | Right
Anchor Horizontal: Left | Center | Right
Anchor Vertical: Top | Center | Bottom
Use Axis: X | Y | Z | XY | XZ | YZ | XYZ
Lock Axis: on
Min: 0.5
Max: 1.5
Randomness: 0.2

- **Anchor Horizontal**: Sets the anchor point for the containers on the horizontal plane.
- **Anchor vertical**: Sets the anchor point for the containers on the vertical plane.
- **Use Axis**: Defines on which axis or axes the containers scale to create the jittering effect.
- **Lock Axis**: With this option enabled, all axes get the same Min, Max and Randomness settings. If you disable randomness, these parameters will be visible for each of the axes and must be set individually.
- **Min**: Sets the minimum scaling for the containers.
- **Max**: Sets the maximum scaling for the containers.
- **Randomness**: Sets the intensity of the jittering movement.

See Also

- Common Container FX Properties

12.3.11 Container FX Plus Plus

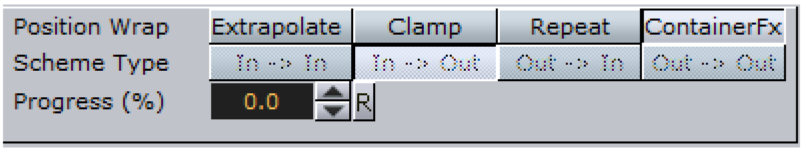
The Container FX Plus Plus plugin allows you to set a number of effects on a container's set.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\CFxPlusPlus.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



- **Position Wrap**
  - **Extrapolate**: Above 100% or below 0% of Path Position will continue the direction of the spline.
  - **Clamp**: Will stop the string on the spline endpoints.
  - **Repeat**: Will cause the string to move to the other spline endpoint as soon as it moves over the endpoint.
  - **ContainerFx**: Will position containers by the relative spline of each container using the ContainerFx parameters.

- **Scheme Type**: Defines how the container-set will look at 0% and 100% progress.
  - **In -> In**: at 0% first container will be at the beginning of the spline, at 100% last container will be at the end of the spline.
  - **In -> Out**: at 0% first container will be at the beginning of the spline, at 100% first container will be at the end of the spline.
  - **Out -> In**: at 0% last container will be at the beginning of the spline, at 100% first container will be at the end of the spline.
  - **Out -> Out**: at 0% last container will be at the beginning of the spline, at 100% last container will be at the end of the spline.

- **Progress (%)**: Animates the progress of the effect(s).

See Also

- Common Container FX Properties

12.3.12 Container FX Rotate

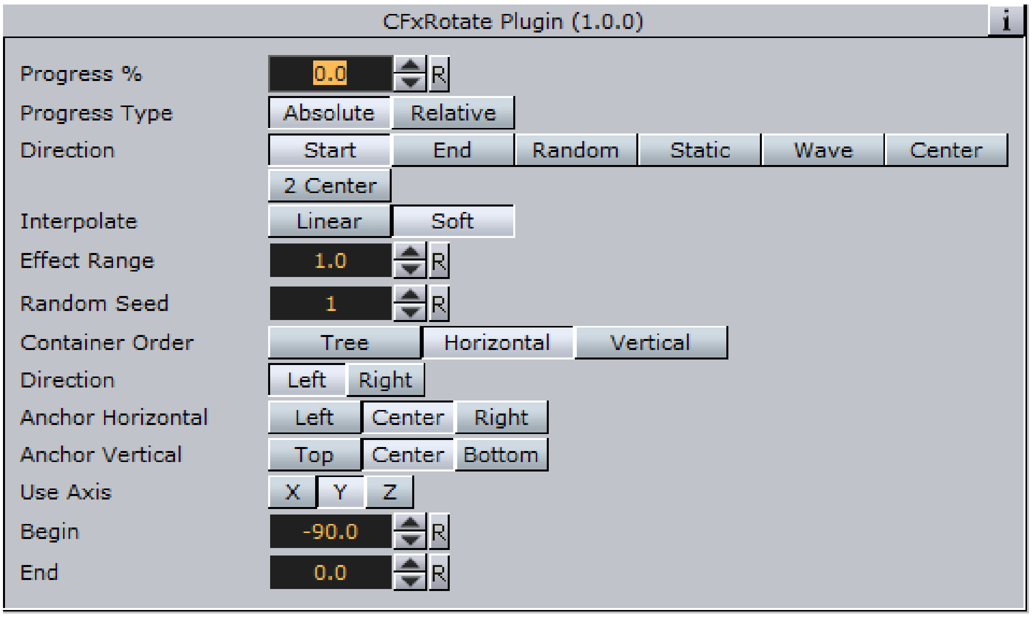
The Container FX Rotate plugin allows you to create an effect where the containers rotate on the X-, Y- or Z-axis.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\CFxRotate.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



CFxRotate Plugin (1.0.0)

Progress %: 0.0
Progress Type: Absolute | Relative
Direction: Start | End | Random | Static | Wave | Center | 2 Center
Interpolate: Linear | Soft
Effect Range: 1.0
Random Seed: 1
Container Order: Tree | Horizontal | Vertical
Direction: Left | Right
Anchor Horizontal: Left | Center | Right
Anchor Vertical: Top | Center | Bottom
Use Axis: X | Y | Z
Begin: -90.0
End: 0.0

- **Anchor Horizontal**: Sets the anchor point for the containers on the horizontal plane.
- **Anchor Vertical**: Sets the anchor point for the containers on the vertical plane.
- **Use Axis**: Defines on which axis the containers rotate.
- **Begin**: Sets the initial rotated position of the containers.
- **End**: Sets the ending rotated position of the containers.

See Also

- Common Container FX Properties
- Expert

12.3.13 Container FX Scale

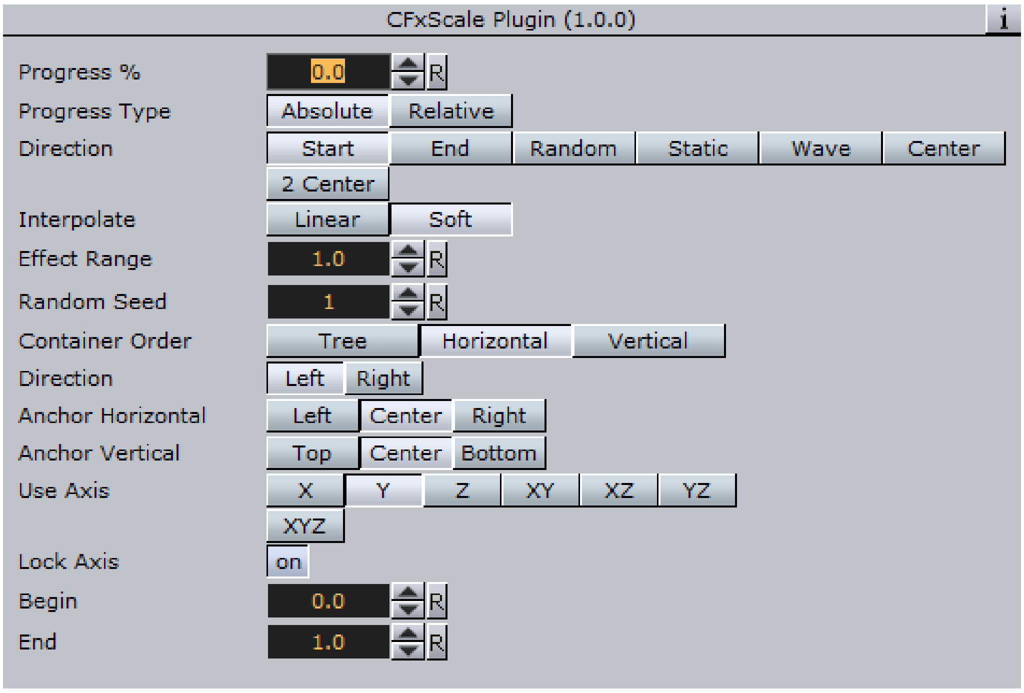
The Container FX Scale plug-in allows you to create a scaling animation of the containers.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\CFxScale.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



CFxScale Plugin (1.0.0)

Progress % 0.0

Progress Type: Absolute | Relative

Direction: Start | End | Random | Static | Wave | Center | 2 Center

Interpolate: Linear | Soft

Effect Range 1.0

Random Seed 1

Container Order: Tree | Horizontal | Vertical

Direction: Left | Right

Anchor Horizontal: Left | Center | Right

Anchor Vertical: Top | Center | Bottom

Use Axis: X | Y | Z | XY | XZ | YZ | XYZ

Lock Axis: on

Begin 0.0

End 1.0

- **Anchor Horizontal**: Sets the anchor point for the containers on the horizontal plane.
- **Anchor Vertical**: Sets the anchor point for the containers on the vertical plane.
- **Use Axis**: Defines on which axis or axes the containers scale.
- **Lock Axis**: With this option enabled the Begin- and End scaling are the same for all the axes. If you disable the option, Begin and End must be set for all axes individually.
- **Begin**: Sets the initial size of the containers.
- **End**: Sets the ending size of the containers.

See Also

- Common Container FX Properties

12.4 Control

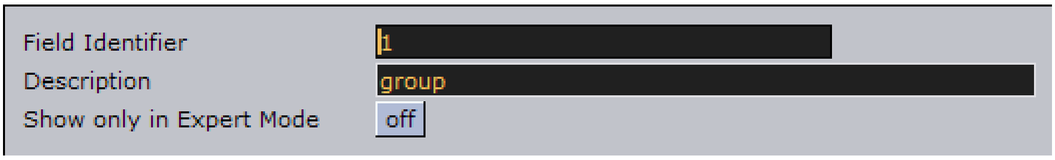
The following container plugins are housed in the Control folder:

- Control Action
- Control Action Table
- Control Audio
- Control Bars
- Control Chart
- Control Clip
- Control Clock
- Control Condition
- Control Container

- Control Data Action
- Control Datapool
- Control DP Object
- Control FeedView
- Control Geom
- Control Hide in Range
- Control Hide on Empty
- Control Image
- Control Keyframe
- Control List
- Control Map
- Control Material
- Control Num
- Control Object
- Control Omo
- Control Parameter
- Control Pie
- Control Scaling
- Control Sign Container
- Control SoftClip
- Control Stoppoint
- Control Text
- Control Targa Alpha
- Control Targa Clip
- Control Video
- Placeholder

12.4.1 Common Control Plug-in Properties

Most control plug-ins have a set of common properties. These are the Field Identifier, Description and Show only in Expert Mode.



Field Identifier 1
Description group
Show only in Expert Mode off

- **Field Identifier**: When making templates for Viz Trio, this should be a numeric value, Range: 1-n. For Viz Content Pilot and Viz Ticker, this can also be a descriptive text string. This value is used by the control clients to identify the editable element in the scene. In Viz Trio the numbers from the control plug-ins are used to create the tab-order between the editable elements.
- **Description**: A description of the tab field or editable object. This will be used as a description for the elements when used in the control clients.
- **Show only in Expert Mode**: Hides the tab-field properties for the user if Expert Mode is enabled in Viz Trio. For more information, read about Viz Trio’s macro commands, and the command *set_expert_mode_enabled*.

12.4.2 Control Action

The Control Action plug-in executes an action, for example Viz Engine commands or Control Object commands, when receiving input on the “input” field.

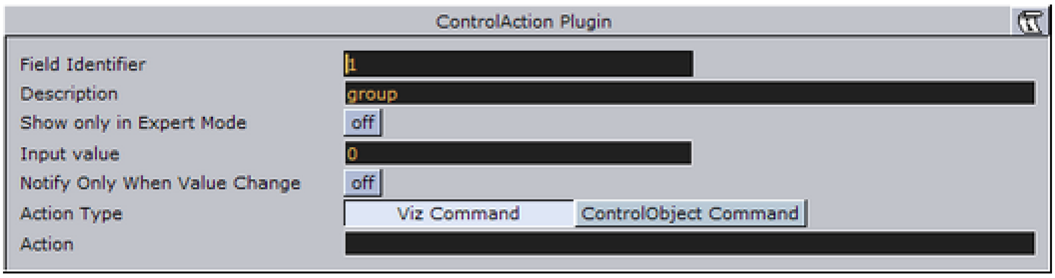
The action can contain more than one command to be called, and the commands must be separated by semicolons. When “Notify Only When Value Change” is checked the actions will only be triggered if the input value differs from the one already stored on the “input” field from previous invocations.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlAction.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ControlAction Plugin

Field Identifier	1
Description	group
Show only in Expert Mode	off
Input value	0
Notify Only When Value Change	off
Action Type	Viz Command / ControlObject Command
Action	

- **Input Value**: Specifies the value input will be compared against
- **Notify Only When Value Change**: Triggers a command when a value is changed.
- **Action Type**
  - **Viz Command**: Triggers an internal Viz Engine command. For example: *“THIS_SCENE*STAGE*DIRECTOR*Audio START”.*
  - **ControlObject Command**: Triggers a ControlObject command. For example: *“ON 1 SET abc”.*
- **Action**: Actions to execute if values match.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.3 Control Action Table

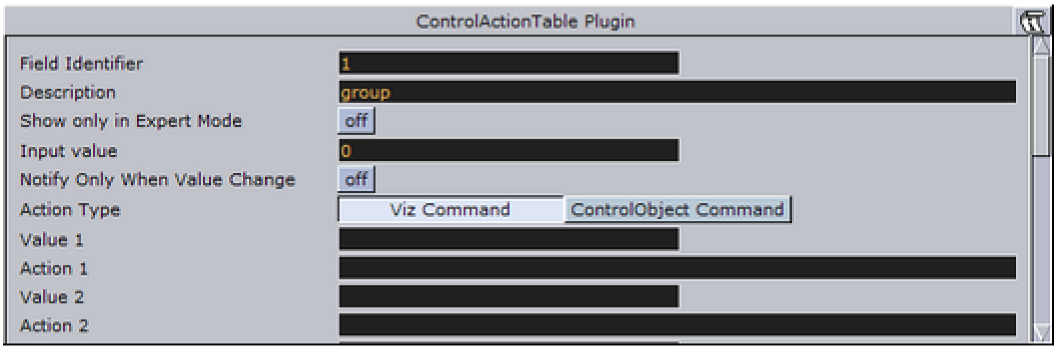
The Control Action Table plug-in allows defining a table of actions similar to Control Action. The actions are associated with a given value. When Control Action Table receives input, it compares the received data to each of the values. If the data matches one of the values the plug-in starts the corresponding action.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlActionTable.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ControlActionTable Plugin

Field Identifier	1
Description	group
Show only in Expert Mode	off
Input value	0
Notify Only When Value Change	off
Action Type	Viz Command / ControlObject Command
Value 1	
Action 1	
Value 2	
Action 2	

- **Input Value**: Specifies the value input will be compared against
- **Notify Only When Value Change**: Triggers a command when a value is changed.
- **Value(n)**: The value that the given action should be associated with.
- **Action(n)**: Actions to execute if value match
- **Default Action**: Will be executed if the input does not match any of the specified values.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.4 Control Audio

The Control Audio plug-in enables the operator of a control application to choose a sound clip.

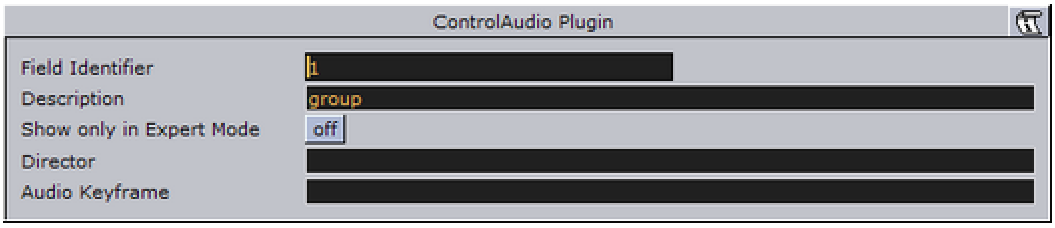
To expose the different properties, and change the audio file in one of the control applications, simply add the Control Audio plug-in to the same container as the Audio plug-in.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlAudio.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ControlAudio Plugin

Field Identifier	1
Description	group
Show only in Expert Mode	off
Director	
Audio Keyframe	

- **Director**: Sets the name of the stage director the audio channel is located below.
- **Audio Keyframe**: Sets the ID/name of the audio clip keyframe.

Note: Under most circumstances there is no need to set director and audio keyframe.

See Also

- Common Control Plug-in Properties
- Control Object
- Audio

12.4.5 Control Bars

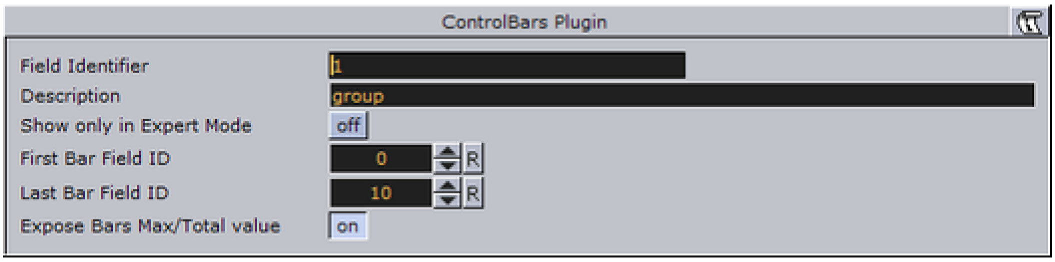
The Control Bars plug-in allows binding of tab field values to the Bar Value fields in the Bar values presenter plug-in. Each bar value property in the Bar Values editor is presented as a separate tab field in Viz Trio.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlBars.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ControlBars Plugin

Field Identifier	1
Description	group
Show only in Expert Mode	off
First Bar Field ID	0
Last Bar Field ID	10
Expose Bars Max/Total value	on

- **First Bar Field ID**: Sets the tab-field ID for the first Bar value.
- **Last Bar Field ID**: Sets the tab-field ID for the Last Bar value
- **Expose Bars Max/Total Value**: If enabled the Bars Max/Total value will be exposed as a separate tab field.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.6 Control Chart

The Control Chart plug-in binds control of chart data to a Control Object. It currently supports the graph geometry plug-in and the bar values, pie values and Vizual Data Tools plug-ins. It serves the purpose to interface all kind of chart data. It covers graph charts, bar charts, pie charts, area charts and generally all kind of plug-in data, which has a table-like representation.

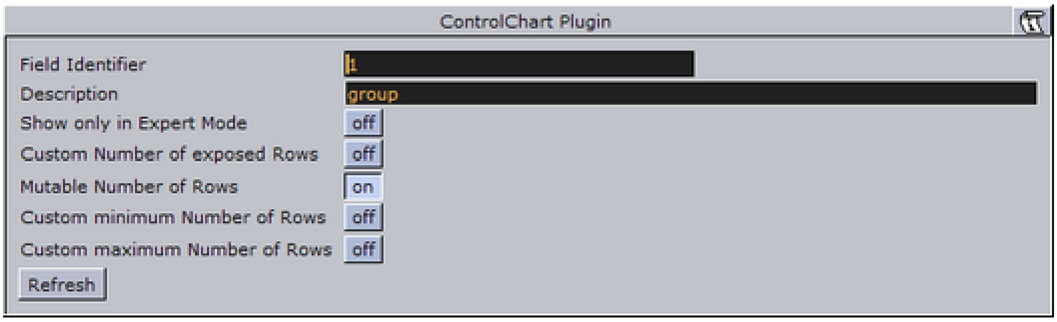
Control chart offers the same kind of interface like control list. That means it delivers a schema specification as type encoding. The chart data is communicated as an XML table value fully compatible to control list values. The supported control chart commands mimic the control list commands.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlChart.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ControlChart Plugin

Field Identifier: 1
Description: group
Show only in Expert Mode: off
Custom Number of exposed Rows: off
Mutable Number of Rows: on
Custom minimum Number of Rows: off
Custom maximum Number of Rows: off

- **Custom Number of Exposed Rows**: Enables the Number of exposed Rows setting.
  - **Number of exposed rows**: Sets the default number of exposed rows for the scene when the number of rows can be changed by the operator. If the number of rows cannot be changed the number of exposed rows is fixed.
- **Mutable Number of Rows**: If set to on, the operator can add or delete rows.
  - **Custom minimum and maximum Number of Rows**: Enables the Minimum and Maximum Number of Rows settings. To the operator, the number of fields can only be added or deleted when inside the range (for example greater than > 2 and less than < 10).

IMPORTANT! If Control Chart is used in combination with Vizual Data Tools plug-ins, make sure the Vizual Data Tools plug-in's Shared Memory is set to Inactive.

See Also

- Common Control Plug-in Properties
- Control Object
- Vizual Data Tools

12.4.7 Control Clip

The Control Clip binds control of an AVI clip played by Mo Viz or Softclip to a Control Object. With the Control Clip plug-in the clips may be changed within the control application.

The Mo Viz plugin enables the user to play media files or media streams inside Viz Engine. Mo Viz uses the Microsoft DirectShow Filtergraph framework to render the media. Therefore it is possible to play for example MPEG and QuickTime files or streams from a server. For the latter to work, the system must be appropriately equipped.

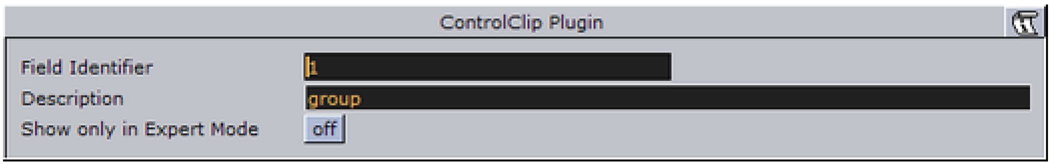
Softclip is a Viz plug-in that can show AVI clips, either projected on a texture, or rendered directly.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlClip.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ControlClip Plugin

Field Identifier: 1

Description: group

Show only in Expert Mode: off

This plug-in does not have any properties or parameters except those that are common to all plug-ins.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.8 Control Clock

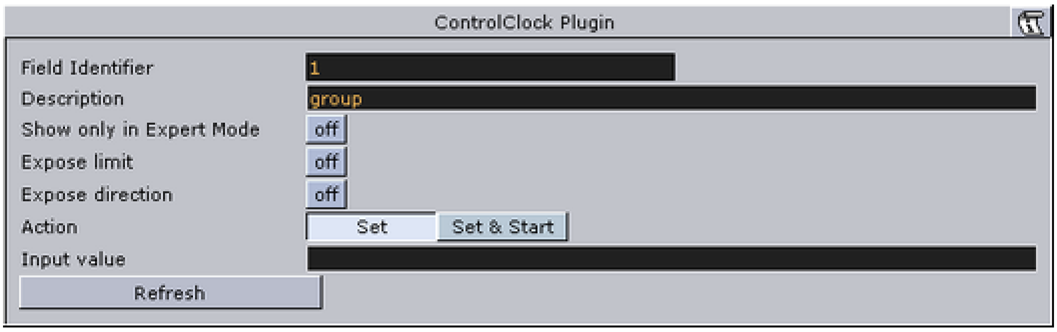
The Control Clock plug-in allows the user to set time for a clock as well as controlling several clock functions. The clock object is a text container in the scene with the clock function enabled.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlClock.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ControlClock Plugin

Field Identifier: 1
Description: group
Show only in Expert Mode: off
Expose limit: off
Expose direction: off
Action: Set | Set & Start
Input value:

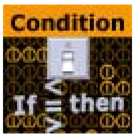
Refresh

- **Expose Limit**: Enables the user set a limit for the clock.
- **Expose Direction**: Enables the user to choose the direction for the clock.
- **Action**: Sets the clock action that will be the default value in the control client's clock editor.
- **Input Value**: Shows the current value for the plug-in. It is not necessary to set any value here. It is normally only used for debugging purposes.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.9 Control Condition



With the Control Condition plug-in action can be triggered based on the input given in a tab-field with the same field identifier. Actions can be Control Object commands in the form like this

```
"ON 2 SET Mike Johnson"
```

or Viz commands like

```
0 RENDERER*STAGE START
```

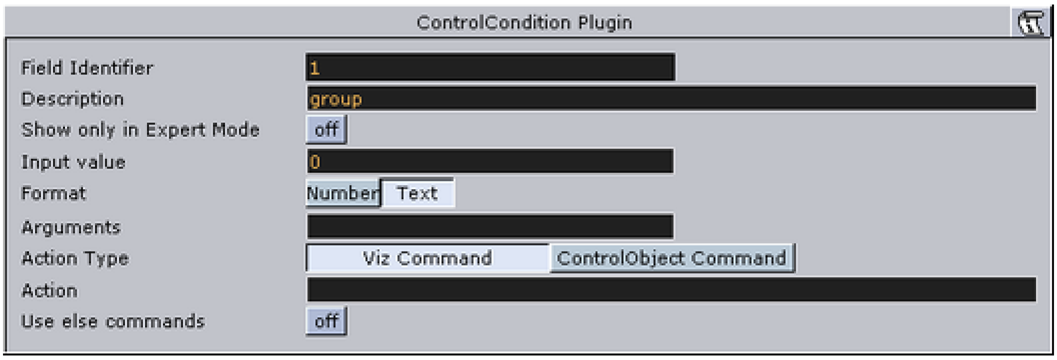
The Viz Engine commands must be prefixed with a zero followed by a space. This is to enable the plug-in to recognize that the command should be interpreted as a Viz Engine command.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlCondition.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ControlCondition Plugin

Field Identifier: 1
Description: group
Show only in Expert Mode: off
Input value: 0
Format: Number Text
Arguments:
Action Type: Viz Command ControlObject Command
Action:
Use else commands: off

- **Input Value**: Shows the current input value to the plug-in.
- **Format**: Sets the format for the input value that it should conditionally trigger on.
- **Condition**: Sets the type of condition if Format is set to number.
- **Arguments**: Sets the arguments for the conditional operation.
- **Action Type** / **Action**: Sets the action that is to be performed when the condition is met. It can be a control object command like “ON 2 SET mytext” or a Viz Engine command prefixed by a zero and a space.
- **Use Else Commands**: When enabled, an *else* action will be triggered when the input data falls within the conditions.
- **Else Action**: Sets the else action that will be triggered when the input data does not fall within the conditions.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.10 Control Container

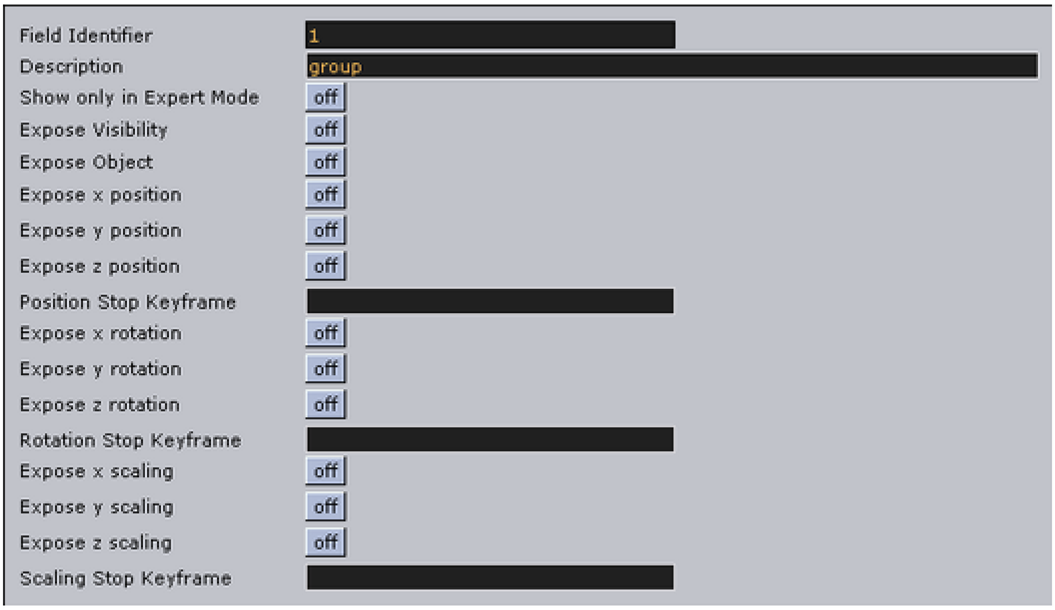
The Control Container plug-in exposes a range of different transformation properties for a container. See the Viz *Artist User’s Guide* for a detailed explanation of all the transformation properties.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlContainer.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



Field Identifier	1
Description	group
Show only in Expert Mode	off
Expose Visibility	off
Expose Object	off
Expose x position	off
Expose y position	off
Expose z position	off
Position Stop Keyframe	
Expose x rotation	off
Expose y rotation	off
Expose z rotation	off
Rotation Stop Keyframe	
Expose x scaling	off
Expose y scaling	off
Expose z scaling	off
Scaling Stop Keyframe	

The following properties can be exposed:

- Visibility
- Object
- X/Y/Z position
- X/Y/Z rotation
- X/Y/Z scaling

For position, rotation, and scaling a stop point keyframe can be specified. This enables the user to control a keyframe in an animation, for instance the end keyframe in a position animation.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.11 Control Data Action

Binds control of a control parameter to a Data Action.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlDataAction.vip`.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.12 Control Datapool

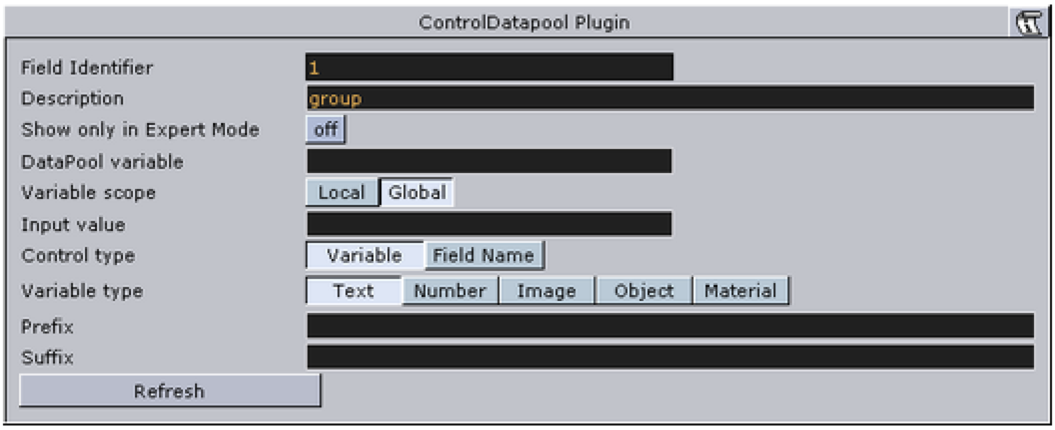
Control data pool can bind a tab field value to a data pool plug-in variable. There exist many kinds of datapool plug-ins. All data pool plug-ins, which are used in a scene, must have its variable listed in the scene plug-in named “data pool”. It is these variables which the control data pool plug-in links a tab-field to. For more information about the datapool plug-ins see the data pool documentation.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlDataPool.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ControlDatapool Plugin

Field Identifier	1
Description	group
Show only in Expert Mode	off
DataPool variable	
Variable scope	Local / Global
Input value	
Control type	Variable / Field Name
Variable type	Text / Number / Image / Object / Material
Prefix	
Suffix	

- **DataPool Variable**: The datapool variable can be controlled by the user.
- **Input Value**: Shows the current input value.
- **Control Type**: Specifies if the tab-field value should be controlled through a datapool variable or a field name.
- **Variable Type**: Allows the user to define the kind of data the referenced variable is linked to. The type selected here will decide what kind of editor will be opened in a Viz Trio client.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.13 Control DP Object

Allows copying or linking of Data Pool objects.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlDPObject.vip`.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.14 Control FeedView

Set the FeedView locator to a given value. Example input values:

```
/nyse/nasdaq/cisco
/nyse/nasdaq/intel
```

If a prefix is set it will be added to the front of the locator key, a prefix of:

```
/nyse/nasdaq
```

With an input of:

```
IBM
```

Will resolve to:

```
/nyse/nasdaq/IBM
```

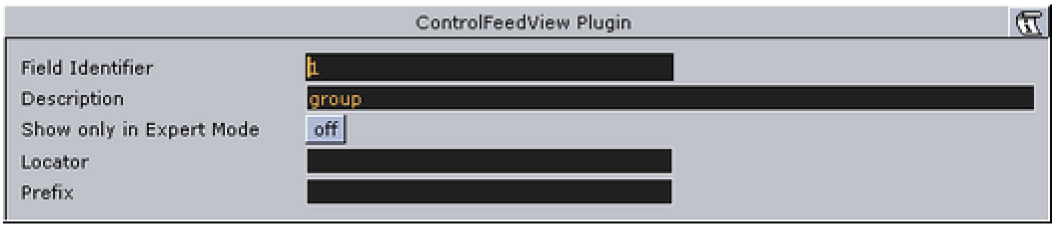
The locator field is the value as received from ControlObject. It will be trimmed of CR/LF before concatenated into a FeedView key.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlFeedview.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ControlFeedView Plugin

Field Identifier: 1
Description: group
Show only in Expert Mode: off
Locator:
Prefix:

- **Locator**: Sets the FeedView locator to a given value.
- **Prefix**: If a prefix is set, it will be added to the front of the locator key.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.15 Control Geom

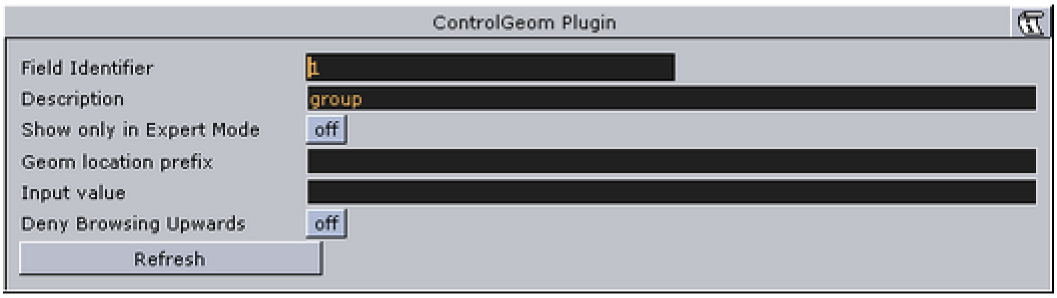
The Control Geom plug-in exposes the control of geometry objects to the user.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlGeom.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ControlGeom Plugin

Field Identifier: 1
Description: group
Show only in Expert Mode: off
Geom location prefix:
Input value:
Deny Browsing Upwards: off

- **Geom location Prefix**: Sets a path to a folder. The geometry location exposed to the control client can be a simple string, instead of a full path. This is relevant if connected to a newsroom system. For example if a user wants to edit a text and geometry on the same tab-field by just entering a text string. If for instance the user have a folder of weather symbols and the control client receives “sunny” from the external system it will load the geometry object with the path: “geom location prefix + sunny”, which then typically would be an image of a shining sun.

- **Input Value**: Shows the current input value. Enter a value to test the relation of max/min. input and scale values.
- **Deny Browsing Upwards**: When enabled, it restricts the user from browsing for objects in folders other than the selected folder and its sub folders.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.16 Control Hide in Range

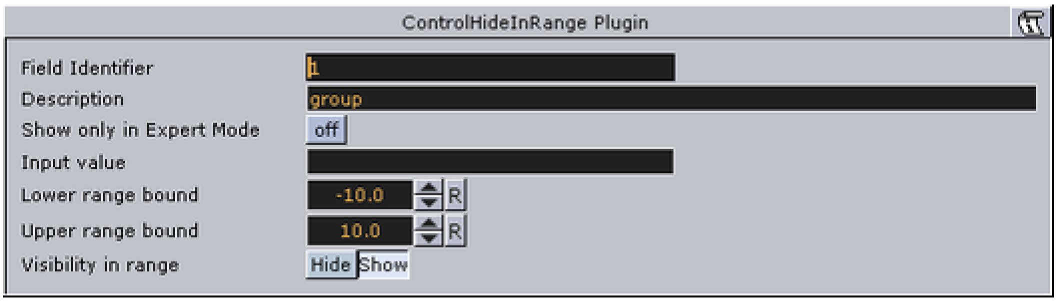
The Control Hide in Range plug-in hides or shows a container if the input value given by the user is within or outside a specified value range. The input value must be from a font container that holds a numeric value, typically with a Control Num plug-in.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlHideinrange.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ControlHideInRange Plugin

Field Identifier: 1
Description: group
Show only in Expert Mode: off
Input value:
Lower range bound: -10.0
Upper range bound: 10.0
Visibility in range: Hide Show

- **Input Value**: Shows the current input value
- **Lower Range Bound**: Sets the upper bound for the range
- **Upper Range Bound**: Sets the lower bound for the range
- **Visibility in Range**: Sets if the container should be visible or hidden when the input value is within the range.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.17 Control Hide on Empty

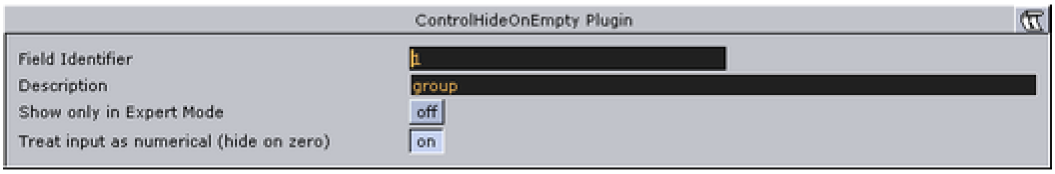
The Control Hide on Empty plug-in performs a very simple function: It hides the container it sits on if the input value is blank (text) or zero (numbers). Use this plug-in to hide the graphics elements if the user does not enter any value in the connected tab-field. Together with the Autofollow plug-in, the plug-in can be used to make intelligent scenes that hide objects that receive no value from the Viz Trio user and also rearrange those elements that have a value or text and are visible. For example when using a lower third scene with a bug and a breaking news bar, the bug and the breaking news bar should not be shown all the time. So instead of having to make several explicit variants simply make one scene and use the plug-in to hide the bug and the breaking news only if the page in Viz Trio have no value in those tab-fields.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlHideOnEmpty.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ControlHideOnEmpty Plugin

Field Identifier	1
Description	group
Show only in Expert Mode	off
Treat input as numerical (hide on zero)	on

- **Input Value**: Shows the current input value
- **Treat Input as Numerical**: When enabled this option will hide the containers when the input value is 0 or 0.0.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.18 Control Image

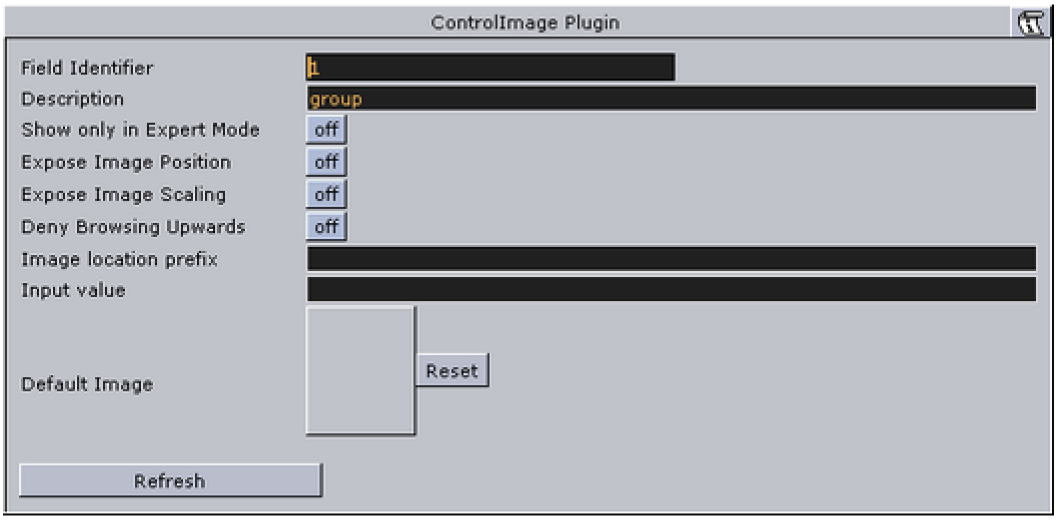
The Control Image plug-in creates an image control in the control clients. The plug-in must be placed on the container that holds the image texture.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlImage.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ControlImage Plugin

Field Identifier: 1

Description: group

Show only in Expert Mode: off

Expose Image Position: off

Expose Image Scaling: off

Deny Browsing Upwards: off

Image location prefix

Input value

Default Image

- **Expose Image Position**: Allows the user to change the position of the image texture.
- **Expose Image Scaling**: Allows the user to change the scaling of the image texture.
- **Deny Browsing Upwards**- When enabled, it restricts the user from browsing for images in folders other than the selected folder and its sub folders.
- **Image location Prefix**: If the full path to a folder is entered, the image location exposed to the control clients can be a simple string, instead of a full path. This is relevant if connected to a newsroom system. For example if a user wants edit a text and an image on the same tab-field by just entering a text string. This is relevant if a user has a folder containing flag images, and the control client receives “US” from the external system it will load the image with the path: “image location prefix + US”, which then typically would be the “Stars and stripes” (flag of the U.S.A).
- **Input Value**: Shows the current value for the plug-in. It is not necessary to set any value here. It is used for debugging purposes.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.19 Control Keyframe

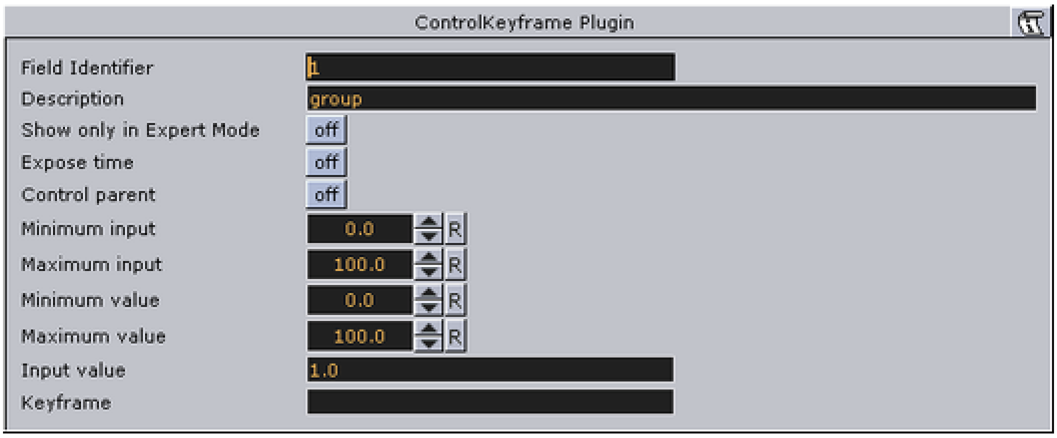
The Control Keyframe plug-in can be used to give the user control over the value of a “single value keyframe” unlike position values for instance where all three axes must be specified. Create an animation, for instance an alpha animation, and give the keyframe to control a name. Put the Control Keyframe plug-in on the object’s container and set the parameters:

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlKeyframe.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ControlKeyframe Plugin

Field Identifier	1
Description	group
Show only in Expert Mode	off
Expose time	off
Control parent	off
Minimum input	0.0
Maximum input	100.0
Minimum value	0.0
Maximum value	100.0
Input value	1.0
Keyframe	

- **Expose Time**: Enables the user to adjust the keyframe along the timeline.
- **Control Parent**: When enabled, this option allows for two Control Keyframe plug-ins to be within one container. Note that in this case the control keyframe plug-ins must be added as sub-containers of the container.
- **Minimum Input**: Sets the minimum allowed input value.
- **Maximum Input**: Sets the maximum allowed input value.
- **Minimum Value**: Sets the minimum value, which will be set relative to the minimum input value.
- **Maximum Value**: Sets the maximum value, which will be set relative to the maximum input value.
- **Input Value**: Shows the current input value. Enter a value to test the relation of max/min. input and scale values.
- **Keyframe Time**: Sets the time value of the keyframe.
- **Keyframe**: Sets the name of the keyframe that is to receive a value by the control client user.

Example for the input values: If the keyframe is an alpha keyframe which in Viz Artist can be float values from 0 to 100, and the following values are set:

- Minimum input = 1
- Maximum input = 10
- Minimum value = 10
- Maximum value = 90.

This means that if the user gives the following input:

- 1, which is the lowest input value allowed, the alpha value in Viz Artist will be set to 10.
- 4, which is somewhere in between, the relative alpha value will be 36,666... .
- 10, which is the highest value allowed, the alpha value is set to 90.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.20 Control List

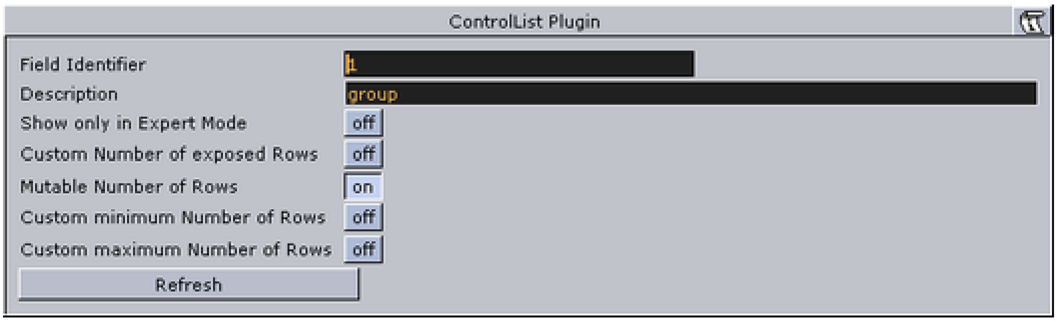
The Control List plug-in allows creation of table controls. By following some design conventions when designing the table in Viz Artist, Viz Trio will display a table editor to the user. This editor supplies a more effective way for filling tables than having one tab field for each table cell.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlList.vip`.

This section contains information on the following topics:

- Properties and Parameters
- Tables and Preview Points
- To design a table scene

Properties and Parameters



ControlList Plugin

Field Identifier: 1
Description: group
Show only in Expert Mode: off
Custom Number of exposed Rows: off
Mutable Number of Rows: on
Custom minimum Number of Rows: off
Custom maximum Number of Rows: off

- **Number of exposed rows**: Sets the default number of exposed rows for the scene when the number of rows can be changed by the operator. If the number of rows cannot be changed the number of exposed rows is fixed.

- **Mutable Number of Rows**: If set to on, the operator can add or delete rows.
- **Minimum and Maximum Number of Rows**: Sets the minimum and maximum number of rows. To the operator, the number of fields can only be added or deleted when inside the range (for example greater than > 2 and less than < 10).

Tables and Preview Points

A table scene might be hard to read if you need to display a large table with many rows (e.g. 20) and/or columns. In such cases it is not uncommon to only show a subset of the information (e.g. 5 rows), and rather animate the scene in order to show the next 5 rows and so on.

However, in such a situation, while editing table data in Viz Trio, you cannot use a basic jump to keyframe event for preview, as this will only show the keyframe that relates to the table's Field identifier (e.g. 3), and not the rows or columns of the table.

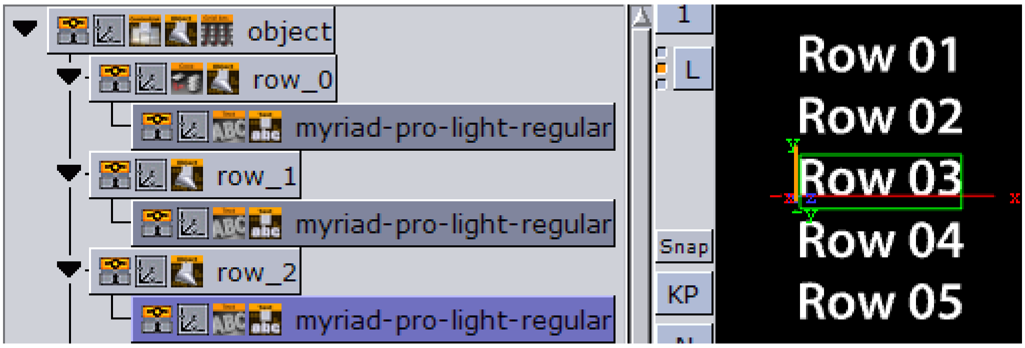
To enable operators to preview the data in the scene, it is therefore possible to add a preview point that extends to the next starting point (i.e. the next 5 rows of the table), using the table's (Control List) Field identifier in addition to the row number.

So, if the identifier of Control List is 03, you have to define tags with names 03.0, 03.5, 03.10 and 03.15 (zero based row index) in order for Viz Trio to detect the tags and jump to them on focusing the according rows in Viz Trio's Control List table editor.

- **Basic jump to keyframe**: When adding keyframe events (stop points, tags,...) and giving them the name of a control plugin identifier, Viz Trio jumps the timeline to that keyframe event when selecting the tab field with that identifier.

- **Extended jump to keyframe**: If the keyframe event name is e.g. `3_005` and there is a control list with identifier `3`, then Viz Trio jumps the timeline to that keyframe event. In this case when the sixth row of the table is focused.

To design a table scene



To design a table scene that will be properly interpreted by the Control List plug-in there are some design conventions that needs to be followed:

1. Add a group as the root container for the table. Give it a descriptive name such as “table”
2. Add the Control List plug-in to the table group container
3. Add a new group as a sub-container to the table container. This container will be the design for the rows in the table. Give it a name such as “row_1”.
4. Add a Control Object plug-in to the newly added row container.
5. As sub-containers of the row container, add text containers with backgrounds to form the design for a single table row with the desired number of columns.
   - Each of the text containers must get a Control Text plug-in and the field identifiers for each must be set from 1-n depending on how many columns the table should have.
   - The Description property in the ControlText plug-ins will be picked up by the Control List plug-in and used as column headers.
6. When the single row looks as it should, duplicate it to make the needed total number of table rows.
   - The *container* plug-in Coco can be used to automate this task.
   - Add the Coco plug-in to the row container, and set the number of copies, for example 9 (1+9=10), and click *Execute*.
7. Arrange the rows
   - The *arrange* plug-in Grid Arrange can be used to automate this task.
   - Add the Grid Arrange plug-in to the table container, and set the number of rows to the same as above and press Enter.
   - Set the row offset and press Enter.

The table element of the scene is ready and when imported into Viz Trio and read, the table tab-field will be displayed with a special table editor. When imported into Viz Template Wizard, a template with a custom table editor will be generated.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.21 Control Map

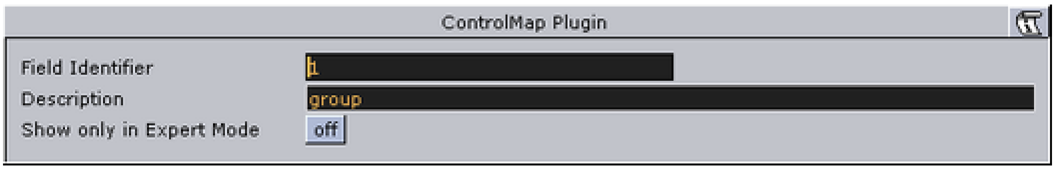
The Map plug-in binds the control of the Viz World Client plug-in XML property to a ControlObject which allows an operator to use the Viz World Client Editor to select maps.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlMap.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ControlMap Plugin

Field Identifier	1
Description	group
Show only in Expert Mode	off

This plug-in does not have any properties or parameters except those that are common to all plug-ins.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.22 Control Material

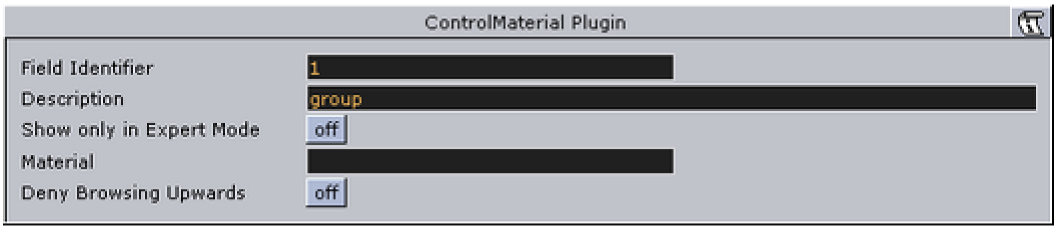
The Control Material plug-in exposes the material control to the Viz Trio user.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlMap.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ControlMaterial Plugin

Field Identifier	1
Description	group
Show only in Expert Mode	off
Material	
Deny Browsing Upwards	off

- **Material**: Sets the path to the material that should be used when the page is imported into the control client.
- **Deny Browsing Upwards**: When enabled, it restricts the user from browsing for materials in folders other than the selected folder and its sub-folders.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.23 Control Num

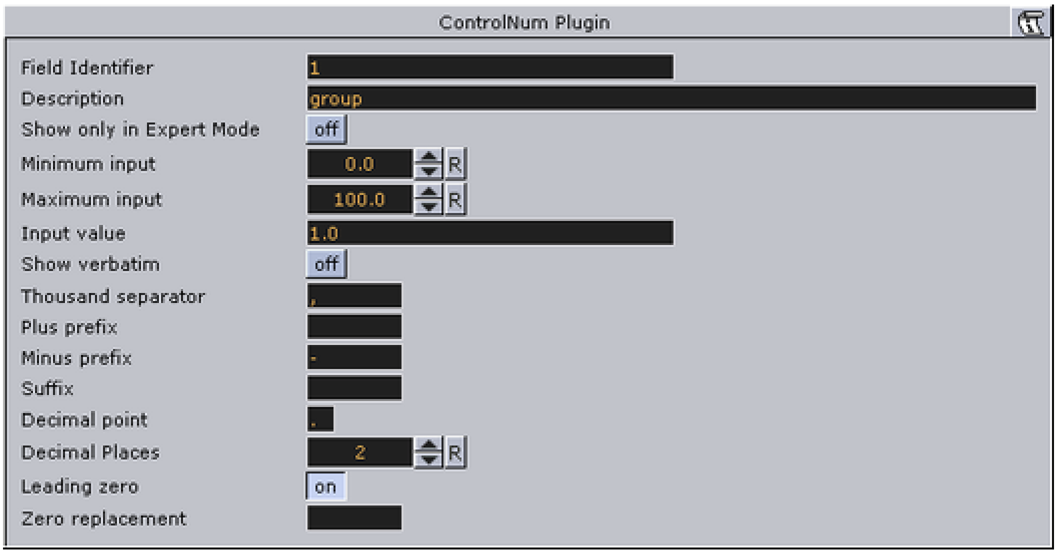
The Control Number plug-in (also known as Control Num) is used to be able to decide how a number input is to be formatted. It can be a value given by the control client user or by any external source. It should be used instead of Control Text when numbers are to be the input value. To make it work, add the plug-in to the container that holds the text object.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlNum.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ControlNum Plugin

Field Identifier	1
Description	group
Show only in Expert Mode	off
Minimum input	0.0
Maximum input	100.0
Input value	1.0
Show verbatim	off
Thousand separator	,
Plus prefix	
Minus prefix	-
Suffix	
Decimal point	.
Decimal Places	2
Leading zero	on
Zero replacement	

- **Minimum Input**: Sets the minimum allowed input value.
- **Maximum Input**: Sets the maximum allowed input value.
- **Input Value**: Shows the current input value. Enter a value to test the number formatting.
- **Show Verbatim**: When enabled, this option disables all formatting.
- **Thousand Separator**: Sets the desired thousand separator for the output.
- **Plus Prefix**: The prefix entered here will be put in front of positive values.
- **Minus Prefix**: The prefix entered here will be put in front of negative values.
- **Suffix**: The string entered here will be put after all values.
- **Decimal Point:** Defines the decimal point for the output.
- **Decimal Places**: Sets the number of decimals that will be shown by the output.
- **Leading Zero**: When enabled, a zero will be put in front of decimal values between 1 and -1.

- **Zero Replacement**: Sets the string to display instead of zero values.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.24 Control Object

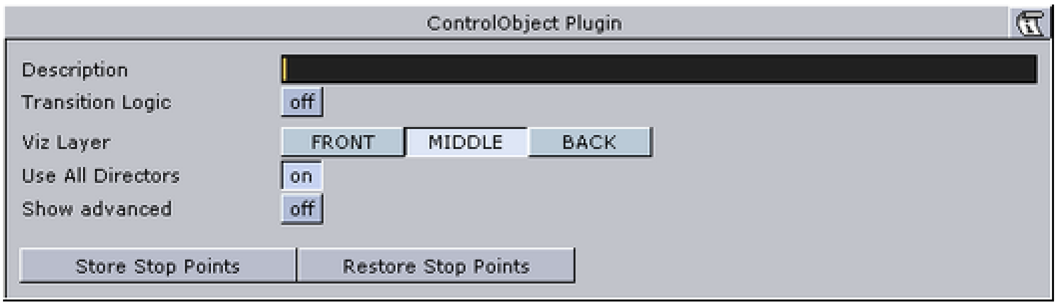
For each Viz Artist scene that is to become a Viz Trio page template, a Viz Content Pilot template or a Viz Ticker template it must have one instance of the Control Object plug-in. For Viz Ticker it is possible to have more than one Control Object plug-in. It should be on the root container for the tab objects in the scene.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlObject.vip`.

This section contains information on the following topics:

- Without Transition Logic
- With Transition Logic Enabled
- Advanced Properties

Without Transition Logic

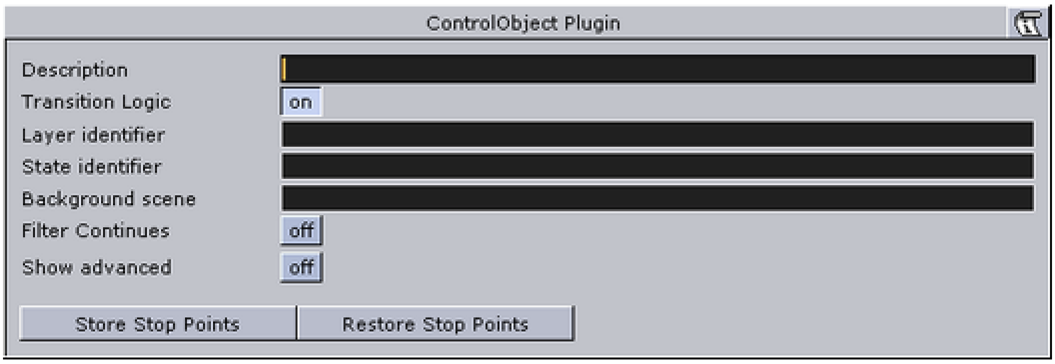


ControlObject Plugin

Description
Transition Logic: off
Viz Layer: FRONT MIDDLE BACK
Use All Directors: on
Show advanced: off

In the simple scene based mode, the plug-in should be put on the root container for the page tab-field containers. A description which will be used as template description in the control clients should also be entered. To load the graphics in another rendering layer than the Middle (default), choose Front or Back. By using different layers up to three elements can be on screen simultaneously without having to design a Transition Logic setup. Be aware that objects in the Back layer of course will be rendered behind objects in the Middle and Front layer and so on, so the scenes must then be designed with that in mind.

With Transition Logic Enabled



With Transition Logic enabled, logical layer and state can be defined for the scene. These will be used by the control clients to find out which transitions and animations to trigger when a page is put on-air or off-air. These transitions and layers must all be defined in the background scene. A background scene name can also be set if a background scene other than the default (named default) is used.

Advanced Properties

By clicking the Show Advanced button in the Control Object editor, a channel property is displayed as well as a text box for manual field definitions. Most of these parameters are used for system integration purposes and are rarely relevant when doing normal scene design for Viz Trio, Viz Content Pilot and Viz Ticker.

- **Channel**: Sets the name of the default playout channel for the template.
- **Manual field definitions**: In the text box, custom fields can be defined by using a fixed syntax: Each entry is a line of text consisting of colon (':') separated fields. The fields for each entry are:

Table 26: Advanced Properties

Field description	Example
Field identifier/key	votes
Property location	$top$label*GEOM*TEXT
Type	integer
Minimum value	0
Maximum value	100
Maximum number of characters	8
Field description	Percentage of votes
Auxiliary field	

- **Remap fields**: For use in scenes with nested control object plug-ins. This feature is mainly for advanced integration use. When turned on, Control Object will give the control clients the list of entries in map, rather than the list of entries registered by control plug-ins below the Control Object in the scene

tree, and manually entered entries from entries. When actions are performed on fields, the Control Object will use map to rename the field identifier before dispatching the action.
- **Store Stop Points**: This mechanism can be used to work around the default behavior of Viz Artist which discards stop-points when sub-trees are merged into objects. It stores information about the stop-points of directors used by the sub- tree in a way that will survive when merging the sub-tree into an object stored in the object pool.
- **Restore Stop Points**: Recreates the stop-points used by the director of an object that has been loaded from the object pool. This uses the information previously stored by Store Stop Points. When used in conjunction with the Toggle plug-in, it will operate on the object's animation director.

See Also

- Common Control Plug-in Properties
- Control Object
- Toggle
- Transition Logic

12.4.25 Control Omo

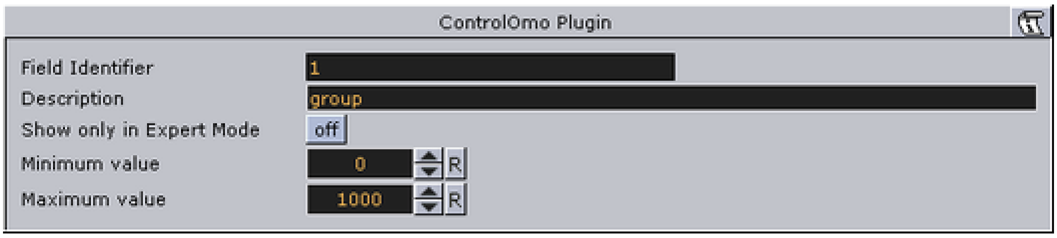
The Control Object moving (Omo) plug-in in Viz Artist gives the possibility to add a group of containers and reveal one at the time. This is done by adding an Omo plug-in on the root container and in its editor specify which of the sub containers to show. The Omo value can be made accessible for the control client by adding the Control Omo plug-in on the same container as the Omo plug-in.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlOmo.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ControlOmo Plugin

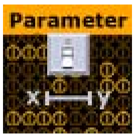
Field Identifier	1
Description	group
Show only in Expert Mode	off
Minimum value	0
Maximum value	1000

- **Minimum Value**: Sets the minimum input value.
- **Maximum Value**: Sets the maximum input value.

See Also

- Common Control Plug-in Properties
- Control Object
- Omo

12.4.26 Control Parameter



The Control Parameter plug-in works with function plug-ins, and is a very flexible plug-in that can be used to control properties that are not exposed in a specific plug-in, and requires some knowledge in the Viz Artist command language. The parameter to control must be identified by its command path relative to the container the plugin is located at. To find the correct command path, open the Show Commands window (button in the lower left corner in Viz Artist) and make changes to the parameter to control. The command path will then be shown among the commands. For instance if the value to be changed is the alpha value in an alpha editor the command window will show the following:

```
receive <-1 SCENE*noname*TREE*#805*ALPHA*ALPHA SET 99.0>
```

#805 is the internal name for the container and the part needed to expose the alpha parameter (ALPHA * ALPHA). The SET part is added by the plug-in. To expose the alpha parameter for a container write ALPHA * ALPHA in the Parameter section in the plug-in and then set the correct data type which in this case is “Float”.

Another example would be to use Control Parameter to expose and control additional values like controlling both Pie Max and Max Value. These parameters are dependent on each other, but both values are not exposed unless the Control Parameter is used. Adjusting the Pie Max value gives the following:

```
receive <5679 SCENE*noname*TREE*#378*FUNCTION*PieValues*DATA GET>
```

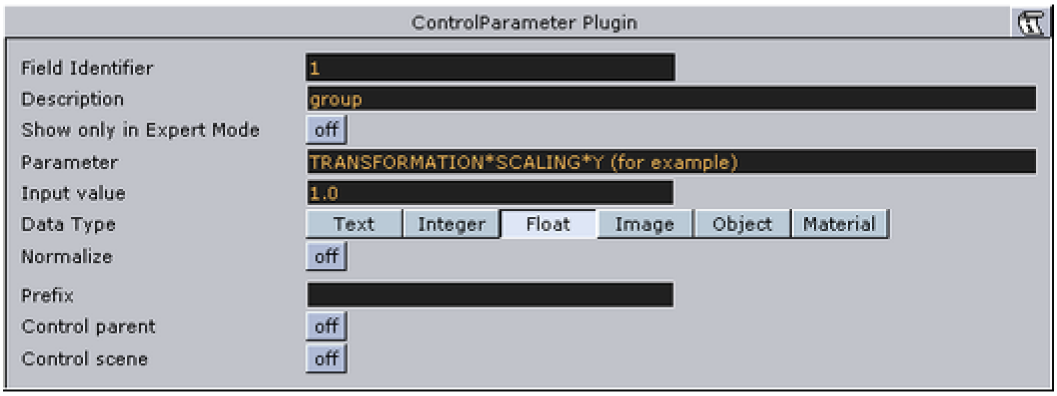
In this case the FUNCTION*PieValues*DATA is the parameter that will expose the object property Pie Max such that it can be controlled through a template.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlParameter.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ControlParameter Plugin

Field Identifier	1
Description	group
Show only in Expert Mode	off
Parameter	TRANSFORMATION*SCALING*Y (for example)
Input value	1.0
Data Type	Text | Integer | Float | Image | Object | Material
Normalize	off
Prefix	
Control parent	off
Control scene	off

- **Parameter**: Sets the parameter path to the property to enable control for.
- **Input Value**: Shows the current input value
- **Data Type**: Sets correct data type depending on the type of parameter to enable control for.
- **Normalize**: Enables normalization options, to normalize the input range for the Viz Trio user.
- **Minimum Input**: Sets the minimum allowed input value.
- **Maximum Input**: Sets the maximum allowed input value.
- **Minimum Value**: Sets the minimum value, which will be set relative to the minimum input value.
- **Maximum Value**: Sets the maximum value, which will be set relative to the maximum input value.
- **Prefix**: Adds a prefix to all values that are sent to the renderer. This can be used to show specific text or it can be used to add a path to an object pool for instance.
- **Control Parent**: When enabled, the parent container is controlled instead.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.27 Control Pie

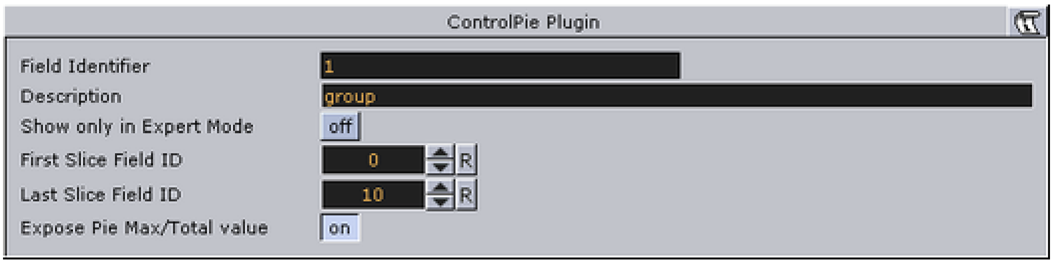
The Control Pie plug-in is able to bind a tab-field to values from PieSlice plug-ins.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlPie.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To build a pie chart

Properties and Parameters

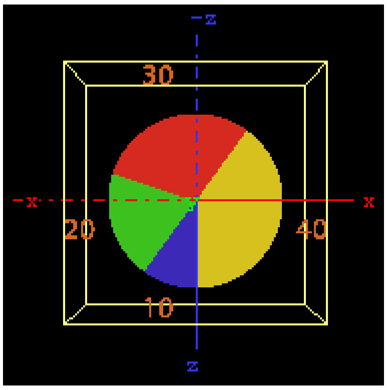


ControlPie Plugin

Field Identifier: 1
Description: group
Show only in Expert Mode: off
First Slice Field ID: 0
Last Slice Field ID: 10
Expose Pie Max/Total value: on

- **First slice ID**: Sets the tab-field number for the first slice (the first container below the group that holds the ControlPie Plugin).
- **Last Slice ID**: Sets the tab-field number for the last slice (the difference between Last and First should correspond to the total number of slices in the pie chart).
- **Expose Pie Max/Total Value**: When enabled, the Viz Trio user can set Max value for a pie slice and Max/Total value for the whole pie chart.

To build a pie chart



1. Add a group container and name it, for instance “pie”. Add the presenter plug-in Pie Values and the control plug-in Control Pie to the group.
2. Add a Cylinder object as a sub-container to the group. Add material and the presenter plug-in Pie Slice to the new container. Name the container “slice” or similar.
   - The Cylinder object is not visible.
3. Open the Pie Values editor and increase the Value0 property a bit. The cylinder becomes visible and its size changes.
4. Open the Cylinder editor and set Centre to “Top”.
5. Open the Transformation Editor for the pie group, and rotate the pie group around 90 degrees on the X-axis to make the cylinder top face the camera.
6. Add a font as sub-container to the slice container, and add a material to make it visible against the background.
7. Open the text editor and set the horizontal justification to Centre. Open the Transformation Editor and rotate the font around the X-axis to -90 degrees and set Y position to 1.0.
8. Scale the font down a bit so it fits over the slice (for example *0.2*).

9. On the slice container, open the PieSlice editor and set Control Text Values to a data type, e.g. None, Integer, Float or Formatted. By choosing Integer or Float the slice text will change, displaying the value for the slice.
10. In the pie slice editor, adjust the Text Offset to position the value label relative to the centre of the pie (for example *70.0*).
11. A single pie slice should be ready. Make the number of slices needed by making copies of the slice container and change their material so they can be differentiated from each other.
12. Set the different values in the PieValues editor to test.

See Also

- Common Control Plug-in Properties
- Control Object
- Pie Slice
- Pie Values

12.4.28 Control Scaling

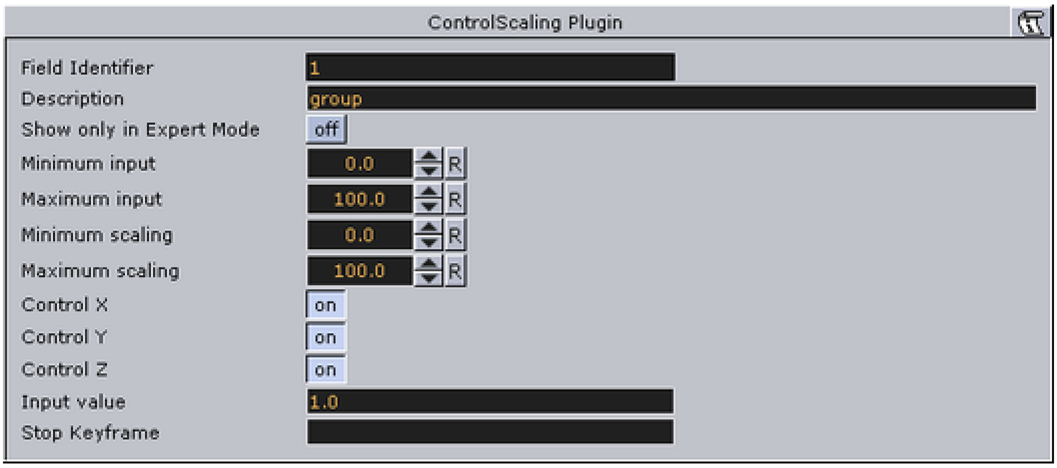
The Control Scaling plug-in enables the Viz Trio user to edit transformation scaling values for objects. Normalization between input and output values can be set by parameters.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlScaling.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ControlScaling Plugin

Field Identifier	1
Description	group
Show only in Expert Mode	off
Minimum input	0.0
Maximum input	100.0
Minimum scaling	0.0
Maximum scaling	100.0
Control X	on
Control Y	on
Control Z	on
Input value	1.0
Stop Keyframe	

- **Minimum & Maximum Input**: Sets minimum and maximum allowed input values.
- **Minimum Scaling**: Sets the minimum transformation value which will be set relative to the minimum input value.
- **Maximum Scaling**: Sets the maximum transformation value which will be set relative to the maximum input value.
- **Control X, Y, Z**- When enabled, it controls the X, Y, and Z-axis independently. The Viz Template Wizard TripletEditor component has three onChange events available that accommodates for these parameters.
- **Input Value**: Shows the current input value. Enter a value to test the relation of max/min. input and scale values.
- **Stop Keyframe**: If the scaled object is animated, the scaling values entered must be put into the end keyframe for the animation. For instance if a user is to create a bar graph that animates from zero to a value specified, then this is the case. Name the end keyframe in the Viz Artist stage and enter that name in the “stop keyframe” field in the control scaling plug-in.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.29 Control Sign Container

The Control Sign Container value can display one of three containers based on the value of a numerical input. The container that holds the plug-in must have three sub-containers.

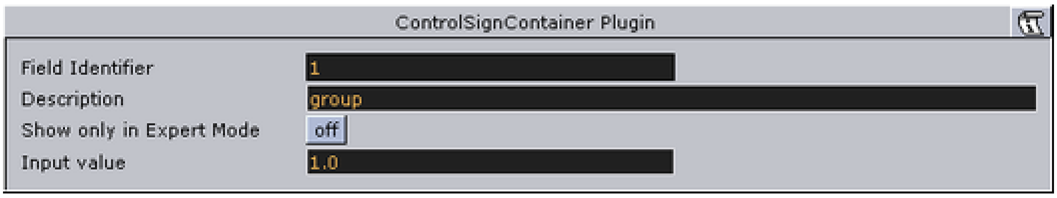
- A negative value will trigger display of the first container.
- A positive value will trigger display of the second container.
- A zero value will trigger display of the third container.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlSigncontainer.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ControlSignContainer Plugin

Field Identifier: 1
Description: group
Show only in Expert Mode: off
Input value: 1.0

- **Input Value**: Shows the current input value. Enter a value here to test that the correct sub-container is displayed.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.30 Control SoftClip

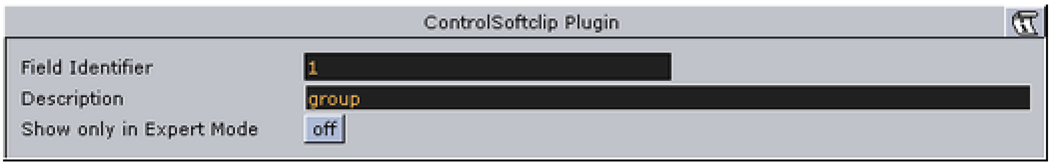
The Control SoftClip plug-in allows the user to control and browse for video clips which are played with the Soft Clip plug-in in Viz Artist. To make it work, simply add the control plug-in on the same container as the Softclip plug-in. Further, specify a Field Identifier and a Description.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlSoftclip.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ControlSoftclip Plugin

Field Identifier: 1
Description: group
Show only in Expert Mode: off

This plug-in does not have any properties or parameters except those that are common to all plug-ins.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.31 Control Stoppoint

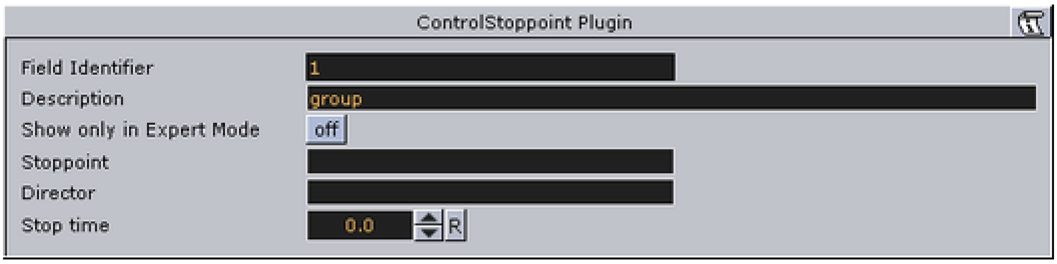
Control Stop point enables control of the time value on a stop point. This can typically be used for setting the length of an animation.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlStop.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ControlStoppoint Plugin

Field Identifier	1
Description	group
Show only in Expert Mode	off
Stoppoint	
Director	
Stop time	0.0

- **Stoppoint**: Sets the name of the stop point to control.
- **Director**: Sets the name of the animation director where the stop point is placed.
- **Stop Time**: Sets the stop time in seconds. This is the value that is set by the Trio operator.

Note: This Control Plug-in does not work with transition logic based scenes.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.32 Control Text

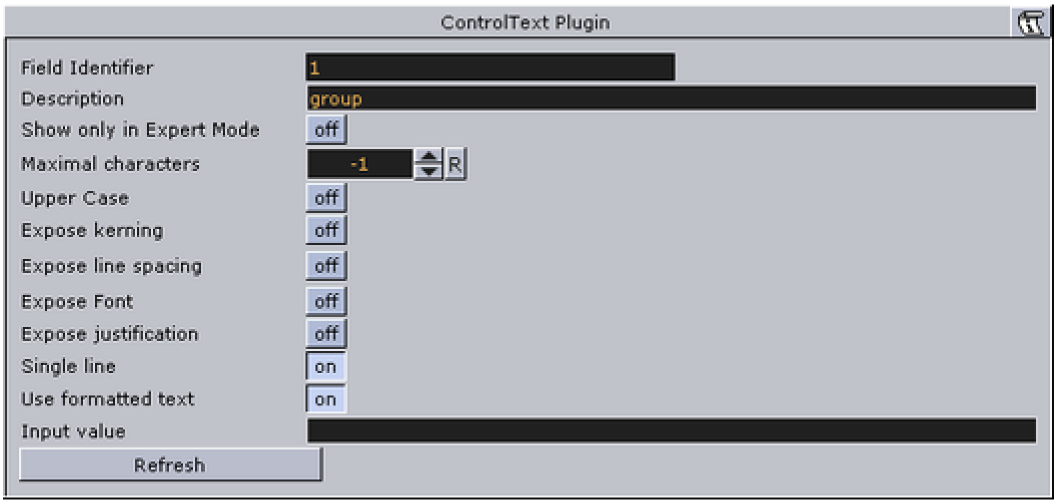
The Control text plug-in enables editing of a text element. Put the plug-in on the text container and set the parameters. The plug-in also works together with all the Text Effects plug-ins.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlText.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ControlText Plugin

Field Identifier	1
Description	group
Show only in Expert Mode	off
Maximal characters	-1
Upper Case	off
Expose kerning	off
Expose line spacing	off
Expose Font	off
Expose justification	off
Single line	on
Use formatted text	on
Input value	

- **Maximal Characters**: Sets the maximum number of characters which can be typed into the text field.
- **Upper Case**: When enabled, lower case input is converted to upper case.
- **Expose Kerning**: When enabled, kerning can be adjusted.
  - Minimum Kerning: Sets the minimum kerning that can be set.
  - Maximum Kerning: Sets the maximum kerning that can be set.
- **Expose Line Spacing**: When enabled, a minimum and maximum value can be specified.
  - Minimum Line Spacing –Sets the minimum line spacing that can be set.
  - Maximum Line Spacing: Sets the maximum line spacing that can be set.
- **Single Line**: Displays the text using one single line by disabling text wrapping.
- **Use Formatted Text**: Enables the use of character specific formatting, i.e. kerning, position, rotation, scaling, alpha, and color.
  - Viz Trio: The user can enter a desired formatting mode by pressing the assigned shortcut key. The Viz Trio user can change the formatting of one or more characters by holding the ALT key down, and shift mode by

toggling the right and left arrow keys. See the Viz *Trio User's Guide* for more information on *Character Formatting*.

- Viz Content Pilot: Not supported.

Note: Formatted text cannot be used together with the TfxWrite text effects plug-in!

- **Input Value**: Shows the current value for the plug-in. It is not necessary to set any value here. It is used for debugging purposes.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.33 Control Targa Alpha

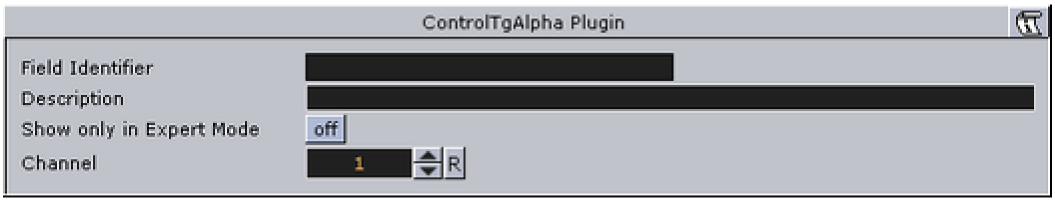
The Control Targa Alpha plug-in binds control of targa video alpha properties to a Control Object.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlTgalpha.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ControlTgAlpha Plugin

Field Identifier	
Description	
Show only in Expert Mode	off
Channel	1

- **Channel**: Sets which channel to control.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.34 Control Targa Clip

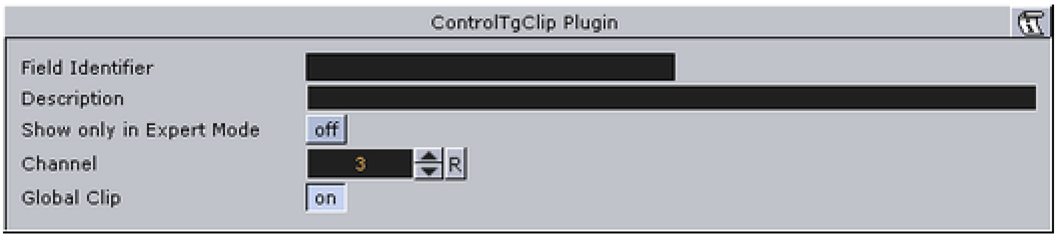
The Control Targa Clip plug-in exposes control over a Targa clip channel. The Viz Trio user will get a Windows file browser where video clips to play can be browsed for.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlTgclip.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ControlTgClip Plugin

Field Identifier

Description

Show only in Expert Mode: off

Channel: 3

Global Clip: on

- **Global Clip**: When enabled, the clip specified will not be added to the stage but will play globally in the scene. If deselected two more fields appear; Director and Device keyframe. The clip will then be loaded at the named keyframe in the specified director and will run when the director is triggered.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.35 Control Video

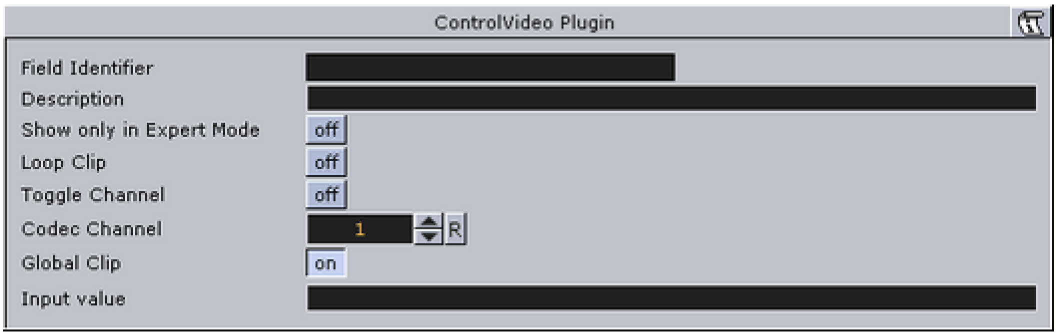
The Control Video plug-in exposes control over a video codec channel. The user will get an editor (e.g. Viz Object Store) where video clip files from Viz Video Hub can be searched for.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlVideo.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



Field Identifier
Description
Show only in Expert Mode: off
Loop Clip: off
Toggle Channel: off
Codec Channel: 1
Global Clip: on
Input value

- **Loop**: Enables or disables continuous looping for a video clip in graphics.
- **Toggle Channel**: Enables a transition logic scene to toggle between two clips. If toggle is enabled a new clip will be loaded creating a smooth transition between the two clips.
- **Codec Channel**: Sets the codec channel for the video board in use.
  - **1**: Is normally used as the default codec channel for full screen video.
  - **2**: Is normally used as the default codec channel for video in graphics.
- **Global Clip**: When enabled, the clip specified will not be added to the stage but will play globally in the scene.
- **Director**: Sets the name of the Director.
- **Clip Keyframe**: Sets the name of the keyframe.

Disabling Global clip will enable the Director and Clip Keyframe fields. Setting Director and Clip Keyframe will load the clip at the named keyframe in the specified director and will run when the director is triggered.

Viz Engine is configured to use 4 video channels, where 2 are dedicated to live feeds, and the remaining 2 are used for video clips and graphics with embedded video clips.

These are the default options:

- A full screen video in codec channel 1, and graphics with video in codec channel 2. Full screen video is by default played out in the BACK layer using a default scene. Default scenes are pre-configured by Vizrt. Graphics are played out in the MIDDLE or FRONT layer.
- Two videos in graphics without the use of full screen video.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.36 Control World

The ControlWoC plugin allows you to expose a set of properties and parameters to the operator. The control is a replacement for the Control Map plugin with much more options and on-the-fly feedback from Viz.

Note: The two controls are not compatible. The new control with not work with Viz Content Pilot, and is only a part of Viz 3.3 and above.

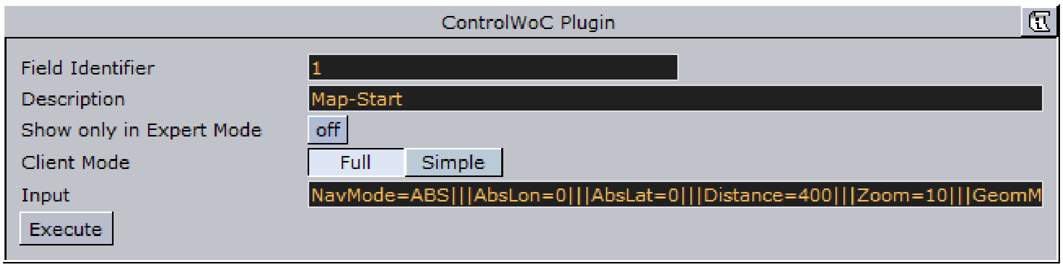
The control can expose different fields based on the container it resides on. When tabbing to the control (in a navigator scene) the camera will jump to map location and all feedback (exact camera position) will be immediate.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ControlCME.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ControlWoC Plugin

Field Identifier: 1
Description: Map-Start
Show only in Expert Mode: off
Client Mode: Full | Simple
Input: NavMode=ABS|||AbsLon=0|||AbsLat=0|||Distance=400|||Zoom=10|||GeomM

- **Client Mode**: Sets the client mode to Full or Simple. This will, when selecting a map, open the full or simple map client in the control application.
- **Input**: Sets the exposed properties and parameters. The field is based on the input from the NavFinder plugin; however, you may also manually change this field.

See Also

- Common Control Plug-in Properties
- Control Object

12.4.37 Placeholder

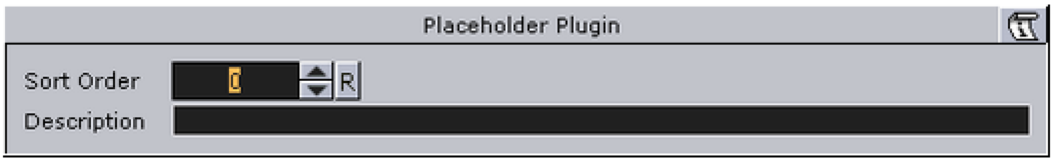
The Placeholder plug-in is used to mark placeholder containers when designing library objects for Viz Trio compositing. A placeholder container will be available as a parent to other compositing objects in the scene tree.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Placeholder.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



Placeholder Plugin

Sort Order 0

Description

- **Sort Order**: Integral sorting order identifier (lower number is sorted first).
- **Description**: A description of the tab field or editable object. This will be used as a description for the elements when used in the control clients.

See Also

- Common Control Plug-in Properties
- Control Object

12.5 Default

The following container plugins are housed in the Default folder:

- Match It
- Object Zoom
- Texture Movie
- Trio Scroll Element
- VCF Parameter

12.5.1 Match It

Sets the axis center (i.e. center for rotations) to the camera and moves the object away from the camera by the specified amount of units. By setting an arbitrary camera distance and initializing the plugin, the container with the Match It plugin will be positioned in the specified distance and any rotation will be done with respect to the camera position. The container’s axis center parameter can also be edited for a more flexible configuration.

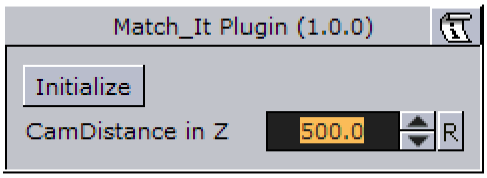
Note: Keep in mind that you have to re-initialize the plugin if the camera is moved.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Match_It.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



CamDistance in Z 500.0

- Initialize: Resets the pivot to the camera and places the object 'CamDistance in Z' units away.
- CamDistance in Z: The distance from the camera on the z axis.

12.5.2 Object Zoom

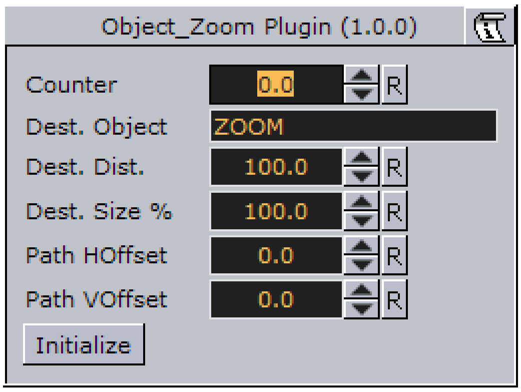
Object Zoom lets an object follow the camera and match its orientation. This creates an effect as if the camera would smoothly zoom to the object.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Object_zoom.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To zoom an object towards the camera

Properties and Parameters



Object_Zoom Plugin (1.0.0)

Counter 0.0

Dest. Object ZOOM

Dest. Dist. 100.0

Dest. Size % 100.0

Path HOffset 0.0

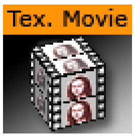
Path VOffset 0.0

- **Counter**: The speed of the zoom operation. Use values from 1-1000.
- **Dest. Object**: The name of the container that will be affected by this plugin.
- **Dest. Dist.**: The distance from the camera at which the object will stop.
- **Dest. Size %**: The scaling of the object.
- **Path HOffset**: Horizontal offset in camera space.
- **Path VOffset**: Vertical offset in camera space.
- **Initialize (button)**: Starts the zoom.

To zoom an object towards the camera

1. Add the Object Zoom plugin to a container.
2. Set the **Dest. Object** parameter to match the containers name.
3. Set the **counter** to at least 1.
4. Click **Initialize**.
5. Go to **Views** and move the camera to see the container zooming towards the camera.

12.5.3 Texture Movie



The Texture movie plug-in gives the user the possibility to create an image animation sequence, using a method much like the one used for making a cartoon film. The basic input for the plug-in is a single image consisting of many tiled, equally sized squares, set up in a matrix. Each square of the image is made up to be a snapshot of an animation sequence, just like each picture frame is on a normal movie film. The image must be made up in advance with the aid of a picture editing program (e.g. Adobe Photoshop).

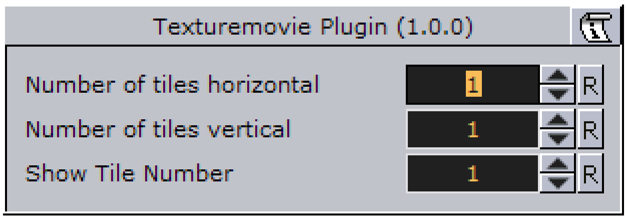
In the property editor you tell the Texture Movie plug-in how many tiled squares there are in the X- and Y-axis. The plug-in is then able to show the tiles of the image one by one. By animating this, a “film like” sequence is created. The plug-in has only a small influence on the rendering performance.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Texturemovie.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a Texture Movie

Properties and Parameters



Texturemovie Plugin (1.0.0)

Number of tiles horizontal 1
Number of tiles vertical 1
Show Tile Number 1

- **Number of tiles horizontal**: The number of picture rows.
- **Number of tiles vertical**: The number of picture columns.
- **Show Tile Number**: The number of the picture to show. The images are numbered from left to right and then top to bottom.

To create a Texture Movie

1. Add the texture containing the image sequence on a container.
2. Add the Texture Movie plugin.
3. Set the tiles parameters.
4. Use **Show Tile Number** to display a certain image from the sequence.

See Also

- Temo

12.5.4 Trio Scroll Element

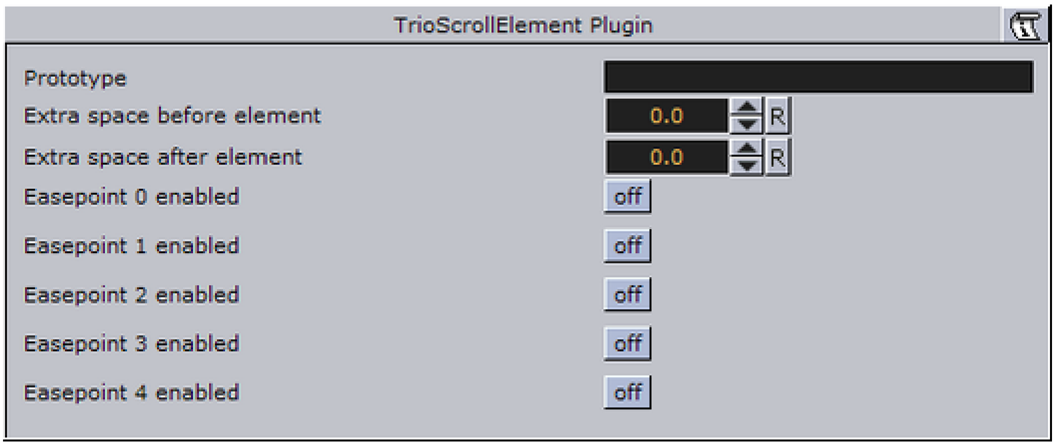
The Trio Scroll Element plug-in is used for data storage by the Trio Scroll, which is a geometry plug-in. It is automatically added to each base element of a Viz Trio scroll scene.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\TrioScrollElement.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



TrioScrollElement Plugin

Prototype	
Extra space before element	0.0
Extra space after element	0.0
Easepoint 0 enabled	off
Easepoint 1 enabled	off
Easepoint 2 enabled	off
Easepoint 3 enabled	off
Easepoint 4 enabled	off

- **Prototype**: Sets the path and the name of the scene.
- **Extra space before element**: Sets the space before the element.
- **Extra space after element**: Sets the space after the element.
- **Easepoint 0-4 enabled**: Enables the easepoint(s).
  - **Identifier**: Sets the identifier for the easepoint.
  - **Slow down length (EaseOut)**: Sets the size in pixels that will be used to ease out the scroll element.
  - **Speed up length (EaseIn)**: Sets the size in pixels that will be used to ease in the scroll element.
  - **Pause time in seconds, or -1 (indefinitely)**: Sets the time in seconds for how long the scroll should wait before continuing the scroll. Note that this will affect the total time the scroll is scheduled to use. See
  - **Element alignment (0.5 for center)**: Sets the alignment of the element relative to the height of the element. 0.5 (equal to 50%) is the default value and represents the middle of the element's bounding box area.
  - **Scroll area alignment (0.5 for center)**: Sets the scroll alignment for the element in the scroll area. 0.5 (equal to 50%) is the default value and represents the middle of the scroll's bounding box area.

See Also

- *Create New Scroll* in the *Viz Trio User's Guide*.
- Trio Scroll

12.5.5 VCF Parameter

The Virtual Camera Flight (VCF) Parameter container plug-in works in conjunction with the VCF scene plug-in. They allow you to create a seamless interpolated transition from a VCF to a real camera: and vice versa. This is only a relevant function to set up, if you have purchased the virtual set expansion components.

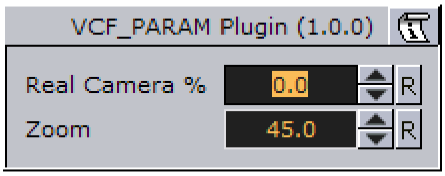
You must have a real camera with data tracking enabled, which is set in remote mode in the camera editor. In case you have several tracked cameras, the virtual camera will interpolate its position to the real camera that is selected on air.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\VCF_Param.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To use the virtual camera flight parameter plug-in

Properties and Parameters


VCF_PARAM Plugin (1.0.0)

Real Camera % 0.0

Zoom 45.0

- **Real Camera %**: Defines the percentage the real camera influences the position of the virtual camera.
- **Zoom**: Sets the actual zoom (FOV-Y) value of the virtual camera

To use the virtual camera flight parameter plug-in

1. Add the VCF scene plug-in to the scene setup plug-ins drop zone.
2. Create a new group in your scene and add two new containers under it.
   - These are to be the objects that will define the virtual camera flight.
   - One will define the position, the other the direction.
3. Name the containers according to the names you entered in the VCF scene plug-in (e.g. T_POS and T_ROT).
4. Click **Initialize** to finish.
5. Add the VCF Parameter container plug-in to the container that holds the position object.
6. Animate your virtual camera flight using the two objects to define position, and direction. Do **not** switch the whole container invisible or else the animation will not run. You may switch the objects to be invisible, at any time.
7. Animate the **Real Camera %** parameter in what frame you want to have the real camera at 0% visibility and when at 100% visibility (the longer the animation is, the smoother the interpolation will be: 100 F should be good).
   - If you start with a virtual camera the first keyframe (0) should have 0% of the real camera, and the last should be 100% of the real camera.
   - The virtual camera will generate a full key (masking the FG), if you choose virtual set in the scene setup: background.
8. Make sure your FG object/talent are not seen while switching to the real camera. They should appear in the frame after the switch.

Note: It is not possible to animate the roll of the camera.

See Also

- VCF
- Creating Animations

12.6 Global

The following container plugins are housed in the Global folder:

- Alpha
- Audio
- Clipper
- Expert
- Extrude
- Glow
- HDR
- Key
- Look-At
- Mask Source and Mask Target
- Lighting
- Z-Sort
- Projector Source and Projector Target
- Script
- Shadow Caster and Shadow Receiver
- Synchronized Properties
- Video Clip
- Window Mask

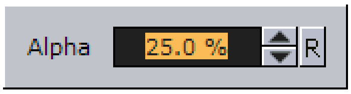
12.6.1 Alpha

The Alpha plug-in adds an alpha channel to the container. The alpha channel defines the degree of transparency for the container and its sub containers. This feature can be used to easily fade in or out a complete part of the scene-tree.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



Alpha 25.0 %

- **Alpha**: Sets the alpha value for the container.

12.6.2 Audio

The Audio plug-in allows a designer to configure audio channels. The Audio plug-in can be applied to any container.

The Audio plug-in has four different audio clip mixing modes; Default, FX and Manual.

This section contains information on the following topics and procedures:

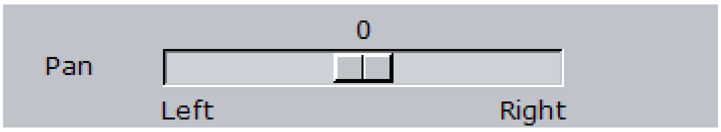
- Default
- FX
- Manual
- To configure audio channels
- To test audio channel setup

Default

The Default mode uses the Channel settings stored in the clip. For example FrontLeft and FrontRight are mixed to the aliases FRONT_LEFT, FRONT_RIGHT and so on.

If Viz Engine is configured with fewer channels, Viz Engine mixes the remaining clip channels according to the channel geometry. For example if Viz Engine only has stereo configured, but the clip used has 7.1 surround sound, Viz Engine mixes LeftBack and LeftMiddle to FRONT_LEFT and so on.

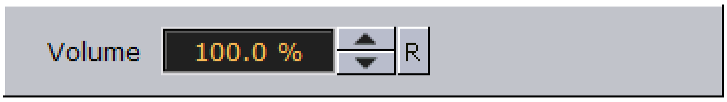
Figure 171: Pan slider, Audio plug-in



Pan: 0

The Pan slider uses audio aliases when adjusting the volume of the left and right speakers. This is valid for multichannel configurations as well. If the slider is moved to the right in a 7.1 configuration LeftBack, LeftMiddle and LeftFront will be muted.

Figure 172: Volume, Audio plug-in



Volume: 100.0 %

The Volume field controls the overall audio volume of all clip channels.

FX

In the FX mixing mode Viz Engine mixes the clip in relation to the listener position which can be defined in the fields X and Y. Viz Engine takes the channel geometry into account.

Manual

In the Manual mixing mode it is possible to manually enter the Out Channel alias for every clip channel. The mask can hold more than one channel. This enables you to mix every clip channel to an arbitrary amount of out channels.

The syntax for the string is as follows:

```
    CHANNEL_ALIAS%VOLUME+CHANNEL_ALIAS%VOLUME…. .
```

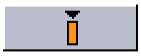
or

```
    [CHANNEL_ALIAS%VOLOUME]+
```

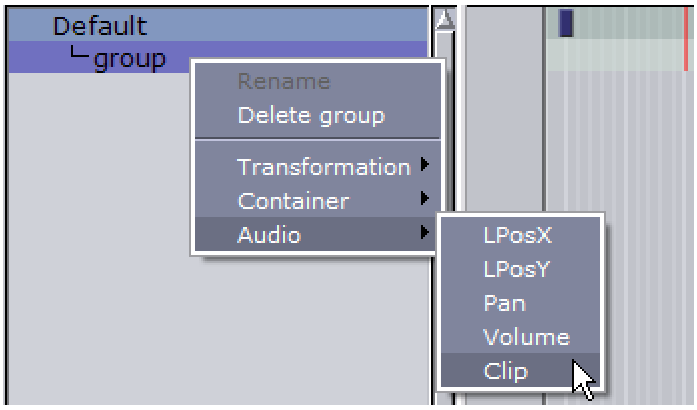
Example: FrontRight%50+EnglishLeft%20

To configure audio channels

1. Add a group container to the Scene tree.



2. Click the **Set Key** button to add a key frame for the container.
3. Add the Audio plug-in to the new group container.
4. Click the **Stage** (not Server/Stage) button, and select the **Dopesheet/Spline** view (CTRL+4).



5. Right-click the group node, and select **Audio** -> **Clip** from the context menu.
6. Click the Server button to return to the Server view.
7. Click the Audio plug-in icon to open the Audio plug-in editor.
8. Click the Manual button to enable manual configuration of Viz Engine’s audio channels.
9. Enter the following channel configurations:
   - Out Channel 0: FrontLeft
   - Out Channel 1: FrontRight
   - Out Channel 2: EnglishLeft
   - Out Channel 3: EnglishRight
   - Out Channel 4: FrenchLeft
   - Out Channel 5: FrenchRight
   - Out Channel 6: GermanLeft

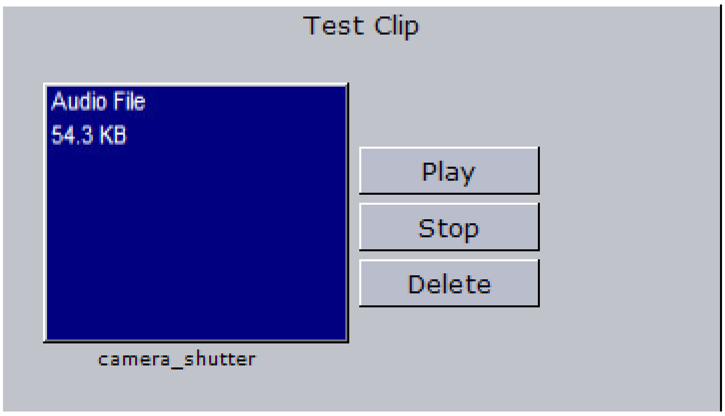
- Out Channel 7: GermanRight

Note: You can use aliases as configured in the Channels tab.

10. Save the scene.

Depending on the settings in the channel configuration a clip will now play the different languages.

To test audio channel setup



1. Create a new scene
2. Add a group container to the scene tree
3. Add the Audio plug-in to the group container
4. Open the Audio plug-in editor, and add an audio clip to the *Test Clip* drop-zone, and click Play

Tip: Always have a set of test clips that will provide audio for the different channel setups.

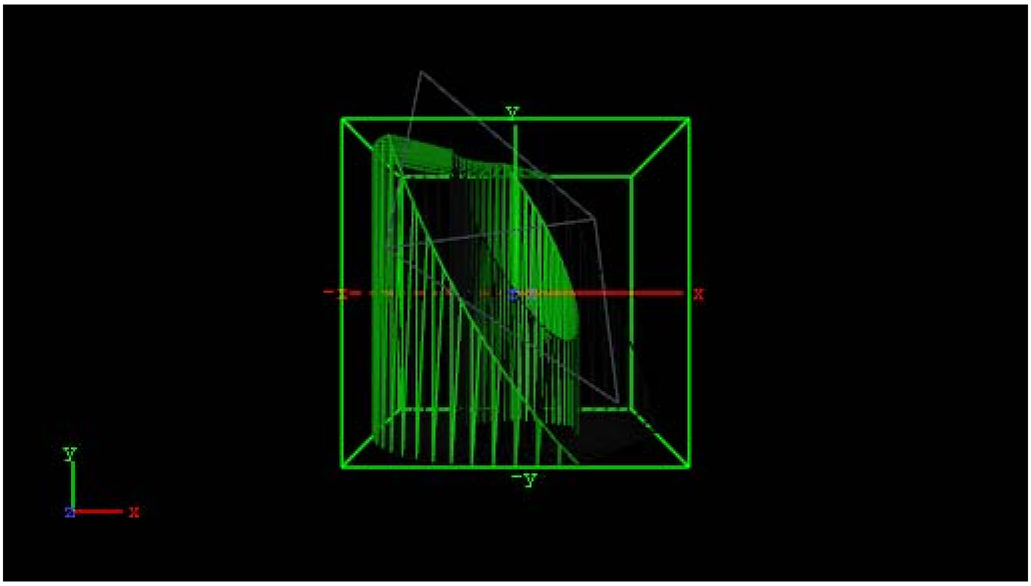
See Also

- Audio Settings
- *Audio Reference* in the *Viz Engine 3 Manual*.

12.6.3 Clipper

Clipper

The Clipper plug-in is an alternative to the mask function. Objects with the clipper plug-in attached that are placed behind the clipper plane and will be masked/clipped. If an object is only partially behind a clipper plane, only the part of the object that are behind the plane will be clipped out, since the clipping is done in true 3D space. The function uses the OpenGL clipping planes. Up to 6 of them may be used. To define a clipper plane use the scene setup.



All the containers that you want to be affected by the clipping plane, you must have the clipper plugin applied and thereafter in the editor specify which of the clipping plane is to mask out/clip the container. It is possible to let a container be affected by multiple clipping planes.

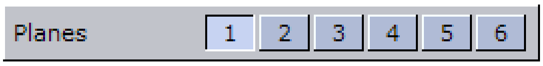
The clipper plug-in can replace mask in many situations. Some of the advantages are compared to normal mask are:

- Clipping is done in true 3D space, while masking is done in 2D (although the 2D mask is created by a 3D object).
- There is no performance hit. In fact there is even a performance gain if parts of the object are clipped. Regular masks have a quite high performance cost.

This section also contains information on the following topics and procedures:

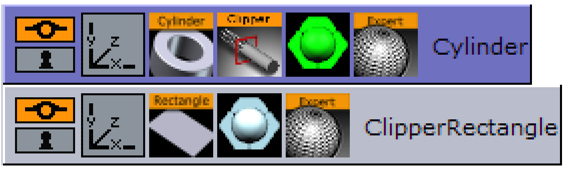
- Properties and Parameters
- To create a clipping plane effect

Properties and Parameters



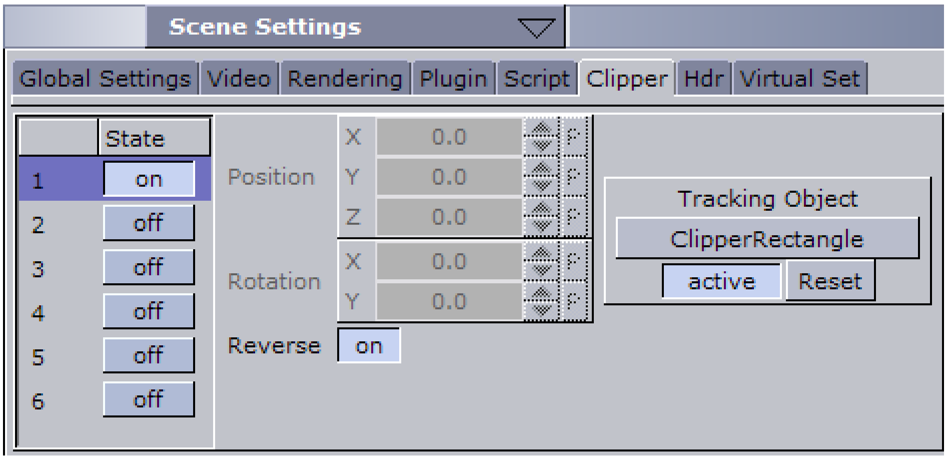
- **Planes**: Enables the planes. By default 1 is enabled.

To create a clipping plane effect



1. Add two group containers to the scene tree (same level) and name them **Cylinder** and **ClipperRectangle**.
2. Add the following to the **Cylinder** container:
   - Cylinder geometry plug-in

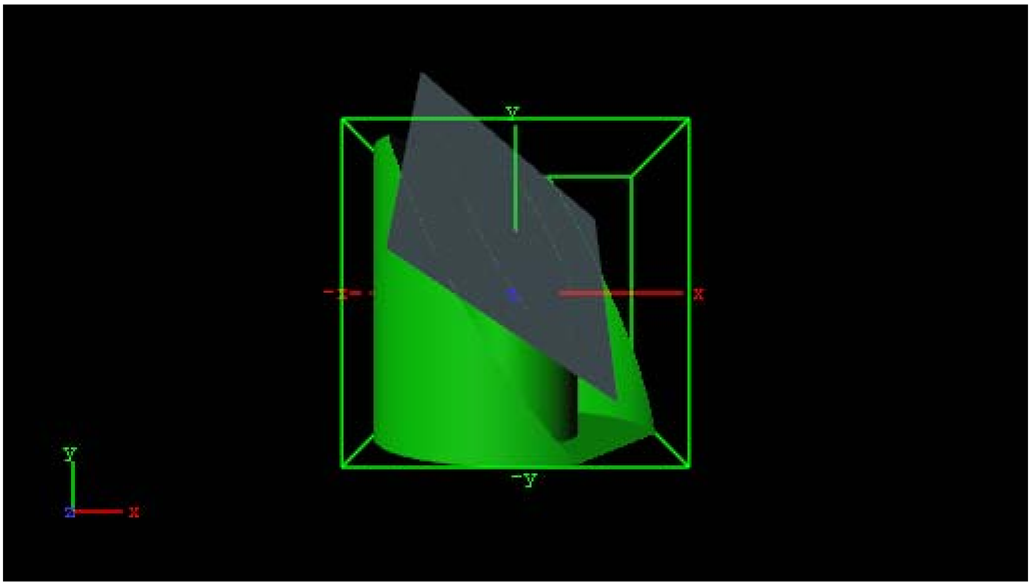
  - Clipper container plug-in
  - Material
  - Optional: Expert container plug-in
3. Add the following to the **ClipperRectangle** container:
  - Rectangle geometry plug-in
  - Material
  - Optional: Expert container plug-in
4. Open the editor for the Cylinder container and set **Scaling** (locked) to 2.0.
5. Open the Cylinder editor and set Hole to 45.0.
6. Open the Clipper editor and enable **Plane 1** (enabled by default).
7. Optional: Open the Expert editor(s) and enable **Back Face** (On) and **Twosided Lighting.**
8. Open the editor for the ClipperRectangle and set the following parameters for the following properties:
  - Set Position Y to 45.0.
  - Set Rotation X to -45.0.
  - Set Rotation Y to 45.0.
  - Set Scaling (locked) to 2.0.



	State
1	on
2	off
3	off
4	off
5	off
6	off

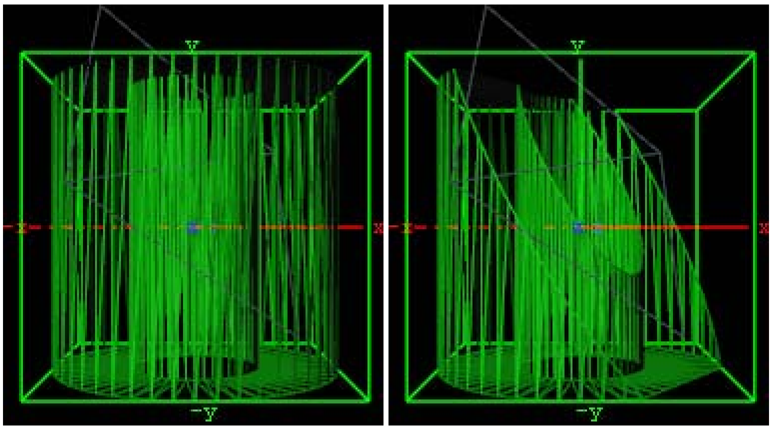
Position X 0.0
Position Y 0.0
Position Z 0.0
Rotation X 0.0
Rotation Y 0.0
Reverse on
Tracking Object ClipperRectangle

9. Open Scene Settings, and select the Clipper tab.
10. Enable **State 1**, and **Reverse**.



11. Drag and drop the ClipperRectangle container onto the Tracking Object drop zone.

Figure 173: Polygons drawn when clipping is enabled

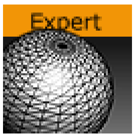


12. Optional: Open the Expert editor(s) and enable the Wireframe mode to see rendering result, and the effect clipping has on the polygons.

See Also

- Clipper

12.6.4 Expert

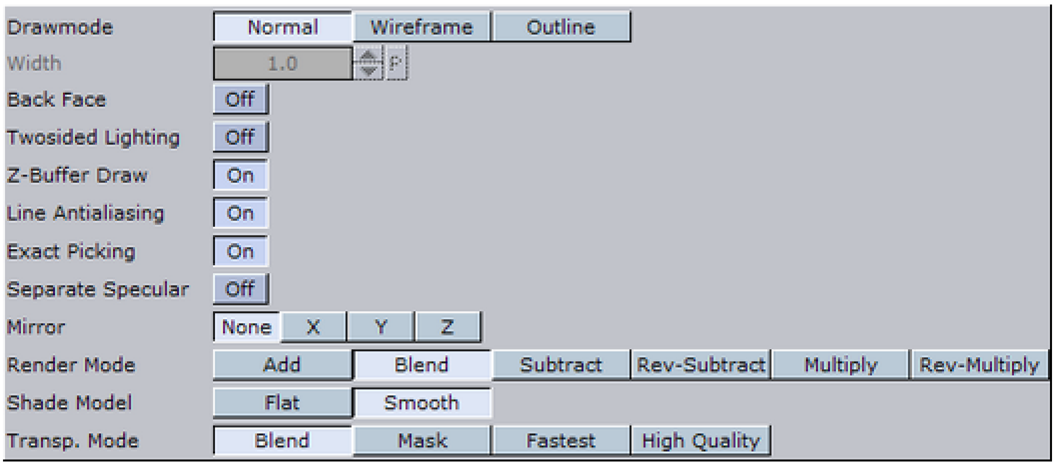


Expert sets some special properties and adds some advanced functions to a container.

This section also contains information on the following topic:

- Properties and Parameters

Properties and Parameters



Drawmode	Normal | Wireframe | Outline
Width	1.0
Back Face	Off
Twosided Lighting	Off
Z-Buffer Draw	On
Line Antialiasing	On
Exact Picking	On
Separate Specular	Off
Mirror	None | X | Y | Z
Render Mode	Add | Blend | Subtract | Rev-Subtract | Multiply | Rev-Multiply
Shade Model	Flat | Smooth
Transp. Mode	Blend | Mask | Fastest | High Quality

- **Draw Mode** you can chose to have the container drawn the following ways:
  - **Normal**: Shows the rendered object with material and/or textures.
  - **Wireframe**: Shows the polygons that are drawn in order to create the object.
  - **Outline**: Shows the outline of the object.
- **Width**: Sets the width of the wires if **Wireframe** or **Outline** draw mode is selected.
- **Back Face**: By enabling this option, an image or a 2D font will be drawn with a backface. This will of course take more performance to render.
- **Twosided Lighting**: With this parameter two-sided lighting can be enabled for a single container instead for a whole scene.
- **Z-Buffer Draw**: Allows you to disable the OpenGL z-buffer draw. This is in some cases the last possibility for solving some z-sort problems. The technical explanation for this is very complex and will not be described here.
- **Line Anti aliasing**: 2D objects on machines that do not have anti-aliasing with multi-sampling, an anti-aliased outline is rendered instead. This extra outline can be removed by disabling this option.
- **Exact Picking**: Enables or disables Exact picking for Events. If Exact Picking is disabled, the bounding box is used instead of a pixel readback.
- **Separate Specular**: Enables specular highlights for textured geometry.
- **Mirror**: Enables you to mirror an image/texture or a 2D font over the X-, Y- and Z-axis.
- **Renderoptions**
  - **Add**: Symbolically: C*A + FC. The source color gets added to the target color. The amount of color that is added depends, as we see from the formula, on the alpha value. However, it is always an addition, so the end result will always be a lighter color than the initial frame content. If the frame color has high values on all three color channels (RGB), you might experience that the addition of the new color takes all channels to values above 255 (saturation). The values will be clamped at 255, which is white.
  - **Blend**: Symbolically: C*A+(1-A)*FC. The new color value gets created as a weighted average of the source and the target. The weight factor is the alpha value of the rendered color. That means that if the incoming color has a very low alpha value, its influence on the new color will be small, and

conversely, if the new color has a high alpha value, its influence on the new color will be bigger.

  - **Subtract**: Symbolically: FC-C*A. The new color is the result of the incoming color being subtracted from the color in the frame buffer and the result is written back into the frame buffer. The alpha value of the incoming color decides how much who gets subtracted. If the incoming color has high values on all three color channels (RGB), you might experience that the subtraction of the new color takes all channel values to 0, and the result is a black color.
  - **Rev-Subtract**: Symbolically: C*A-FC. This is a reversed version of subtractive. The color in the frame buffer gets subtracted from the incoming color and the result is written back into the frame buffer.
  - **Multiply**: Symbolically: CA*FC. The new color and alpha get multiplied with the existing values in the frame buffer. The formula presupposes that the colors and alpha re described as values between 0 and 1. A color rendered with multiply will always result in a darker color than both the color being rendered and the color in the frame buffer.
  - **Rev-Multiply**: This is a reversed version of multiply.
- **Shade Model**
  - **Flat**: Shows the single polygons the texture is made of.
  - **Smooth**: Calculates a smooth texture.
- **Transp. Mode**
  - **Blend**: Is the standard option.
  - **Mask**: Uses a raster.
  - **Fastest**: Uses the transparency mode which takes the least time to render.
  - **High Quality**: Uses the transparency mode which delivers the best looking result.

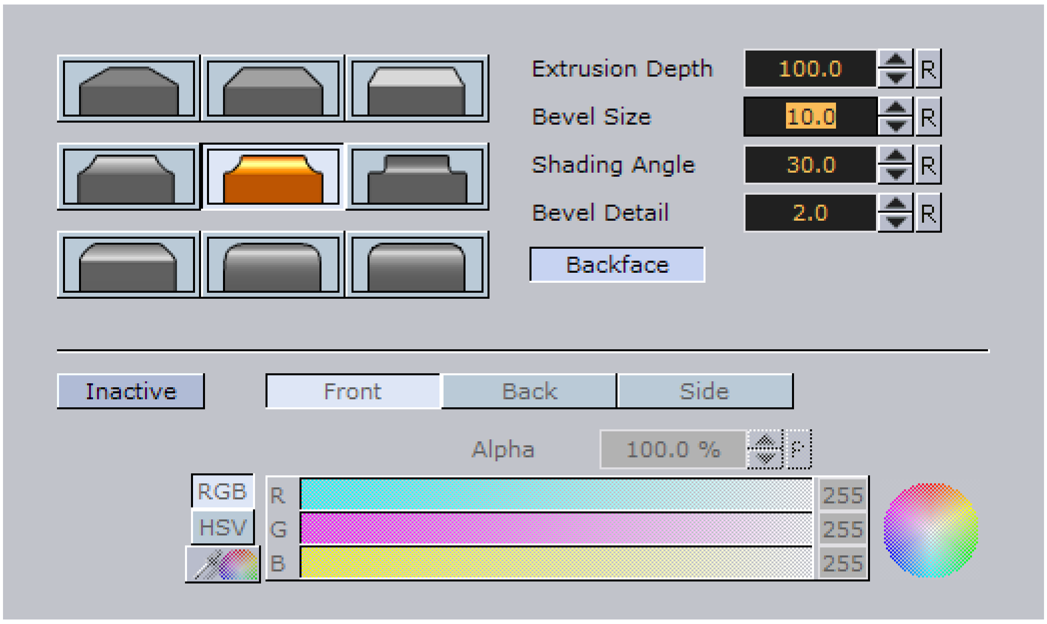
12.6.5 Extrude

An Extrude plug-in can be explained simply as extruding or sweeping the 2D element through space along its Z-axis. The sweeping path the element is followed during this process is used to create a surface. A a result a 3D element with front, back and sides is created from the 2D element.

This section also contains information on the following topics:

- Properties and Parameters
- To extrude a 2D object

Properties and Parameters



Extrusion Depth 100.0

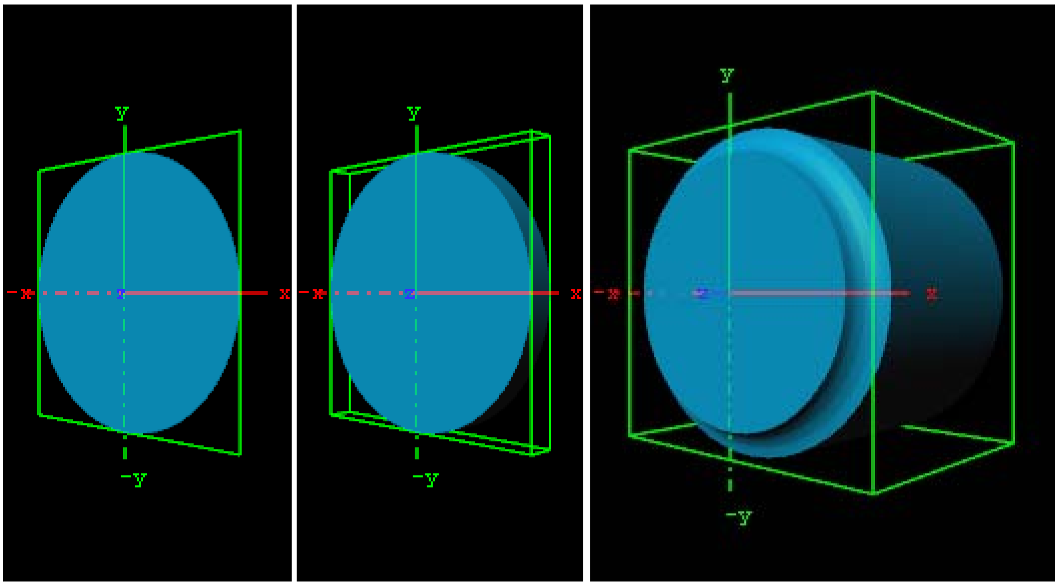
Bevel Size 10.0

Shading Angle 30.0

Bevel Detail 2.0

- **Bevel Size**: The buttons marked with different bevel types are used to set the type of bevel to use on the figure.
- **Extrusion Depth**: Sets the extrusions depth on the Z-axis.
- **Bevel Size**: Sets the size of the bevel. This must be set to some value to enable the selection of bevel type.
- **Shading Angle**: Sets the angle of the shading function. To “erase out” unevenness, try increasing the shading angle.
- **Bevel Detail**: Sets the degree of detail of the bevel. The lover the value is set the more detailed the bevel is constructed. The more detailed the bevel is, the smaller tilings its constructed from.
- **Backface**: Allows you to enable or disable visualization of the backface.
- If colors are set to **Active**, you can edit the elements **Front**, **Back** and **Side** color by using the sliders or changing the values.

To extrude a 2D object



1. Create a group and add the Circle geometry plug-in and material to it.
2. Add the Extrude plug-in to the same container.
3. Open the Extrude editor and select one of the bevel effects, and set Extrusion depth to 100.0 and Bevel Size to 10.0.

12.6.6 Glow

Makes the lit faces of objects glow.

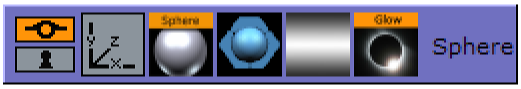
This section also contains information on the following topics:

- Properties and Parameters
- To add glow

Properties and Parameters

- **Blur Width**: Sets the width of the blur (i.e. the size of the glow).
- **Strength**: Sets the intensity/strength of the glow.
- **Object Strength**: Sets the intensity/strength of the glow on the faces of the object.
- **Use Color**: When enabled (On), it is possible to set the color of the glow. If not the glow will be based on the Material added to the object.
- **Textured**: Uses the color of the texture to determine glow instead of diffuse factor.

To add glow



1. Add a Sphere geometry plug-in to the scene tree.
2. Add material and/or texture to the Sphere container.
3. Add the Glow plug-in.
4. Open the Glow editor and set the following parameters:
   - Set Blur Width to 20.0.
   - Set Strength to 6.0.
   - Set Object Strength to 0.0.
   - Enable Use Color and set the color parameter.

12.6.7 HDR

Enables High Dynamic Range Rendering/Imaging for the selected container. This improves the contrast of the generated scene in a much more realistic way.

This section also contains information on the following topic:

- Properties and Parameters
- To add HDR

Properties and Parameters

This plug-in does not have any properties or parameters.

To add HDR

- Drag the plugin on a container and modify the HDR settings under Scene setting/hdr. You can specify a diffuse and a Cubic Reflection image.

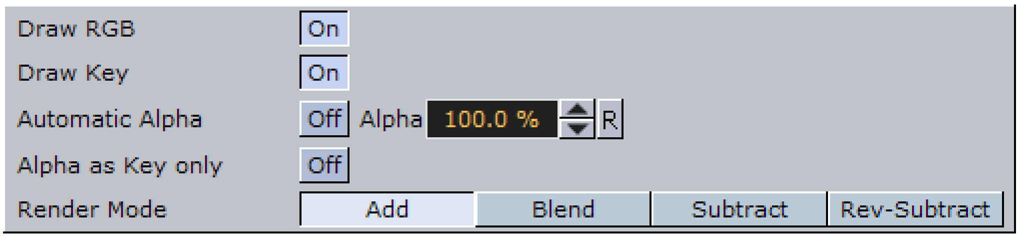
12.6.8 Key

Key adds a Key signal to a container.

This section also contains information on the following topic:

- Properties and Parameters

Properties and Parameters



Draw RGB On

Draw Key On

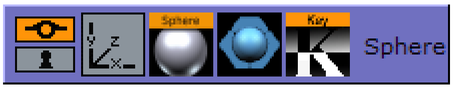
Automatic Alpha Off Alpha 100.0 % R

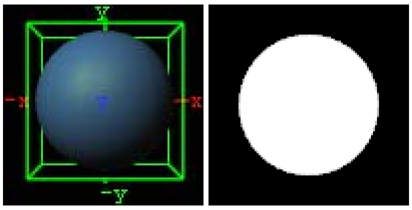
Alpha as Key only Off

Render Mode Add Blend Subtract Rev-Subtract

- **Draw RGB**: Disable this if you do not want the graphical object to be shown, just the key signal.
- **Draw Key**: Enables/disables the key signal.
- **Automatic Alpha**: If you disable the automatic alpha option, you can set the value manually. This is normally done to obtain some level of transparency so that, for example, video background is visible through the key object.
- **Alpha as Key only**: Takes the alpha value of the object and renders the key with this alpha instead of the alpha value of the key function attached. The object itself is rendered opaque. This is usually used to render transparent objects for keyed graphics. The “color mixing” of the object in the scene and the for example video background is done by the external keyer so the object must be rendered non transparent.
- **Render Mode**: This option sets the mode in which the alpha values of the key element you are rendering should be mixed with the alpha values already existing in the frame buffer. Be aware that with the subtractive and the rev-subtractive, mode the order in which the objects are being rendered then is crucial. So make sure the z-sort is correct. The mixing is being performed on a per pixel basis (in the formulas the range of the alpha values is from 0-1, instead of 0-100 as in the value field). If we say A is the current source alpha value being rendered now and FA is the alpha of the frame buffer contents (target), the different modes can be explained as following:
  - **Add**: Symbolically: A + FA. The source alpha gets added to the target alpha. As the formula shows it is an addition, so the end result will always be a higher alpha value than the initial frame content. If the frame alpha has a high value, you might experience that the addition gives a total value above 1. This will be clamped at 1 which is opaque.
  - **Blend**: Symbolically: A+(1-A)*FA. The new alpha value gets created as a weighted average of the source and the target. If we as an example say that the key being rendered has alpha 1 and the frame alpha has 0.5, the formula will be like this: 1+(1-0.5)*0.5 = 0.75.
  - **Subtract**: Symbolically: FA-A. The new alpha value is the result of the incoming alpha being subtracted from the alpha in the frame buffer and the result is written back into the frame buffer.
  - **Rev-Subtract**: Symbolically: A-FA. This is a reversed version of subtractive. The alpha in the frame buffer gets subtracted from the incoming alpha and the result is written back into the frame buffer.

To add key



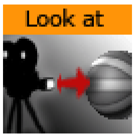


1. Open a scene, and add the Key plug-in to a container with an object on it.
2. Click the **Key** button on the left side of the Scene Editor.

See Also

- Global Settings

12.6.9 Look-At

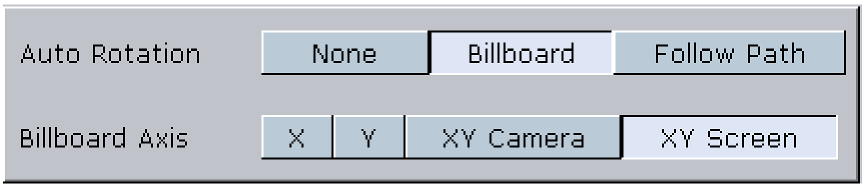


The look at plug-in is used to rotate a container so that it faces the direction you define. This is especially valuable in animations.

This section also contains information on the following topic:

- Properties and Parameters

Properties and Parameters



Auto Rotation: None | Billboard | Follow Path

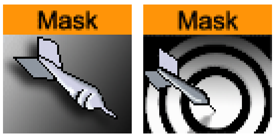
Billboard Axis: X | Y | XY Camera | XY Screen

- **Auto Rotation**
  - **None**: No rotation.
  - **Billboard**: Will rotate the container to face the camera.
  - **Follow Path**: Will rotate the container to justify at the animation path.
- **Billboard Axis**: Is enabled when selecting **Billboard**.
  - **X**: The container will rotate about its X-axis to face the camera, even if camera moves
  - **Y**: The container will rotate about its Y-axis to face the camera, even if camera moves
  - **XY Camera**: The container will rotate about its X- and Y-axis to face the camera, even if it moves
  - **XY Screen**: The container will rotate about its X and Y-axis to face the camera if the camera orbits the container only.

See Also

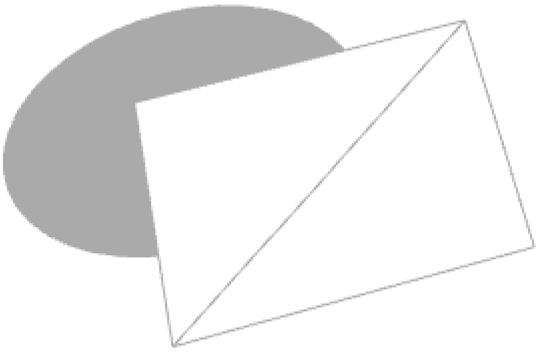
- Creating Animations

12.6.10 Mask Source and Mask Target

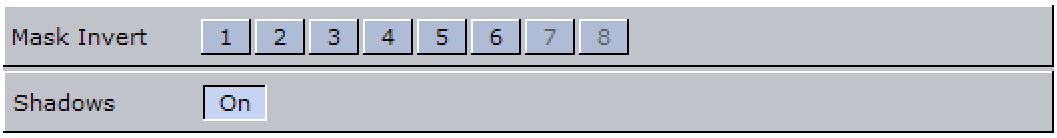


With the mask plug-in you can make one container act as a mask for others. The mask container is called the source and the containers being affected by the source mask are called targets. A container being affected by a mask becomes transparent where the mask covers it.

There is one editor for the mask source and another for the mask target where you can define by which layer the object is affected.



You can use up to eight different target layers and source layers for one object. This means that targets with the layer 1 set will be affected by all masks which have the layer selected. If a mask has the layers 1, 2 and 3 selected it will be affect all targets that have 1, 2 or 3 selected.



Mask Invert 1 2 3 4 5 6 7 8

Shadows On

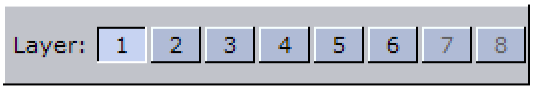
The layer 7 and 8 may not be available. This is because global shadow settings by default are enabled. Shadows use two layers of the mask plug-in, but can be switched off under the Global Settings to enable the use of all eight layers.

Under the Global Settings you can also define if a mask layer is to be drawn inverted. Normally the mask cuts a hole in the mask target where it covers it. If you select a layer to be drawn inverted the opposite situation is the case. Only where the mask covers the mask target will the target be visible.

This section also contains information on the following topics:

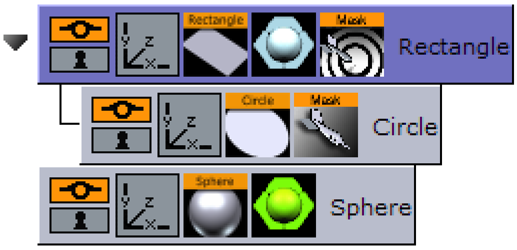
- Properties and Parameters
- To add mask

Properties and Parameters

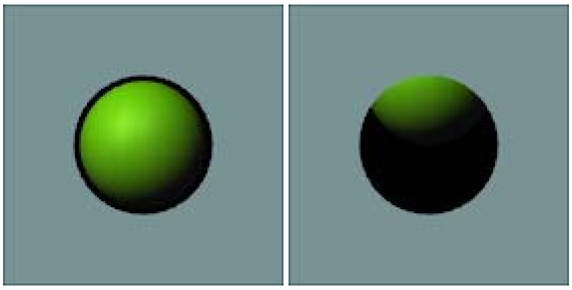


- **Layer**: Sets the layer the mask should have an effect on.

To add mask



1. Add a Rectangle geometry to the scene tree, and add material and the Mask Target plug-in to it.
2. Open the Rectangle editor and set Width to 200.0 and Height to 200.0.
3. Add a Circle geometry as a sub container to the Rectangle container, and add the Mask Source plug-in to it.
4. Add a Sphere geometry to the scene tree at the same level as the Rectangle container.
5. Add material to the sphere.
6. Open the transformation editor for the Sphere and set Position Z to -50.0.



7. Adjust Position Y see the effect of the mask.

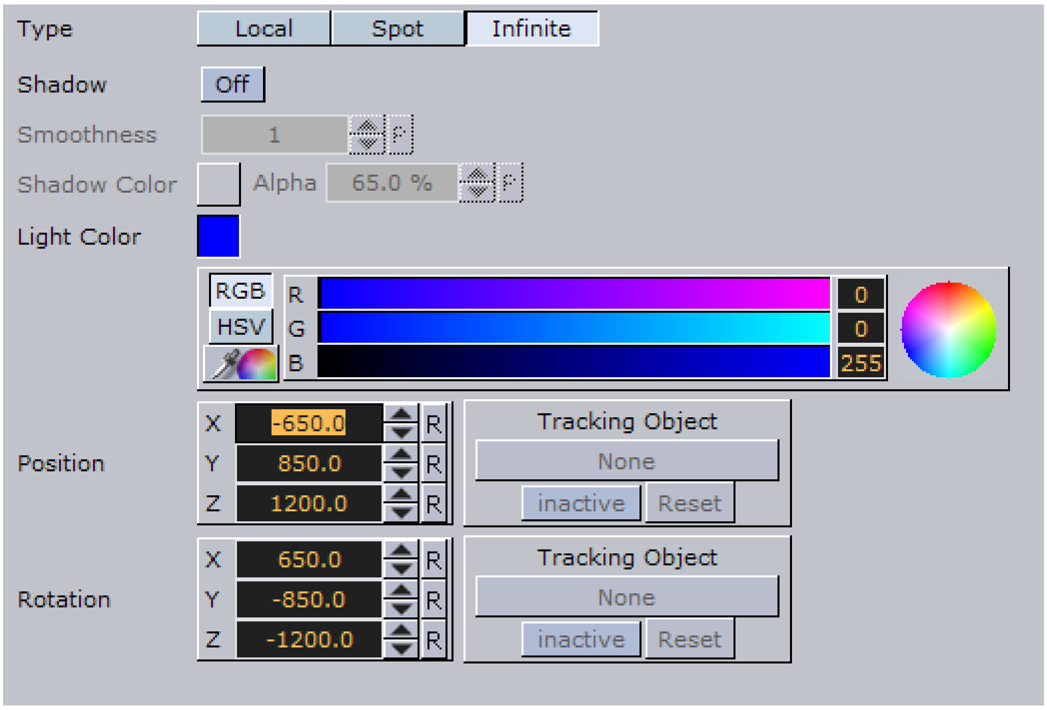
12.6.11 Lighting

Adds individual lighting per object.

This section also contains information on the following topics:

- Properties and Parameters
- To add mask

Properties and Parameters



Type: Infinite

Shadow: Off

Smoothness: 1

Shadow Color — Alpha: 65.0 %

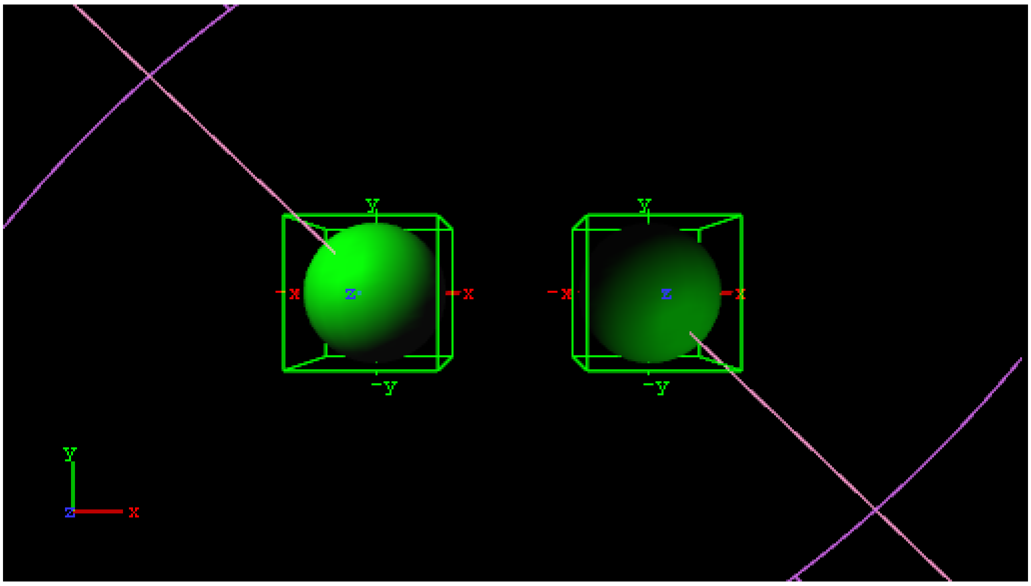
Light Color: R 0, G 0, B 255

Position: X -650.0, Y 850.0, Z 1200.0 — Tracking Object: None

Rotation: X 650.0, Y -850.0, Z -1200.0 — Tracking Object: None

- **Type**: There are three light source types:
  - **Local**: The Local light source is a positional light. It is near or within the scene, and the direction of its rays is taken into account in lighting calculations. Local lights have a greater performance cost than directional lights, due to the additional calculations. A real life equivalent to a local light source is a light bulb. The Local light source has properties for color and position.
  - **Spot**: The Spot light source emits a cone of light. The only objects that are illuminated are those within the cone. The Spot light source has properties for color, position, rotation, and attenuation.
  - **Infinite**: The Infinite light source is a directional light. It is considered to be an infinite distance away from the objects in the scene. Because of the distance, the rays of light are considered parallel by the time they reach the object. A real life equivalent to an infinite light source is sunlight. The Infinite light source has properties for color, position, and rotation. Infinite is selected by default.
- **Color**: Sets the light source color.
- **Position**: Sets the position of the light source along the X, Y, and Z axis.
- **Rotation**: Sets the values for Pan, Tilt, and Twist for Spot or Infinite light sources.
- **Spot**: Sets the concentration of the light within the light cone of the Spot light source. When set to zero, the whole light cone has the same intensity. If set greater than zero, the intensity decreases away from the centre.

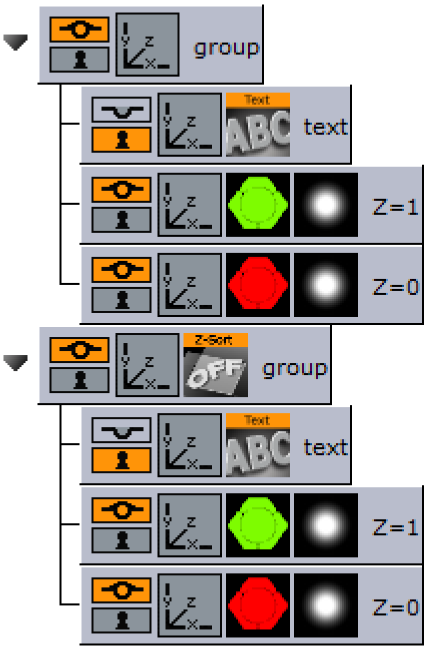
To add lighting



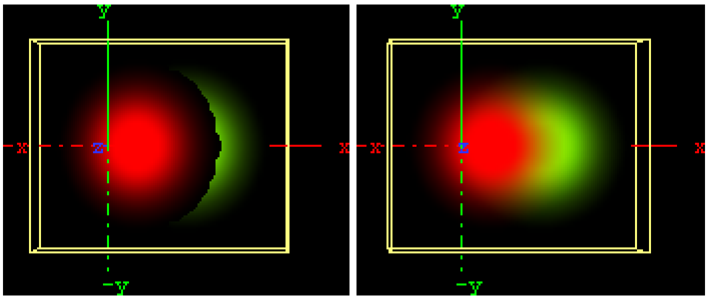
1. Open the Light Editor, and disable global light settings.
2. Add a Sphere geometry to the scene tree, and add material and the Lighting plug-in to it.
3. Duplicate the Sphere container and place it at the same level (root) as the other container.
4. Open the transformation editor and set Position X to -100.0 for the first container and Position X to 100.0 for the second.
5. Open the Lighting editors the first container one and set the following parameters:
   - Set Type to Spot.
   - Set Light Color to 0,255,0.
6. Open the Lighting editors the second container one and set the following parameters:
   - Set Type to Spot.
   - Set Light Color to 0,255,0.
   - Set Position X and Y to 650.0 and -850.0, respectively.
   - Set Rotation X and Y to -650.0 and 850.0, respectively.

12.6.12 Z-Sort

Sorting the objects of a scene on the Z-axis is essential for any 3D program in order to render correctly. The renderer starts by rendering the object furthest behind, and then mixing in objects closer and closer to the camera. Sometimes the machine gets “confused” with what should be rendered first and last because it calculates this from what is the center of the object’s Z position.



A typical example of a Z sort problem is when two colored objects have rotated their Y axis to for example -12.0. In this case the default behavior for a render engine is to render the first object (green) with a Z position 0.0 first, and then the second object (red) with Z position 1.0 last.



However, when objects are rotated the output will be wrong because the red object should be rendered before the green due to the rotation of the objects. This can be corrected by adding the Z-Sort plug-in, which lets users manually sort objects using the logic of the tree structure. Hence, what is placed at the top of a group in the scene tree gets rendered first, and what is placed below will be rendered after.

Sorting is solved by rendering the second (red) object first, and then the first (green) last. Be aware that although the red object has a Z position of 1.0 and the green 0.0, the red object must be rendered before the green because of the overlapping rotation (Y -12.0). Normally objects that are furthest behind should be placed at the top in the scene tree.

The Z-Sort plug-in disables the automatic Z-sort for a container. How the elements in the container get sorted in the Z-buffer is then dictated by the order of the objects in the scene tree.

12.6.13 Projector Source and Projector Target

The projector and the projector target is used to project something on something else (e.g. an image onto a geometry).

A projector container creates a planar projection of their texture onto container(s) with the projection target function attached. In principle it works similar to a slide-projector. For objects that do not have texture coordinates, the projector function can be used to apply texture to such objects. It also gives the possibility to have two textures on a single object, the objects own texture and the projected texture.

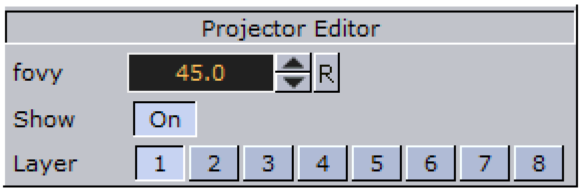
The projector and the projection target can be set up to react on different layers similar to shadows and masks.

Note: The projector plug-in does not work on text.

This section also contains information on the following topics:

- Properties and Parameters
- To project a texture onto an object

Properties and Parameters



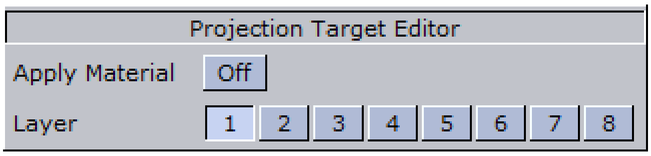
Projector Editor

fovy 45.0 R

Show On

Layer 1 2 3 4 5 6 7 8

- **fovy**: Sets the field of view for the projector container. The higher the fovy is set, the larger the projected texture on the receiver will be. You can picture it as the zoom function of a slide projector.
- **Show**: Allows you to hide/show the projector object.
- **Layer**: Lets you select by which layers the projector and receiver are to perform the function. You can use up to 8 different projector layers and source layers for one object. This means that targets with the layer 1 set will be affected by all projectors which have the corresponded layer selected. If a projector has the layers 1, 2 and 3 selected it will affect all targets with 1, 2 or 3 selected.



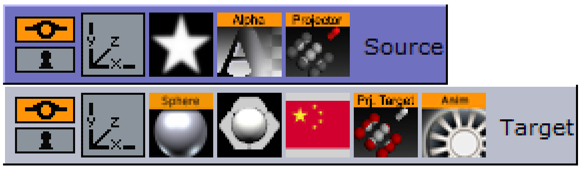
Projection Target Editor

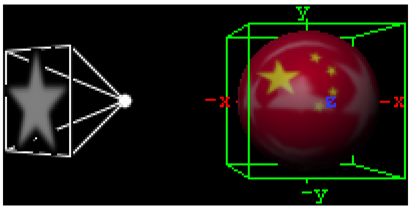
Apply Material Off

Layer 1 2 3 4 5 6 7 8

- **Layer**: Lets you select by which layers the projector and receiver are to perform the function. You can use up to 8 different projector layers and source layers for one object. This means that targets with the layer 1 set will be affected by all projectors which have the corresponded layer selected. If a projector has the layers 1, 2 and 3 selected it will affect all targets with 1, 2 or 3 selected.

To project a texture onto an object

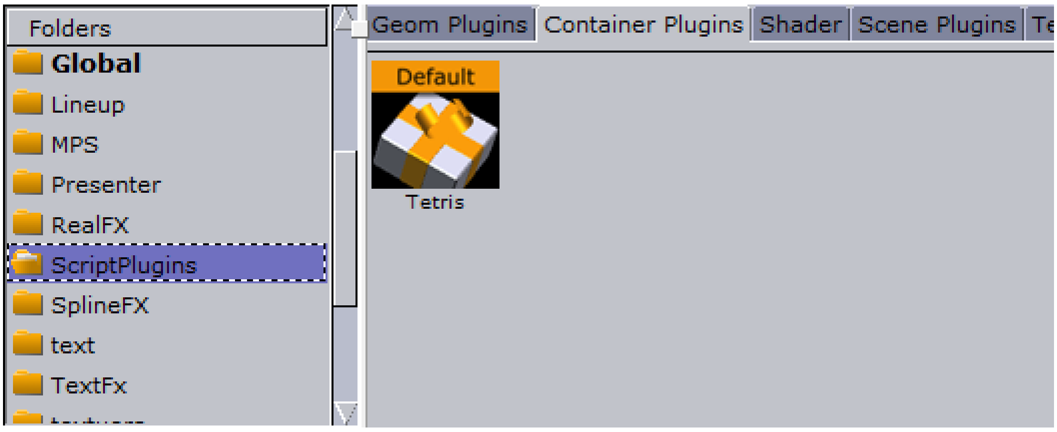




1. Add a group container to the scene tree, add the Projector and Alpha plug-in to it, an image/texture, and name it **Source**.
2. Open the Alpha editor and set the alpha value to 50.0%.
3. Open the transformation editor for the Source container and set Rotation Y to 135.0.
4. Add another group to the scene tree, and name it **Target.**
5. Open the transformation editor for the Target container, and set Position X to 130.0.
6. Add the Sphere geometry plug-in, material, image/texture and the Projector Target plug-in to the Target container.
7. *Optional*: Animate the sphere.
8. Open the Projector editor and **disable** the **Show** property.

12.6.14 Script

The script functionality in Viz is a plug-in in itself. Scripts can be added to a container or the Scene Settings, and is stored with the scene. It is possible to archive and import scripts as part of a scene archive as with any other plug-in.



In addition it is possible to create your own script plug-ins by dragging a compiled script into the **ScriptPlugins** folder. The ScriptPlugins folder is a special folder that allow users to create and name their own plug-ins. Note that script plug-ins are saved to the same plug-in folder where Viz has its own plug-ins.

Viz plug-ins are located in the following folder: `C:\Program Files\Vizrt\Viz3\plugin\`

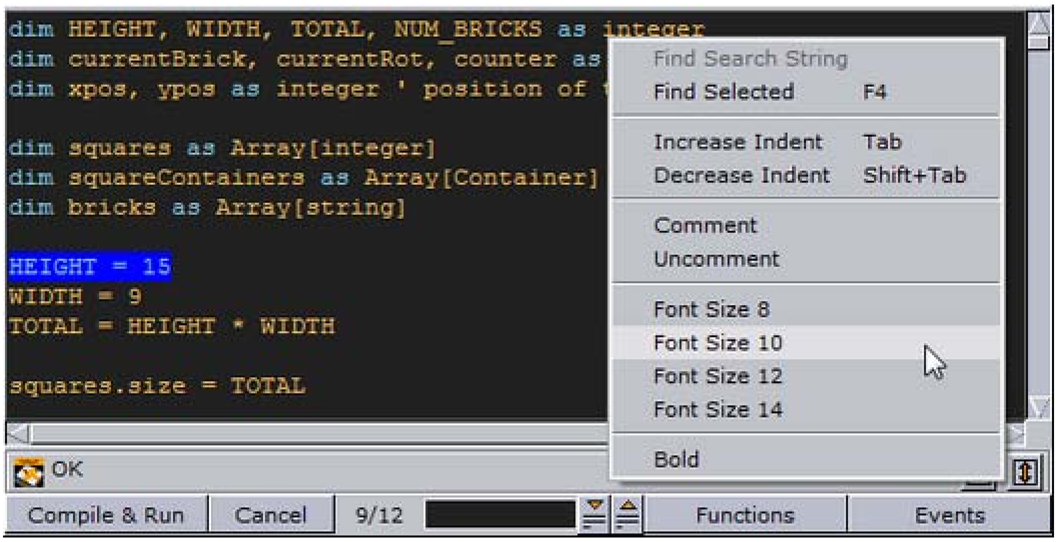
A script plug-in is not stored in the database, and as a consequence not distributed with an archive because it is code and not data.

There is an important difference between a script that is not converted into a plug-in, and a script that is. As long as it is not converted the content of the script code is data within the “Script” plug-in, just like the width property is data within for example a geometry plug-in. Once the script is converted to a plug-in it is a plug-in itself, like any other plug-in.

This section also contains information on the following topics:

- Properties and Parameters
- To create a plug-in from a script
- To edit a script plug-in

Properties and Parameters



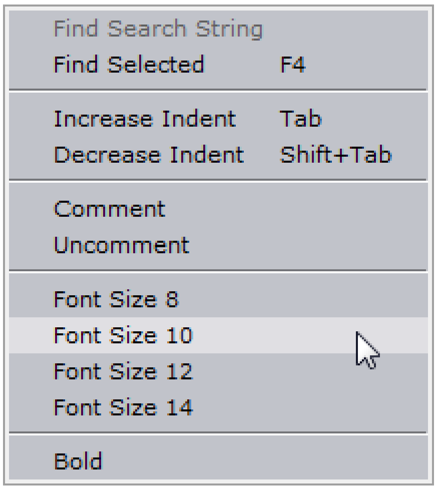
```
dim HEIGHT, WIDTH, TOTAL, NUM_BRICKS as integer
dim currentBrick, currentRot, counter as
dim xpos, ypos as integer ' position of

dim squares as Array[integer]
dim squareContainers as Array[Container]
dim bricks as Array[string]

HEIGHT = 15
WIDTH = 9
TOTAL = HEIGHT * WIDTH

squares.size = TOTAL
```

↕ ↔

- **Arrow buttons**: Clicking the vertical arrow buttons detach or attach the Script Editor from or to the Viz interface. While detached the script editor can be scaled to any size.
- **Compile & Run**: checks the code entered and translates it into byte code, so the script can be run. If an error occurs, you will find a message shown in the text above.
- **Cancel**: Ignores the last source code changes and compiles previous version.
- **Find** enables you to search for a string in the script. To do so, enter the string you search for in the input box and press *F3*.
  Alternatively you can select a string in the script editor and search for this string by pressing *F4*.

- **Functions**: Shows a list of used functions in the script (both built-in and self defined ones). Select a function in the list and you will jump there in the script editor.
- **Events**: Provides a list of built-in events. Select one there to add it to the script.

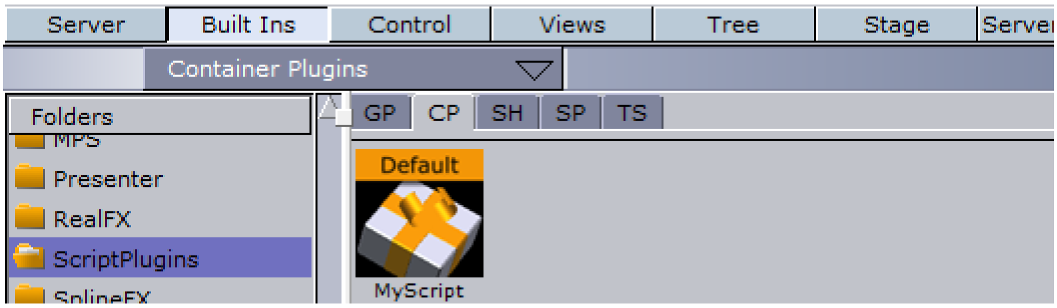
If you select a string in the script editor and do a right-click, a shortcut menu will open. Here you can select one of the following actions:



- **Find Search String**: Is only available if you have entered a string into the **Find** input box. If you select this, the search string will be found.
- **Find Selected (F4)**: Performs a search for the next occurrence of the selected string.
- **Increase Indent (Tab)**: Increases the indent of the selected text.
- **Decrease Indent (Shift+Tab)**: Decreases the indent of the selected text.
- **Comment**: Comments the selected text.
- **Uncomment**: Uncomments the selected text.
- **Font Size**: Sets the font size. Available options are; 8, 10, 12 or 14.
- **Bold**: Sets the font to bold.

To create a plug-in from a script

After you have finished creating your script, you may want it to save it as a plug-in in order to use it for other scenes.



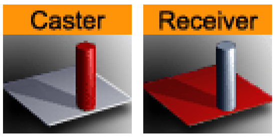
1. Open the Container Plugins directory and select the folder **ScriptPlugins**.
2. In the Script Editor, click the **Compile & Run** button.

3. Drag and drop the Save Script icon (located above **Compile & Run** button in the Script Editor) onto the ScriptPlugins folder.
4. Enter a name for the plug-in, and click **OK**.
   - The plug-in is saved to the plugin

To edit a script plug-in

1. Add a group container to the scene tree, and add the Script plug-in to it.
2. Open the Script Editor, and drag and drop the script from the ScriptPlugins folder onto the script editor.

12.6.15 Shadow Caster and Shadow Receiver



The Shadow Caster and Receiver plug-ins are used to enable an object’s shadow to reflect on another object. To create a shadow element in a scene, two containers at a minimum must be used. The caster plug-in must be attached to one container and the receiver function to the other.

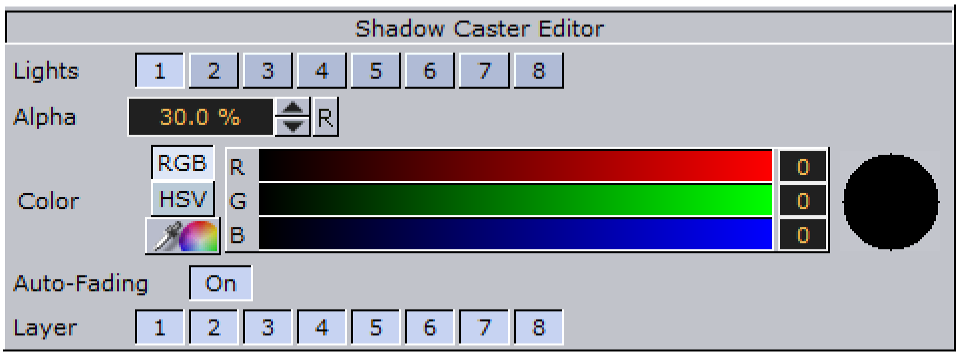
The container acting as the caster must be positioned between the receiver container and a light source.

Objects that can function as a shadow receiver should be a built in 2D geometry or a font. An imported 2D will not work.

This section also contains information on the following topics:

- Properties and Parameters
- To cast a shadow

Properties and Parameters



Shadow Caster Editor

Lights 1 2 3 4 5 6 7 8

Alpha 30.0 % R

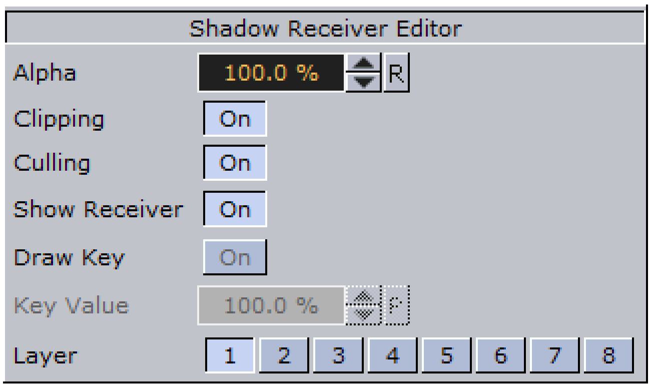
Color RGB HSV R 0 G 0 B 0

Auto-Fading On

Layer 1 2 3 4 5 6 7 8

- **Lights**: Allows you to select which light sources are to shine on the caster object and thereby producing the shadow/shadows.
- **Alpha**: Sets the alpha value and hence the transparency of the shadow being cast on the receiver objects.
- **Color**: Allows you to define the shadows color.

- **Auto-Fading**: With this option enabled, if you fade out a caster object either by using the alpha function or by adding a material with a low alpha value, the shadows created by the caster will fade correspondingly. The option is enabled by default, since it looks right that the shadow fades out when the caster object fades out.
- **Layer**: Allows you to select by which layer(s) the caster will cast shadow. You can select up to 8 layers. Each layer corresponds to the layer selected in the Receiver objects. If you want all shadow casters to cast shadow on all receivers, you can set all layers on all casters and receivers to 1. If you want one caster to create shadow on some receivers and another caster to create shadow on different receivers, you select different layers for the sets of casters and corresponding receivers. Casters and receivers can have multiple layers enabled at the same time so you can set up a great number of combinations of caster/receiver combinations.



Shadow Receiver Editor

Alpha 100.0 %

Clipping On

Culling On

Show Receiver On

Draw Key On

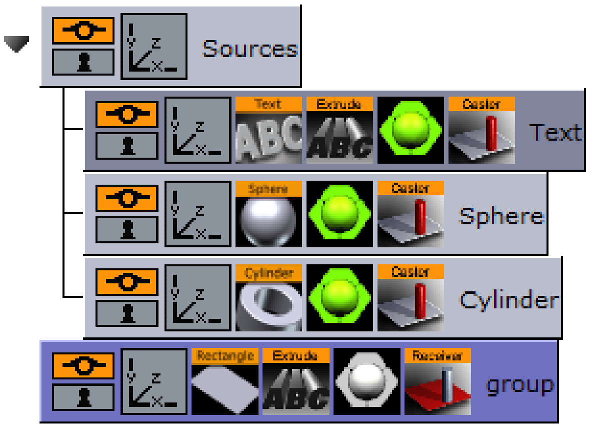
Key Value 100.0 %

Layer 1 2 3 4 5 6 7 8

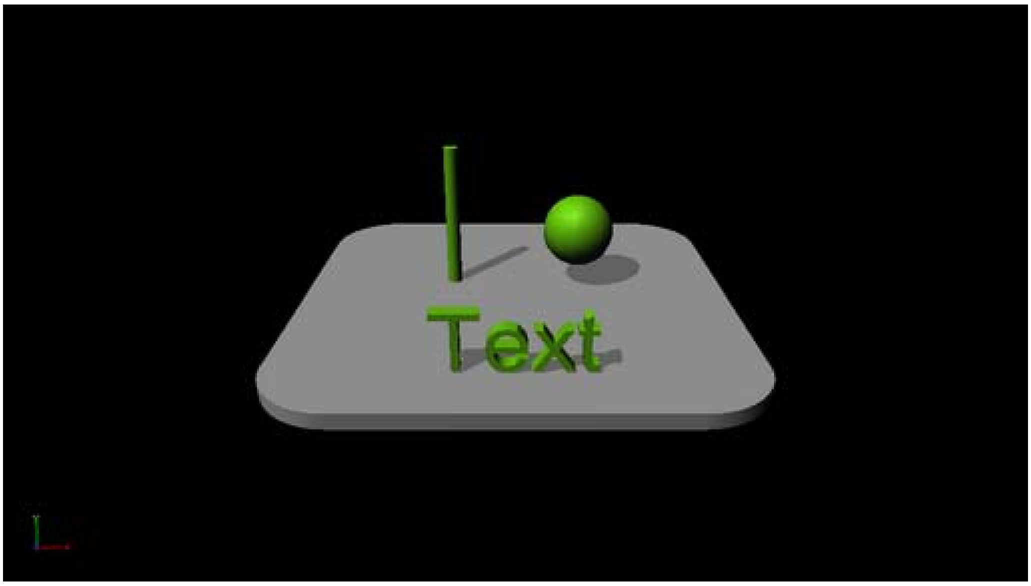
- **Alpha**: Sets the alpha value.
- **Clipping**: With clipping enabled the shadow will be cut on the border of the object. With clipping disabled, the shadow will be rendered onto the background outside the object, which normally looks wrong. Rendering with the clipping options set takes more resources than without. If you for instance have an infinite floor or a room with walls, you can switch of clipping without seeing the “false” shadows, and thereby save some rendering performance.
- **Culling**: With culling enabled, Viz Artist will decide in advance of the rendering process if the shadow will be visible or if it will be hidden by other objects. If that is the case it will skip it in the render process. If you know for sure that the shadow for an object in a scene will be rendered visible for the whole object, you can disable culling to save some rendering time, but the effect is marginal.
- **Show Receiver**: Hides or shows the receiver object. If it is hidden, the shadow will still be visible.
- **Draw Key**: Adds key to the shadow. When enabled, the receiver must be hidden in order for the key to be drawn.
    - **Key Value**: Sets the key value.
- **Layer**: Allows you to select by which layers the receiver will receive shadow. You can select up to 8 layers. Each layer corresponds to the layer selected in the caster objects. If you want all shadow casters to cast shadow on all receivers, you can set all layers on all casters and receivers to 1. If you want one caster to create shadow on some receivers and another caster to create shadow on different receivers, you select different layers for the sets of casters and corresponding receivers. Casters and receivers can have multiple

layers enabled at the same time so you can set up a great number of combinations of caster- and receiver combinations.

To cast a shadow



1. Add a group container to the scene tree, and name it Sources.
2. Add a font, Sphere and Cylinder geometries as a sub container of Sources, and add material and the Caster plug-in to it.
3. Add a group container at the same level as the Sources container, and add a rectangle geometry, material and the Receiver plug-in to it.



4. Position the rectangle and the other objects such that the Sphere, Cylinder and text casts a shadow on the rectangular surface.

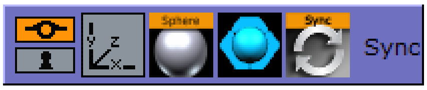
CAUTION! Containers are rendered twice when shadows are used.

12.6.16 Synchronized Properties

The Synchronized Properties plug-in can be used to synchronize container property changes within the same scene loaded on a cluster of Viz Engines. For example: A renderer with touch screen controls a stereoscopic interactive scene which is rendered on 2 other engines. All 3 engines need to perform all property or transformation changes at the same time.

This feature is available in the latest viz 3.3 versions and works for all container properties and plugin parameters.

To synchronize scene property changes on multiple Viz Engines



- Add the Synch. Properties plug-in on all containers that should be synchronized (no scripting is needed).

Note: The Synchronized Property plug-in does not inherit to child containers to keep network traffic low.

See Also

- To synchronize multiple Viz Engines

12.6.17 Video Clip

The Video Clip plug-in works in combination with the Control Video plug-in (not with the Control Targa Clip). It allows you to use clip keyframes in the stage for clip playback. When used with Transition Logic the keyframes can be merged with the object.

Note: The Control Targa Clip only allows the use of global clip channels when used in transition logic scenes, or clip keyframes when used in standalone scenes.

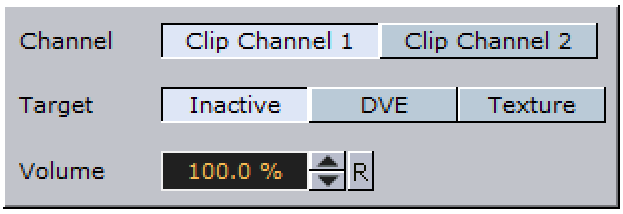
When Video Clip is used in transition logic scenes it allows a Toggle between the two clip channels. So a single object will playback any clip, and the Toggle plug-in in a transition logic scene will make sure that the channels toggle correctly, which again allows for transitions between two running clips. The plugin also toggles the clip texture if texture mode i selected.

The plug-in is commonly used in a video workflow using Viz Video Hub. It automatically adds a Clip Channel plug-in. The clip channel has to be activated using the Video Input configuration section.

This section also contains information on the following topics:

- Properties and Parameters
- To add a video clip

Properties and Parameters



Channel: Clip Channel 1 | Clip Channel 2

Target: Inactive | DVE | Texture

Volume: 100.0 % R

- **Channel**: Sets the clip channel to be used. Available options are; Clip channel 1 and 2.
- **Target**: Sets the clip mode. Setting target to DVE adds default keyframes to the stage. Setting target to Texture adds the Clip Channel texture plug-in and default keyframes to the stage. Available options are; Inactive, DVE or Texture.
- **Volume**: Sets the clip volume in percentage.

To add a video clip

1. Check the configuration to see if clip channels are activated.
2. Add the Video Clip plug-in onto a geometry (e.g. a rectangle).
3. Select Clip Channel 1 and Type Texture.
4. Switch to the Stage and select the Clip under VideoClip.
    - On the right side the clip to be played can be selected.
    - Additional keyframes can be added to play different clips.

See Also

- Implementing Video
- Animation Stage
- Toggle
- Control Video
- Transition Logic

12.6.18 Window Mask

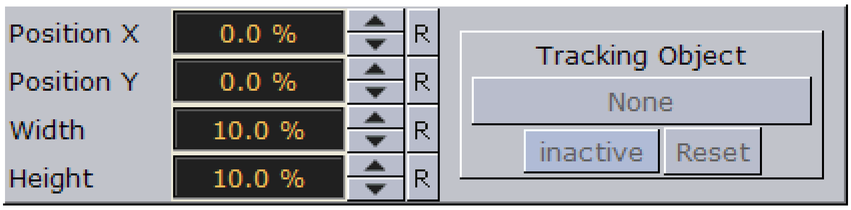
Window Mask limits the area into which a sub container of the Window Mask plug-in is rendered to a user-defined rectangular section of the output.

It is possible to add a tracking object. If a tracking object is used, it's bounding box is projected onto the screen and the resulting rectangle is used as a mask.

This section also contains information on the following topics:

- Properties and Parameters
- To use the Window Mask

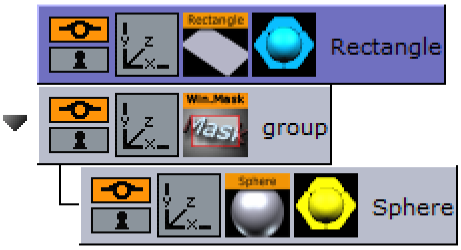
Properties and Parameters

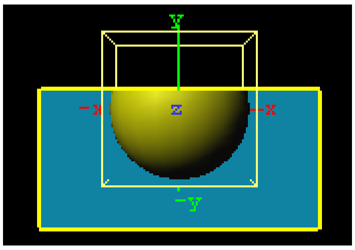


Position X 0.0 % R
Position Y 0.0 % R
Width 10.0 % R
Height 10.0 % R

- **Position X**, **Position Y**: Specifies the upper left corner of the rectangle. The values are scaled in such a way that (0, 0) refers to the upper left and (100, 100) to the bottom right corner of the screen.
- **Width**, **Height**: Specifies the dimensions of the rectangle in percentages of the screen dimensions.
- **Tracking Object**: Provides an alternative to setting the position and dimensions manually. When you drag a container from the scene tree onto this drop zone, the rectangle will automatically be adjusted to match the position and size of that container. **Inactive** will toggle tracking on and off. To remove the container click on **Reset**.

To use the Window Mask





1. Add a Rectangle geometry to the scene tree, and add material to it.
2. Open the Rectangle editor and set Width to 200.0.
3. Open the transformation editor and set Position Y to -36.
4. Add a new group, and add the Window Mask plug-in to it.
5. Add a Sphere geometry as a sub container of the group container, and add material to it.
6. Open the Window Mask editor and drag and drop the Rectangle container onto the Tracking Object drop zone.
    - This should give you a sphere that is partly masked due to the rectangle object.

12.7 MPS

The following container plugins are housed in the MPS folder:

- ExposeToVizkyAPI
- MPSImageOptimization
- MPSUI_CheckBox
- MPSUI_PushButton
- MPSUI_RadioGroup
- MPSUI_RadioButton

12.7.1 ExposeToVizkyAPI

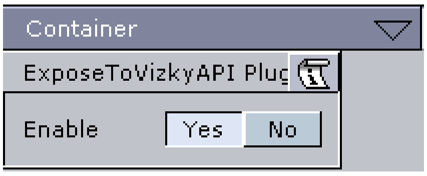
Used in conjunction with the Control Text plugin. It allows you to use JavaScript to change text values directly in scenes loaded in Vizky instead of using GOD to update MPS files with new data.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\ExposeToVizkyAPI.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To use the ExposeToVizkyAPI plugin
- Using JavaScript
- JavaScript Example

Properties and Parameters



Container

ExposeToVizkyAPI Plug

Enable Yes No

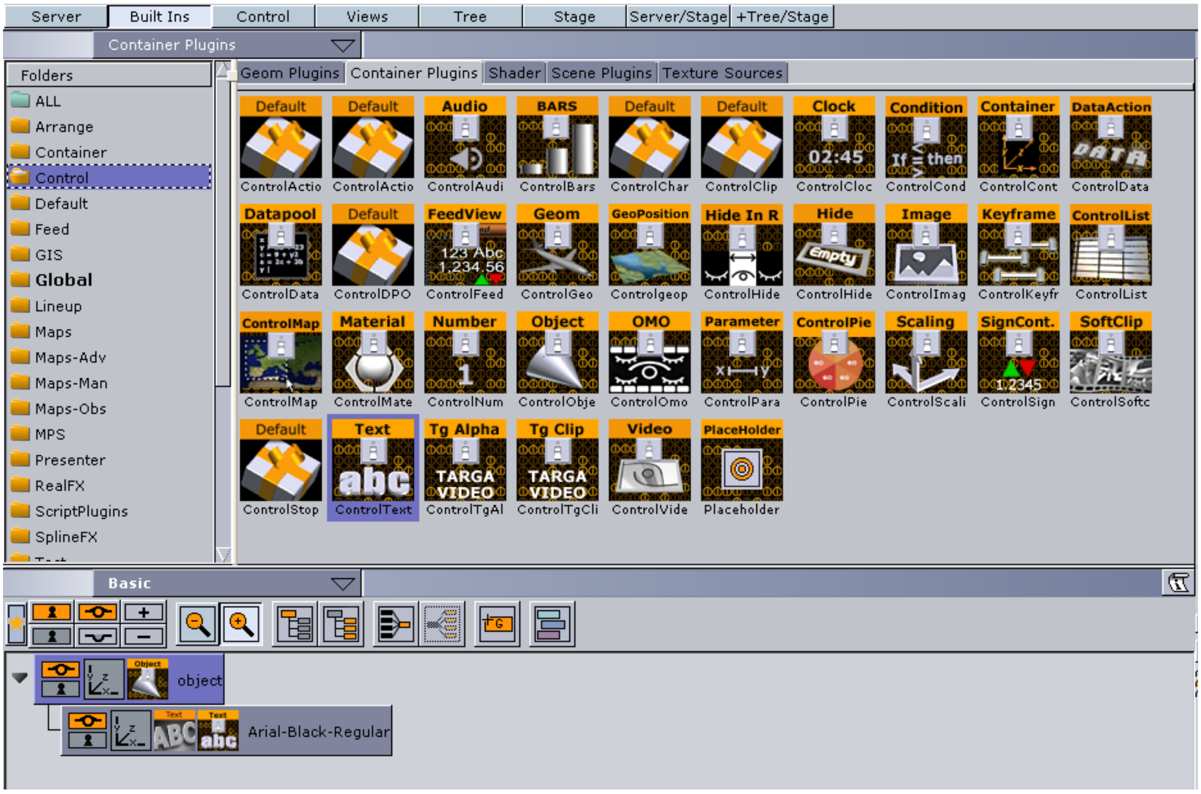
The only parameter is Enable. **Yes** enables and **No** disables.

The default setting is **Yes**.

To use the ExposeToVizkyAPI plugin

1. Click the Create Group button to create a root container. See Adding Empty Containers.
2. In the main menu, click **Server**, then click the **F** tab for fonts.
3. Select a font and drag it into the scene tree as a sub container of the group just created.

4. In the main menu, click **Built Ins**, then click the **Container Plugins** tab.
5. In the Folders pane to the left, click **Control**.
6. Select the Control Text plugin and drag it into the font container.
7. Define the ControlText field identifier. This is the reference used in the JavaScript.
8. Rename the root container to **object**.



9. In the Folders pane to the left, click **MPS.**
10. Select the **ExposeToVizkyAPI** plugin and drag into the font container.
11. Save the scene.

Using JavaScript

Refer to the *MPS Client Manual*. The relevant chapters for using JavaScript and the Vizky APIs are as follows:

- *Vizky Java Scripting Guide*
- *Vizky Client API Overview*

JavaScript Example

```
Control.SetControlChannelData(<InstanceSceneID>, <Control Text Name>,
<Text Value>)
```

This command is used when you want to change the value of the text which was composed with the ExposeToVizkyAPI plugin.

There are three parameters:

1. InstanceSceneID

From Import and Load Vizky API commands. Both must be used when downloading MPS files.

```
ExternalSceneID = Control.Import(StationID, MPSFileLocation);
InstanceSceneID = Control.Load(StationID, ExternalSceneID);
```

The InstanceSceneID is used as the main parameter with many functions in JavaScript. Other examples are StartDirector and ResetDirector control animation in downloaded MPS files.

2. Control Text Name

The control text Field Identifier's name. This refers to the text whose value you intend to change, identified by the ControlText plugin's Field Identifier.

3. Text Value

The new value to replace or change, identified by the ControlText plugin's Field Identifier.

See Also

- *MPS Client Manual*

12.7.2 MPSImageOptimization

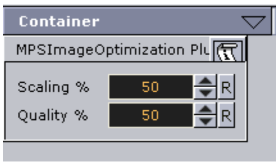
This plugin adjusts the image scaling and quality within a scene. Note that higher image size and quality will mean larger MPS files (and may affect performance)

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\MPSImageOptimization.vip`.

This section contains information on the following topics:

- Properties and Parameters
- Best Practices
- To optimize images

Properties and Parameters



Container

MPSImageOptimization Plu

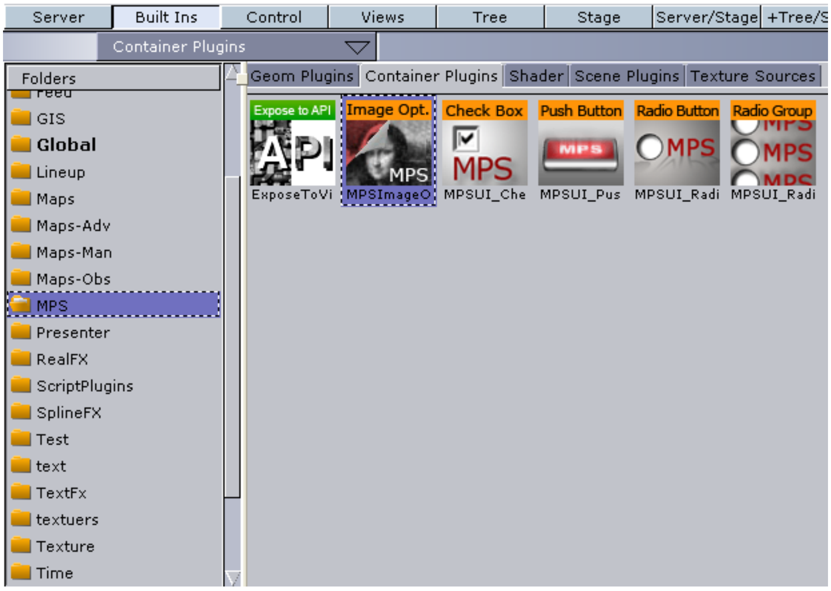
Scaling % 50 R

Quality % 50 R

- **Scaling**: A higher value makes for a larger image. A lower value makes for a smaller image.
- **Quality**: A higher value makes for higher quality. A lower value makes for lower quality.

To optimize images

1. From the main menu, click **Built Ins** and from the drop-down menu select **Container Plugins**.
2. In the Folders pane to the left, click **MPS**.
3. Drag the **ImageOptimization** plugin into the container.



4. Double-click the icon in the container to configure.
5. Adjust the Scaling and Quality as necessary.

Best Practices

To avoid MPS files that are too large, apply this plugin to the image on the exact container you need it, rather than on any parent container.

See Also

- *MPS Client Manual*

12.7.3 MPSUI_CheckBox

This plugin creates check-box for yes/no options. The first mouse click will play the object's animation forward; a second click will play the animation in reverse.

Use this plugin to create interactive buttons. In a scene, it appears as a toggle switch.

When implemented, you can select only one director for each instance of the plugin. If you have more than one instance of this plugin in your scene, you can click multiple buttons to view all animations. They function like a check-box option, that is to say, that you can make multiple selections simultaneously.

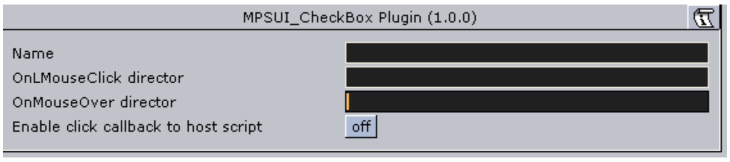
Clicking on the enabled portion in the scene sends the `Control.GetInteractivityObject()` parameter in the relevant Vizky Callback Function to JavaScript.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\MPSCheckBox.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To implement the MPSUI_CheckBox plugin
- Best Practices

Properties and Parameters


MPSUI_CheckBox Plugin (1.0.0)

Name

OnLMouseClick director

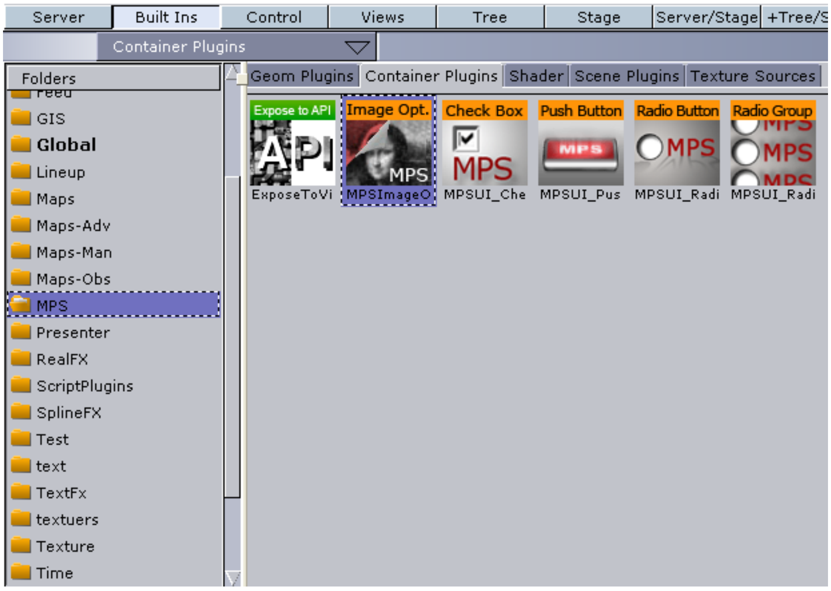
OnMouseOver director

Enable click callback to host script: off

- **Name**: The name assigned in this field can be called in the HTML JavaScript function. To get this name in the resulting JavaScript, call the API function `GetInteractivityObject`.
- **OnLMouseClick director**: The name assigned in this field indicates the animation director name activated by a left-click of the mouse.
- **OnMouseOver director**: The name assigned in this field indicates the animation director name activated by moving the mouse over the object.
- **Enable click callback to host script**: When set to **On**, the Vizky client will recognize Vizky Callback Functions.

To implement the MPSUI_CheckBox plugin

1. From the main menu, click **Built Ins** and from the drop-down menu select **Container Plugins**.
2. In the Folders pane to the left, click **MPS**.
3. Drag the **MPSUI_CheckBox** plugin into the container.



4. Double-click the icon in the container to configure the Properties and Parameters.
5. Create the director's animation.
6. In the lower-mid panel, click the E button to handle script/plug-in events.
7. Mouse over the scene and click the button to test the animation.

   The animation will play when you click the button. The animation plays in reverse when you click the button again.

Best Practices

Check-box is akin to an on/off switch. You click once to run a director; you click the same one again to run it in reverse. However, if you do not click again to play that director in reverse, when you click another check-box in the scene, the animations may conflict. Therefore, if you wish to have the animation reversed automatically upon clicking another option, use the MPSUI_RadioButton instead.

See Also

- *Vizky Callback Functions* in the *MPS Client Manual*

12.7.4 MPSUI_PushButton

This plugin creates interactive buttons, with mouse over and click functionality. In one modality, when clicking and holding down the button, the animation will play forward, and when the mouse is released, it will play in reverse. The other modality allows you to click once to play the animation; a second click does not play it in reverse.

Use this plugin to create interactive buttons. In a scene, it appears as a push button.

When implemented, you can select only one director for each instance of the plugin. If you have more than one instance of this plugin in your scene, you can click multiple buttons to view all animations.

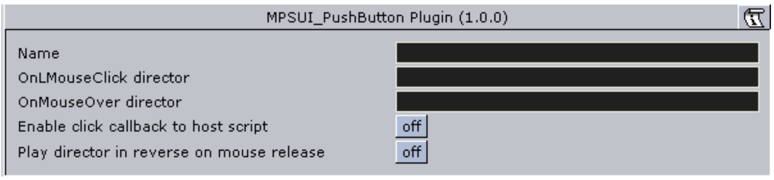
Clicking on the enabled portion in the scene sends the `Control.GetInteractivityObject()` parameter in the relevant Vizky Callback Function to JavaScript.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\MPSPushButton.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To implement the MPSUI_PushButton plugin
- Best Practices

Properties and Parameters



MPSUI_PushButton Plugin (1.0.0)

Name

OnLMouseClick director

OnMouseOver director

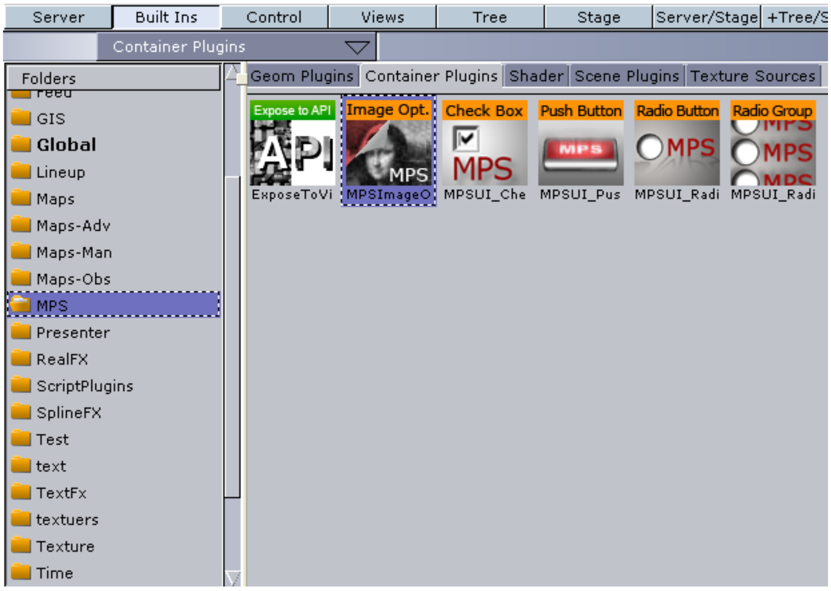
Enable click callback to host script: off

Play director in reverse on mouse release: off

- **Name**: The name assigned in this field can be called in the HTML JavaScript function. To get this name in the resulting JavaScript, call the API function `GetInteractivityObject`.
- **OnLMouseClick director**: The name assigned in this field indicates the animation director name activated by a left-click of the mouse.
- **OnMouseOver director**: The name assigned in this field indicates the animation director name activated by moving the mouse over the object.
- **Enable click callback to host script**: When set to **On**, the Vizky client will recognize Vizky Callback Functions.
- **Play director in reverse on mouse release**: When set to **On**, the director will play in reverse when the mouse is released.

To implement the MPSUI_PushButton plugin

1. From the main menu, click **Built Ins** and from the drop-down menu select **Container Plugins**.
2. In the Folders pane to the left, click **MPS**.
3. Drag the **MPSUI_PushButton** plugin into the container.



4. Double-click the icon in the container to configure the Properties and Parameters.
5. Create the director's animation.
6. In the lower-mid panel, click the E button to handle script/plug-in events.
7. Mouse over the scene and click the button to test the animation.

   The animation will play when you click and hold the button.

Best Practices

Keep animations short and simple with this plugin, as users will not usually hold down the mouse button.

See Also

- *Vizky Callback Functions* in the *MPS Client Manual*

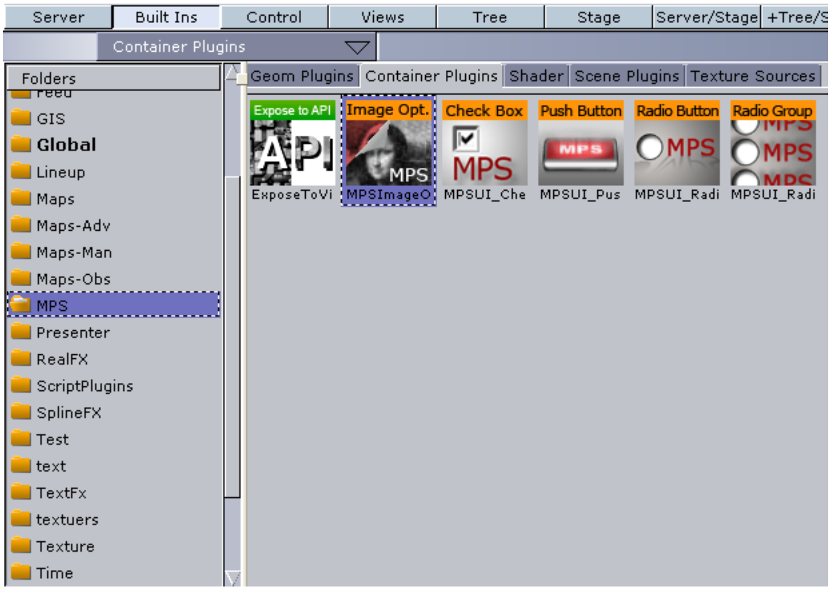
12.7.5 MPSUI_RadioGroup

This plugin enables a series of radio buttons to select an option on screen. The nature of this plugin requires that parent-child container be created to govern the properties and parameters. When the MPSUI_RadioGroup plugin is dragged onto the parent folder, all child folders receiveMPSUI_RadioButton. Furthermore, the configuration is done in each respective child container only. As such, there are no parameters to configure in MPSUI_RadioGroup.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\MPSRadioGroup.vip`.

To implement MPSUI_RadioGroup

1. Click the Create Group button to create a root container. See Adding Empty Containers.
2. Create as many child containers below the parent for options to be offered in the scene.
3. From the main menu, click **Built Ins** and from the drop-down menu select **Container Plugins**.
4. In the Folders pane to the left, click **MPS**.
5. Drag the **MPSUI_RadioGroup** plugin into the parent container.



6. Configure the respective child containers. See MPSUI_RadioButton.

12.7.6 MPSUI_RadioButton

This plugin creates radio button to select an on-screen option. When clicking on one radio button, its animation will play forward, while the other radio buttons do nothing. When another button is selected, the previous animation will play in reverse as the current selected animation is played.

You do not necessarily need to drag MPSUI_RadioButton onto any container, as it is inherited from the parent container when MPSUI_RadioGroup is implemented. If however you want to create more options, i.e. child containers, then you can independently implement MPSUI_RadioButton onto subsequently created child containers.

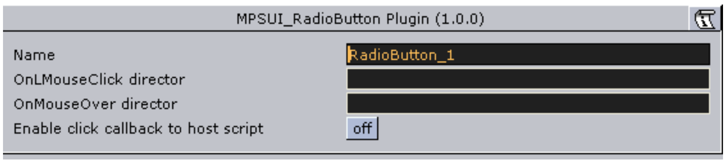
Clicking on the enabled portion in the scene sends the `Control.GetInteractivityObject()` parameter in the relevant Vizky Callback Function to JavaScript.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\MPSRadioButton.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To configure MPSUI_RadioButton parameters
- To implement MPSUI_RadioButton on child containers
- Best Practices

Properties and Parameters



MPSUI_RadioButton Plugin (1.0.0)

Name	RadioButton_1
OnLMouseClick director	
OnMouseOver director	
Enable click callback to host script	off

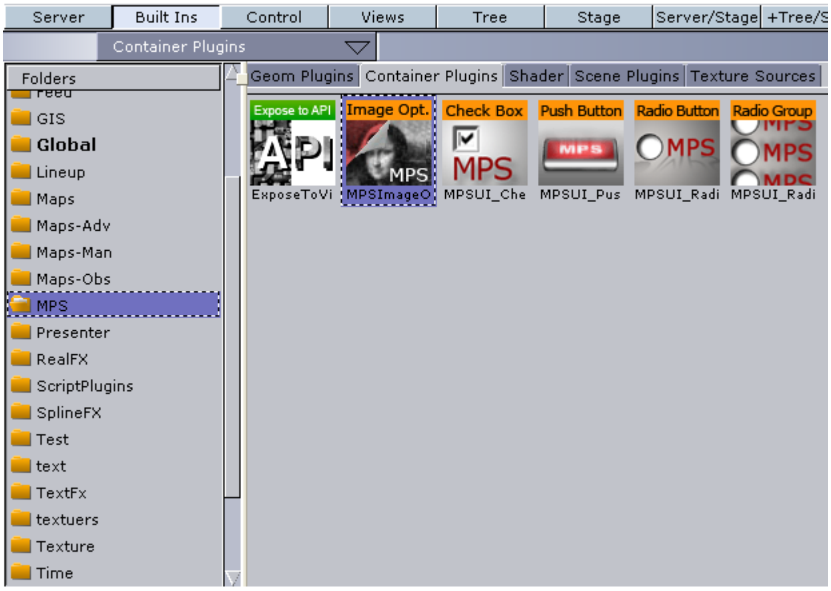
- **Name**: The name assigned in this field can be called in the HTML JavaScript function. To get this name in the resulting JavaScript, call the API function `GetInteractivityObject`.
- **OnLMouseClick director**: The name assigned in this field indicates the animation director name activated by a left-click of the mouse.
- **OnMouseOver director**: The name assigned in this field indicates the animation director name activated by moving the mouse over the object.
- **Enable click callback to host script**: When set to **On**, the Vizky client will recognize Vizky Callback Functions.

To configure MPSUI_RadioButton parameters

1. Perform the procedure To implement MPSUI_RadioGroup.
2. Click on the MPSUI_RadioButton icon in the child container.
3. Configure the respective Properties and Parameters.
4. Repeat for each respective child container.

To implement MPSUI_RadioButton on child containers

1. Under your MPSUI_RadioGroup parent container, create another child container.
2. From the main menu, click **Built Ins** and from the drop-down menu select **Container Plugins.**
3. In the Folders pane to the left, click **MPS.**
4. Drag the **MPSUI_RadioButton** plugin into the newly created child container.



5. Configure the respective Properties and Parameters.

Best Practices

When you click a button, it will play the director, and when you click another it will play the first one in reverse as it plays the next one. Therefore, keep animation movements simple in order to minimize visual conflicts with the other animations in the scene.

See Also

- MPSUI_RadioGroup
- *Vizky Callback Functions* in the *MPS Client Manual*

12.8 Presenter

The following container plugins are housed in the Presenter folder:

- Bar
- Bar Value
- Bar Values
- Pie Slice
- Pie Values

12.8.1 Bar

The Bar plugin tells the Bar Values plugin that the container is part of the bar chart and it also tells which elements of the geometry on the container to control.

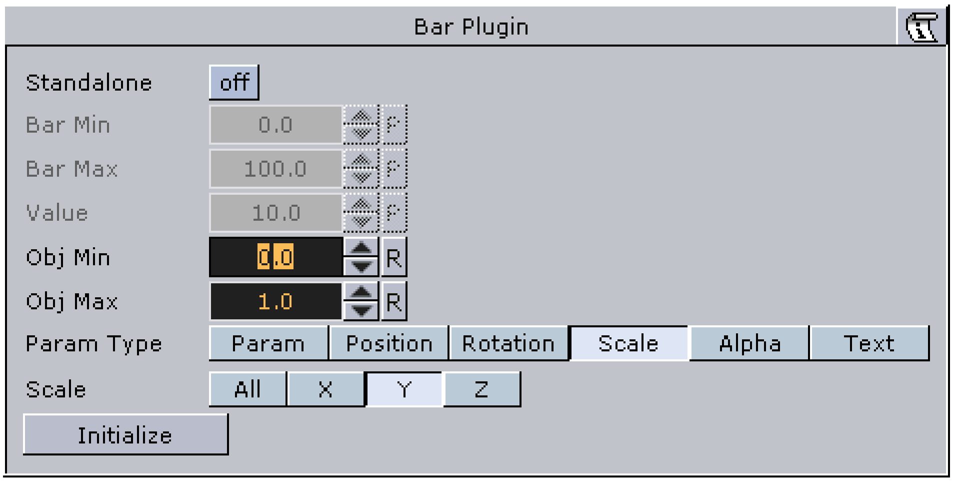
Each container that is to be a bar, must hold a geometry and a Bar plugin. Bar can be used as a standalone plugin.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\Bar.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



Bar Plugin

Standalone: off
Bar Min: 0.0
Bar Max: 100.0
Value: 10.0
Obj Min: 0.0
Obj Max: 1.0
Param Type: Param | Position | Rotation | Scale | Alpha | Text
Scale: All | X | Y | Z

- **Standalone**: If the Bar plugin is going to be used without a Pie Values plugin to set the values, enable this button. It will enable you to set the value directly on the plugin and you can define normalization factors between the input values and the value that is being set on the geometry.
- **Bar Min** / **Max**: If in standalone, these values set the minimum and maximum values for the bar. These values are connected to the Obj Min/Max values and by adjusting these together you can get the desired effect between input values and visual effect.
- **Value**: Sets the scaling value of the object.
- **Obj Min** / **Max**: Maps the input values to the native values on the object being controlled by the plug-in. Adjust these together with Bar Max/Min on the Bar Values plugin or on the Bar plugin itself if in standalone mode, to get the desired effect between input values and visual effect.
- **Param Type**: Sets the type of value to control on the object.
  - **Param**: Controls a named parameter on the object. Use the console to find the correct name for the parameter: Enable “Show commands”, Open the object’s editor, change the parameter and see in the console what the parameter name is.
  - **Position**: Position values can be set to control the X, Y or Z axis.
  - **Rotation**: Rotation values can be set to control the X, Y or Z axis.
  - **Scale**: Scale values can be set to control All, X, Y or Z axis.
  - **Text**: Text values can be shown as Integer, Float or Formatted text.
- **Initialize**: Enables the plug-in to initialize and gain control of the parameters.

See Also

- Bar Value
- Bar Values
- Vizual Data Tools

12.8.2 Bar Value

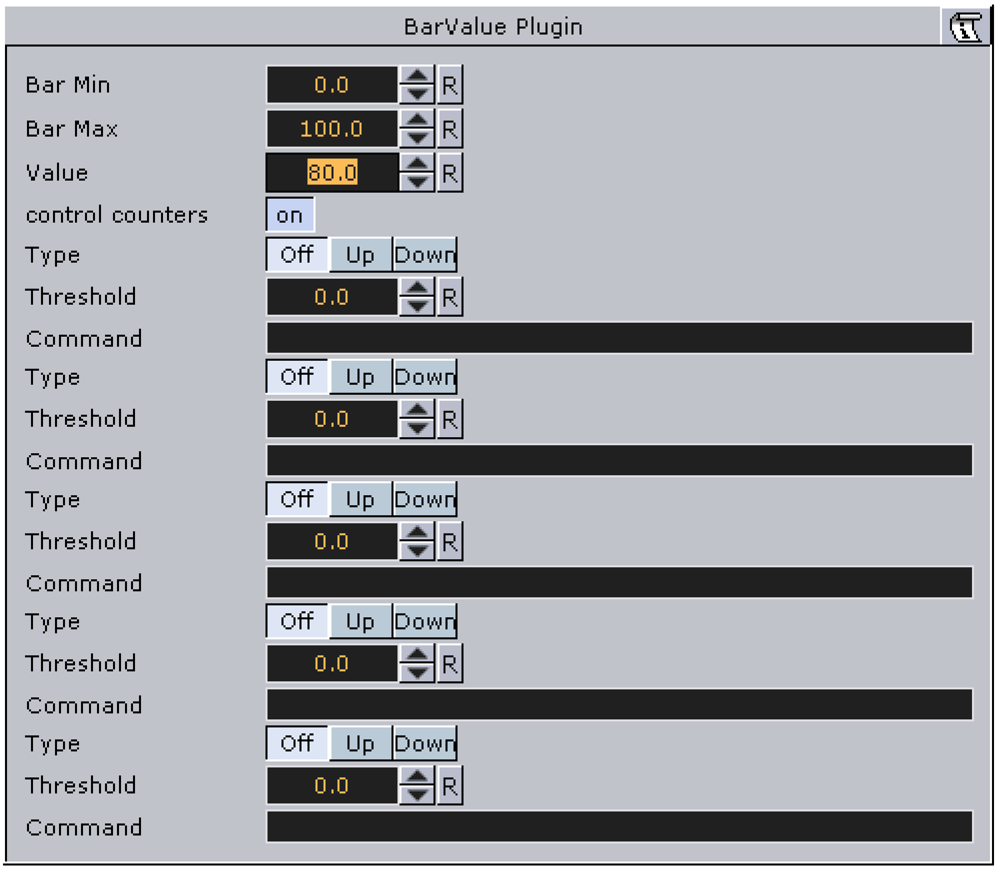
Bar Value controls the value of all Bar plug-ins and Counters of its child nodes. In addition a set of commands can be specified that are executed when the value passes certain thresholds.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\BarValue.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



BarValue Plugin

Bar Min: 0.0
Bar Max: 100.0
Value: 80.0
control counters: on
Type: Off Up Down
Threshold: 0.0
Command:
Type: Off Up Down
Threshold: 0.0
Command:
Type: Off Up Down
Threshold: 0.0
Command:
Type: Off Up Down
Threshold: 0.0
Command:
Type: Off Up Down
Threshold: 0.0
Command:

- **Bar Min**: Minimum Value of the bar.

- **Bar Max**: Maximum Value of the bar.
- **Value**: The actual value of the bar.
- **control counters**: States if counters should be affected. If this is checked, Value will be displayed in all sub containers with Counter plug-ins.
- **Type**: Determines whether the command should fire if the threshold is passed from below, or from above. Off does not execute the command at all. Available options are Off, Up and Down.
- **Threshold**: The threshold against Value is compared.
- **Command**: The Viz command that will be executed when the threshold is passed.

To create a bar that displays its value

1. Create a Container and add the Bar Value plugin to it.
2. Open the Bar Value editor, and check control counters.
3. Set Threshold to 20, Type to Up and set a command (e.g. `MAIN*ONAIR GET_INFO`).
4. Add two sub-containers.
5. Add a Font and a Counter plugin to the first sub-container.
6. Add a Cube and a Bar plugin to the second sub-container.
7. Set the Cube's Y-Axis Center to (B)ottom.
8. Open the Bar plugin editor and set Param Type to Scale and Scale to Y.
9. Modify Bar Value's Value field.
   - If the value exceeds 20 the information will be written to the console.

See Also

- Bar
- Bar Values
- Vizual Data Tools

12.8.3 Bar Values

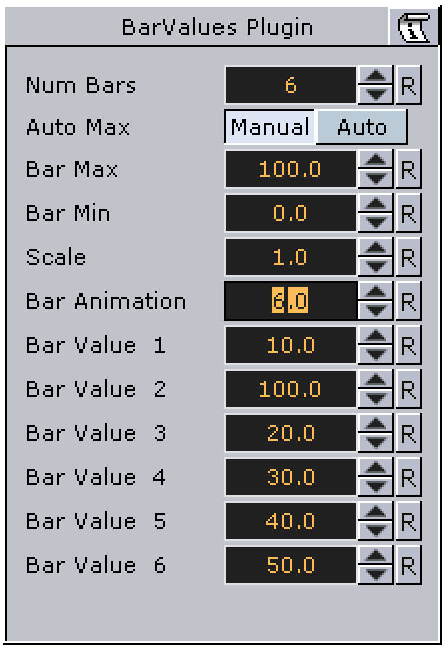
The Bar Values plugin communicates with all sub containers that hold a Bar plugin and lets you set values for each of them.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\BarValues.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



BarValues Plugin

Num Bars 6

Auto Max Manual Auto

Bar Max 100.0

Bar Min 0.0

Scale 1.0

Bar Animation 6.0

Bar Value 1 10.0

Bar Value 2 100.0

Bar Value 3 20.0

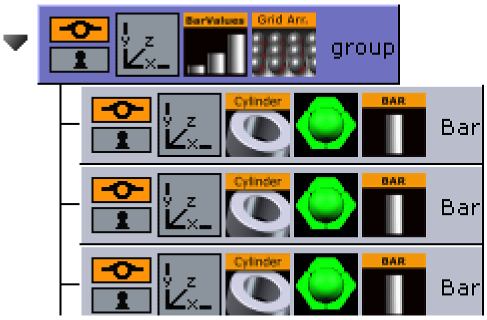
Bar Value 4 30.0

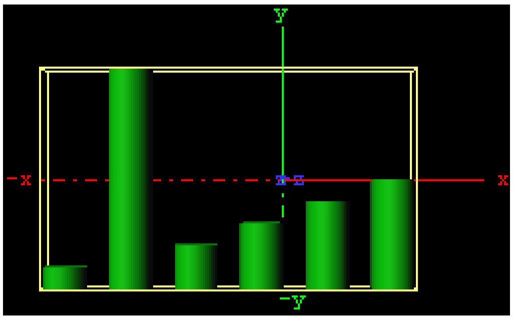
Bar Value 5 40.0

Bar Value 6 50.0

- **Num Bars**: Sets the number of bars to be controlled by the Bar Values plug-in. Maximum number of bars is 25.
- **Auto Max**: Allows the user to switch between Manual and Auto mode. Auto mode disables the Bar Max value.
- **Bar Max**: Sets the maximum values of the bar.
- **Bar Min**: Sets the minimum value of the bar.
- **Scale**: Sets the scale of the object.
- **Bar Animation**: Sets the number of bars to be animated.
- **Bar Value 1-25**: Sets the value for the bar(s).

To create a bar chart using the Bar Values plug-in





1. Add a group container to the scene tree, and add the Bar Values and Grid Arrange plug-in to it.
2. Add a group container as a sub container to the first, and name it **Bar**.
3. Add material, the Cylinder and Bar plug-in to the Bar container.
4. Open the **Cylinder** editor and set Center to **Bottom**.
5. Open the **Bar** editor, set Param Type to **Scale** and Scale to **Y**, and click **Initialize**.

6. Create and place a number of copies of the Bar container (e.g. 5) at the same level as the Bar container.
7. Open the Grid Arrange editor and set **Number of Columns** to 6 and **Column offset** to `30.0`.
8. Open the Bar Values editor and set **Num Bars** to 6 and **Bar Animation** to 6.

See Also

- Vizual Data Tools
- Control Bars
- Bar
- Bar Value
- Pie Slice
- Pie Values

12.8.4 Pie Slice

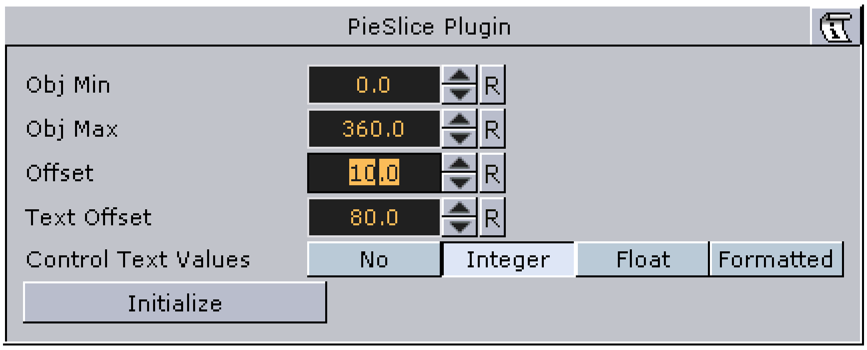
Use the Pie Slice plug-in with the Pie Values plug-in to create and animate a pie chart of up to 20 slices.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\PieSlice.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



PieSlice Plugin

Obj Min 0.0

Obj Max 360.0

Offset 10.0

Text Offset 80.0

Control Text Values: No | Integer | Float | Formatted

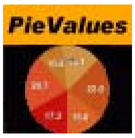
- **Obj Min** / **Max**: Maps the input values to the values on the cylinder object. Adjust these together with Bar Max/Min on the Pie Values plugin to get the desired effect between input values and visual effect.
- **Offset**: Offsets the slice from the center of the pie chart.
- **Text Offset**: Offsets the text label from the center of the pie chart.
- **Control Text Values**: Sets the type of text. Available options are No, Integer, Float and Formatted.

- **Initialize**: Initializes the value of the pie slice.

See Also

- Vizual Data Tools
- Control Bars
- Bar
- Bar Value
- Bar Values
- Pie Values

12.8.5 Pie Values



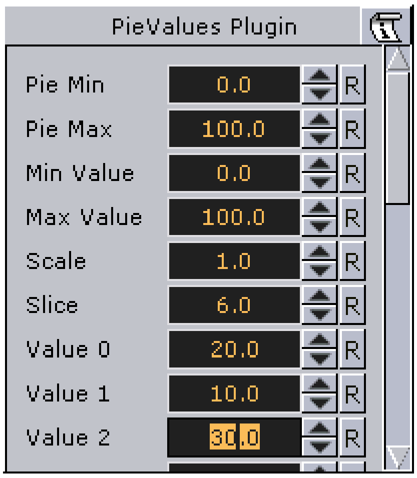
Use the Pie Values plug-in with the Pie Slice plug-in to create and animate a pie chart of up to 20 slices.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\PieValues.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a pie chart using the Pie Values plug-in

Properties and Parameters

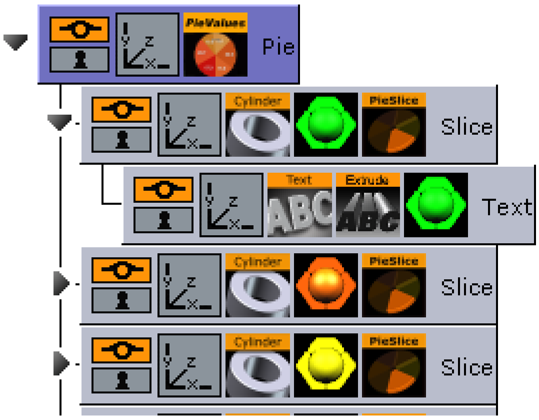


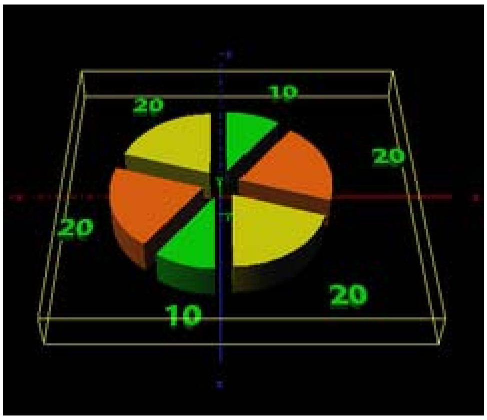
PieValues Plugin

Pie Min	0.0
Pie Max	100.0
Min Value	0.0
Max Value	100.0
Scale	1.0
Slice	6.0
Value 0	20.0
Value 1	10.0
Value 2	30.0

- **Pie Min and Max**: Sets the minimum and maximum values for the pie size and the range of the size.
- **Min and Max Value**: Sets the range of the displayed pie.
- **Scale**: Sets the scale of the pie. Default is 1.
- **Slice**: Sets the number of slices to be shown as part of the pie.
- **Value 0 - 19**: Sets the slice value. Available number of slices is 20.

To create a pie chart using the Pie Values plug-in





1. Add a group container to the scene tree, and name it **Pie**.
2. Open the transformation editor and set **Rotation X** to **45.0**, and **Scaling Y** (single) to **0.2**.
3. Add the Pie Values plug-in to the Pie container.
4. Open the Pie Values editor and set **Slice** to **6.0**, and enter the following values in **Value 0** to **5**; 10.0, 20.0, 20.0, 10.0, 20.0, 20.0.
5. Add a sub container to the Pie container and name it **Slice**.
6. Add the Cylinder geometry plug-in, material and PieSlice plug-in to the Slice container.
7. Open the PieSlice editor and set Offset to 10.0 and Text Offset to 80.0.
8. Add a sub container to the slice container and name it **Text**.
9. Open the transformation editor for the Text container and set **Scaling** (locked) to **0.2** and **Rotation X** to -90.0.
10. Add a font, material and the Extrude plug-in to the Text container.
11. Open the Extrude editor and set **Extrusion Depth** to **100.0**.
12. Create and place a number of copies of the Slice container (e.g. 5) at the same level as the Slice container.
13. Change the colors of the pie slices to distinguish each slice.

See Also

- Vizual Data Tools
- Control Pie
- Bar
- Bar Value
- Bar Values
- Pie Slice

12.9 Script Plugins

This is an empty folder for you to save your own Script plug-ins into.

12.10 Spline FX

The following container plugins are housed in the SplineFX folder:

- 2D Follow

12.10.1 2D Follow

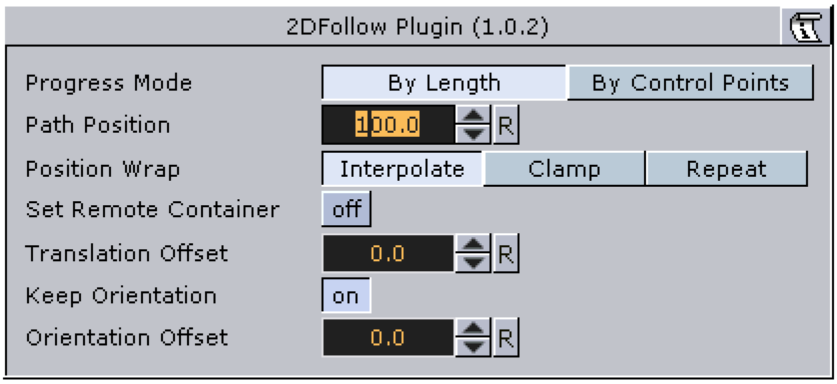
The 2D Follow function works together with the 2D Ribbon. What the plug-in basically does is to automatically create an animation path that follows the form of a 2D Ribbon. The 2DRibbon must be created first.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\2DFollow.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To set up the 2D Follow plugin

Properties and Parameters

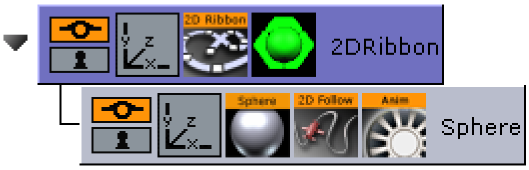


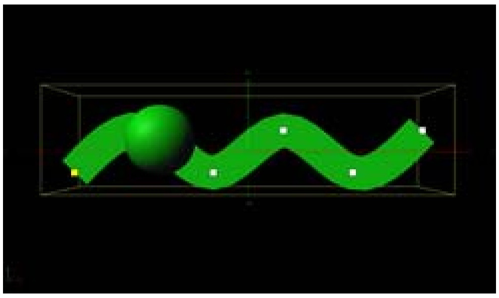
2DFollow Plugin (1.0.2)

Progress Mode: By Length | By Control Points
Path Position: 100.0
Position Wrap: Interpolate | Clamp | Repeat
Set Remote Container: off
Translation Offset: 0.0
Keep Orientation: on
Orientation Offset: 0.0

- **Path Position**: Sets the position for the object along the path of the 2D Ribbon object. Animate this value to get an animation in the stage.
- **Position Wrap**: Can be set to **Interpolate**, **Clamp** or **Repeat**.
- **Translation Offset**: Sets an offset between the object and the 2D Ribbon path.
- **Keep Orientation**: If enabled, the orientation between the object and the 2D Ribbon path is kept as the object moves along the path.
- **Orientation Offset**: Offsets the orientation between the object and the 2D Ribbon path.

To set up the 2D Follow plugin





1. Add the 2D Ribbon geometry to the scene tree.
2. Open the 2D Ribbon editor and enable **Show Control Point Values**.
3. Set alternating values of 30.0 and -30.0 to the Y axis values, creating a wave shape.
4. Add the element(s) (e.g. a Sphere) you want to animate along the 2D Ribbon’s path as a sub container to the 2D Ribbon container.
5. Add the 2D Follow plug-in to the container.
6. Open the 2D Follow editor and animate its Path Position value from 0.0 to 100.0.

See Also

- 2D Ribbon
- Creating Animations

12.11 Text

The following container plugins are housed in the Text folder:

- Mark Text

12.11.1 Mark Text

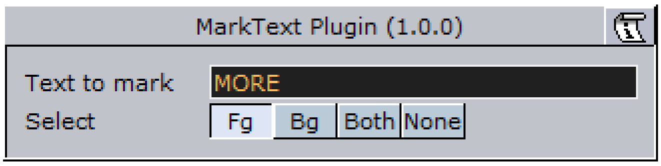
The Mark Text plug-in allows you to underline words in a text string with a.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\markText.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To mark text

Properties and Parameters



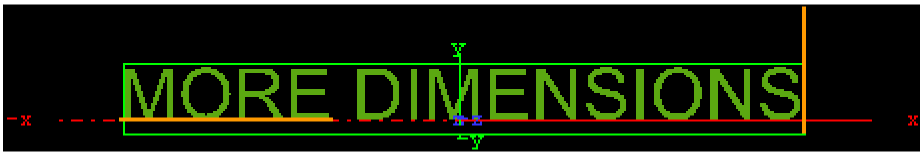
MarkText Plugin (1.0.0)

Text to mark: MORE

Select: Fg | Bg | Both | None

- **Text to Mark**: Sets the text that should be marked. Note that the text is case sensitive.
- **Select**: Selects where the underline should be placed. Available options are foreground (Fg), background (Bg), Both and None. Background will underline the text's shadow.

To mark text



1. Add a group container to the scene tree and name it Text.
2. Add a font to the Text container.
3. Open the font editor and enter a text string.
4. Add the Mark Text plug-in to the Text container.
5. Open the Mark Text editor and enter a word from your text string.

Note: Do not change the font type between texture and geometry as this will not affect the whole string when using the Mark Text plug-in.

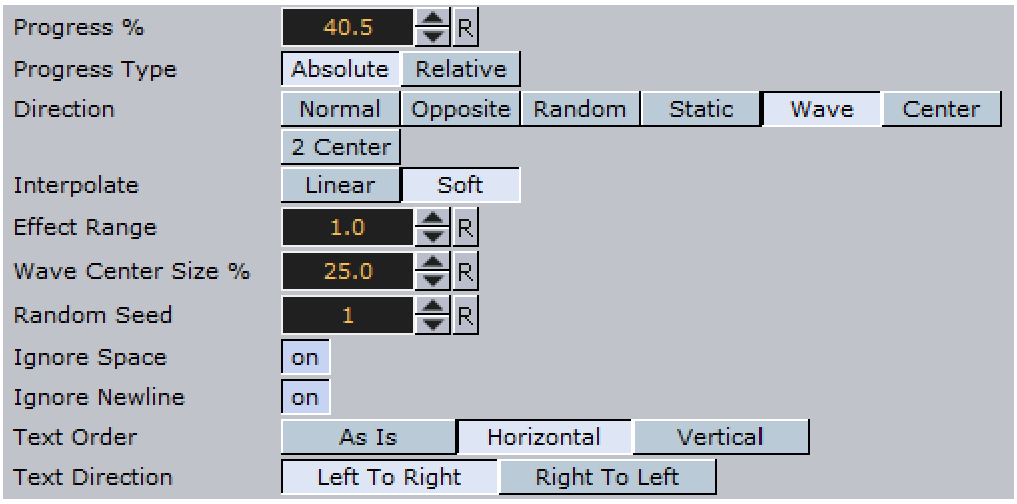
12.12 Text FX

The following container plugins are housed in the TextFX folder:

- Common Text FX Properties and Parameters
- Convert Case
- Text FX Alpha
- Text FX Arrange
- Text FX Color
- Text FX Color Per Vertex
- Text FX Emoticons
- Text FX Explode
- Text FX Jitter Alpha
- Text FX Jitter Position
- Text FX Jitter Scale
- Text FX Plus Plus
- Text FX Rotate
- Text FX Scale

- Text FX Size
- Text FX Slide
- Text FX Vertex Explode
- Text FX Write

12.12.1 Common Text FX Properties and Parameters



Progress %	40.5 R
Progress Type	Absolute / Relative
Direction	Normal / Opposite / Random / Static / Wave / Center / 2 Center
Interpolate	Linear / Soft
Effect Range	1.0 R
Wave Center Size %	25.0 R
Random Seed	1 R
Ignore Space	on
Ignore Newline	on
Text Order	As Is / Horizontal / Vertical
Text Direction	Left To Right / Right To Left

The following properties and parameters are common to most of the Text FX plug-ins:

- **Progress %**: 0% progress is the beginning of the effect, 100% the end. Animate this value from 0% to 100% to see the effect or from 100% to 0% to animate the effect backwards.
- **Progress Type**
  - **Absolute**: 100% progress will animate all characters of a text, regardless how many characters it has.
  - **Relative**: 100% progress animates 10 characters. This is needed to adjust the timing of several text objects with different sizes. The effect speed should be for example 5 characters per second, so the animation must be from 0% to 100% in two seconds. This will work for text with 10 characters or less. If you want to use longer texts, animate the progress value over 100% (10% for each character).
- **Direction**: Sets the direction of the text effect sequence, you can choose between the following options:
  - **Left**: Starts with the first character.
  - **Right**: Starts with the last character
  - **Random**: Uses a random order.
  - **Static**: All characters are processed at the same time.
  - **Wave**: Starts with the first character, animates the effect from 0: 100% and then down again to 0%.
  - **Center**: Starts the effect from the center of the text.
  - **2 Center**: Starts the effect at the same time from the beginning and the end of the text. They meet at the center.
- **Interpolate**: Choose between a **soft** or a **linear** interpolation of the transition from character to character.

- **Effect Range**: Defines how many characters are processed at the same time. If for example the Effect Range is set to 4, and you manually increase the progress value, you will see that when the fifth character starts to be processed, the first is finished, when the sixth starts, the second is finished, and so on.
- **Wave Center Size %** this parameter is only visible when **Wave** is chosen as the effect direction. Its effect is only visible when the effect range is set relatively high. It defines how quickly the effect is to reach the top of the wave sequence or in other words, how wide the wave top is to be. If for instance the **Effect Range** is set to 10 and the **Wave Center Size %** is set to 80, all characters except the one at each end will be at the top of the wave sequence. The one at the left end will be on its way up and the one on the right end will be on its way down.
- **Random Seed**: Specifies a seed for the random number generator when a random direction is chosen. Even though Viz Artist uses random numbers, the animation for a specific random seed will always look the same. This is typically useful if you combine two different text effects.
- **Ignore Space**: Ignores space when animating the effect.
- **Ignore Newline**: Ignores new lines when animating the effect.
- **Text Order**: Sets the text order for the effect. Available options are As Is, Horizontal or Vertical. Horizontal and Vertical enables the Text Direction options.
- **Text Direction**: Sets the horizontal and vertical text direction (see Text Order). When the text order is set to Horizontal, the text direction must be set to Left to Right or Right to Left. When the text order is set to Vertical, the text direction must be set to Up to Down or Down to Up.

12.12.2 Convert Case

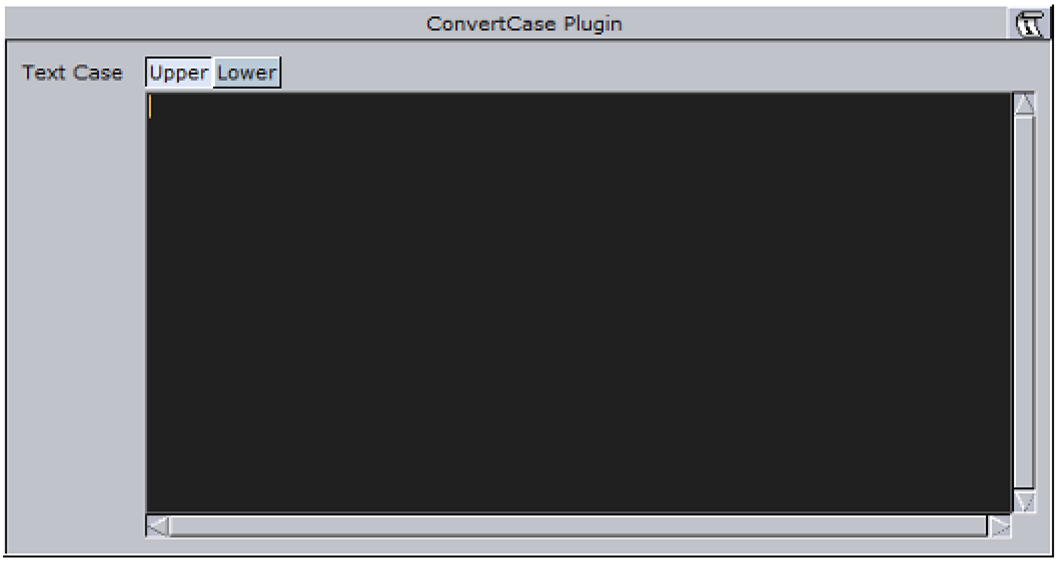
Converts ASCII strings to upper or lower case. Current container must hold a string.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\ConvertCase.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To convert a string to upper or lower case

Properties and Parameters



- **Upper/Lower**: Converts the text for the specific container to upper or lower case.

```
FUNCTION*ConvertCase*text_case - int; 0: upper, 1: lower
```

```
FUNCTION*ConvertCase*text - string
```

To convert a string to upper or lower case

1. Add a group to the scene tree, and add a font to it.
2. Add the Convert Case plug-in to the same container.
3. Open the Convert Case editor, and enter text in the editor.
4. Click the Upper button to convert to upper case text, or click the Lower button to convert to lower case text.

12.12.3 Text FX Alpha

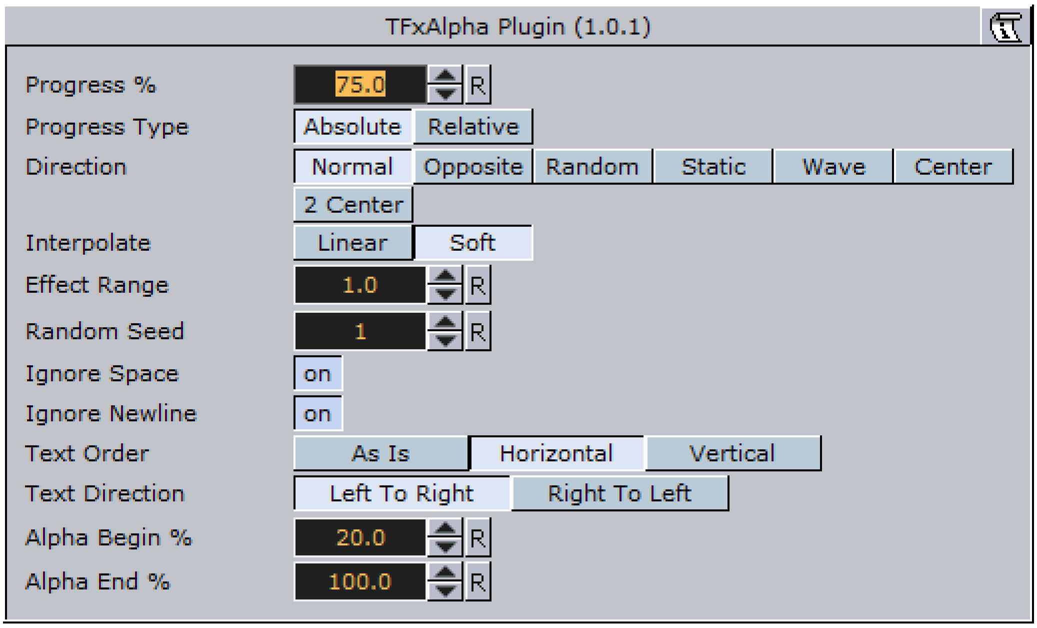
The Text FX Alpha plug-in is a text effect that creates a fade in effect for the text characters. The effect sequence can be set to go many different ways. A fade in and out effect can also be achieved.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\TextFxAlpha.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters


TFxAlpha Plugin (1.0.1)

Progress % 75.0 R
Progress Type: Absolute | Relative
Direction: Normal | Opposite | Random | Static | Wave | Center | 2 Center
Interpolate: Linear | Soft
Effect Range 1.0 R
Random Seed 1 R
Ignore Space on
Ignore Newline on
Text Order: As Is | Horizontal | Vertical
Text Direction: Left To Right | Right To Left
Alpha Begin % 20.0 R
Alpha End % 100.0 R

- **Alpha Begin %**: Sets the alpha level of the character at 0% effect.
- **Alpha End %**: Sets the alpha level of the character at 100% effect.

Common Text FX Properties and Parameters.

12.12.4 Text FX Arrange

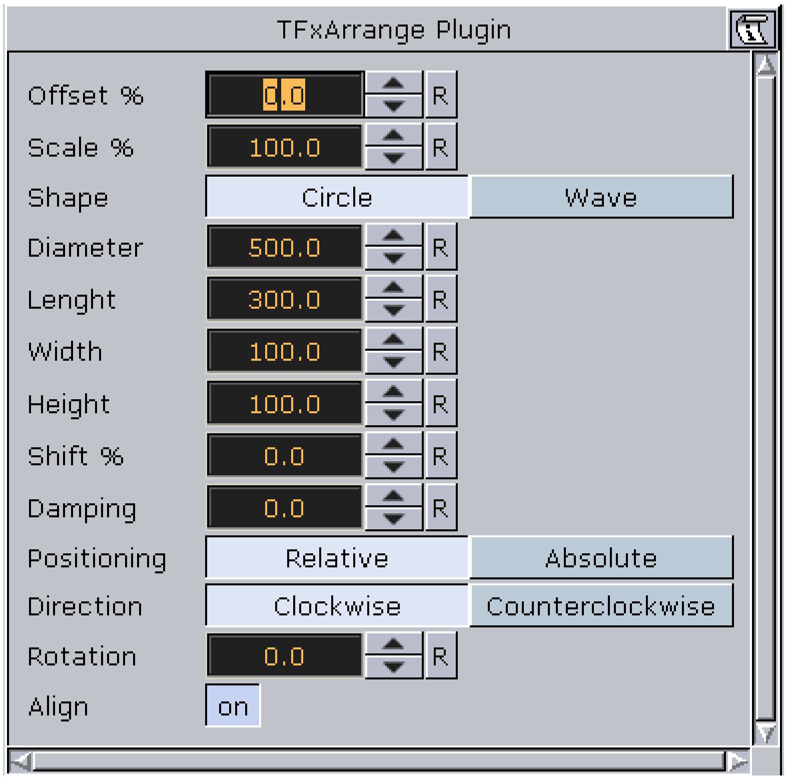
The Text FX Arrange plug-in allows you to arrange characters in either a circular- or a wave shape. The characters can be animated on the selected shape by animating the offset value.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\TextFxArrange.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



TFxArrange Plugin

Offset % 0.0 R
Scale % 100.0 R
Shape Circle Wave
Diameter 500.0 R
Lenght 300.0 R
Width 100.0 R
Height 100.0 R
Shift % 0.0 R
Damping 0.0 R
Positioning Relative Absolute
Direction Clockwise Counterclockwise
Rotation 0.0 R
Align on

- **Offset %**: Moves the characters on the shape. 100% means one full rotation of the text on the circle.
- **Scale %**: Sets the text on the shape. The parameter does not scale the characters itself, but the kerning of the characters.
- **Shape**: Changes the shape of the character layout. The options are **Circle** or **Wave.**
- **Rotation**: Rotates the characters on the X-axis.
- **Align**: Rotates the characters on the Z-axis to align the X-axis with the tangent of the shape at the characters position.
- **Circle Parameters**
  - **Diameter** sets the diameter of the circular shape
  - **Positioning** defines the position of the text on the circular shape. **Relative** means that the text spacing of the text object is maintained. **Absolute** means that the text is evenly distributed out on the circle.
  - **Direction** sets the direction of the text on the circle to either **Clockwise** or **Counterclockwise.**
- **Wave Parameters**
  - **Length** sets the amplitude/wave length. A high value creates many and small waves, a low value fewer and bigger waves.
  - **Width** the width of the wave.
  - **Height** sets the height of the waves without altering the number of waves as the **Length** does.
  - **Shift %** the wave usually starts at the height 0. The shift value moves that point, thereby moving the texts position on the wave structure. Setting **Shift %** to 100% brings you back to the beginning.
  - **Damping** can be used to modify the amplitude of the wave curves.

12.12.5 Text FX Color

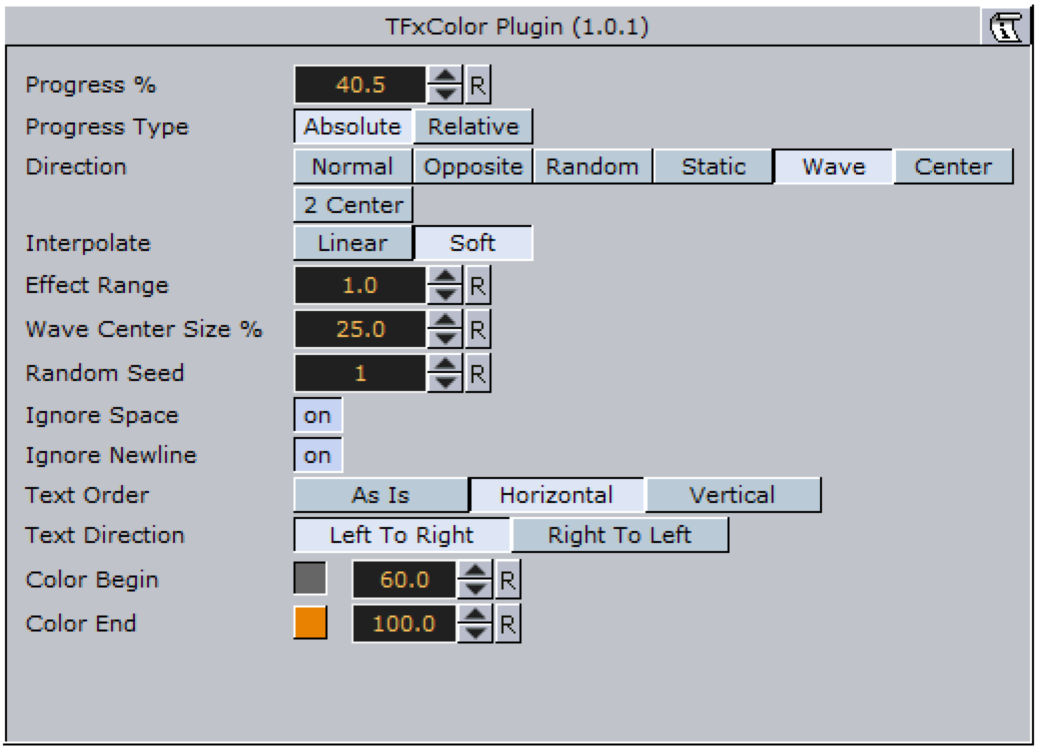
Adds a color effect to the text.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\TextFxColor.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



TFxColor Plugin (1.0.1)

Progress %: 40.5
Progress Type: Absolute | Relative
Direction: Normal | Opposite | Random | Static | Wave | Center | 2 Center
Interpolate: Linear | Soft
Effect Range: 1.0
Wave Center Size %: 25.0
Random Seed: 1
Ignore Space: on
Ignore Newline: on
Text Order: As Is | Horizontal | Vertical
Text Direction: Left To Right | Right To Left
Color Begin: 60.0
Color End: 100.0

- **Color Begin**: Sets the initial color before applying the effect.
- **Color End**: Sets the color the effect should apply.

Common Text FX Properties and Parameters.

12.12.6 Text FX Color Per Vertex

Adds a color per vertex effect to the text.

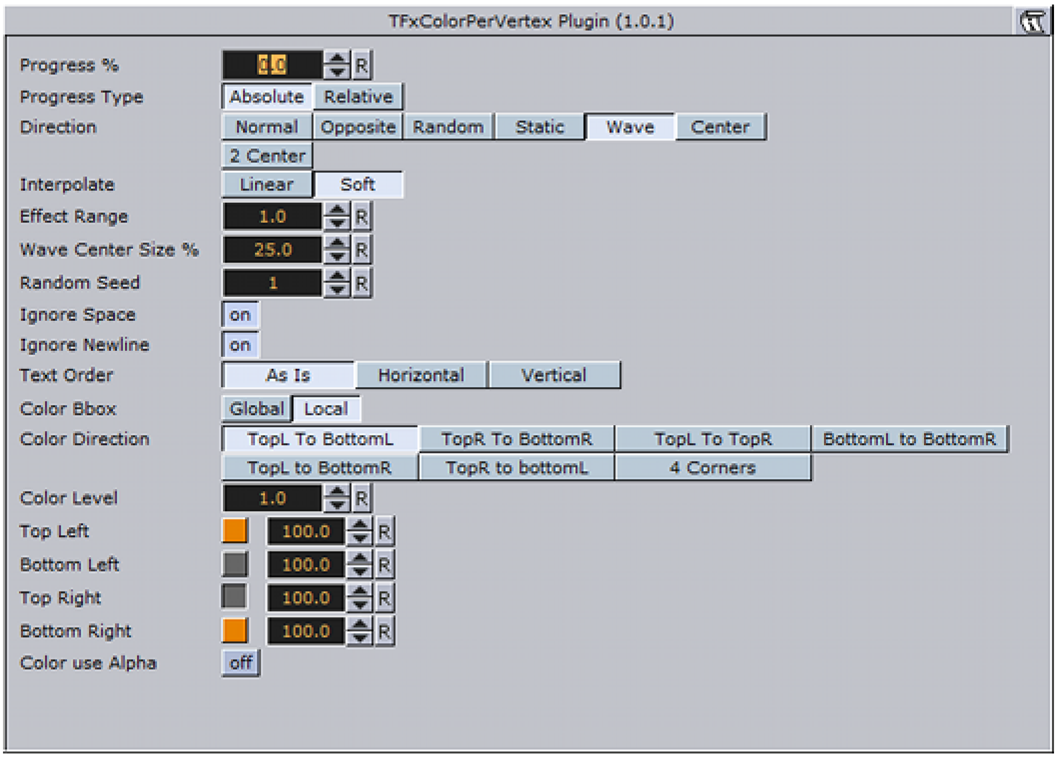
Note: Works only if the text is set to texture (does not work with geometry text).

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\TextFxColorPerVertex.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



TFxColorPerVertex Plugin (1.0.1)

Progress % 0.0
Progress Type: Absolute | Relative
Direction: Normal | Opposite | Random | Static | Wave | Center | 2 Center
Interpolate: Linear | Soft
Effect Range 1.0
Wave Center Size % 25.0
Random Seed 1
Ignore Space on
Ignore Newline on
Text Order: As Is | Horizontal | Vertical
Color Bbox: Global | Local
Color Direction: TopL To BottomL | TopR To BottomR | TopL To TopR | BottomL to BottomR | TopL to BottomR | TopR to bottomL | 4 Corners
Color Level 1.0
Top Left 100.0
Bottom Left 100.0
Top Right 100.0
Bottom Right 100.0
Color use Alpha off

- **Color Bbox**: Defines the reference bounding box, global for the entire text, or local for each character.
- **Color Direction**: Defines the color gradient shape. Available options are; top left to bottom left, top right to bottom right, top left to top right, bottom left to bottom right, top right to bottom left and 4 corners.
- **Color Level**: Sets the level of the color (0.1 to 1.9). Default is 1.0.
- **Top and Bottom Left and Right**: Sets the color and alpha value for each edge.
- **Color use Alpha**: Enables alpha.

Common Text FX Properties and Parameters.

12.12.7 Text FX Emoticons

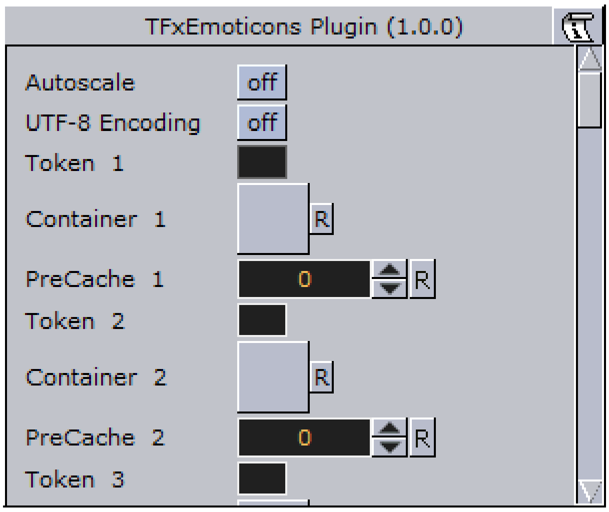
The Text FX Emoticons allows you to create an emoticon container, add it to the Emoticons plug-in and replace characters (tokens) in a text string with an assigned container.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\TextFxEmoticons.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



TFxEmoticons Plugin (1.0.0)

Autoscale off
UTF-8 Encoding off
Token 1
Container 1
PreCache 1 0
Token 2
Container 2
PreCache 2 0
Token 3

- **Autoscale**: Scales the emoticon containers so their height above the baseline (0) is identical to the height of character A.
- **UTF-8 Encoding**: Enables UTF-8 encoding for the text effect.
- **Token 1 - n**: The placeholder for an emoticon.
- **Container 1 - n**: References the container holding the emoticon. The container should be similar to the layout of a glyph.
- **Precache 1-n**: - Enables Viz to cache the objects.

12.12.8 Text FX Explode

The Text FX Explode plug-in creates an explosion like function where the characters get thrown away from their initial position. The speed, direction and spread of the moving characters can be altered with parameters.

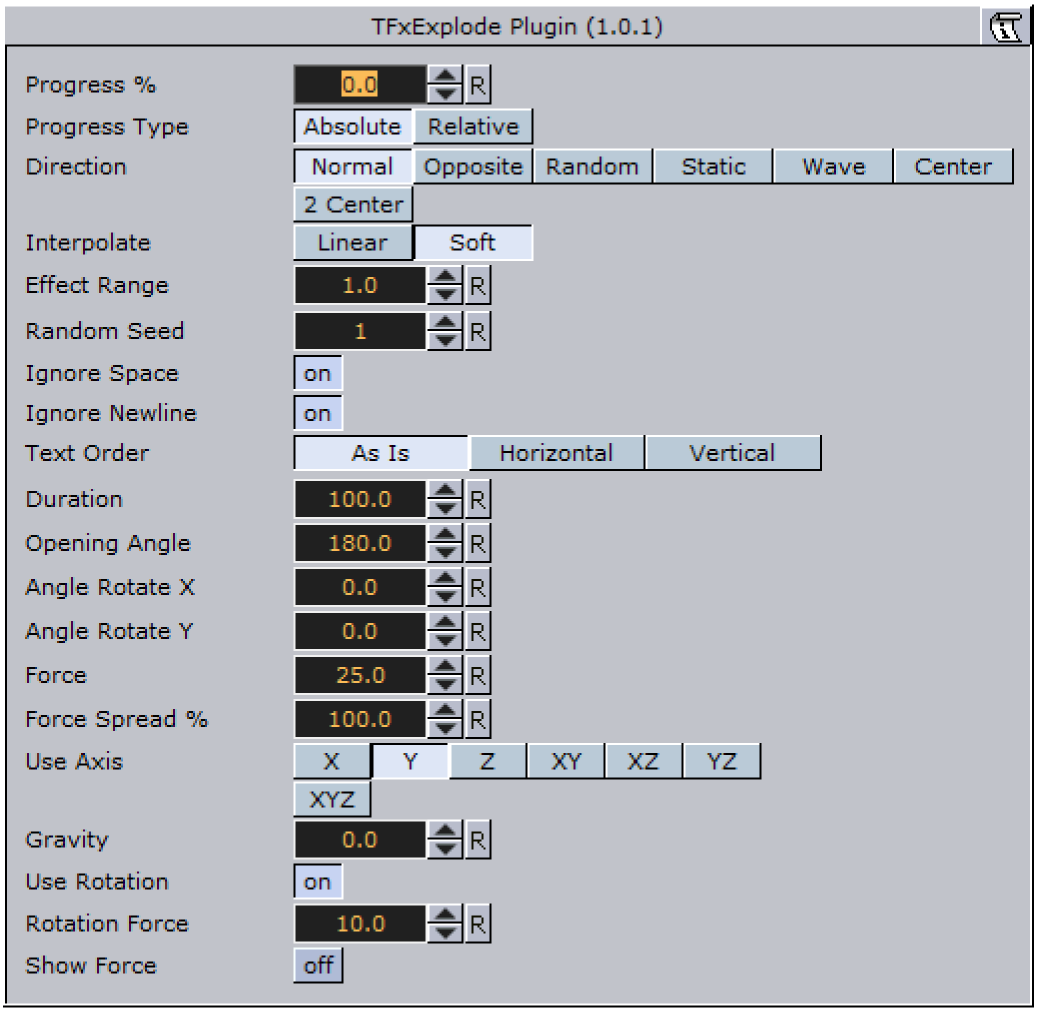
Note: Works only if the text is set to texture (does not work with geometry text).

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\TextFxExplode.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



TFxExplode Plugin (1.0.1)

Progress %	0.0 R
Progress Type	[✓ Absolute] Relative
Direction	[✓ Normal] Opposite Random Static Wave Center 2 Center
Interpolate	Linear [✓ Soft]
Effect Range	1.0 R
Random Seed	1 R
Ignore Space	on
Ignore Newline	on
Text Order	[✓ As Is] Horizontal Vertical
Duration	100.0 R
Opening Angle	180.0 R
Angle Rotate X	0.0 R
Angle Rotate Y	0.0 R
Force	25.0 R
Force Spread %	100.0 R
Use Axis	X [✓ Y] Z XY XZ YZ XYZ
Gravity	0.0 R
Use Rotation	on
Rotation Force	10.0 R
Show Force	off

- **Duration**: Defines the duration of the progressing for each of the characters. If you want a longer progressing, increase this value. You could achieve the same by making the gravity stronger and animating the progress slower, but it is more easy to increase the duration instead if your explode effect is too short.
- **Opening Angle**: Sets the angle for the spread of the characters. 0 will send them straight up, 360 will spread them in a circular shape.
- **Angle Rotate X**: Rotates the opening angle around the X-axis.
- **Angle Rotate Y**: Rotates the opening angle around the Y-axis.
- **Force**: Sets the force that throws away the characters. A high force will make them go far away, conversely a low force will create only a small motion of the characters.
- **Force Spread %**: Sets a variation of the force among the characters.
- **Use Axis**: Allows you to select on which axis or combination of axes the characters are to spread along.
- **Gravity**: Sets a gravity force that influence the path of the characters to end up going downwards. The higher the value is set, the faster each character will divert from its initial path and start going downwards.
- **Use Rotation**: With this option enabled, the characters will rotate as they are being thrown away from their initial position.

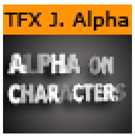
- **Rotation Force**: Sets the degree of rotation as the characters are being thrown away.
- **Show Force**: With this option enabled, lines showing the characters path and speed will be visible in the render output.

Common Text FX Properties and Parameters.

See Also

- Text FX Vertex Explode

12.12.9 Text FX Jitter Alpha



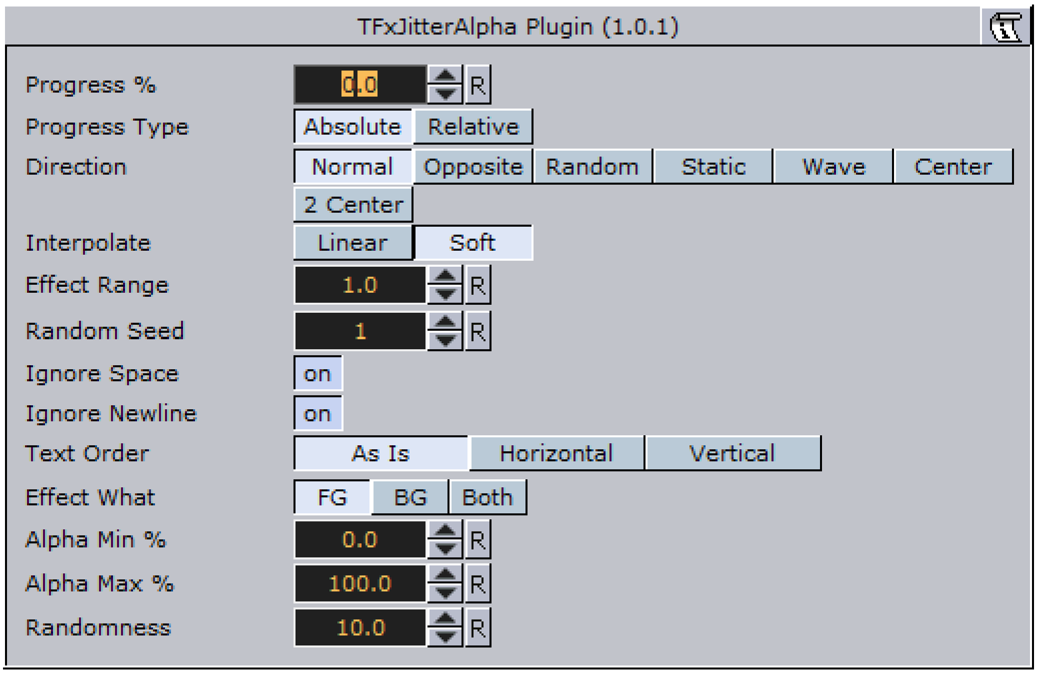
The Text FX Jitter Alpha plug-in creates a jittering motion of the characters by randomly changing the alpha value of each character. The degree of change and the start sequence of the jittering can be altered. To use the plug-in, add it onto a container with a font. To create an animation, animate the progress. Other values can of course be animated as well.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\TextFxJitterAlpha.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



TFxJitterAlpha Plugin (1.0.1)

Progress % 0.0

Progress Type: Absolute | Relative

Direction: Normal | Opposite | Random | Static | Wave | Center | 2 Center

Interpolate: Linear | Soft

Effect Range 1.0

Random Seed 1

Ignore Space on

Ignore Newline on

Text Order: As Is | Horizontal | Vertical

Effect What: FG | BG | Both

Alpha Min % 0.0

Alpha Max % 100.0

Randomness 10.0

- **Effect What:** Defines where the effect should have an effect. Available options are background (BG), foreground (FG) or both.
- **Alpha Begin**: Sets the alpha level of the character at 0% effect.
- **Alpha End**: Sets the alpha level of the character at 100% effect.
- **Randomness**: sets the intensity of the jittering alpha changes.

12.12.10 Text FX Jitter Position

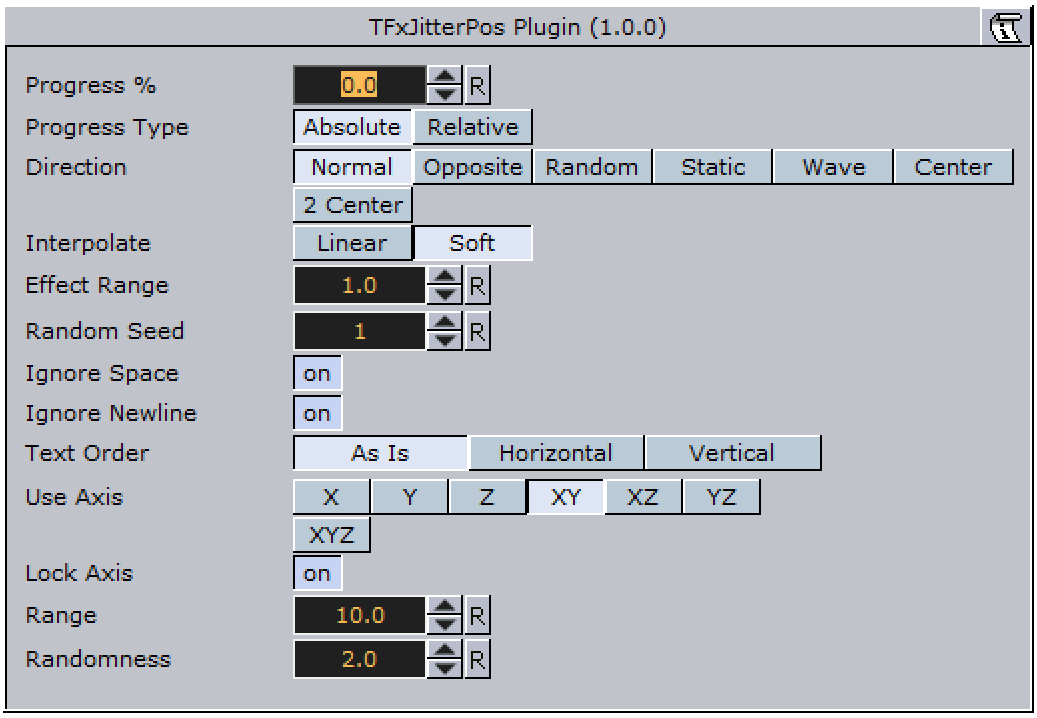
The Text FX Jitter Position plug-in creates an jittering motion of the characters by randomly changing the position of each character. The degree of position change and the starting sequence of the jittering can be altered. To use the function add it onto a container which holds a font. To create an animation, animate the progress. Other values can of course be animated as well.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\TextFxJitterPosition.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters


TFxJitterPos Plugin (1.0.0)

Progress % 0.0
Progress Type: Absolute | Relative
Direction: Normal | Opposite | Random | Static | Wave | Center | 2 Center
Interpolate: Linear | Soft
Effect Range 1.0
Random Seed 1
Ignore Space on
Ignore Newline on
Text Order: As Is | Horizontal | Vertical
Use Axis: X | Y | Z | XY | XZ | YZ | XYZ
Lock Axis on
Range 10.0
Randomness 2.0

- **Use Axis** defines on which axis or axes the characters moves to create the jittering effect.
- **Lock Axis** with this option enabled **Range** and **Randomness** will be the same for all axes. If you disable it, **Range** and **Randomness** for each of the axes will be displayed and you can set them individually.
- **Range** sets the range of the jittering movement.
- **Randomness** sets the intensity of the jittering movement.

Common Text FX Properties and Parameters.

12.12.11 Text FX Jitter Scale

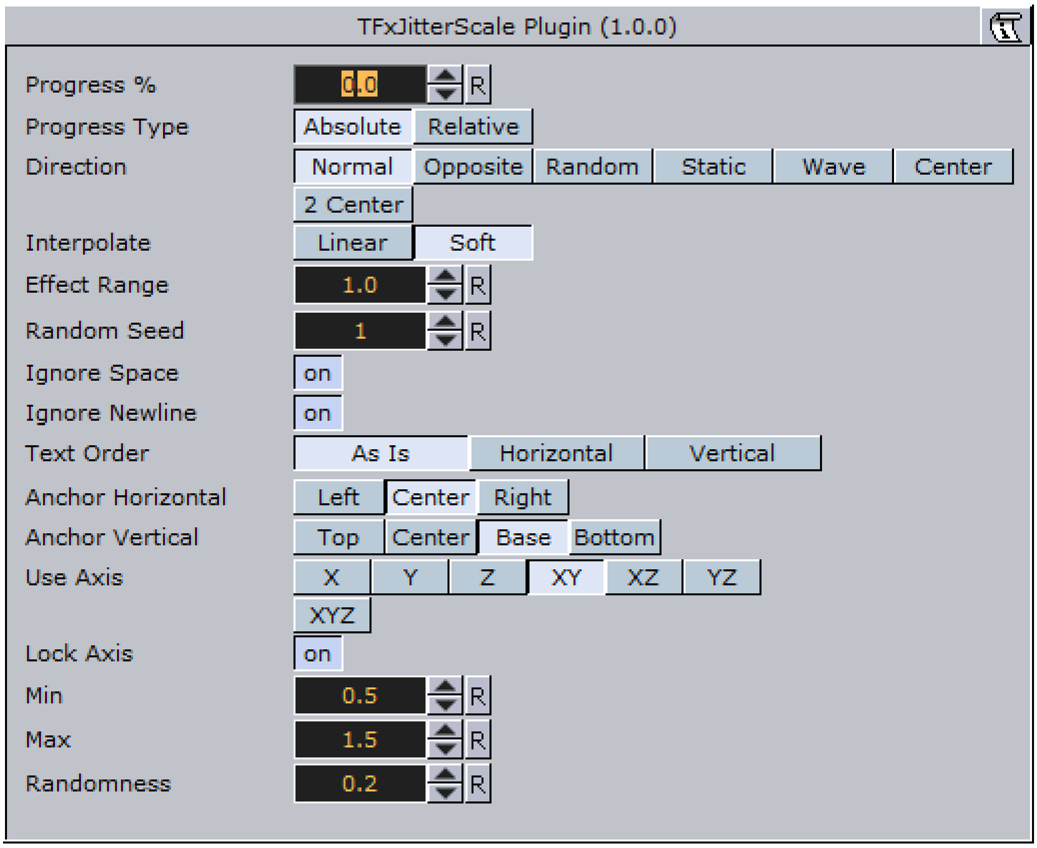
The Text FX Jitter Scale animates a jittering effect on the scale of the character.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\TextFxJitterScale.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



TFxJitterScale Plugin (1.0.0)

Progress % 0.0 R

Progress Type: Absolute | Relative

Direction: Normal | Opposite | Random | Static | Wave | Center | 2 Center

Interpolate: Linear | Soft

Effect Range 1.0 R

Random Seed 1 R

Ignore Space on

Ignore Newline on

Text Order: As Is | Horizontal | Vertical

Anchor Horizontal: Left | Center | Right

Anchor Vertical: Top | Center | Base | Bottom

Use Axis: X | Y | Z | XY | XZ | YZ | XYZ

Lock Axis on

Min 0.5 R

Max 1.5 R

Randomness 0.2 R

- **Anchor Horizontal**: Sets the anchor point for the characters on the horizontal plane.
- **Anchor vertical**: Sets the anchor point for the characters on the vertical plane.
- **Use Axis**: Defines on which axis or axes the characters scale to create the jittering effect.
- **Lock Axis**: With this option enabled, all the axes get the same **Min**, **Max** and **Randomness** settings. If you disable randomness, these parameters will be visible for each of the axes and must be set individually.
- **Min**: Sets the minimum scaling for the characters.
- **Max**: Sets the maximum scaling for the characters.
- **Randomness**: Sets the intensity of the jittering movement.

Common Text FX Properties and Parameters.

12.12.12 Text FX Plus Plus

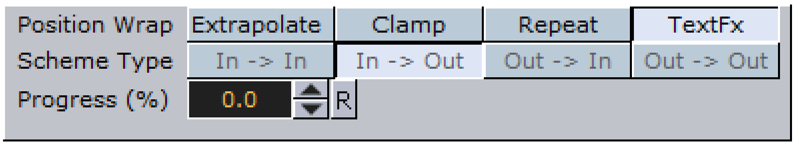
The Text FX Plus Plus plug-in allows you to set a number of effects on a text string.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\TextFxPlusPlus.vip`.

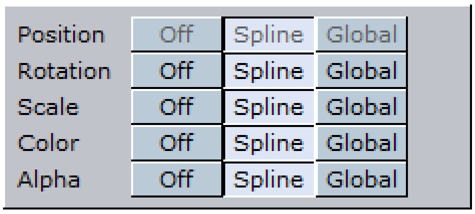
This section contains information on the following topics:

- Properties and Parameters

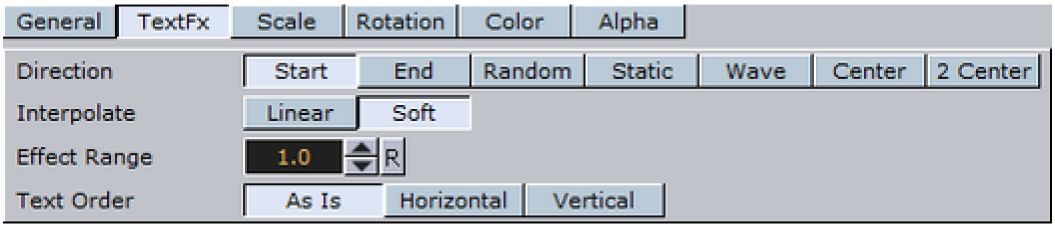
Properties and Parameters



- **Position Wrap**
  - **Extrapolate**: Above 100% or below 0% of Path Position will continue the direction of the spline.
  - **Clamp**: Will stop the string on the spline endpoints.
  - **Repeat**: Will cause the string to move to the other spline endpoint as soon as it moves over the endpoint.
  - TextFx: Will position characters by the relative spline of each character using the common TextFx parameters.
- Scheme Type: Defines how the text will look at 0% and 100% progress.
  - **In -> In**: at 0% first character will be at the beginning of the spline, at 100% last character will be at the end of the spline.
  - **In -> Out**: at 0% first character will be at the beginning of the spline, at 100% first character will be at the end of the spline.
  - **Out -> In**: at 0% last character will be at the beginning of the spline, at 100% first character will be at the end of the spline.
  - **Out -> Out**: at 0% last character will be at the beginning of the spline, at 100% last character will be at the end of the spline.
- **Progress (%)**: Animates the progress of the effect(s).



- **Position**: Positions the characters using the spline as a position axis.
- **Rotation**: Rotates the characters using the spline as a rotation axis.
- **Scale**: Scales the character’s X, Y and Z-axis using the spline as a scaling graph.
- **Color**: Applies Color Change on character.
- **Alpha**: Applies Alpha on the characters using the spline as a alpha graph.



Direction: Start

Interpolate: Linear

Effect Range: 1.0

Text Order: As Is

- **General**: Enables rotation, offset and align properties.
  - **Rotate X/Y/Z**: Adds extra Rotation to each character.
  - **Text Offset**: Sets the text offset.
  - **Align**: Rotates the characters to follow the spline movements.
- **TextFX**: See Common Text FX Properties and Parameters.
  - **Direction**: Sets the direction of the text effect sequence. Normal starts with the first character (e.g. left). Opposite starts with the last character (e.g. right). Random uses a random order (see Random Seed). Static means that all characters are processed at the same time. Wave starts with the first character, animates the effect from 0 to 100% and then down again to 0%. Center starts the effect from the center of the text. 2 Center starts the effect at the same time from the beginning and the end of the text. They meet at the center.
- **Scale**: Defines the scale graph to apply on the characters for each axis.
- Rotation: Defines the rotation effect to apply on the characters for each axis.
- Color: Defines the color gradient effect to apply on the characters.
- Alpha: Defines the alpha effect to apply on the characters.

12.12.13 Text FX Rotate

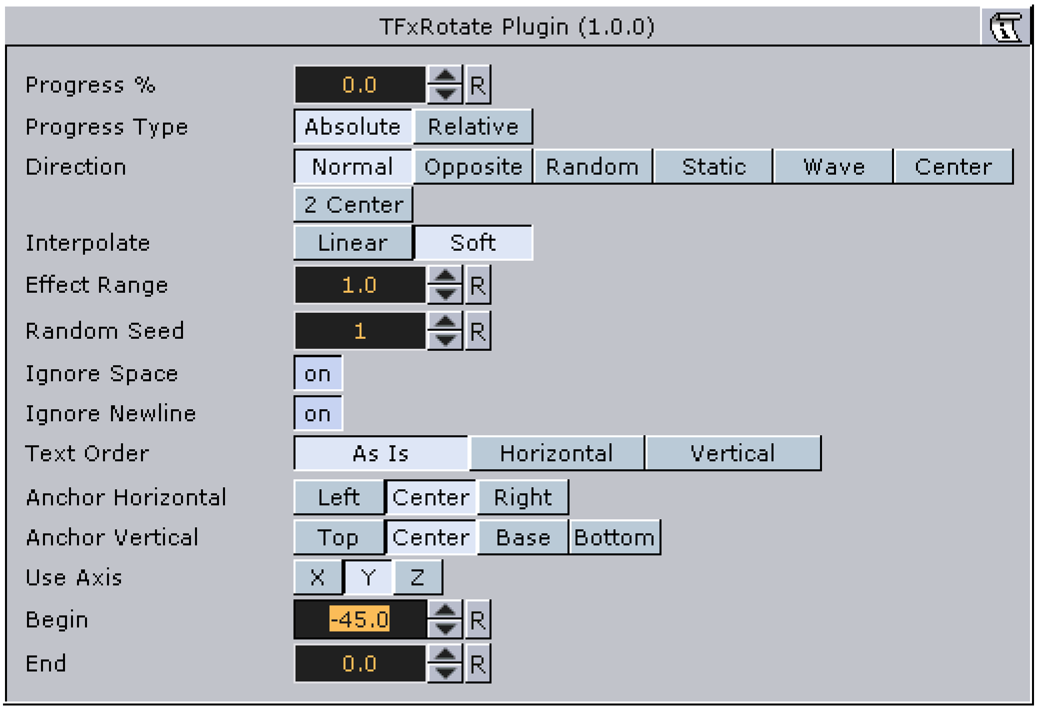
The Text FX Rotate plug-in allows you to create an effect where the characters rotate on the X-, Y- or Z-axis.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\TextFxRotate.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



TFxRotate Plugin (1.0.0)

Progress % 0.0 R
Progress Type: Absolute | Relative
Direction: Normal | Opposite | Random | Static | Wave | Center | 2 Center
Interpolate: Linear | Soft
Effect Range 1.0 R
Random Seed 1 R
Ignore Space on
Ignore Newline on
Text Order: As Is | Horizontal | Vertical
Anchor Horizontal: Left | Center | Right
Anchor Vertical: Top | Center | Base | Bottom
Use Axis: X | Y | Z
Begin -45.0 R
End 0.0 R

- **Anchor Horizontal**: Sets the anchor point for the characters on the horizontal plane.
- **Anchor Vertical**: Sets the anchor point for the characters on the vertical plane.
- **Use Axis**: Defines on which axis the characters rotate.
- **Begin**: Sets the initial rotated position of the characters.
- **End**: Sets the ending rotated position of the characters.

Common Text FX Properties and Parameters.

See Also

- Expert

12.12.14 Text FX Scale

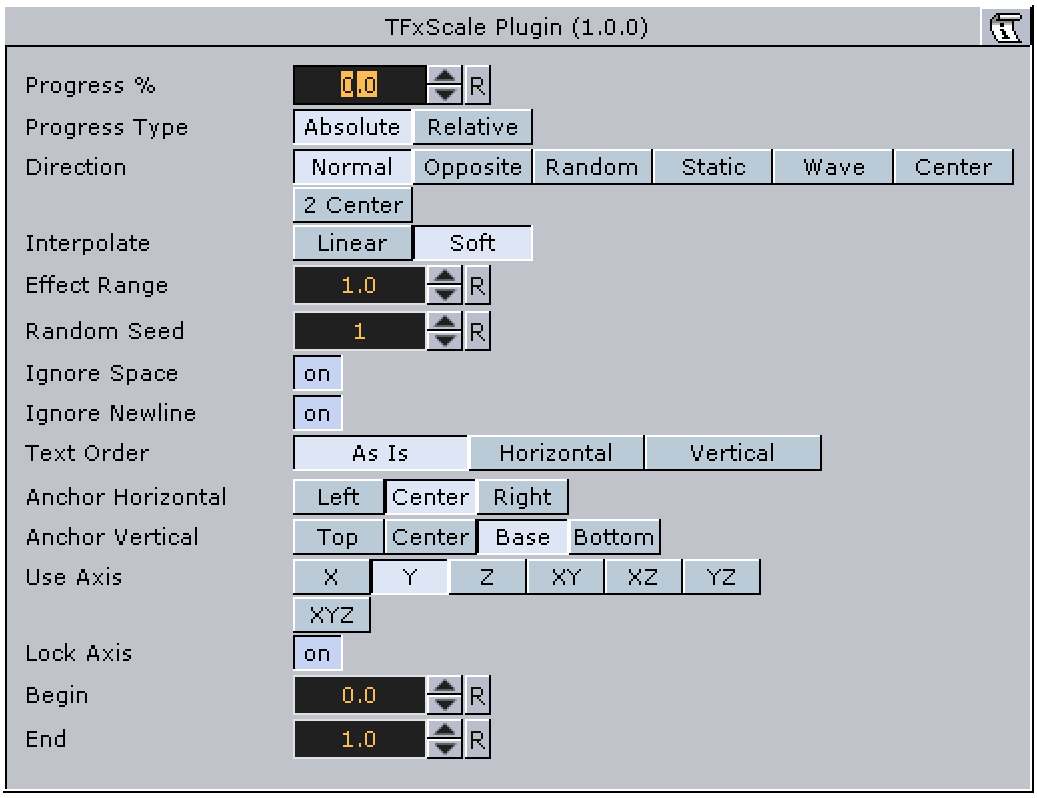
The Text FX Scale plug-in allows you to create a scaling animation of the characters.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\TextFxScale.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



TFxScale Plugin (1.0.0)

Progress % 0.0

Progress Type: Absolute | Relative

Direction: Normal | Opposite | Random | Static | Wave | Center | 2 Center

Interpolate: Linear | Soft

Effect Range 1.0

Random Seed 1

Ignore Space on

Ignore Newline on

Text Order: As Is | Horizontal | Vertical

Anchor Horizontal: Left | Center | Right

Anchor Vertical: Top | Center | Base | Bottom

Use Axis: X | Y | Z | XY | XZ | YZ | XYZ

Lock Axis on

Begin 0.0

End 1.0

- **Anchor Horizontal**: Sets the anchor point for the characters on the horizontal plane.
- **Anchor Vertical**: Sets the anchor point for the characters on the vertical plane.
- **Use Axis**: Defines on which axis or axes the characters scale.
- **Lock Axis**: With this option enables the **Begin**- and **End** scaling are the same for all the axes. If you disable the parameter, **Begin** and **End** must be set for all the axes individually.
- **Begin**: Sets the initial size of the characters.
- **End**: Sets the ending size of the characters.

Common Text FX Properties and Parameters.

See Also

- Text FX Size

12.12.15 Text FX Size

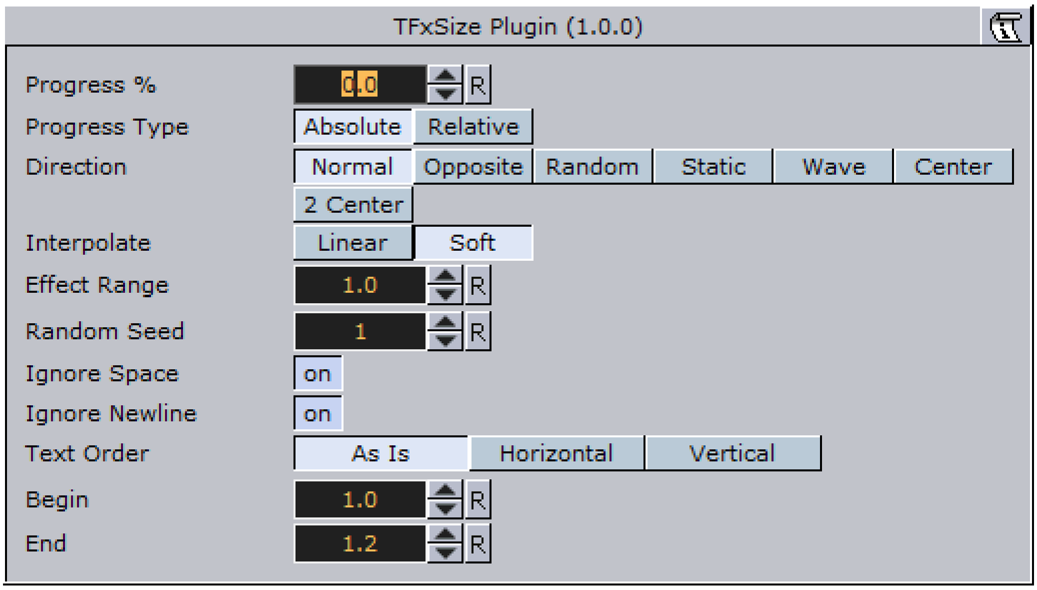
The Text FX Scale plug-in allows you to create a sizing animation of the characters.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\TextFxSize.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



TFxSize Plugin (1.0.0)

Progress % 0.0 R
Progress Type Absolute Relative
Direction Normal Opposite Random Static Wave Center 2 Center
Interpolate Linear Soft
Effect Range 1.0 R
Random Seed 1 R
Ignore Space on
Ignore Newline on
Text Order As Is Horizontal Vertical
Begin 1.0 R
End 1.2 R

- **Begin**: Sets the initial scale of the text before the effect.
- **End**: Sets the final scale of the text after the effect.

Common Text FX Properties and Parameters.

See Also

- Text FX Scale

12.12.16 Text FX Slide

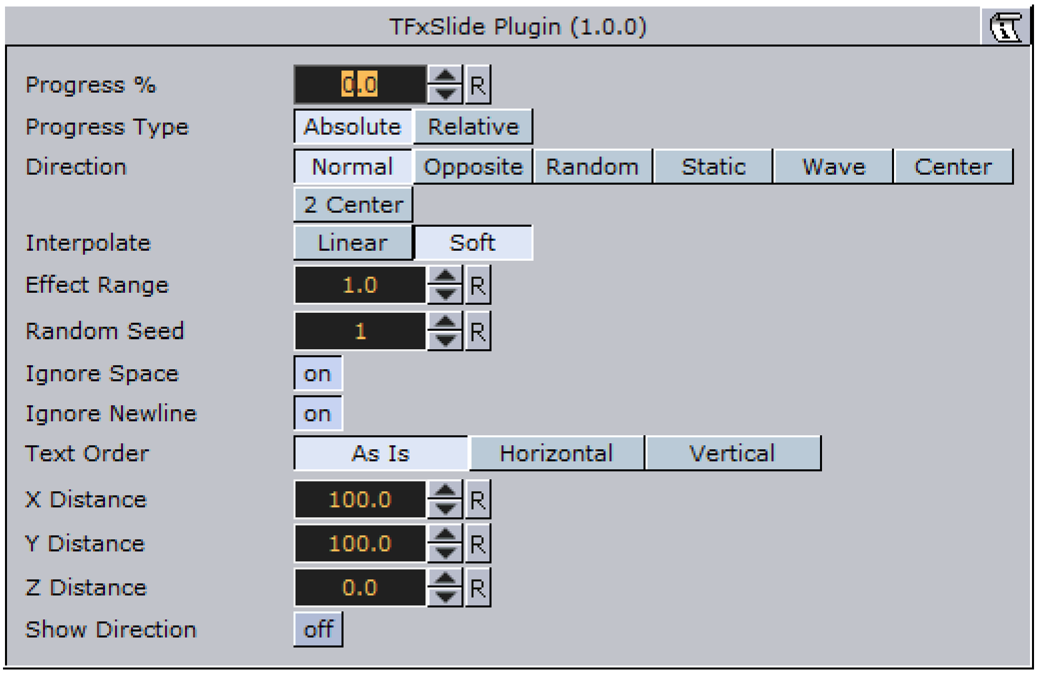
The Text FX Slide plug-in allows you to create a sliding animation of the characters on the X, Y and Z-axis.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\TextFxSlide.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



TFxSlide Plugin (1.0.0)

Progress % 0.0 R

Progress Type: Absolute | Relative

Direction: Normal | Opposite | Random | Static | Wave | Center | 2 Center

Interpolate: Linear | Soft

Effect Range 1.0 R

Random Seed 1 R

Ignore Space on

Ignore Newline on

Text Order: As Is | Horizontal | Vertical

X Distance 100.0 R

Y Distance 100.0 R

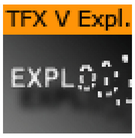
Z Distance 0.0 R

Show Direction off

- **X, Y and Z Distance**: Defines direction and distance of moved text.
- **Show Direction**: Enables and disables view of 'movement' vector. Note that direction will not be shown in On Air mode.

Common Text FX Properties and Parameters.

12.12.17 Text FX Vertex Explode



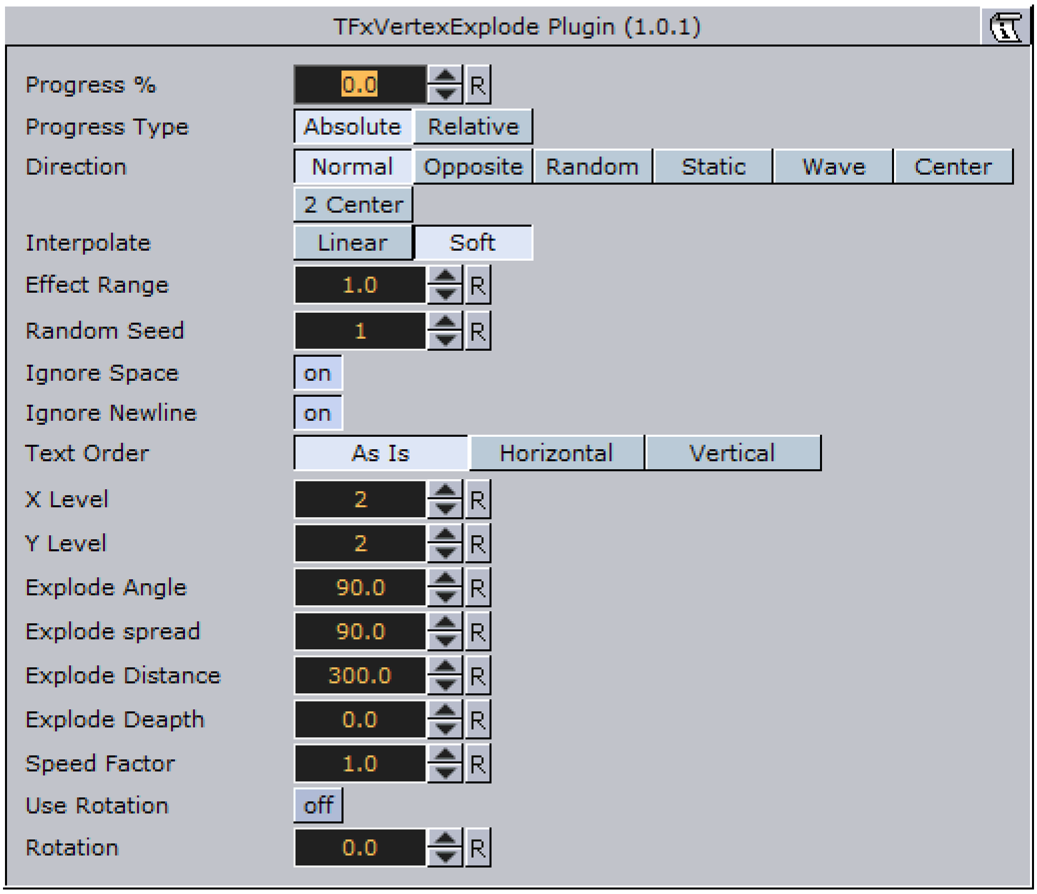
The Text FX Vertex Explode plug-in allows you to create an effect where text characters are exploded with a vertex effect.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\TextFxVertexExplode.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



TFxVertexExplode Plugin (1.0.1)

Progress %	0.0
Progress Type	Absolute / Relative
Direction	Normal / Opposite / Random / Static / Wave / Center / 2 Center
Interpolate	Linear / Soft
Effect Range	1.0
Random Seed	1
Ignore Space	on
Ignore Newline	on
Text Order	As Is / Horizontal / Vertical
X Level	2
Y Level	2
Explode Angle	90.0
Explode spread	90.0
Explode Distance	300.0
Explode Deapth	0.0
Speed Factor	1.0
Use Rotation	off
Rotation	0.0

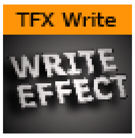
- **X, Y and Z Distance**: Defines direction and distance of moved text.
- **Show Direction**: Enables and disables view of 'movement' vector. Note that direction will not be shown in On Air mode.

Common Text FX Properties and Parameters.

See Also

- Text FX Explode

12.12.18 Text FX Write



The Text FX Write plug-in allows you to create an effect where text characters get displayed as if they were written by the keyboard in real time, a sort of Typewriter effect.

The text field in the font editor does not need to contain any text, since the text for this plug-in is entered in the plug-in's own editor.

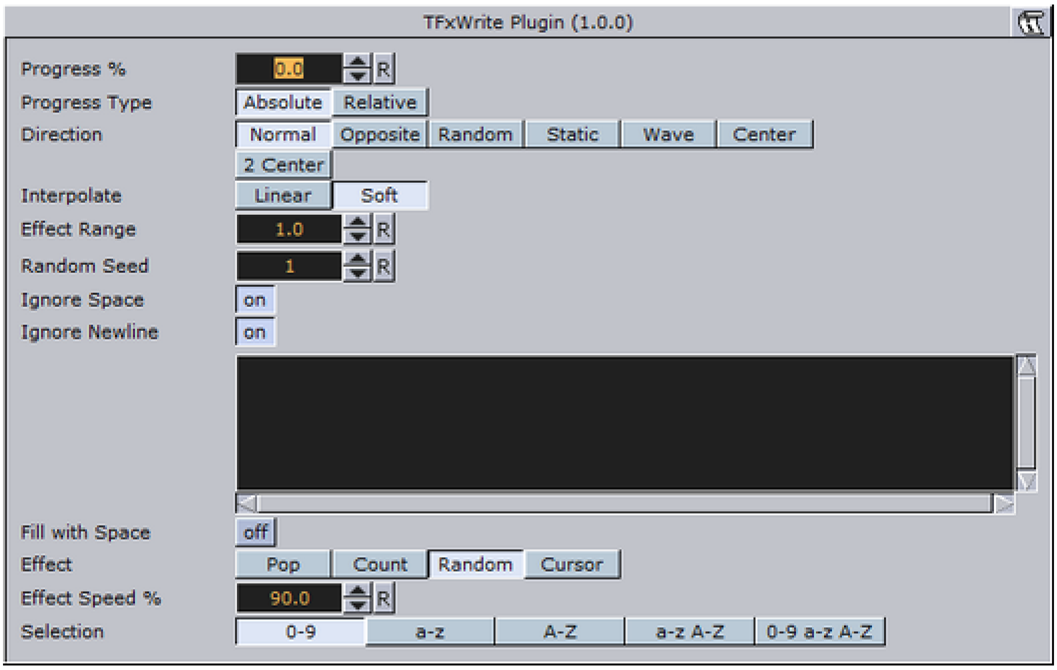
Note: Text FX Write expects UTF-8 encoded text as input if a font encoding other than *Default* is selected. See Font Options.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\TextFxWrite.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



TFxWrite Plugin (1.0.0)

Progress % 0.0

Progress Type: Absolute | Relative

Direction: Normal | Opposite | Random | Static | Wave | Center | 2 Center

Interpolate: Linear | Soft

Effect Range 1.0

Random Seed 1

Ignore Space on

Ignore Newline on

Fill with Space off

Effect: Pop | Count | Random | Cursor

Effect Speed % 90.0

Selection: 0-9 | a-z | A-Z | a-z A-Z | 0-9 a-z A-Z

- **Fill with Space**: With this option enabled, the function will add a space for every character not yet written. The bounding box of the text is then sized correctly from the start. If you use an **Auto Follow** function connected to the text, enabling this option will create a better result.
- **Effect**: Allows you to select an effect to be executed as each character gets written on the screen. There are four effects to choose from and they have in addition their own parameters.
    - **Pop**: Is the basic effect where the characters just pop up on the screen without any other sign being displayed.
    - **Count**: Creates a counting sequence by showing a random symbol before showing the character to be displayed.
    - **Random**: This (in combination with effect range) writes random characters before the final character is displayed. Use a non proportional font when you use this function to get the better looking effect.
    - **Cursor**: Displays a cursor that writes the characters. When you select this option, a text field opens below where you can choose which symbol to use as cursor.

- **Effect Speed**: Is an additional parameter that is visible if you have selected **Count** or **Random.** It sets the speed of the effect that shows the preceding symbol.
- **Selection**: Is an additional parameter to the **Random** effect. It allows you to select between some preset character ranges to use for the effect.
- **Cursor**: Allows you to enter the character which is to be used as a cursor.

Common Text FX Properties and Parameters.

12.13 Texture

The following container plugins are housed in the Texture folder:

- Grabbit
- Graffiti
- Image Clip
- LightBlur
- MoViz
- Noise
- Soft Clip

12.13.1 Grabbit

With the Grabbit plug-in you are able to use captured video inside Viz Engine. Grabbit uses the Microsoft DirectShow Filter graph framework to capture the video.

You can use devices which are supported by DirectShow and are self-contained such as capture cards or web cameras. Self-contained means the capture device does not need any further input.

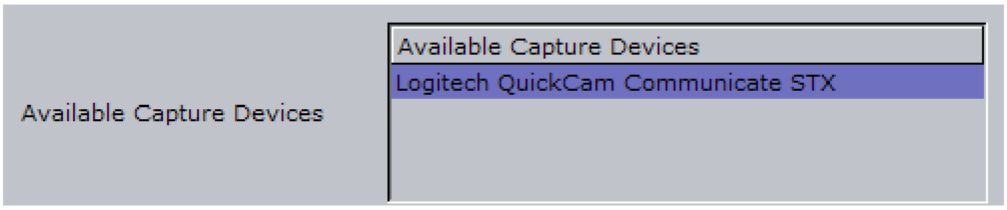
The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\Grabbit.vip`.

This section contains information on the following topics:

- Properties and Parameters
- Required Software
- Best Practices
- Known Limitations
- To capture a video stream using Grabbit
- To capture a video stream using Grabbit with a VfW supported device

Properties and Parameters

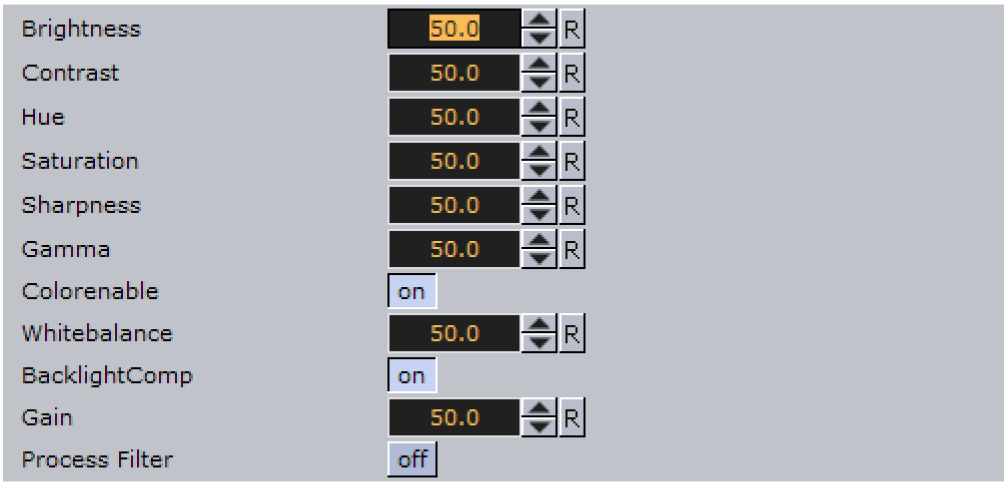
Following is a list of all parameters of the plug-in. Depending on the state of the plug-in, they may be hidden. Some of them are not visible at all.



Available Capture Devices

Available Capture Devices
Logitech QuickCam Communicate STX

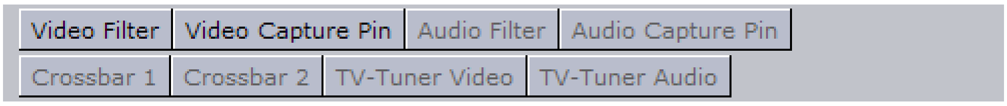
- **Available Devices**: Lets you decide which video capture device to use. This list is created during loading of the plug-in. That means if you have a capture device which is not known at startup of the renderer it will not show up. On the other hand it does not know if a capture device is already in use by some other plug-in. Since the list is static is just shows all capture devices which were available at startup. If the parameter is changed and the plug-in is already started it will stop the capturing before loading the new capture device. Changes are effective immediately.
  - Parameter name: `CaptureDevices (list)`.
- **Selected Capture Device**: Note that this parameter is hidden from the user interface. Contains the simple name for the chosen capture device from the list. This parameter decides which DirectShow Device is taken for capturing. If the parameter is changed and the plug-in is already started it will stop the capturing before loading the new capture device.
  - Parameter name: `CaptureDevice (string)`.
- **Capture Format**: Note that this parameter is hidden from the user interface. Human readable string that represents the current capturing format. It shows the width by height, frame rate, video subtype, and bit depth. It is a read only resource and is changed with the configuration dialogs below. This should really be an opaque data type.
  - Parameter name: `CaptureFormat (string)`.



Brightness	50.0
Contrast	50.0
Hue	50.0
Saturation	50.0
Sharpness	50.0
Gamma	50.0
Colorenable	on
Whitebalance	50.0
BacklightComp	on
Gain	50.0
Process Filter	off

Parameters are shown if the filter supports the ProcAmp interface.

- **Brightness**: Changes the brightness. The range is relative and the value is re-calculated internally according to the possibilities (min/max/stepping) of the device. Changes are effective immediately but not for already captured video.
  - Parameter Name: `Brightness`.
  - Type: float. Range: [0..100]. Default: 50.
- **Contrast**: Changes the contrast. The range is relative and the value is re-calculated internally according to the possibilities (min/max/stepping) of the device. Changes are effective immediately but not for already captured video.
  - Parameter Name: `Contrast`.
  - Type: float. Range: [0..100]. Default: 50.
- **Hue**: Changes the hue. The range is relative and the value is re-calculated internally according to the possibilities (min/max/stepping) of the device. Changes are effective immediately but not for already captured video.
  - Parameter Name: `Hue`.
  - Type: float. Range: [0..100]. Default: 50.
- **Saturation**: Changes the saturation. The range is relative and the value is re-calculated internally according to the possibilities (min/max/stepping) of the device. Changes are effective immediately but not for already captured video.
  - Parameter Name: `Saturation`.
  - Type: float. Range: [0..100]. Default: 50.
- **Sharpness**: Changes the sharpness. The range is relative and the value is re-calculated internally according to the possibilities (min/max/stepping) of the device. Changes are effective immediately but not for already captured video.
  - Parameter Name: `Sharpness`.
  - Type: float. Range: [0..100]. Default: 50.
- **Gamma**: Changes the gamma. The range is relative and the value is re-calculated internally according to the possibilities (min/max/stepping) of the device. Changes are effective immediately but not for already captured video.
  - Parameter Name: `Gamma`.
  - Type: float. Range: [0..100]. Default: 50.
- **Colorenable**: Enables or disables color. Changes are effective immediately but not for already captured video.
  - Parameter Name: `Colorenable`.
  - Type: toggle button. Default: on.
- **Whitebalance**: Changes the whitebalance. The range is relative and the value is re-calculated internally according to the possibilities (min/max/stepping) of the device. Changes are effective immediately but not for already captured video.
  - Parameter Name: `Whitebalance`.
  - Type: float. Range: [0..100]. Default: 50.
- **Backlight Compensation**: Enables or disables the backlight compensation. Changes are effective immediately but not for already captured video.
  - Parameter Name: `BacklightComp`.
  - Type: toggle button. Default: on.
- **Gain**: Changes the gain. The range is relative and the value is re-calculated internally according to the possibilities (min/max/stepping) of the device. Changes are effective immediately but not for already captured video.
  - Parameter Name: `Gain`.
  - Type: float. Range: [0..100]. Default: 50.



Parameters are shown if the filter supports the DirectShow dialog interface. Since the dialogs are handled by Windows and not the Viz interface they may open in the background. If so please bring the dialog to the front and make your settings. Make sure that the dialog is closed before you change anything in the Viz interface.

The buttons are disabled if there is no such dialog. Some implementations hint that a dialog is available even if it is not. In that case the button is not disabled but no dialog window shows.

- Video Filter: The dialog window may be behind the renderer GUI after activation. Please be careful to close the dialog window. Otherwise the renderer and the GUI may lock up. Changes are effective immediately.
  - Parameter Name: `ButtonVideoFilter`.
  - Type: push button.
- Video Capture Pin: If available choose RGB24 or UYV as format. Other settings may not work. The dialog window may be behind the renderer GUI after activation. Please be careful to close the dialog window. Otherwise the renderer and the GUI may lock up. Changes are effective immediately.
  - Parameter Name: `ButtonVideoCapPin`.
  - Type: push button.
- Audio Filter: The dialog window may be behind the renderer GUI after activation. Please be careful to close the dialog window. Otherwise the renderer and the GUI may lock up. Changes are effective immediately.
  - Parameter Name: `ButtonAudioFilter`.
  - Type: push button.
- Audio Capture Pin: The dialog window may be behind the renderer GUI after activation. Please be careful to close the dialog window. Otherwise the renderer and the GUI may lock up. Changes are effective immediately.
  - Parameter Name: `ButtonAudioCapPin`.
  - Type: push button.
- Crossbar 1: The dialog window may be behind the renderer GUI after activation. Please be careful to close the dialog window. Otherwise the renderer and the GUI may lock up.
  - Parameter Name: `ButtonCrossbar1`.
  - Type: push button.
- Crossbar 2: The dialog window may be behind the renderer GUI after activation. Please be careful to close the dialog window. Otherwise the renderer and the GUI may lock up.
  - Parameter Name: `ButtonCrossbar2`.
  - Type: push button.
- TV-Tuner Video: The dialog window may be behind the renderer GUI after activation. Please be careful to close the dialog window. Otherwise the renderer and the GUI may lock up.
  - Parameter Name: `ButtonTvVideo`.
  - Type: push button.

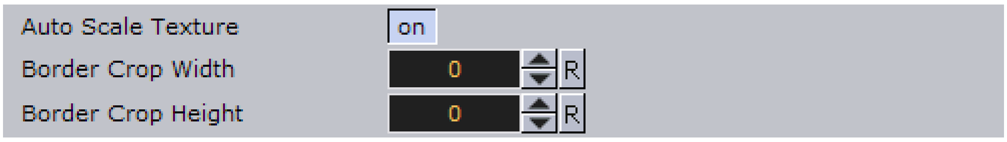
- **TV-Tuner Audio**: The dialog window may be behind the renderer GUI after activation. Please be careful to close the dialog window. Otherwise the renderer and the GUI may lock up.
  - Parameter Name: `ButtonTvAudio`.
  - Type: push button.

Video for Windows Dialogs:

Parameters are shown if the filter supports the Video for Windows (VfW) dialog interface. Since the dialogs are handled by Windows and not the Viz interface they may open in the background. If so please bring the dialog to the front and make your settings. Make sure that the dialog is closed before you change anything in the Viz interface.

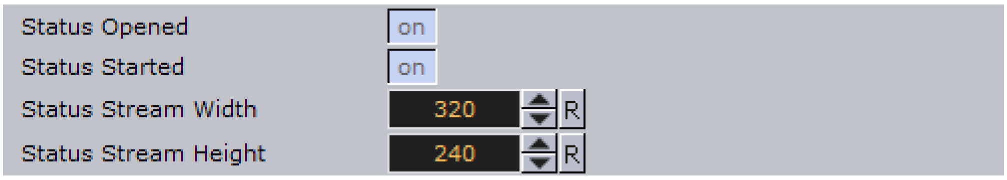
Buttons are disabled if there is no such dialog. Some implementations hint that a dialog is available even if it is not. In that case the button is not disabled but no dialog window shows.

- **Cap Source**: If the filter supports the VfW Source dialog this button is present. It can be activated if the capturing is stopped otherwise the button is grayed out. The dialog window may be behind the renderer GUI after activation. please be careful to close the dialog window. otherwise the renderer and the GUI may act in a strange way. Changes are effective immediately.
  - Parameter Name: `ButtonCaptureSource`.
  - Type: push button.
- **Cap Format**: If the filter supports the VfW Format dialog this button is present. It can be activated if the capturing is stopped otherwise the button is greyed out. The dialog window may be behind the renderer GUI after activation. please be careful to close the dialog window. otherwise the renderer and the GUI may act in a strange way. Changes are effective immediately.
  - Parameter Name: `ButtonCaptureFormat`.
  - Type: push button.
- **Cap Display**: If the filter supports the VfW Display dialog this button is present. It can be activated if the capturing is stopped otherwise the button is grayed out. The dialog window may be behind the renderer GUI after activation. please be careful to close the dialog window. otherwise the renderer and the GUI may act in a strange way. Changes are effective immediately.
  - Parameter Name: `ButtonCaptureDisplay`.
  - Type: push button.


Auto Scale Texture: on
Border Crop Width: 0
Border Crop Height: 0

- **Auto Scale Texture**: If enabled the captured video is scaled according to the underlying geometry. Changes are effective immediately but not for already captured video.
  - Parameter Name: `AutoScale`.
  - Type: toggle button. Default: on.
- **Border Crop Width**: Crop pixels on the horizontal sides of the video. This parameter is shown only if Auto Scale Texture is enabled. Changes are effective immediately but not for already captured video.

    - Parameter Name: BorderCrop_x.
    - Type: integer. Range: [0..0x7fff]. Default: 0.
- **Border Crop Height**: Crop pixels on the vertical sides of the video. This parameter is shown only if Auto Scale Texture is enabled. Changes are effective immediately but not for already captured video.
    - Parameter Name: BorderCrop_y.
    - Type: integer. Range: [0..0x7fff]. Default: 0.



Status Opened on
Status Started on
Status Stream Width 320
Status Stream Height 240

- **Status Opened**: Shows if the capture graph has been built successfully.
    - Parameter Name: StatusOpened.
    - Type: toggle button (display only).
- **Status Started**: Shows if the capture graph has been started.
    - Parameter Name: StatusStarted.
    - Type: toggle button (display only).
- **Status Stream Width**: Shows the captured video width.
    - Parameter Name: StatusStream_x.
    - Type: integer (display only).
- **Status Stream Height**: Shows the captured video height.
    - Parameter Name: StatusStream_y.
    - Type: integer (display only).

- **Start**: Starts capturing. Feedback is given through Status Started and Status Stream Width/Height. If capturing does not start check the Status Opened and Connected. Capturing cannot commence if the filter graph is not connected. Changes are effective immediately.
    - Parameter Name: ButtonStart.
    - Type: push button.
- **Stop**: Stops capturing. You need to do this in order to select the Vfw Dialog buttons. Changes are effective immediately.
    - Parameter Name: ButtonStop.
    - Type: push button.
- **Clear**: Clears the texture to opaque white. You may need to hit the button twice for the change to show. Changes are effective immediately.
    - Parameter Name: ButtonClear.
    - Type: push button.

Required Software

- DirectShow version 9.0c. It is strongly recommended to have a clean DirectShow installation to start with. Additional source filters may be added later. Find more Information at the
- DirectShow of VfW compatible capture card.

Best Practices

GraphEdit is a tool from the Microsoft DirectShow SDK that allows you to visualize the default Filtergraph that DirectShow builds in order to render the Media. If you can render the Media inside GraphEdit it should play in Viz Engine, too. If it does not play in GraphEdit it will certainly not play in Viz Engine.

Known Limitations

- Dynamic listbox for capture devices.
- Handle the capture device list dynamically.
- ProcAmp settings should be updated from dialog settings as well.
- Query for availability in the various ProcAmps.
- The Capture Device must be known to the system at application start. When using USB devices this means they must be connected before Viz Engine is started.

To capture a video stream using Grabbit

1. Add the Rectangle geometry to the scene tree, and add the Grabbit plug-in to it.
2. Select the capture device you want to work with from the **Available Capture Devices** list.
   - You should get a positive **Status Opened** feedback.

Note: Do not select a device which is currently in use (either by another instance of this plug-in or by another application).

3. If you have a device that supports the capture format enumeration and the ProcAmp interface you will get a screen similar to the following.
4. Set the **Capture Format** by using the dialogs behind the configuration buttons Video Filter, and so on.
5. Select **Auto Scale Texture** if you want the texture position and scaling set by this plug-in so that the video fills out the entire geometry. If you have selected this feature you may want to crop something on the horizontal or vertical sides of the video. Use the *Border Crop Width* and *Border Crop Height* for this effect.
6. Click the **Start** button to start capturing. You should get a positive *Status Started* feedback. In addition the *Status Stream Width* and *Status Stream Height* should reflect the dimensions of the captured video.
7. Adjust the effect settings. Note that the effect settings are highly dependant on the selected device. You might want to experiment with them. Some parameters have discreet values as for example the *Gamma* parameter.

To capture a video stream using Grabbit with a VfW supported device

Note: If you have a device which supports the VfW Dialogs interface you will most probably see that your device is tagged with VfW.

Once you have selected the capture device it needs to be configured. You do this by clicking the buttons that leads to the VfW dialog windows.

1. Click **Cap Source** and **Cap Format** and set the parameters according to your needs.
    - The values you set here cannot be changed via the usual control mechanism. You need to pay close attention that the dialog window may open in the behind the Viz GUI.
    - Make sure that each dialog window is closed before any other changes are made in the viz GUI.
    - You should get a positive **Status Connected** feedback. If you want to go back to change something here the graph must be in Stopped mode.
2. Select **Auto Scale Texture** if you want the texture position and scaling set by this plugin so that the video fills out the entire geometry.
    - If you have selected this feature you can use the **Border Crop Width** and **Border Crop Height** to crop something on the horizontal or vertical sides of the video.
3. Click the **Start** button to start capturing.
    - You should get a positive **Status Started** feedback. In addition the **Status Stream Width** and **Status Stream Height** should reflect the dimensions of the captured video.
4. Click the **Stop** button to stop the capturing.
    - This will set it back to Stopped mode.
    - You should see the feedback in the **Status Started** toggle.
    - Once the capturing is stopped you can use the **Clear** button to set the underlying texture to opaque white.

12.13.2 Graffiti

The Graffiti Texture plug-in allows telestration on top of flat containers, such as rectangles. Telestration is done by drawing with a brush shape using a mouse, 6DOF device or multi touch device. The plug-in can also recognize some rendered shape and replace the hand-drawn element with the recognized shape (circle, ellipse, cross, arrow).

The plug-in automatically adds a texture placeholder (i.e. an image) to the container the plug-in resides.

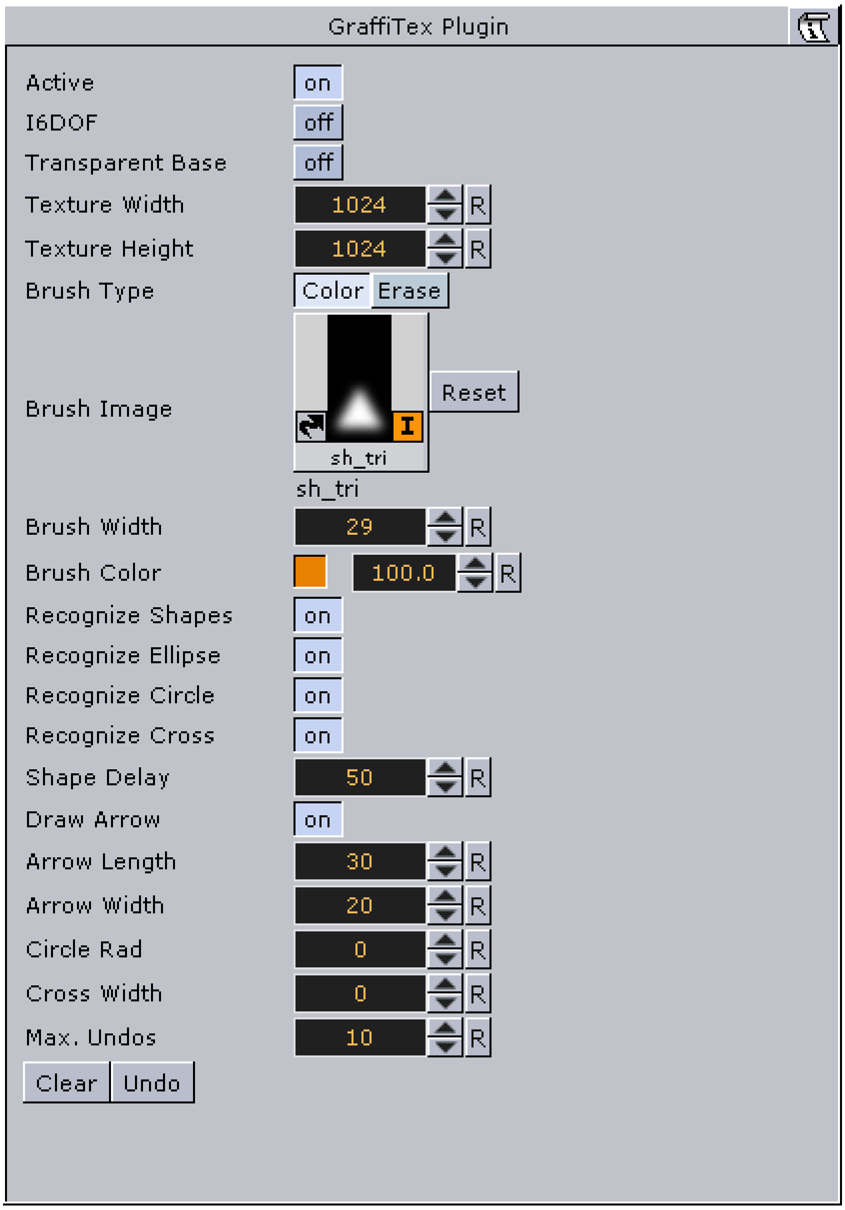
Note that there are two graffiti plug-ins, one that can be used globally as a scene plug-in, and this one that is applied to the container level.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\GraffiTex.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a container level graffiti effect

Properties and Parameters



GraffiTex Plugin

Active: on
I6DOF: off
Transparent Base: off
Texture Width: 1024
Texture Height: 1024
Brush Type: Color Erase
Brush Image: sh_tri
Brush Width: 29
Brush Color: 🟧 100.0
Recognize Shapes: on
Recognize Ellipse: on
Recognize Circle: on
Recognize Cross: on
Shape Delay: 50
Draw Arrow: on
Arrow Length: 30
Arrow Width: 20
Circle Rad: 0
Cross Width: 0
Max. Undos: 10

- **Active**: Enable/disable drawing
- **I6DOF**: Specifies whether input comes from mouse of 6DOF
- **Transparent Base**: Specifies whether the base color of the container will be the background of the rendered brush or whether the brush will be drawn on a transparent background
- **Texture Width, Texture Height**: Dimensions of the texture used for drawing canvas
- **Brush Type**: Color or eraser brush
- **Brush Image**: Optional image that will determine the shape of the brush. If empty, a round brush will be used
- **Brush Width**: Width of the brush in pixels. Visible only if the color brush is selected
- **Eraser Brush Width**: Width of the eraser in pixels. Visible only if the eraser brush is selected

- **Brush Color**: Color of the brush
- **Recognize Shapes**: Shape recognition mode (on/off)
- **Recognize Ellipse**: Specifies whether shape recognition will try to recognize ellipse shape
- **Recognize Circle**: Specifies whether shape recognition will try to recognize circle shape
- **Recognize Cross**: Specifies whether shape recognition will try to recognize cross shape
- **Shape Delay**: Number of frames to wait from mouse up before trying to recognize shapes
- **Draw Arrow**: Specifies whether non-recognizable shapes will be converted to an arrow
- **Arrow Length**: Length of arrow head
- **Arrow Width**: Width of arrow head
- **Circle Rad**: Radius of the circle replacing a recognized circle. If zero, the radius of the recognized circle will be used
- **Cross Width**: Width of the cross replacing the recognized cross shape. If zero, the width of the recognized cross shape will be used
- **Max. Undos**: Max number of undo operations
- **Clear (button)**: Clears the canvas
- **Undo (button)**: Undo
- **Brush Width**: Width of the brush in pixels. Visible only if the color brush is selected
- **Eraser Brush Width**: Width of the eraser in pixels. Visible only if the eraser brush is selected
- **Brush Color**: Color of the brush
- **Recognize Shapes**: Shape recognition mode (on/off)
- **Recognize Ellipse**: Specifies whether shape recognition will try to recognize ellipse shape
- **Recognize Circle**: Specifies whether shape recognition will try to recognize circle shape
- **Recognize Cross**: Specifies whether shape recognition will try to recognize cross shape
- **Shape Delay**: Number of frames to wait from mouse up before trying to recognize shapes
- **Draw Arrow**: Specifies whether non-recognizable shapes will be converted to an arrow
- **Arrow Length**: Length of arrow head
- **Arrow Width**: Width of arrow head
- **Circle Rad**: Radius of the circle replacing a recognized circle. If zero, the radius of the recognized circle will be used
- **Cross Width**: Width of the cross replacing the recognized cross shape. If zero, the width of the recognized cross shape will be used
- **Max. Undos**: Max number of undo operations
- **Clear (button)**: Clears the canvas
- **Undo (button)**: Undo

To create a container level graffiti effect

- Add the plug-in to the container, set the plug-in Properties and Parameters, set Viz in On Air Mode and start drawing.

See Also

- Graffiti (scene plug-in)

12.13.3 Image Clip

The Image Clip plug-in is designed to play back a sequence of still images (e.g. TGA or TIFF) rather than playing a movie file. It also supports alpha channel and various play modes (looping, swing, etc). The plug-in loads all images to RAM, then loads only one image at a time to texture memory. The download time is almost linear to the size of each frame. Preparation for this plug-in should be uncomplicated as most video applications has support for saving out TIFF or Targa sequences. Since it plays the sequence from memory, one obvious benefit is that once loaded into memory it does not access the hard drive at all. So if it is needed to play back ten different sequences at once, ImageClip is pretty much the only solution. It is ideal for looping small animations such as logos. The drawback is that it consumes system memory.

Remember to keep track of how much memory each sequence needs, to avoid running out of memory on the Viz Engine machine. Running low on memory will make the system start caching parts of memory to the hard drive and it may not play the sequence and graphics in real-time (see Properties and Parameters and the Memory Type setting).

Image clip should not be used with very large clips in Continuous and Array mode. Since all images are loaded into memory a large number of images or a large image size would require large amounts of memory.

Memory can be calculated as follows:

- Number of images * Image width * Image height * 3 (or 4 when using alpha)

The use of Thread mode solves the memory issue, but it will take longer to request a frame. Therefore it should be used only in a slow motion image clip.

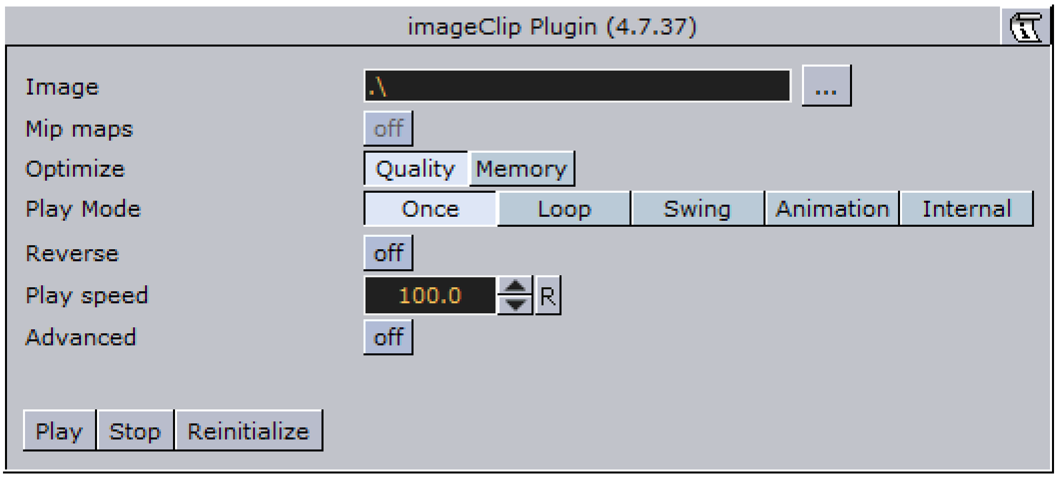
Images for an image sequence should be placed in specific folders. For example if you out many flag image sequences, you would have a separate folder for each flag image sequence. Make sure that the images are named sequentially e.g. “england001.tif”, ”england002.tif”, ”england003.tif” etc.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\ImageClip.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To add an image clip

Properties and Parameters



imageClip Plugin (4.7.37)

Image: .\
Mip maps: off
Optimize: Quality | Memory
Play Mode: Once | Loop | Swing | Animation | Internal
Reverse: off
Play speed: 100.0
Advanced: off

- **Image**: Sets the image path and the first image you would like to use in your animation. Do not use clip names with numbers (except the counter (e.g. 000, 001, 002, etc.)).
- **Mip maps**: Enables the use of Mip maps. This is possible when the Scale option is enabled, about + 30% and – 30% in performance.
- **Optimize**: Optimizes on quality or memory usage.
- **Play Mode**: Sets the play mode. Available modes are Once, Loop, Swing, Animation and Internal.
- **Animation Position**: Sets the animation position (e.g. where to start and stop the animation). The position is a counter for the number of images in the folder referenced by the Image setting. This setting is enabled when Play Mode is set to Animation.
- **Reverse**: Reverses the animation (except for in the Animation play mode).
- **Play Speed**: Sets the speed of the animation.
- **Advanced**: Enables the advanced settings (see below).
- Time Mode: Enables time specific settings.
- **Stand Alone Image**: Enables the same clip to be played in different speeds.
- First Image: Defines the first image of the animation.
- Nof Image: Defines the number of images, relative to the First Image, that should be part of the animation.
- **Crop**: Crops the image in percent from the left, right, bottom and top side.
- **Scale By**: Enables the scale setting.

  - **Scale**: Scales all images to the closest power of 2. When not selected, automatic texture coordinates will be applied so the image that will fit the texture.
- **Keep Under**: It forces the image size. For example if you have selected 64, the image will be trimmed to the size 64 x 64.
- **Memory Type**: Sets the memory type.
  - **Continuous**: Uses one big chunk of memory to store all images.
  - **Array**: Uses divided chunks of memory to store all images.
  - **Thread**: Loads only requested frame on-the-fly, with [n] frames loading time limit.
- **Play (button)**: Plays the animation.
- **Stop (button)**: Stops the animation.
- **Reinitialize (button)**: Reinitializes all settings.

To add an image clip

1. Add a container to the scene tree, and add the Image Clip plug-in to it.
2. Click the Browse ... button to locate the folder that houses the images.
3. Select the first image in the sequence, and click OK.
   - The image sequence now loads into the ImageClip plug-in, and is visible in the scene.
4. Click the Play button to test the image sequence.
   - If you change the image sequence and you need to reload it, click the Reinitialize button.
5. Change the Play Mode to Loop to play the image sequence continuously.
   - The image sequence will start playing automatically.
6. Change the Play Mode to Animation to animate the image sequence frame position to control the sequence within the Stage Editor Area.
   - The image sequence now stops playing and a new property is revealed that is called Animation Position. This refers to the frame number that is currently displayed for the image sequence.
7. Animate the Animation Position value.
   - The Animation Position maximum value is restricted to the number of images.
8. Another possibility is to create a file with a "*.vln" extension. This file includes the base path and also the names of the images to load. In this case the images must not have a counter number in their filename. You can handle this file as an ordinary text file.
9. Load the “vln - file” instead of loading an image file located in a directory.

Example:

```
BASE_PATH 'C:/clip/images'
{
   'radar_200504110800.png' 2005_04_11_10:00
   'radar_200504110815.png' 2005_04_11_10:15
   'radar_200504110830.png' 2005_04_11_10:30
   'radar_200504110845.png' 2005_04_11_10:45
   'radar_200504110900.png' 2005_04_11_11:00
   'radar_200504110915.png' 2005_04_11_11:15
   'radar_200504110930.png' 2005_04_11_11:30
   'radar_200504110945.png' 2005_04_11_11:45
}
```

Note: Viz Engine will not interpret between keyframes, so if you choose to slow down an animation then the motion will not be smooth.

See Also

- Implementing Video
- MoViz
- Soft Clip
- Background Clip

12.13.4 LightBlur

Allows the user to configure and create a light blurred image.

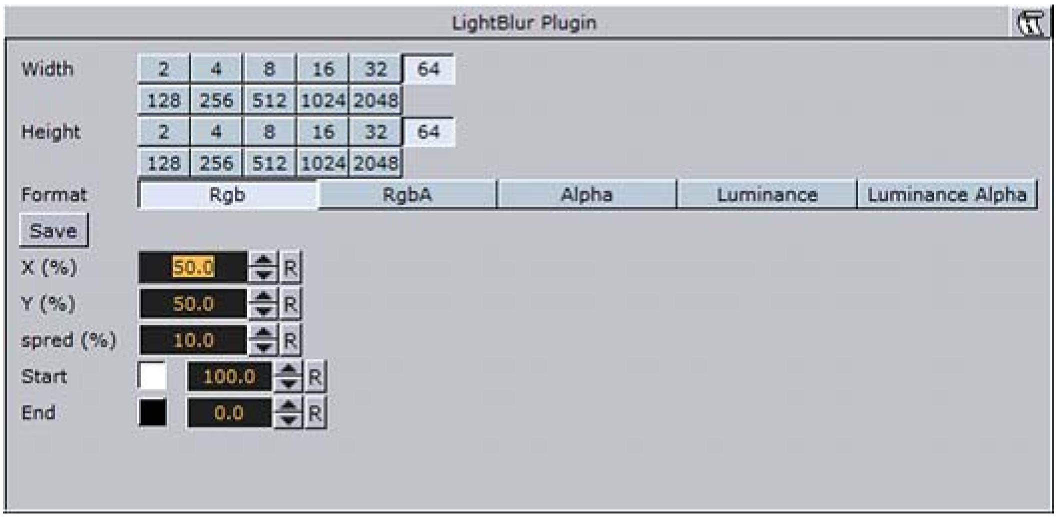
IMPORTANT! The light blur functionality is now a built in feature of the Viz core. For future use it is therefore recommended to use the Dynamic Texture plug-in.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\lightblur.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



LightBlur Plugin

Width: 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 | 1024 | 2048 (64 selected)

Height: 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 | 1024 | 2048 (64 selected)

Format: Rgb | RgbA | Alpha | Luminance | Luminance Alpha (Rgb selected)

Save

X (%): 50.0

Y (%): 50.0

spred (%): 10.0

Start: □ 100.0

End: ■ 0.0

- **Width and Height**: Sets the width and height of the blurred light. Available options are 2 to 2048.

- **Format**: Sets the image format. Available options are RGB, RGBA, Alpha, Luminance and Luminance Alpha.
- **Save (button)**: Saves the blur as an image.
- **X and Y %**: Sets the position of the blur on the X and Y axis (50% places the blur at the origin).
- **Spread %**: Sets the spread of the blur.
- **Start**: Sets the start color of the blur.
- **End**: Sets the end color of the blur.

12.13.5 MoViz

With the MoViz plug-in you are able to play media files or streams inside Viz Engine. MoViz is neither a QuickTime player nor an AVI player. It uses the Microsoft DirectShow Filter Graph framework to play the media files, which means it can play both mentioned types, but only certain installed Filter Graphs (some of the QuickTime and some AVI). MoViz is not natively QuickTime, it uses the AVI wrappers and filters for QuickTime and AVIs. This means that you cannot play all QuickTime codecs, only the ones that are supported within DirectX wrappers and filters.

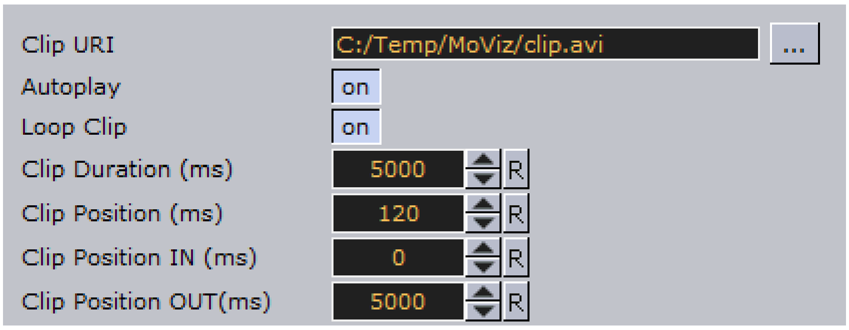
Note: Throughout this plug-in documentation, media file and media stream is referred to as media and may be used interchangeably unless stated otherwise.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\MoViz.vip`.

This section contains information on the following topics:

- Properties and Parameters

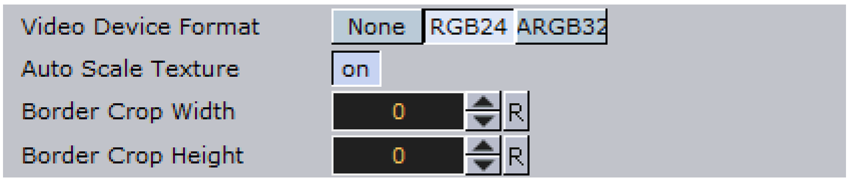
Properties and Parameters



Clip URI	C:/Temp/MoViz/clip.avi
Autoplay	on
Loop Clip	on
Clip Duration (ms)	5000
Clip Position (ms)	120
Clip Position IN (ms)	0
Clip Position OUT(ms)	5000

- **Clip URI**: You can load clips from the file system by using the 'Browse'-Button, or you can enter the location directly. It takes the form of a valid uri (e.g. http://... mms://...). If the parameter is changed and the plug-in is already started it will stop the playback before loading a new clip. Changes are effective immediately. It is important that all video clips from the file system are stored locally on the V: drive of the machine that is used to play out the graphics. Remember to save the clip locally on the client machines with preview (e.g. Viz Trio) and use the same V: drive so that the clip can be seen in preview.

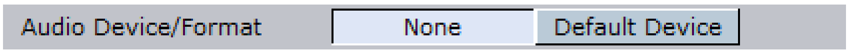
  - Parameter Name: `ClipURI`.
  - Type: string.
- **Autoplay**: Enable or disable automatic start of the clip after it has loaded here. Changes are effective immediately.
  - Parameter Name: `Autoplay`.
  - Type: toggle button. Default: off.
- **Loop Clip**: Enable or disable a looped playback here. If the clip position is at the clip out position or EOF it will rewind to the clip in position. Changes are effective for the next EOF, clip out position.
  - Parameter Name: `Loop`.
  - Type: toggle button. Default: off.
- **Clip Duration (ms)**: Displays the duration of the clip in units of milliseconds. The parameter is for display only. Make no changes to this parameter. This parameter makes only sense with media that supports this kind of information.
  - Parameter Name: `Duration`.
  - Type: integer (read only).
- **Clip Position (ms)**: Gives the current position of the clip in units of milliseconds. When in pause mode this parameter may be changed to seek to the gives time. Do not change this parameter when not in pause mode. The position may be changed between Clip Position IN and Clip Position OUT. This parameter makes only sense with media that supports this kind of information. Changes are effective immediately.
  - Parameter Name: `Position`.
  - Type: integer. Range: [IN..OUT]. Default: 0.
- **Clip Position IN (ms)**: Gives the first frame where the clip should start in units of milliseconds. This parameter makes only sense with media that supports this kind of information. Changes are effective immediately.
  - Parameter Name: `PositionIn`.
  - Type: integer. Range: [0..Duration]. Default: 0.
- **Clip Position OUT (ms)**: Gives the last frame where the clip should stop in units of milliseconds. This parameter makes only sense with media that supports this kind of information. Changes are effective immediately.
  - Parameter Name: `PositionOut`.
  - Type: integer. Range: [0..Duration]. Default: Duration.



Video Device Format	None RGB24 ARGB32
Auto Scale Texture	on
Border Crop Width	0
Border Crop Height	0

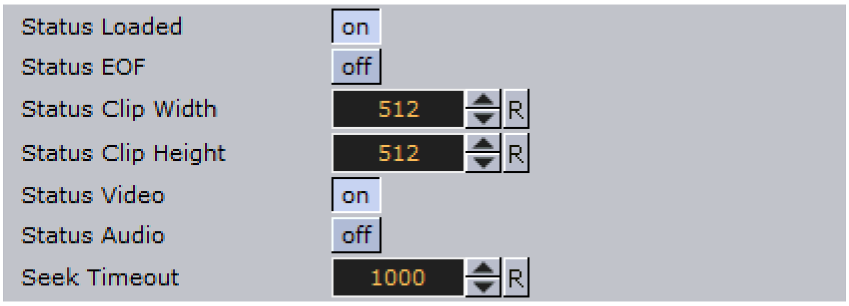
- **Video Device Format**: Lets you decide which video format to use for rendering. Changes are effective at load time.
  - Parameter Name: `VideoFormat`.
  - Type: radio button.

  - **None**: The video part is not rendered.
  - **RGB24**: The video part is delivered to Viz using RGB24 format.
  - **ARGB32**: The video part is delivered to Viz using ARGB32 format.
- **Force Opaque**: If the clip has an alpha channel and is selected (ARGB32) you can clear the alpha value to 100% opaque here. Changes are effective for each frame delivered.
  - Parameter Name: `ForceOpaque`.
  - Type: toggle button. Default: off.
- **Auto Scale Texture**: If enabled the captured video is scaled according to the underlying geometry. Changes are effective immediately but not for already delivered video.
  - Parameter Name: `AutoScale`.
  - Type: toggle button. Default: on.
- **Border Crop Width**: Crop pixels on the horizontal sides of the video. This parameter is shown only if Auto Scale Texture is enabled. Changes are effective immediately but not for already delivered video.
  - Parameter Name: `BorderCrop_x`.
  - Type: integer. Range: [0..0x7fff]. Default: 0.
- **Border Crop Height**: Crop pixels on the vertical sides of the video. This parameter is shown only if Auto Scale Texture is enabled. Changes are effective immediately but not for already delivered video.
  - Parameter Name: `BorderCrop_y`.
  - Type: integer. Range: [0..0x7fff]. Default: 0.



Audio Device/Format: None | Default Device

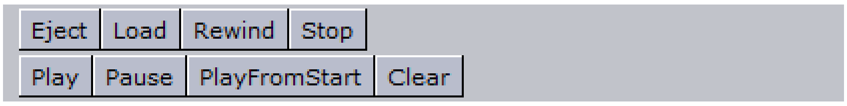
- **Audio Device Format**: Lets you decide which audio device and format to use for rendering. This may be useful until audio via viz is supported although it is not recommended due to performance and frame drop issues. Changes are effective immediately but not for already delivered audio.
  - Parameter Name: `AudioFormat`.
  - Type: radio button.
  - **None**: The audio part is not rendered.
  - **Default Device**: The audio part is rendered using the default DirectShow renderer.



Status Loaded	on
Status EOF	off
Status Clip Width	512
Status Clip Height	512
Status Video	on
Status Audio	off
Seek Timeout	1000

- **Status Loaded**: Shows if the clip has loaded.
  - Parameter Name: `StatusLoaded`.
  - Type: toggle button (display only).
- **Status EOF**: Shows if the clip is at EndOfFile.
  - Parameter Name: `StatusEof`.

- Type: toggle button (display only).
- **Status Clip Width**: Shows the video width.
  - Parameter Name: `StatusClip_x`.
  - Type: integer (display only).
- **Status Clip Height**: Shows the video height.
  - Parameter Name: `StatusClip_y`.
  - Type: integer (display only).
- **Status Video**: Shows if the clip has a usable Video component.
  - Parameter Name: `StatusVideo`.
  - Type: toggle button (display only).
- **Status Audio**: Shows if the clip has a usable audio component.
  - Parameter Name: `StatusAudio`.
  - Type: toggle button (display only).



- **Eject**: Unloads the clip and destroys the graph. Changes are effective immediately.
  - Parameter Name: `ButtonEject`.
  - Type: push button.
- **Load**: Loads the given clip and builds the graph. Changes are effective immediately.
  - Parameter Name: `ButtonLoad`.
  - Type: push button.
- **Rewind**: Rewinds the clip to the beginning when supported. Changes are effective immediately.
  - Parameter Name: `ButtonRewind`.
  - Type: push button.
- **Stop**: Puts the graph in stop mode. Changes are effective immediately.
  - Parameter Name: `ButtonStop`.
  - Type: push button.
- **Play**: Puts the graph in play mode. i.e. start playback. Changes are effective immediately.
  - Parameter Name: `ButtonPlay`.
  - Type: push button.
- **Pause**: Puts the graph in pause mode. Changes are effective immediately.
  - Parameter Name: `ButtonPause`.
  - Type: push button.
- **Play From Start**: Like Play but rewinds the clip to the beginning first. Changes are effective immediately.
  - Parameter Name: `ButtonPlayFromStart`.
  - Type: push button.
- **Clear**: Clears the texture to opaque white. You may need to hit the button twice for the change to show. Changes are effective immediately.
  - Parameter Name: `ButtonClear`.
  - Type: push button.

See Also

- Implementing Video
- Advanced Issues with Video Codecs
- Image Clip
- Soft Clip

12.13.6 Noise

Allows the user to configure and create a noise image.

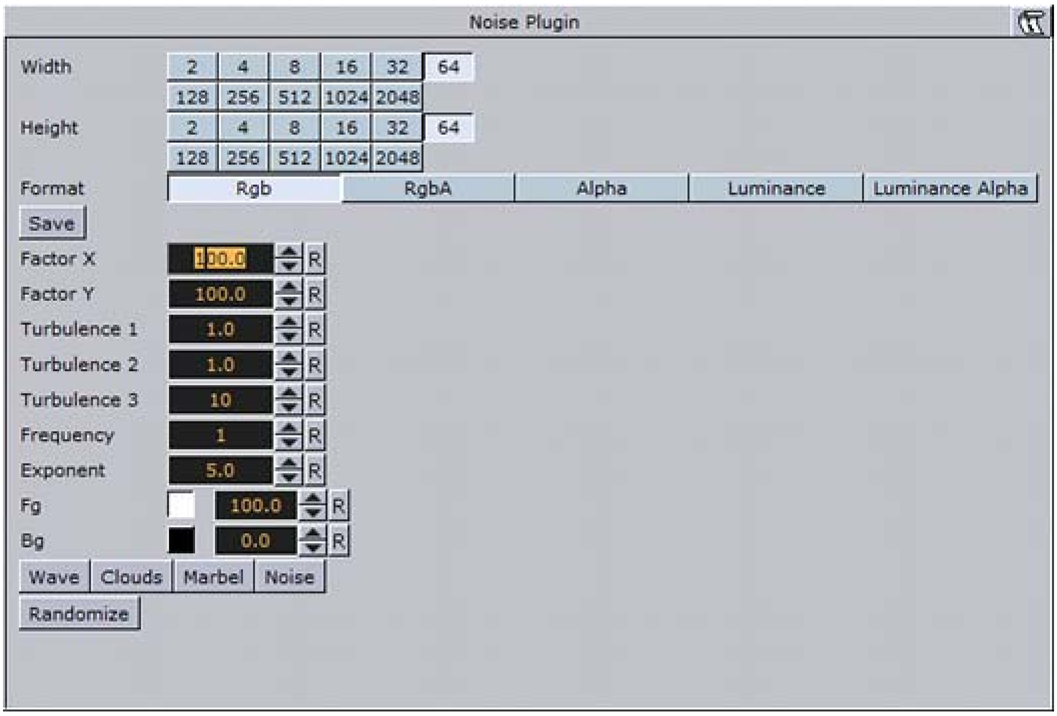
IMPORTANT! The noise functionality is now a built in feature of the Viz core. For future use it is therefore recommended to use the Dynamic Texture plug-in.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\noise.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



Noise Plugin

Width: 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 | 1024 | 2048 (selected: 64)

Height: 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 | 1024 | 2048 (selected: 64)

Format: Rgb | RgbA | Alpha | Luminance | Luminance Alpha (selected: Rgb)

Save

Factor X: 100.0

Factor Y: 100.0

Turbulence 1: 1.0

Turbulence 2: 1.0

Turbulence 3: 10

Frequency: 1

Exponent: 5.0

Fg: 100.0

Bg: 0.0

Wave | Clouds | Marbel | Noise

Randomize

- **Width and Height**: Sets the width and height of the noise. Available options are 2 to 2048.

- **Format**: Sets the image format. Available options are RGB, RGBA, Alpha, Luminance and Luminance Alpha.
- **Save (button)**: Saves the noise as an image.
- **Factor X and Y**: Sets the noise factor on the X and Y axis (range is from 0.000 to 1000.000).
- **Turbulence 1, 2 and 3**: Sets the turbulence of the noise.
- **Frequency**: Sets the frequency of the noise.
- **Exponent**: Sets the exponent of the noise.
- **Fg**: Sets the foreground color.
- **Bg**: Sets the background color.
- **Wave, Clouds, Marble and Noise (buttons)**: Allows the user to select between four different presets.
- **Randomize (button)**: Sets random factor, turbulence and frequency. Randomize works together with the Wave, Clouds, Marble and Noise selections.

12.13.7 Soft Clip

The Soft Clip is used for playing video clips projected on a texture, and supports AVI files only. It is recommended to use no compression when rendering clips to use with the Soft Clip plug-in. This format supports RGB and RGBA (alpha channel).

In its simplest form, the Soft Clip plug-in can exist on an empty container to play back a video clip in Viz Artist or Viz Engine. Other plug-ins can be mixed with Soft Clip to obtain different results. For example you can apply a Soft Clip plug-in to a Sphere and add a Material to affect the overall shape, color and shading of the video clip.

With Soft Clip you are able to control the video clip within the stage, unlike with the Clip Channel method. The video clip size can also be whatever you decide, however, it is recommended that you keep the dimensions in multiples of 8 (e.g. 128x128 or 256x512) for performance purposes.

Use of the Soft Clip plug-in might have an affect on performance and quality, hence, it is recommended to use the Performance Bar when working with the Soft Clip plug-in.

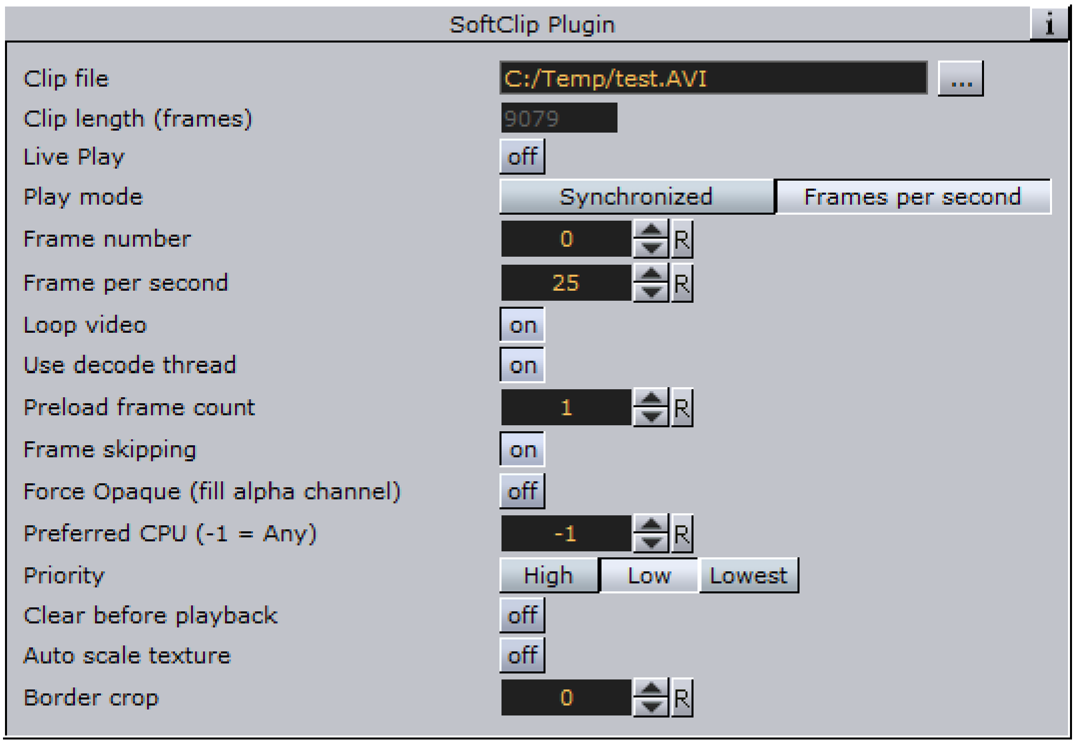
Important to note is that Viz Engine keeps all designer/operator/automation loaded scenes - and plug-ins - in memory. As a consequence, if you load 30 different templates that load 30 different scenes - that each has a single Soft Clip plug-in on it, then Viz Engine will have thirty Soft Clip plug-ins with 30 threads activated in memory. If every plug-in have specified that it should have 10 preloaded frames, then Viz Engine will have 300 uncompressed decoded frames preloaded in memory. This can take a substantial amount of memory and the combination of having that much data loaded and that many threads running, is likely to affect performance. This workflow situation has to be taken into consideration when designing and testing the design. Therefore, unloading scenes and cleaning up the Viz Engine renderers (a function in many Vizrt control client programs) should be used.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\SoftclipTexture.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To use Soft Clip

Properties and Parameters



SoftClip Plugin

Clip file: C:/Temp/test.AVI
Clip length (frames): 9079
Live Play: off
Play mode: Synchronized | Frames per second
Frame number: 0
Frame per second: 25
Loop video: on
Use decode thread: on
Preload frame count: 1
Frame skipping: on
Force Opaque (fill alpha channel): off
Preferred CPU (-1 = Any): -1
Priority: High | Low | Lowest
Clear before playback: off
Auto scale texture: off
Border crop: 0

- **Clip file**: Use this function to browse for the clip you want to play out. NOTE: It is important that all video clips are stored locally on the V: drive of the machine used to play out the graphics. Remember to save the clip locally on the client machines with preview (e.g. Viz Trio) and use the same V: drive so that the clip can be seen in preview.
- **Clip length**: Displays the clip’s number of frames.
- **Live Play**: Ensures that the clip will be played and synchronized to the frame rate of the Viz Engine. This is set in Viz Config / Output Format options. Disabling Live Play will allow the clip playback to be keyframe animated.
- **Play mode**: Synchronize means the clip will run synchronized with the scene playout.   This is set in Viz Config / Output Format options. Frames per second allows you to adjust for different speeds, set in the next parameter:
  - **Frame number**: Enables the frame numbers to be manually animated10
  - . This is an alternative to playing the whole clip.
  - **Frames per second**: Makes you able to customize the frame rate.

Note: There is no frame interpolation done when you change the frame rate from the default specified.

- **Loop Video**: Enabling this option means the video will loop automatically.
- **Use decode thread**: Enabling this option will split the processing of the clip evenly between the machine's processors. It is recommended to have this on. On multi-processor machines this will gain performance, but threading makes scheduling less predictable and may cause the Viz Engine frame rate to be less stable on a single CPU machine.
- **Preload frame count**: Works as a buffer to enable smoother playback of the clip. If using Live Play, the Preload frame count can help reducing spike-loads and avoid dropped frames.
- **Frame skipping**: With frame skipping enabled, Viz Engine will skip frames if it needs, to try and maintain real-time graphics playout. With frame skipping disabled the clip will play out in full, as fast as possible but will not try to maintain real-time playout. (Graphics will stagger on the output if the renderer drops out of real-time.) When using the video card's ring buffer functionality the SoftClip frame skipping should be turned off.
- **Force Opaque**: When running videos where the codec leaves the alpha channel blank, the video rendering may become completely transparent. This option will fill the alpha channel, thus making the video completely opaque.
- **Preferred CPU**: This function enables the user to specify a particular processor to handle video play out. The recommended CPU value is based on the machine spec you are running, if you have a quad core processor the CPU value can be set to 4, but to avoid any errors it would be recommended to set that value to 2. If you have a lower spec machine then the recommended value is 1.
- **Priority**: Sets the priority of drawing the video to a custom level in Viz Engine giving it more or less processor priority.
- **Auto scale texture**: This scales the video clip to the size of the underlying rectangle on which the video is placed in the container.
- **Border crop**: When using the auto scaling over, how much of the border to crop. Due to bilinear texture interpolation, you might see that color from the underlying texture is bleeding into the video texture area. Use this parameter to crop away the affected border pixels.

To use Soft Clip

1. Add a group container to the scene tree, and add the Soft Clip plug-in to it.
2. Open the Soft Clip editor.
3. Click the **Browse ...** button to select and load a clip.
    - Loading a clip will make it visible in the scene. By default, the Soft Clip plug-in has the Live Play and Loop video settings enabled, hence, the clip will instantly start playing (and looping) in the render window.
4. *Optional*: Enable the **Use decode thread** option.
5. Disable the **Live Play** option to enable the **Frame number** setting.
6. Animate the Frame number setting (e.g. from frame 10 to 30).

Note: Viz Engine will not interpret between keyframes, so if you choose to slow down an animation then the motion will not be smooth.

See Also

- Implementing Video
- Advanced Issues with Video Codecs
- Image Clip
- MoViz

12.14 Time

The following container plugins are housed in the Time folder:

- Analog Watch

12.14.1 Analog Watch

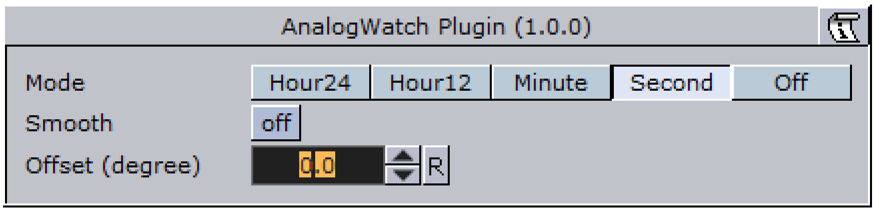
The Analog Watch plug-in lets you create a real-time animated clock of objects.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\AnalogWatch.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create an analog watch

Properties and Parameters


AnalogWatch Plugin (1.0.0)

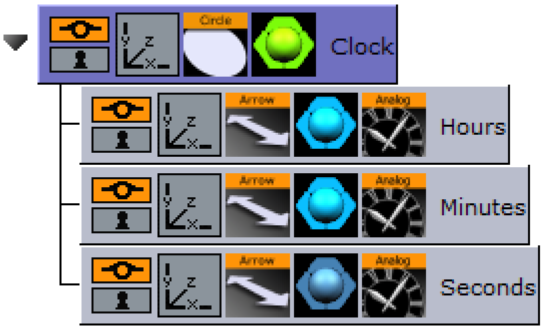
Mode: Hour24 | Hour12 | Minute | [✓ Second] | Off

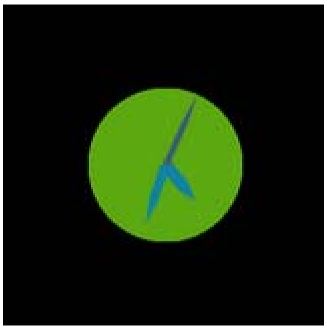
Smooth: off

Offset (degree): 0.0

- **Mode**: Sets the mode. Available options are Hour24, Hour12, Minute, Second and Off.
- **Smooth**: Minute hands usually rotate smoothly, second hands do that in some cases. Check this button to enable smooth rotation. Hour hands always rotate smoothly, so this switch is inactive if the plugin is set to Hour24 or Hour12 mode.
- **Offset**: 1200 usually is equal to a rotation of 0 degrees. Use this value to change that.
  - 1 hour in Hour24 mode: 15 degrees.
  - 1 hour in Hour12 mode: 30 degrees.
  - 1 minute in Minute mode: 6 degrees.
  - 1 second in Second mode: 6 degrees.

To create an analog watch





1. Add a **Circle** geometry to the scene tree, add material to it, and name it **Clock**.
2. Open the **Circle editor** and set scaling to **2.0**.
3. Add three **Arrow** geometries as sub containers of the **Clock** container, add material to them, and name the first **Hours, the** second **Minutes**, and the third as **Seconds.**
4. Open the **Arrow editor** for Hours and set the following properties and parameters:
   - Style1 to Flat.
   - Width and Arrow Width to 4.0
   - Percent to 30.0.
5. Open the **Arrow editor** for Minutes and set the following properties and parameters:
   - Style1 to Flat.
   - Width and Arrow Width to 3.0
   - Percent to 40.0.
6. Open the **Arrow editor** for Seconds and set the following properties and parameters:
   - Style1 to Flat.
   - Width and Arrow Width to 2.0
   - Percent to 50.0.
7. Add the Analog Watch plug-in to the Hours, Minutes and Seconds containers.
8. Open the Analog Watch editor for the Hours container and set it to Hour24 or Hour12.
9. Open the Analog Watch editor for the Minutes container and set it to Minute.
10. Open the Analog Watch editor for the Seconds container and set it to Second.

See Also

- Clock Rotation
- System Time

12.15 Tools

The following container plugins are housed in the Tools folder:

- Advanced Counter
- Autofollow

- Autorotate
- Bounding Actions
- Clock Rotation
- Cloner
- Colorize
- Counter
- Heartbeat
- Image Link
- Jack
- Level Of Detail (LOD)
- Magnify
- Max Size
- Max Size Lines
- Omo
- Pablo
- Parliament
- Rotations Order
- Slide Show
- System Time
- Temo
- Text BG
- Text Link
- Text Parameters

12.15.1 Advanced Counter

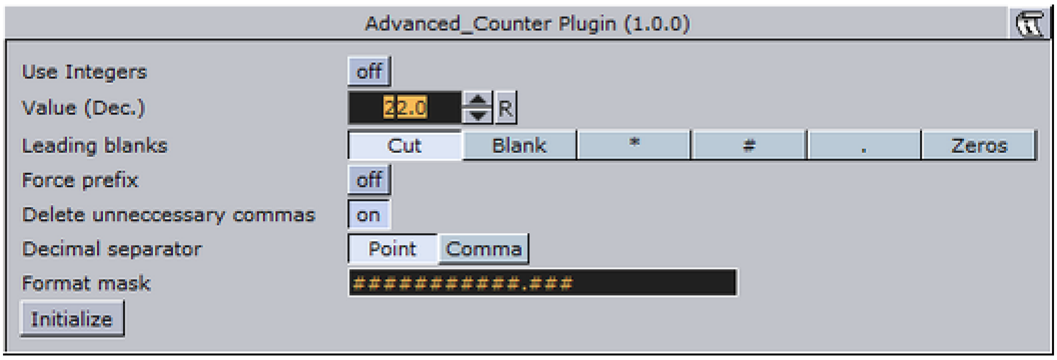
The Advanced Counter plug-in allows you to easily create an animated counting sequence. It shows values with decimals and it has the possibility to create a user defined format mask. Furthermore it can have prefixed values.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\Advanced_Counter.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create an advanced counter

Properties and Parameters



Advanced_Counter Plugin (1.0.0)

Use Integers: off

Value (Dec.): 22.0 R

Leading blanks: Cut | Blank | * | # | . | Zeros

Force prefix: off

Delete unneccessary commas: on

Decimal separator: Point | Comma

Format mask: ###########.###

Initialize

- **Use Integers**: Enables the use of integers only.
- **Value (Dec.)**, **Value (Int.)**: Sets the current value of the counter. Animate this value to create the counting sequence.
- **Leading blanks**: you can here choose what the counter does with leading blanks. They can either be **Cut** away or put in as **Blank**, asterisk, hash, dot or zeros.
- **Force prefix**: Enables both positive and negative values to have a prefix, not only negative values as is the default (+/-).
- **Delete unnecessary commas**: Removes superfluous commas when a specified format mask has more commas than the number needs. If you for instance have defined the mask: “$###,###,###,###.##” and you enter the number 4120.37, it will be displayed as “$,,4,120.37”. If you enable this option, the number will get a correct format: $4,120.37
- **Decimal separator**: Switches the decimal separator between point and comma.
- **Format mask**: Allows you to define a format mask. Each hash symbolizes an element of the total number. To alter, add or remove hashes. You can also add constant values, for example DM, NOK etc.
- **Initialize**: Starts the counter.

To create an advanced counter

1. Add a group container to the scene tree.
2. Add a font and material to the group container.
3. Add the Advanced Counter plug-in to the group container.
4. Open the Advanced Counter editor and animate the parameters.

12.15.2 Autofollow

The Autofollow plug-in is used to connect one object to another. This object is from then on dependent on the connected plug-in. Depending on the axis of your Object to follow, the bounding box will grow into the reverse direction. If you want to follow for example a text object, which writes from left to right, you have to set the axis of your text object to the left side, because the bounding box will grow to the right side. You can select for the following object either one axis

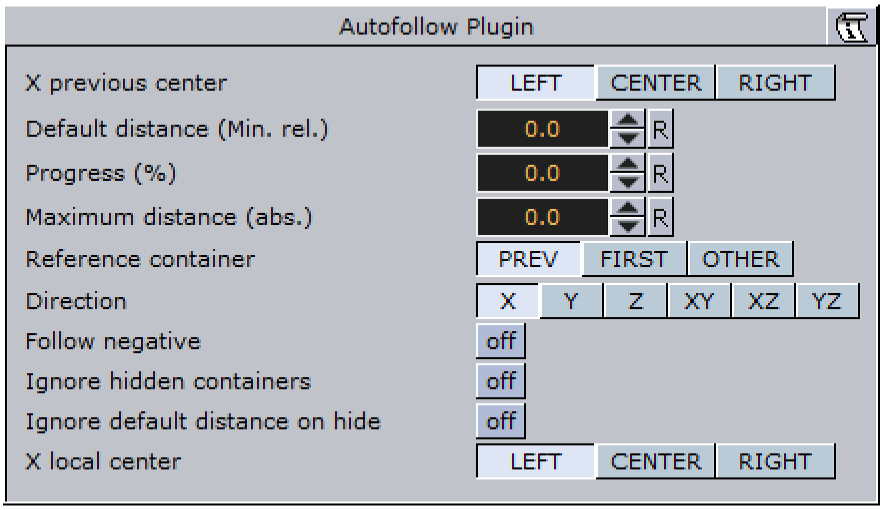
(X,Y,Z) or two (XY, XZ, YZ). Drag the Autofollow plug-in over the object container which should be dependent. It will be automatically be attached to the previous object in the scene tree. If you are not happy with this, it is possible to change it and to relate it to another object in the scene tree. Either you choose the first container or a complete different one.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\AutoFollow.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



Autofollow Plugin

X previous center: LEFT | CENTER | RIGHT

Default distance (Min. rel.): 0.0 R

Progress (%): 0.0 R

Maximum distance (abs.): 0.0 R

Reference container: PREV | FIRST | OTHER

Direction: X | Y | Z | XY | XZ | YZ

Follow negative: off

Ignore hidden containers: off

Ignore default distance on hide: off

X local center: LEFT | CENTER | RIGHT

- **X Previous Center**: In this section you can choose the alignment of your related object.
- **Default Distance (Min)**: Here you can set a distance from object to the related one. This is a relative value, which is dependent on the size of the related container.
- **Progress (%)**: This feature is useful, when you want to animate the related object. In this case you can set a default distance value and a maximum distance. What you need to do is just set keyframes on the progress. At the beginning of the animation you have 100%, e.g., and at the end 0%. The object will run from the maximum to the default distance value.
- **Maximum Distance**: Here you can set the maximum distance from the object to the related one. This is a absolute value, which is independent on the size of the related container.
- **Reference Container**: With these buttons (Previous, First, Other) you can change the related object. The default preference is 'Previous', where the previous container in the scene-tree is selected. If you choose 'Other', you have to drag your wished reference-container onto the empty button, which appears when you select 'Other'.
- **Direction**: These buttons give you the chance to choose in which axes there should be a dependency. You have the choice between X,Y,Z,XY,XZ or YZ.

- **Follow Negative**: If you select this feature, all numerical values are negated. E.g., when you change 'Default Distance (Min)' in a positive value, your object will be positioned in negative direction of the selected axis or axes.
- **Ignore hidden containers**: If the reference container is hidden, search for the next unhidden container depending on the reference container setting (only if set to previous or first).
- **Ignore default distance on hide**: This feature is useful, when you hide the reference container and do not want the default distance to be considered, in this case it will be null. If you unhide the reference container, the default distance will appear again.
- **X Local Center**: Choose your alignment for the local center in X-direction.

See Also

- Merging Containers

12.15.3 Autorotate

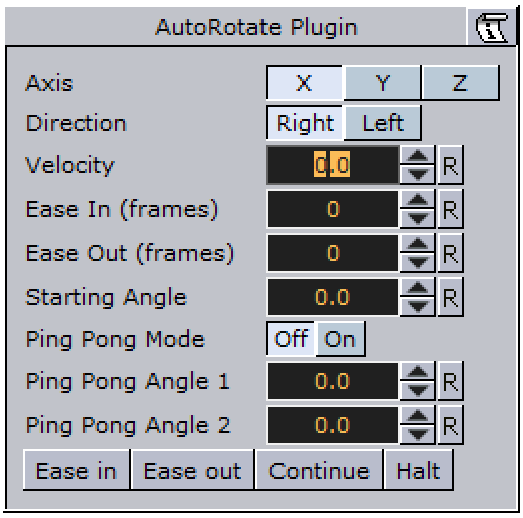
With the Autorotate function, a container can be set to continuously rotate round one of its axes.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\AutoRotate.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



AutoRotate Plugin

Axis	X Y Z
Direction	Right Left
Velocity	0.0
Ease In (frames)	0
Ease Out (frames)	0
Starting Angle	0.0
Ping Pong Mode	Off On
Ping Pong Angle 1	0.0
Ping Pong Angle 2	0.0

- **Axis**: Selects the rotation axis either **X**, **Y** or **Z**.
- **Direction**: Selects the direction of the rotation, either **Right** or **Left**.

- **Velocity**: Sets the speed of the rotation. The unit is degrees per frame.
- **Ease In (frames)**: Sets the number of frames the object will use to accelerate smoothly from no motion to the speed set in **Velocity**.
- **Ease Out (frames)**: Sets the number of frames the object will use to retard from the rotation speed to a stop.
- **Starting Angle**: Sets the angle from the objects current position, from which the object is to start from.
- **Ping Pong**: Enables a “ping pong” motion where the object rotates forward and backward between two angles on the axis.
- **Ping Pong Angle 1**: Sets angle 1 for the “Ping Pong Mode”.
- **Ping Pong Angle 2**: Sets angle 2for the “Ping Pong Mode”.
- **Ease in**: Initializes the rotation from the starting angle selected and with the **Ease in** value selected.
- **Ease out**: Stops the rotation. The stop motion is influenced by the **Ease out** parameter.
- **Continue**: Starts the rotation from the point where it was halted. The value of **Ease in** is disregarded.
- **Halt**: Stops the rotation without regard to **Ease out**.

12.15.4 Bounding Actions

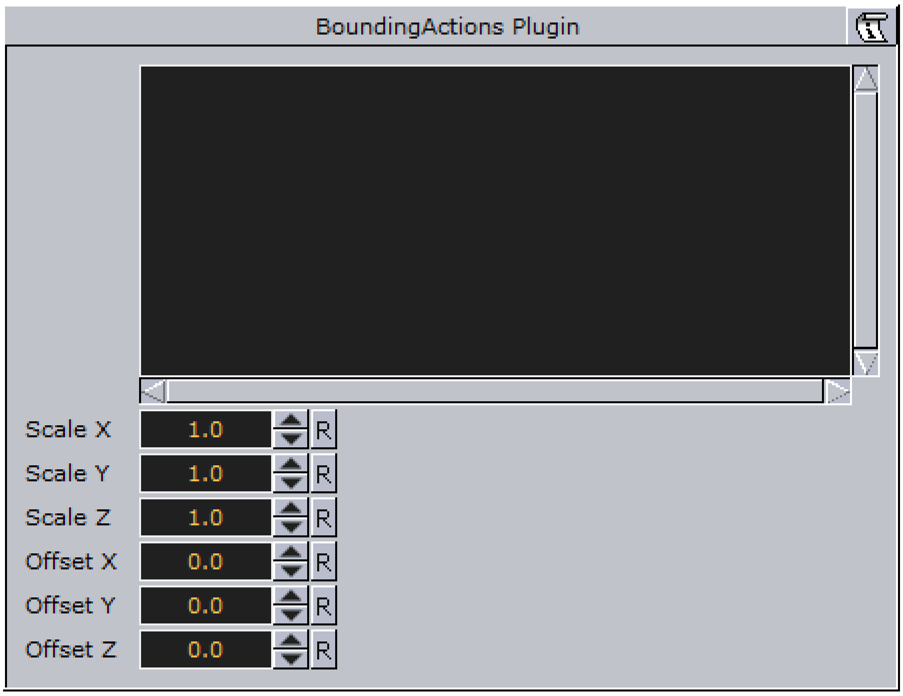
The Bounding Actions plug-in enables you to run Viz actions depending on the size of the bounding box.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\BoundingActions.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



BoundingActions Plugin

Scale X	1.0
Scale Y	1.0
Scale Z	1.0
Offset X	0.0
Offset Y	0.0
Offset Z	0.0

- **Text box**: The command, that will be executed every time the bounding box changes.
- **Scale X, Y and Z**: Applies a scaling to the values %dx %dy %dz.
- **Offset X, Y and Z**: Adds an offset to the values %dx %dy %dz.

Syntax:

- **%dx**: Substitutes the width of the bounding box.
- **%dy**: Substitutes the height of the bounding box.
- **%dz**: Substitutes the depth of the bounding box.
- **%xbool**: 0 if dx <= 0 else 1.
- **%ybool**: 0 if dy <= 0 else 1.
- **%zbool**: 0 if dz <= 0 else 1.
- **%container**: Path of the container the bounding action is applied to.
- **%container[path]**: Relative path of the container the bounding action is applied to.

Example:

```
THIS_SCENE*TREE*#297*TRANSFORMATION*POSITION SET %dx %dy %dz
```

Example:

```
%container[$textnode1$textnode2]*GEOM*TEXT SET %xbool
```

To use bounding actions

1. Drag the plug-in onto a container.
2. Insert a command in the Text box (see examples above).
   - Now every time the bounding box changes (a sub container is moved outside the current bounding box or the bounding box plug-in is used to modify it), this command is executed.

12.15.5 Clock Rotation

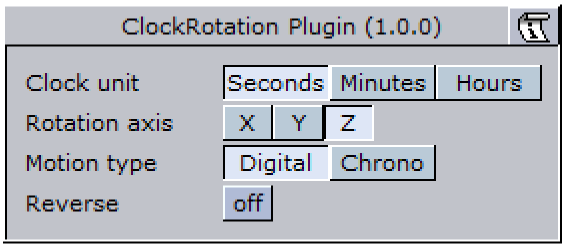
Animates any object as a rotating clock.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\ClockRotation.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters


ClockRotation Plugin (1.0.0)

Clock unit: Seconds | Minutes | Hours

Rotation axis: X | Y | Z

Motion type: Digital | Chrono

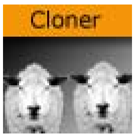
Reverse: off

- **Clock Unit**: Sets the units for the rotation of the object. Available options are Seconds, Minutes and Hours
- **Rotation axis**: Sets the axis for the rotation of the object. Available options are X, Y and Z
- **Motion type**: Sets the motion type for the rotated object. Digital will show a ticking motion, and Chrono a smooth motion.
- **Reverse**: Sets the direction of the rotation. When disabled (off) the rotation is clockwise, and when enabled (on) the rotation is counter-clockwise.

See Also

- Analog Watch
- System Time

12.15.6 Cloner



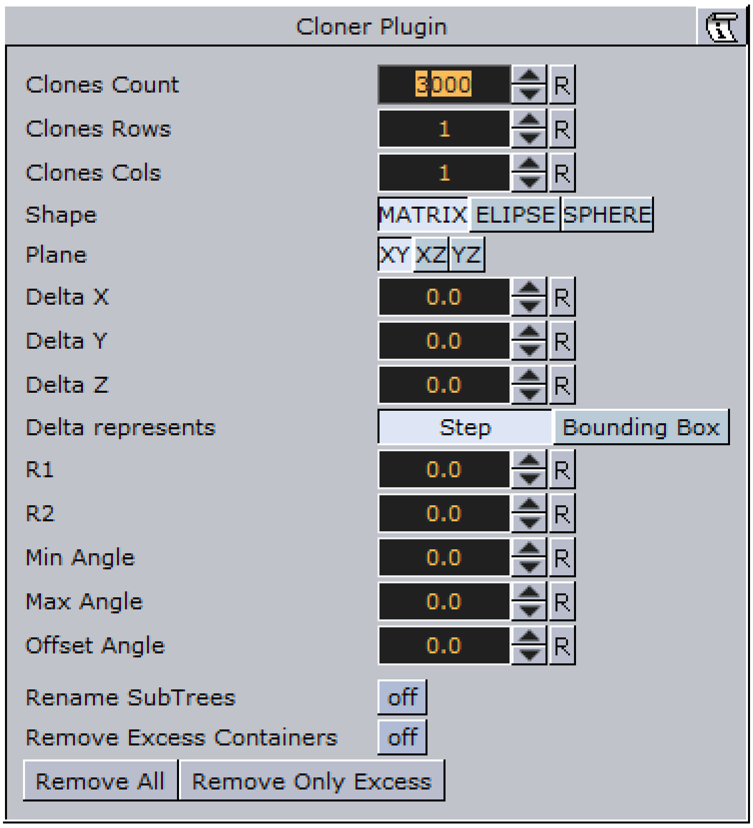
Creates a number of clones of a given container (target).

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\Cloner.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters


Cloner Plugin

Clones Count 3000 R
Clones Rows 1 R
Clones Cols 1 R
Shape MATRIX ELIPSE SPHERE
Plane XY XZ YZ
Delta X 0.0 R
Delta Y 0.0 R
Delta Z 0.0 R
Delta represents Step Bounding Box
R1 0.0 R
R2 0.0 R
Min Angle 0.0 R
Max Angle 0.0 R
Offset Angle 0.0 R
Rename SubTrees off
Remove Excess Containers off
Remove All | Remove Only Excess

- **Clones Count**: Specifies the max. number of clones to be created.
- **Clones Rows**: Defines, together with Clones Cols, how clones are being laid out. Layout depends on the Shape parameter which can be Matrix (grid), Ellipse and Sphere.
- **Clones Cols**: See Clones Rows description above.
- **Shape**: Sets the different shapes. Available options are Matrix, Ellipse and Sphere.
- **Rename Subtrees**: Clones will be labeled "group_x_y" per default. If this flag is set, clones will be labeled "<name of target>_x_y". Only takes effect upon future changes in the subtree.
- **Rename Excess Containers**: Suppose a matrix of 10x10 clones. If you now change the matrix to be made of 5x5 clones you will be left with 75 orphans, which will be removed if this flag is set. Only takes effect upon future changes in the subtree.
- **Remove All (button)**: Remove all clones right now.
- **Remove Only Excess (button)**: Remove excess clones right now.

Matrix and Ellipse shape specific settings:

- **Plane**: Specifies along which axes, newly generated clones are going to be laid out. Default is XY, alternatives are XZ and YZ.

- **Delta X, Y and Z**: Specifies an relative offset in X,Y and Z axis between each clone. Actually defines how closely clones are being packed into the matrix or ellipse.
- **Delta represents**: **Step** directly affects the translation matrix of the clones 1:1. For example a value of 200 for Delta X means that clone A (located at position 0) will be followed by clone B translated 200 units in X. The container's transformation matrix is being modified accordingly. **Bounding Box** does not affect the clones transformation matrix 1:1, but instead applies the transformation relative to the bottom-left edge of the target's bounding box in matrix mode. Suppose a rectangle of 100x100 is used as target. Delta X of 100 in Step mode will cause all clones to be positioned with an offset of 100 units in X. In Bounding Box mode a value of 100 will produce an effective offset of only 50.
- **R1 and R2**: Specifies values of radius 1 and radius 2, defining the ellipse. Used in ellipse mode.
- **Min and Max Angle**: Specifies an open ellipse from for example 45° to 175° instead of 0° to 360°. Used in ellipse mode.
- **Offset Angle**: Rotates the individual clones along the axis. This is not affected by setting of Plane. If the ellipse is laid out in XY this parameter will affect the Z-axis. Used in ellipse mode.

Sphere shape specific settings:

- **Radius**: Specifies the radius of the sphere composed by the given number clones.
- **Min and Max Azimuth**: Same as Min and Max angle for the ellipse.
- **Min and Max Pitch**: Specifies the position of the poles of the sphere. The poles can only be shifted towards the center of the sphere. This will actually result in a sphere with its poles being cropped.

To clone container

1. Add a new group to the scene tree, and add the Cloner plug-in to it.
2. Add the object (target) to be cloned as the first child of this group.
   - By modifying parameters Clones Count, Rows and Cols a corresponding number of clones will be created.
   - Operating in shape mode Matrix you can alter Delta X, Y and Z parameters in order to create a layout for the newly generated clones.

12.15.7 Colorize

The colorize plug-in allows you to transform the colors of containers within a container group. If you for instance have 15 sub containers in a group, the colorize plug-in assign numbers to each container by the order they have in the group. You can then define for each sub container which color it should have to start with and which color it should change to.

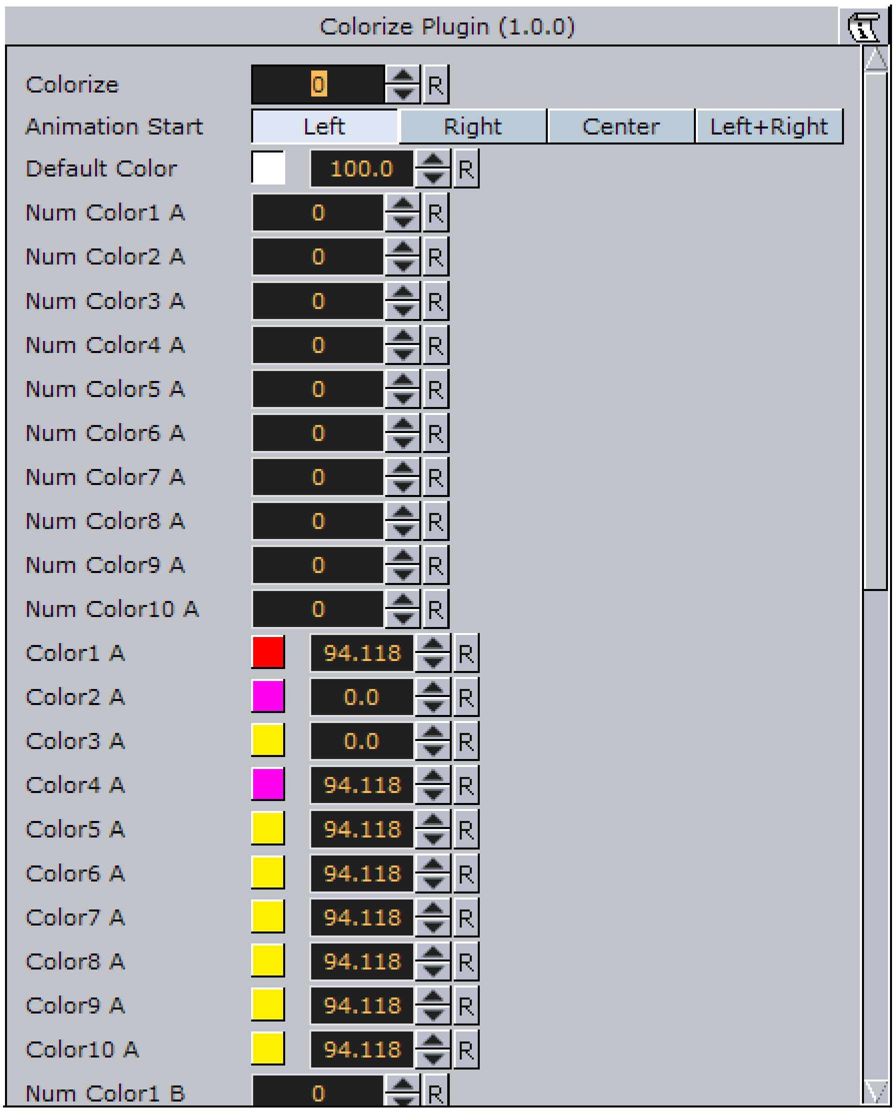
The plug-in can define up to 10 colors. Each color is able to influence more than one container. The Num Color1-10 parameters allow you to link multiple containers to one color in the colorize plug-in. Each of the sub containers must have its own material for the plug-in to work.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\Colorize.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



Colorize Plugin (1.0.0)

Colorize		0
Animation Start	Left / Right / Center / Left+Right	
Default Color	⬜	100.0
Num Color1 A		0
Num Color2 A		0
Num Color3 A		0
Num Color4 A		0
Num Color5 A		0
Num Color6 A		0
Num Color7 A		0
Num Color8 A		0
Num Color9 A		0
Num Color10 A		0
Color1 A	🟥	94.118
Color2 A	🟪	0.0
Color3 A	🟨	0.0
Color4 A	🟪	94.118
Color5 A	🟨	94.118
Color6 A	🟨	94.118
Color7 A	🟨	94.118
Color8 A	🟨	94.118
Color9 A	🟨	94.118
Color10 A	🟨	94.118
Num Color1 B		0

- **Colorize**: Runs the color transformation. One number corresponds to one container.
- **Animation Start**: Allows you to select the starting point for the color change.

- **Default Color**: Sets the color for containers in the group which are not selected for color transformation. To set it, use the color editor below or drag a material from the server view onto the small square.
- **Num Color1 A to 10 A**: Assigns the starting colors to containers, from which the color change starts. The value sets the number of containers to be linked with the corresponding color in the Color1 A: 10 A parameters. The function starts counting from the first undefined container. This means that, if color 1 A has the value 2, color 2 A has the value 3 and 3 A has the value 1, the two first containers get the properties of color 1 A, container 3: 5 get the properties of color 2 A and container 6 gets the properties for color 3 A.
- **Color1 A to 10 A**: Sets the colors for 1 A: 10 A. You can set each color using two methods:
  - Either select the color by clicking the color icon and then set the color properties at the color editor at the bottom, or
  - Drag a material from the server view onto the color icon of the color you want to set.
- **Num Color1 B to 10 B**: Assigns the ending color to containers, to which the color change ends. The value sets the number of containers to be linked with the corresponding color in the Color 1 B: 10 B parameters. The function starts counting from the first undefined container. This means that, if color 1 B has the value 2, color 2 B has the value 3 and 3 B has the value 1, the two first containers get the properties of color 1 A, container 3: 5 get the properties of color 2 B and container 6 gets the properties for color 3 B.
- **Color1 B to 10 B**: Sets the color for 1 B: 10 B. You can set each color using two methods:
  - Either select the color by clicking the color icon and then set the color properties at the color editor at the bottom, or
  - Drag a material from the server view onto the color icon of the color you want to set.
- **Rebuild**: After having made changes to colors and assigning containers, click rebuild to apply the changes.

12.15.8 Counter

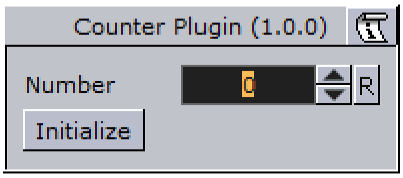
The counter is a simple plug-in that can create a count up or count down at any given range and at any given speed. It shows integral numbers.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\Counter.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To use the counter

Properties and Parameters


Counter Plugin (1.0.0)

Number 0

- **Number**: Sets the current number of the counter. Animate this value to create the wanted count-up or -down.
- **Initialize**: Starts the operation.

To use the counter

- Add the Counter plug-in to a container with a font, and click Initialize.

12.15.9 Heartbeat

The Heartbeat plug-in creates a hear beat animation. Includes animation of:

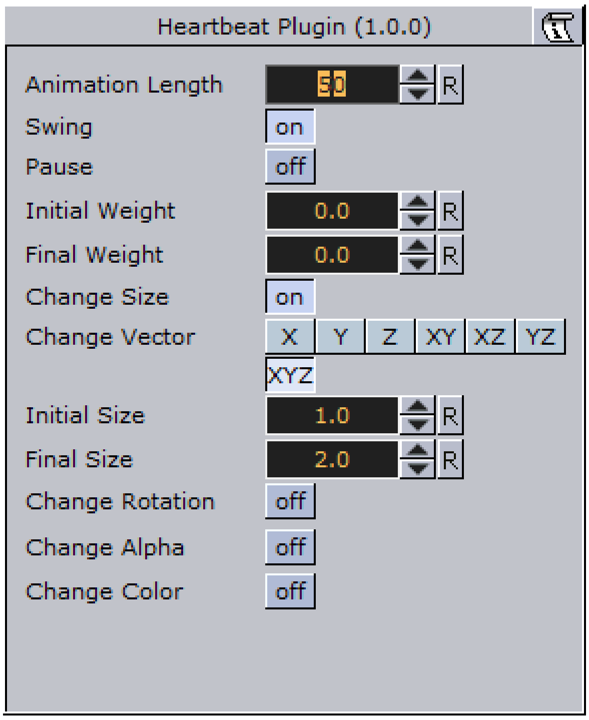
- Size, along different axes, axis-pairs or all axes.
- Rotation, around three main axes.
- Alpha.
- Color.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\Heartbeat.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



Heartbeat Plugin (1.0.0)

Animation Length 50 R

Swing on

Pause off

Initial Weight 0.0 R

Final Weight 0.0 R

Change Size on

Change Vector X Y Z XY XZ YZ XYZ

Initial Size 1.0 R

Final Size 2.0 R

Change Rotation off

Change Alpha off

Change Color off

- **Animation Length**: Sets the length of the animation. Parameter Name: `animLength`.
- **Swing**: Enables the object to have a swinging action (back and forth). Parameter Name: `swing`.
- **Pause**: Enables the object to pause before repeating the action. Parameter Name: `pause`.
  - **Pause Length**: Sets the length of the pause. Parameter Name: `pauseLength`.
- **Change Size**: Enables the change size parameters for changing the vector(s), initial and final weight of the object's animation. Parameter Name: `changeSize`.
  - **X, Y, Z, XY, XZ, YZ, XYZ**: Sets animation vector(s). Parameter Name: `changeVector`.
  - **Initial Weight**: Sets the initial weight of the object. The higher the value the slower the animation will be at first. Parameter Name: `initialWeight`.
  - **Final Weight**: Sets the final weight of the object. Initial Weight. Parameter Name: `finalWeight`.
- **Change Rotation**: Enables the rotation parameters changing the rotation axes, initial and final angles. Parameter Name: `changeRotation`.
  - **X, Y, Z**: Rotates the object on the X, Y or Z axis. Parameter Name: `rotAxes`.
  - **Initial Angle**: Sets the initial angle of the object. Parameter Name: `initialAngle`.
  - **Final Angle**: Sets the final angle of the object. Parameter Name: `finalAngle`.
- **Change Alpha**: Enables the alpha parameters changing the initial and final alpha values. Parameter Name: `changeAlpha`.
  - **Initial Alpha**: Sets the initial alpha value. Parameter Name: `initialAlpha`.
  - **Final Alpha**: Sets the final alpha value. Parameter Name: `finalAlpha`.

- **Change Color**: Enables the color parameters changing the color of the object. Parameter Name: `changeColor`.
  - **Source Color**: Sets the source color. Parameter Name: `sourceColor`.
  - **Target Color**: Sets the target color. Parameter Name: `targetColor`.

12.15.10 Image Link

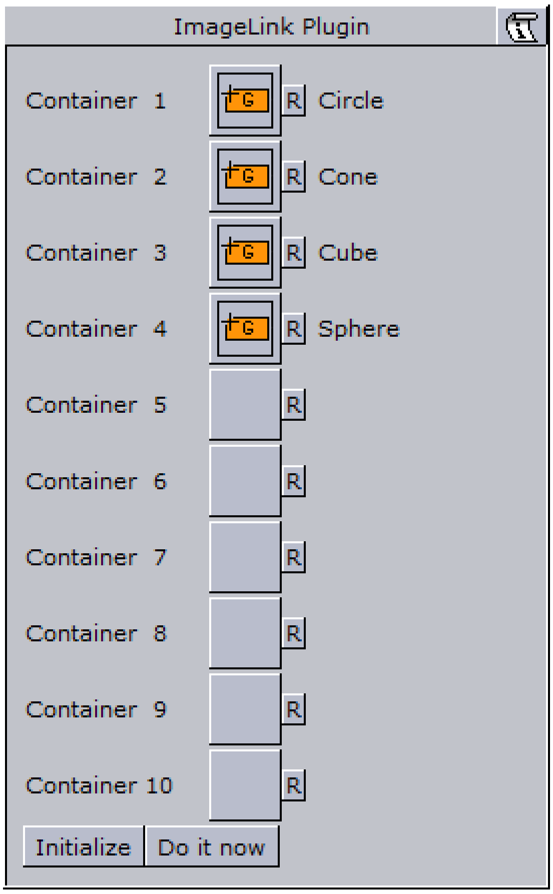
The Image Link plug-in copies the image/texture from the source container to up to 10 linked containers. The image you load in the source container will be repeated in all the linked containers.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\ImageLink.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To link an image to multiple containers

Properties and Parameters



- **Container 1: 10**: Drop zones for the containers you want to link.
- **Initialize**: Initializes the plug-in.
- **Do it now**: Performs the operation immediately.

To link an image to multiple containers

1. Add a group container to the scene tree.
2. Add the Image Link plug-in and the source image to the group container.
3. Add a number of containers to the scene tree with a dummy image on each.
4. Open the Image Link editor.
5. Drag and drop the containers with the dummy images onto the Container 1-n drop zones.
6. Click **Initialize** to apply change.

See Also

- Text Link

12.15.11 Jack

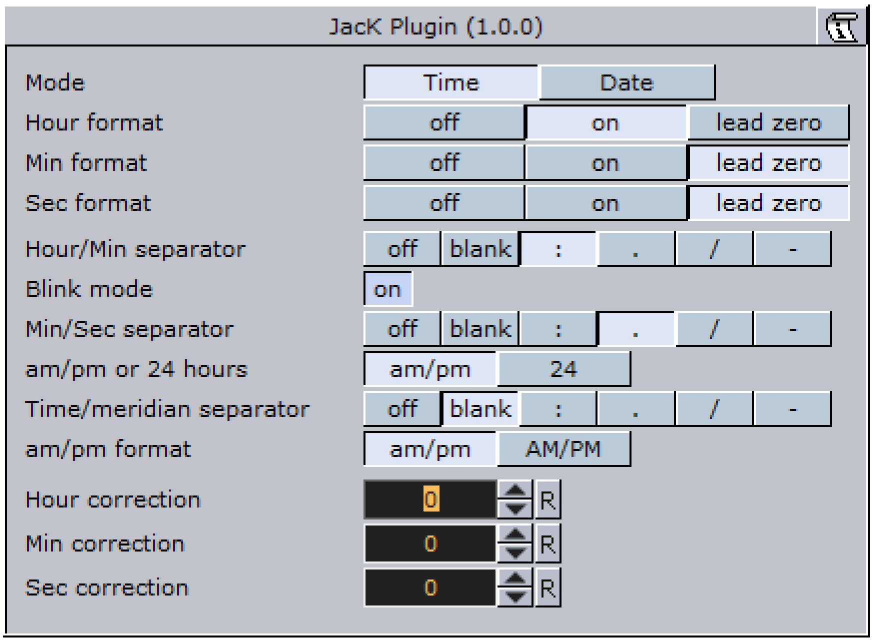
The Jack plug-in allows you to display a digital clock and date. The plug-in is highly customizable allowing you to set your own time and date formats and correction values.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\Jack.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



JacK Plugin (1.0.0)

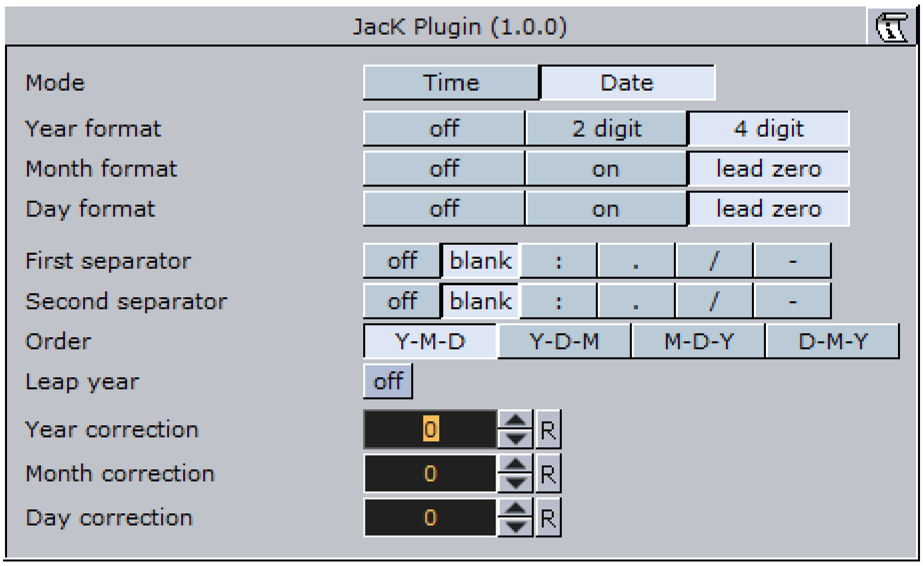
Mode: Time | Date
Hour format: off | on | lead zero
Min format: off | on | lead zero
Sec format: off | on | lead zero
Hour/Min separator: off | blank | : | . | / | -
Blink mode: on
Min/Sec separator: off | blank | : | . | / | -
am/pm or 24 hours: am/pm | 24
Time/meridian separator: off | blank | : | . | / | -
am/pm format: am/pm | AM/PM
Hour correction: 0 R
Min correction: 0 R
Sec correction: 0 R

Time mode has the following properties and parameters:

- **Hour format**: Sets how to display hours (either off, on or with a leading zero if the value is less than 10).
- **Min format**: Sets how to display minutes (see Hour format).
- **Sec format**: Sets how to display seconds (see Hour format).
- **Hour/Min separator**: Sets character separation for hours and minutes.
- **Blink mode**: If set to on the separator between hours and minutes will blink, this means - every odd second it will be turned off.
- **Min/Sec separator**: Sets a separator for minutes and seconds.

Note: Although a format is switched off, the separator will still be displayed.

- **am/pm or 24 hours**: Switches between the am/pm and 24 hour format
- **Time/meridian separator**: Sets a separator for the am/pm hour format.
- **am/pm format**: Sets the case of am/pm to upper or lower.
- **Hour correction**: Enables you to correct the hour value of the system time. Useful for displaying the local time of a location in a different time zone.
- **Min correction**: Sets correction value for minutes (see Hour correction).
- **Sec correction**: Sets correction value for seconds (see Hour correction).



JacK Plugin (1.0.0)

Mode: Time [✓ Date]
Year format: off | 2 digit | [✓ 4 digit]
Month format: off | on | [✓ lead zero]
Day format: off | on | [✓ lead zero]
First separator: off | [✓ blank] | : | . | / | -
Second separator: off | [✓ blank] | : | . | / | -
Order: [✓ Y-M-D] | Y-D-M | M-D-Y | D-M-Y
Leap year: off
Year correction: 0
Month correction: 0
Day correction: 0

Date mode has the following properties and parameters:

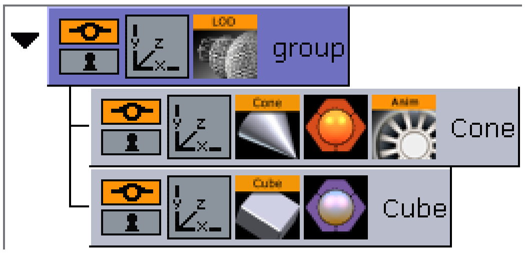
- **Year format**: Sets how to display the year format (off, 2 digits or 4 digits, please keep in mind that in the year 10000 this will be 3 digits or 5 digits, due to the algorithm used).
- **Month format**: Sets how the month is displayed, either off, on or with a leading zero from january to september.
- **Day format**: Sets how the day is displayed (see Month format).
- **First separator**: Sets the separator character for the first two elements of the date. If the dot value is selected an additional blank character will be inserted automatically.
- **Second separator**: Sets the separator character for the last two elements of the date.
- **Order**: Sets the order of the elements of the date.
- **Leap year**: Corrects the date value with correction values. If enabled, February 29th is used in the date calculation.
- **Year correction**: Sets a correction value for the year.
- **Month correction**: Sets a correction value for the month.
- **Day correction**: Sets a correction value for the day.

12.15.12 Level Of Detail (LOD)

Level Of Detail (LOD) is a mechanism for controlling the level of complexity of a 3D object, depending on camera position and object size. As an object becomes smaller on the screen, the objects polygons become smaller. At a certain size those polygons are small enough to be replaced by a lower number of larger polygons, without loosing realism. The goal is to maintain the overall shape of the object so it will not change significantly, when the number of polygons are changing.

Viz Artist has a built-in LOD mechanism for text objects and internal objects, such as Sphere, Cylinder and so on. This mechanism changes the tessellation of the object, based on its size on screen. However, there is no LOD mechanism for imported objects and models. This is where the LOD plug-in can be used. The LOD plug-in enables switching between different representations (levels) of the same object, based on its distance from the camera and on the zoom level of the camera. A switching range is assigned to each representation: to determine the specific distance from camera in which it will be replaced by the next representation.

In addition to that, the zoom of the camera is taken into consideration; objects that are far away from the camera will look closer (and larger) at a narrow zoom position, or smaller at wide zoom. To accommodate for different zoom values, the LOD plug-in assumes that the switching ranges are correlative to a reference zoom value. When the actual zoom value differs from the reference zoom value, the switching ranges are automatically adjusted.



In order to use the LOD plug-in on an object, you have to import several representations of that object from an external source. The different representations of the object have to be placed under a group node, in the order of detail, this means the most detailed level will be on the top of the group. When the LOD plug-in is placed on the group container it will make only one of its children (levels) visible at a given time, based on the distance between the center of the object and the camera, as well as the current zoom value, and other selectable parameters of the plug-in.

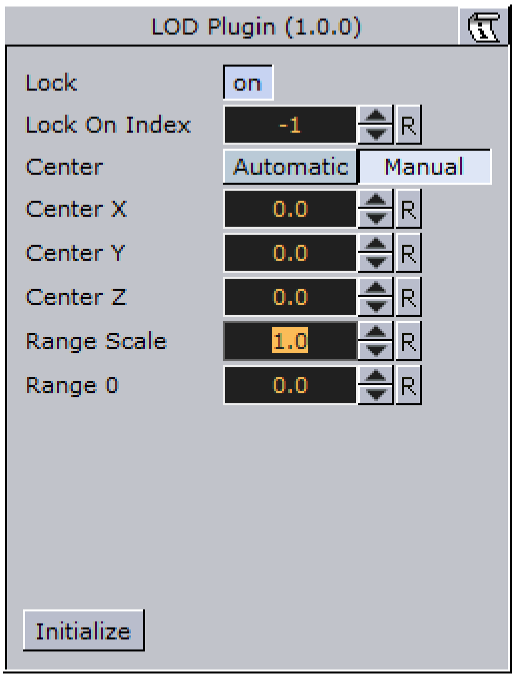
The LOD container plug-in works in conjunction with the Level Of Detail (LOD) Manager scene plug-in that controls all the LOD plug-ins in the scene.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\LOD.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters


LOD Plugin (1.0.0)

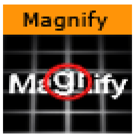
Lock: on
Lock On Index: -1
Center: Automatic / Manual
Center X: 0.0
Center Y: 0.0
Center Z: 0.0
Range Scale: 1.0
Range 0: 0.0

- **Lock**: In certain cases it is desirable to disable the distance switching and lock the LOD plug-in to show one of the levels.
- **Lock On Index**: Displays index of the selected level when **Lock** is **on**.
- **Center**: Defines whether the distance from the camera will be measured from the origin of the object (**Automatic**) or from a specific point in the object, for example one of the corners (**Manual**)
- **Center X**, **Y** and **Z**: Sets the point from which the distance from the camera is measured (in object coordinates), if the **Center** parameter is set to **Manual**.
- **Range Scale**: Sets a scale factor on the switching ranges. This scale factor allows shortening or lengthening the distance at which object representations are switched.
- **Range 0**: Sets the switching ranges. Those are the distances at which object representations are switched. The number of entries is n+1 where n is the number of children in the group on which the LOD plugin is assigned
- **Initialize**: Re-initializes the LOD plug-in. This is required if the number of children of the LOD group has changed after the LOD plug-in was assigned to the group.

See Also

- Level Of Detail (LOD) Manager

12.15.13 Magnify



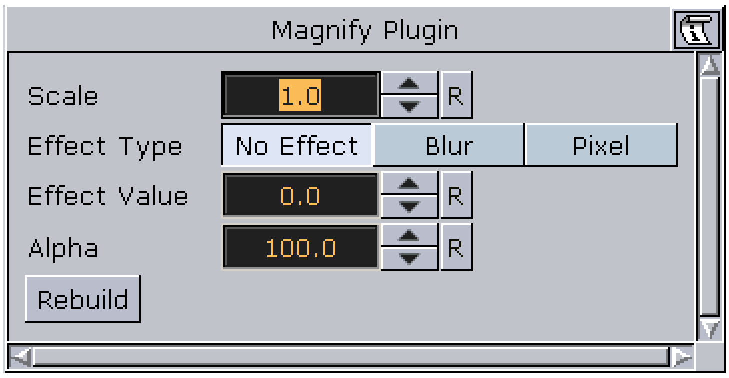
The Magnifying plug-in offers the possibility to create a simulation of a looking glass and other similar effects. At an area defined by an overlying object an image can be shown in a magnified way.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\Magnify.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters


Magnify Plugin

Scale 1.0 R

Effect Type: No Effect | Blur | Pixel

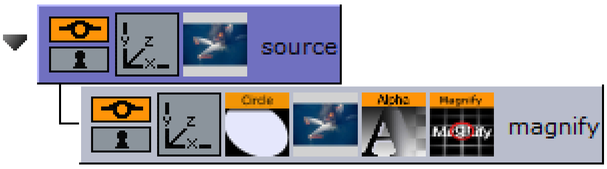
Effect Value 0.0 R

Alpha 100.0 R

Rebuild

- **Scale**: Sets the scaling of the magnifying effect
- **Effect Type**: Adds an extra effect, either **Blur** or **Pixel**. This only works on onux.
- **Effect Value**: Sets the value for the selected extra effect.
- **Alpha**: Sets the alpha value for the container.
- **Rebuild**: Applies the changes to the texture.

To magnify a texture



1. Add a container to the scene tree, and name it **source**.
2. Add the image or texture that is to be magnified to the source container.
3. Add a sub container to the source container, and name it **magnify**.
4. Add the same image or texture to the magnify container that was used in the source container.
5. Add the Circle geometry to the magnify container.
6. Open the transformation editor for the magnify container and set Position Z to 1.0.
7. Add the Magnify plug-in to the magnify container.
8. Open the Magnify editor and set Scale to 2.0.

12.15.14 Max Size

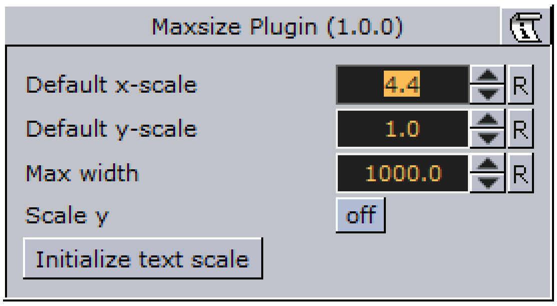
The Max size plug-in is used to set max size parameters for text objects. This allows you to have control over the space the text should use such that it does not “overflow” the designated text area.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\Maxsize.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To set max size

Properties and Parameters



Maxsize Plugin (1.0.0)

Default x-scale 4.4 R

Default y-scale 1.0 R

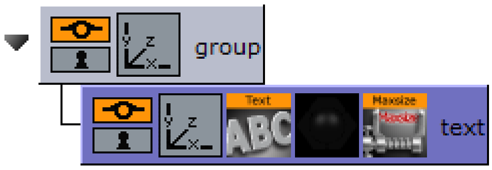
Max width 1000.0 R

Scale y off

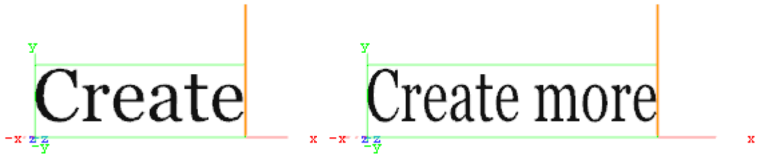
Initialize text scale

- **Default x-scale**: Sets the default x-scale.
- **Default y-scale**: Sets the default y-scale.
- **Max width**: Sets the max width.
- **Scale y**: Scales the y-value accordingly when max width is reached.
- **Initialize text scale**: Initializes the plug-in parameters.

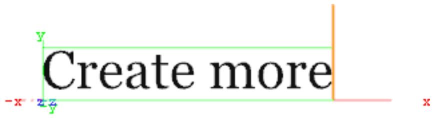
To set max size



1. Add a group container to the scene tree
2. Add a group container as a sub container of the first (scaled to 1,1,1) and name it **text**
3. Add a font (e.g. Georgia Regular) to the text container
4. Add the **MaxSize plugin** to the text container
5. Click the MaxSize plugin and set **Max width** to **200.0**



6. Click the font and type *Create*
7. Now type *Create more*
   - This will automatically scale the font’s X value as it limits itself to the Max width



8. *Optional*: Click the MaxSize plugin and enable the **Scale Y** parameter to automatically scale the font’s Y value

See Also

- Max Size Lines

12.15.15 Max Size Lines

The Max Size Lines plug-in solves the problem of using Text FX with multiline text in right-to-left languages.

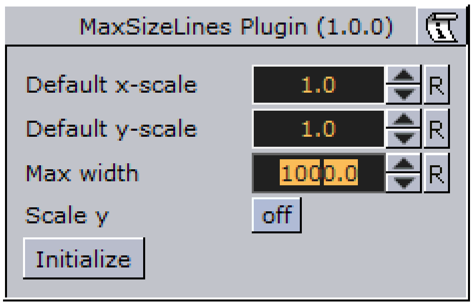
Note that each line of text must be in separate containers, and organized as a sub group of the container holding the Max Size Lines plug-in.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\MaxSizeLines.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



MaxSizeLines Plugin (1.0.0)

Default x-scale 1.0 R

Default y-scale 1.0 R

Max width 1000.0 R

Scale y off

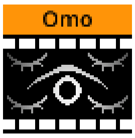
Initialize

- **Default x-scale**: Sets the default x-scale.
- **Default y-scale**: Sets the default y-scale.
- **Max width**: Sets the max width.
- **Scale y**: Scales the y-value accordingly when max width is reached.
- **Initialize text scale**: Initializes the plug-in parameters.

See Also

- Max Size

12.15.16 Omo



The Object moving (Omo) plug-in gives the user the possibility to create a very realistic animation of a complex imported 3D object, where the object not just changes position or rotates, but where it changes form and moves in a realistic way, like a man walking or a fish swimming.

The plug-in can also create animation sequences with other sorts of objects. Anything you can put in a container can be used as an element in an Omo animation process.

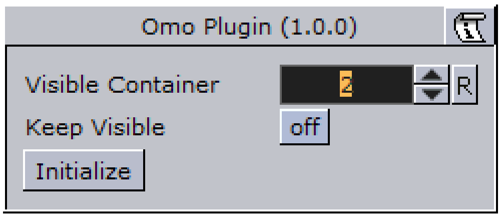
The plug-in hides all sub containers and shows them one by one. The one to be shown is controlled by the **Visible Container** parameter. By animating this value an animation sequence of all the containers is made. The plug-in can also be used to show one group at a time.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\Omo.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters

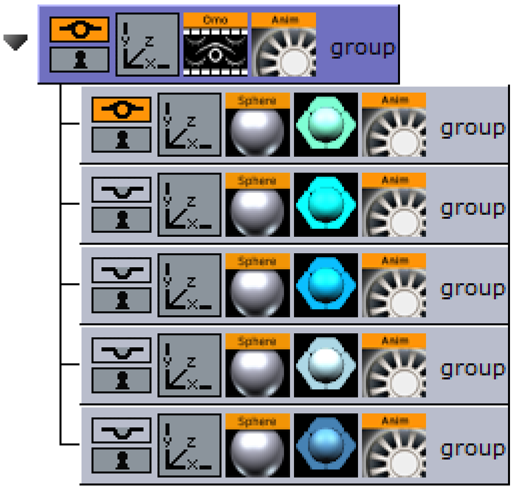

Omo Plugin (1.0.0)

Visible Container: 2

Keep Visible: off

- **Visible Container**: Allows you to select which container that is to be visible.
- **Keep Visible**: Enables you to keep containers visible after they have been revealed one by one.
- **Initialize**: Initializes the Omo plug-in. All containers in the group are then hidden, except the one selected by the **Visible Container** parameter.

To animate one object at a time



1. Create a group and add the Omo plug-in to it.
2. Create a number of sub containers to the root container.
3. Add a Sphere geometry and material to each sub container.
4. *Optional*: Animate the sphere.

5. Open the Omo editor, click the Initialize button, and animate the Visible Container parameter.

See Also

- Control Omo

12.15.17 Pablo

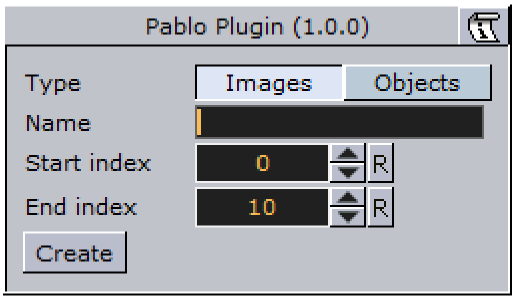
Pablo is an object import plug-in that enables you to import multiple indexed images/objects from the pool under the local container Pablo is added to.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\Pablo.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



Pablo Plugin (1.0.0)

Type: Images | Objects

Name:

Start index: 0

End index: 10

- **Type**: Select wether you want to import images (from the image pool) or 3D objects (from the object pool).
- **Name**: Sets the name of the images/objects you want to import, including path and index number format. Index number format is specified with characters. Use multiple characters for index numbers with leading zeroes. The name must match the path and name of an existing image/object in one of the pools.
- **Start index**: Sets the index number of the first object to be imported.
- **End index**: Sets the index number of the last file to be imported.

12.15.18 Parliament

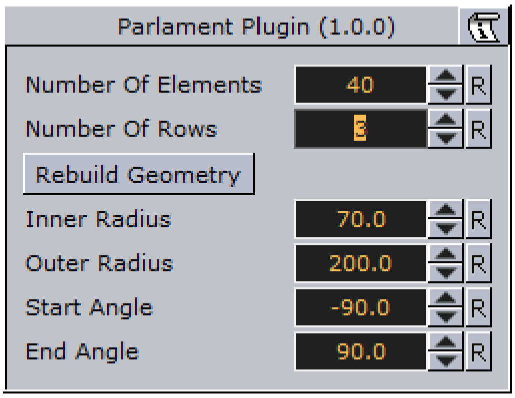
The Parliament plug-in is specially developed for creating visualizations of parliament seats in graphics designed for election programs, but it can of course be used in other ways. The plug-in creates a “parliament like” seating structure using an object of your selection.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\Parlament.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a parliament shape

Properties and Parameters



Parlament Plugin (1.0.0)

Number Of Elements 40

Number Of Rows 3

Rebuild Geometry

Inner Radius 70.0

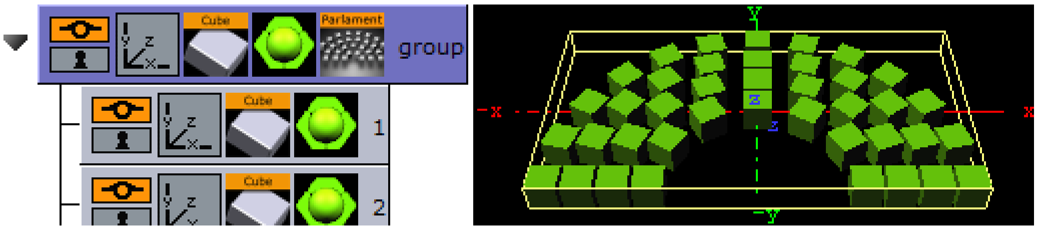
Outer Radius 200.0

Start Angle -90.0

End Angle 90.0

- **Number of Elements**: Allows you to set the number of seats to be created in the parliament structure.
- **Number of Rows**: Allows you to decide how many rows the parliament seats should be distributed over.
- **Scale**: Sets the Scaling (locked) of the generated sub containers. Default is 1.0.
- **Rebuild Geometry**: After having made changes to either **Number of Elements** or **Number of rows**, you must click on this button to apply the changes.
- **Inner Radius**: Sets the inner radius of the parliament.
- **Outer Radius**: Sets the outer radius.
- **Start Angle**: Sets the starting angle which by default is -90° from the Y-axis.
- **End Angle**: Sets the ending angle, default 90°.

To create a parliament shape



1. Add a group to the scene tree, add material, rotate and set parameters (except Position and Scaling).
2. Add the Parliament plug-in to the container, and click the Initialize button.
   - By default the seating is created with 40 seats in 3 rows. The plug-in then by default creates 40 sub containers that are copies of the core-object.
   - All the newly created sub containers have the same properties as the source object, except from Position and Scaling (if set). Position is set by the plug-in to create the parliament structure.
3. Open the transformation editor for the first sub container and set Scaling (locked) to 0.2.
4. Still having the transformation editor open for the first sub container, select all other sub containers and drag and drop the Scaling property onto on of the selected sub containers.
   - This will set the same Scaling parameters for all selected sub containers.

Note: If the object you used for creating the seats contains many polygons, the total product of the parliament container can quickly be too heavy to render.

12.15.19 Rotations Order

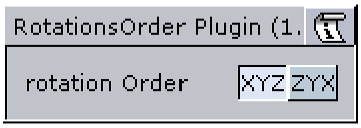
Rotations Order changes the rotation order of a container in Viz Artist to match the Softimage XSI rotation order. This plug-in is automatically added to some containers when Softimage XSI scenes are imported into Viz Artist.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\rotationsOrder.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters


RotationsOrder Plugin (1.

rotation Order: XYZ ZYX

- **Rotation Order**: Sets the rotation order to either **XYZ** or **ZYX**.

12.15.20 Slide Show

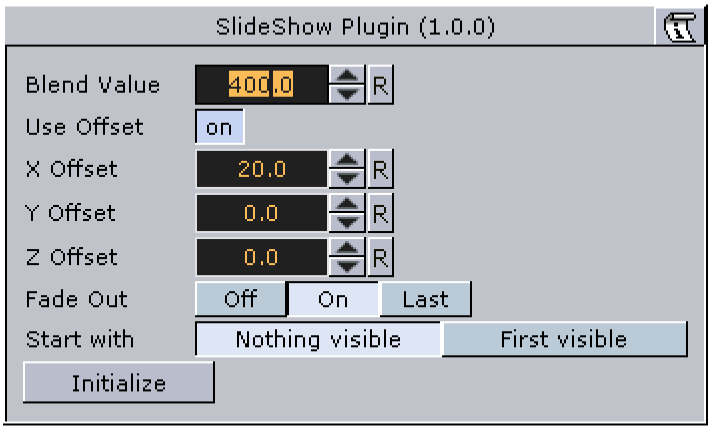
The Slide Show plug-in allows you to create a sequence where a group of containers fade in and out one by one in the same manner as a normal slide show.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\SlideShow.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters


SlideShow Plugin (1.0.0)

Blend Value: 400.0

Use Offset: on

X Offset: 20.0

Y Offset: 0.0

Z Offset: 0.0

Fade Out: Off / On / Last

Start with: Nothing visible / First visible

- **Blend Value**: Controls the slide show sequence. Animate this value to create the change from slide to slide.
- **Use Offset**: Enable this option if the slides have the same initial position and you want them to move along some of the axes as the slides are being shown.
- **X**, **Y** and **Z Offset**: Sets the offset values of each slide calculated form the one previously shown.
- **Fade Out**: May be set **Off**, **On** (every slide) or **Last** (last slide only)
- **Start with**: Sets what should be visible if slide show is not started.

To create a slide show

1. Add a group to the scene tree, and name it **slideshow**.

2. Add a number of sub containers that will be used to house the slide(s).
3. Add the Slide Show plug-in to the slideshow container.
4. Position the sub containers in advance using the transformation editors for each container, or open the Slide Show editor and offset them to create a moving sequence.
5. Animate the blend value to create the slide show sequence.
6. Click the Initialize button to initialize the plug-in.
   - Each of the sub containers automatically get an alpha icon.

12.15.21 System Time

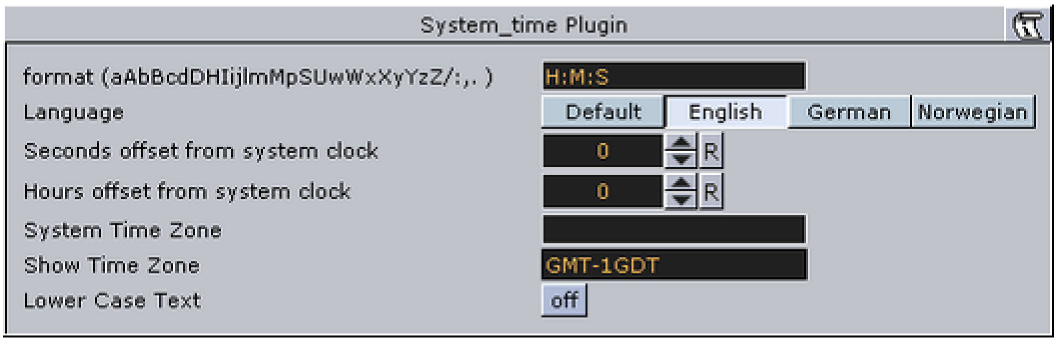
The system time plug-in allows you to create a variety of time and date settings based on the system time.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\System_time.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



System_time Plugin

format (aAbBcdDHIijlmMpSUwWxXyYzZ/:,.): H:M:S
Language: Default | English | German | Norwegian
Seconds offset from system clock: 0
Hours offset from system clock: 0
System Time Zone:
Show Time Zone: GMT-1GDT
Lower Case Text: off

- **format (aAbBcdDHIijlmMpSUwWxXyYzZ/:,.)**: Sets the format of the system time.
- **Language**: Sets the language format of the system time. Available options are Default (system specific), English, German and Norwegian.
- **Seconds offset from system clock**: Sets the number of seconds the system clock should be offset. Default is 0.
- **Hours offset from system clock**: Sets the number of hours the system clocl should be offset. Default is 0.
- **System time zone**: Sets the system time zone.
- **Show Time Zone**: Shows the result of the system time zone.
- **Lower Case Text**: Displays all text in lower case.

Time Zone: tzn[+|-]hh[:mm[:ss]][dzn]

- tzn: Time zone name, e.g. PST
- hh,mm,ss: Offsets from UTC
- dzn: Daylight-saving-time zone, e.g. PDT

Examples: EST5EDT, PST5PDT, GST-1GDT

See Also

- Analog Watch
- Clock Rotation

12.15.22 Temo

The Temo plugin gives the user the possibility to create an image animation sequence, using a method much like the one used for making a cartoon film. The basic input for the plug-in is a single image consisting of many tiled and equally sized squares set up in a matrix.

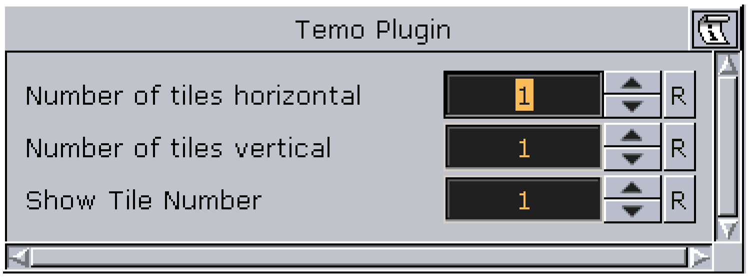
Each square of the image is made up to be a snapshot of an animation sequence, just like each picture frame is on a normal film. The image must be made up in advance with the aid of a image editing tool. In the property editor you tell the Temo plug-in how many tiled squares there are in the X- and the Y-axis. The plug-in is then able to show the tiles of the image one by one. By animating this, a “film like” sequence is created. The plug-in has only a small influence on the rendering performance.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\Temo.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



Temo Plugin

Number of tiles horizontal: 1

Number of tiles vertical: 1

Show Tile Number: 1

- **Number of tiles horizontal**: Here you define the number of tiles in the X-axis of the source image.
- **Number of tiles vertical**: Here you define the number of tiles in the Y-axis of the source image.

- **Show Tile Number**: Selects the tile to be shown. The range of tile numbers is the product of the two above parameters. If there are 3 in the horizontal and 4 in the vertical plane, the total is of course 12. The tile with the coordinates X1Y1 is shown first, then X2Y1 and so on. By animating the whole range of this value, a “film animation” can be created. The quality of the animation depends of course on the number of tiles and on the degree of transformation from one tile to the next throughout the whole sequence.

12.15.23 Text BG

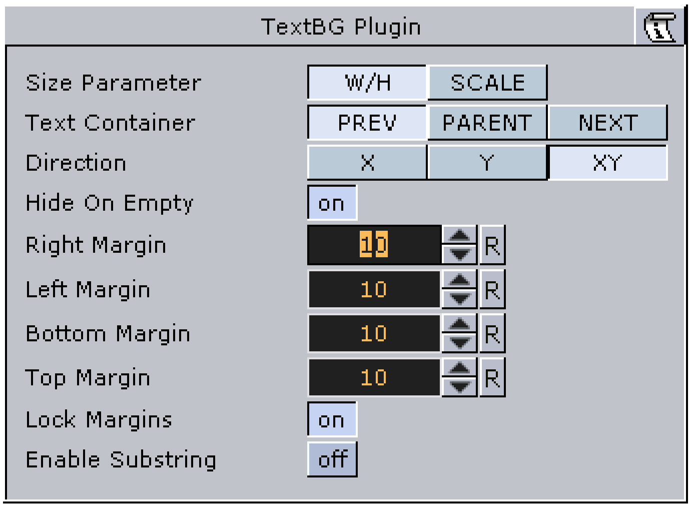
The Text BG plug-in works with Rectangle, Noggi, Fade Rectangle and Cube.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\TextBG.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To add background to text

Properties and Parameters



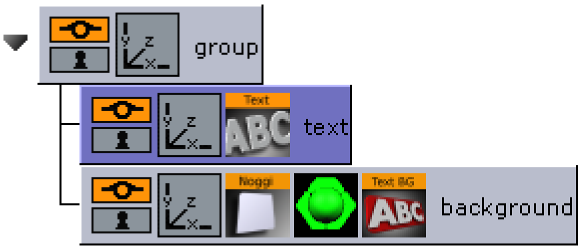
TextBG Plugin

Size Parameter	W/H SCALE
Text Container	PREV PARENT NEXT
Direction	X Y XY
Hide On Empty	on
Right Margin	10 R
Left Margin	10 R
Bottom Margin	10 R
Top Margin	10 R
Lock Margins	on
Enable Substring	off

- **Size Parameter**: Sets the size of the background to adjust according to width and height or scale.
- **Text Container**: Sets the Text BG plug-in to use the background in the PREV (previous), PARENT or NEXT container.
- **Direction**: Sets the direction in which the background should “grow” in order to adjust it self to the text. Available options are only the X and Y axis or the XY-axes in combination.
- **Hide On Empty**: Hides the background object if there are no text.
- **Right, Left, Bottom and Top Margin**: Sets the margin between the text and the background.

- **Lock Margins**: Sets the margin to be the same for all.
- **Enable Substring**: Enables the First and Last Character options.
  - **First Character**: Sets on which character the background should start animating.
  - **Last Character**: Sets on which character the background should stop animating.

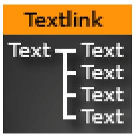
To add background to text



MORE
MORE CONTROL

1. Add a group container to the scene tree.
2. Add two sub containers to the group container and name them text and background.
3. Add a font to the text container.
4. Open the transformation editor for the text container and set Position Z to 1.0.
5. Add a geometry (e.g. Noggi), material and the Text BG plug-in to the background container.
6. Open the Text BG editor and make sure that the Text Container property is set to PREV (reads previous container).
7. Open the text editor and enter some text (e.g. MORE CONTROL), and see that the background automatically resizes the background geometry according to the text.

12.15.24 Text Link



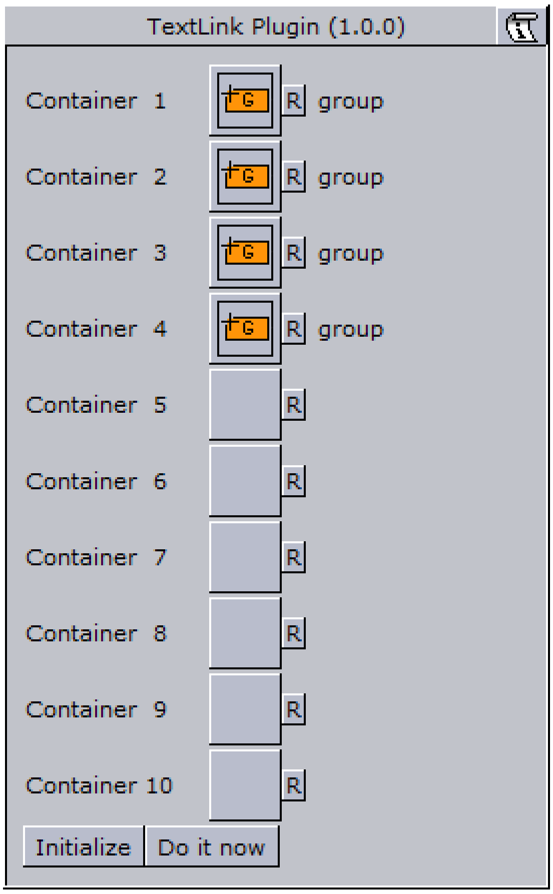
The Text Link plug-in copies the text from the source container to up to 10 inked containers. The text you enter in the source container will be repeated in all the linked containers. The linked containers must contain a font.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\TextLink.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To link text to multiple containers

Properties and Parameters



- **Container 1**: **10**: Drop zones for the containers you want to link.
- **Initialize**: Initializes the plug-in.
- **Do it now**: Preforms the operation immediately.

To link text to multiple containers

1. Add a group container to the scene tree.
2. Add the Text Link plug-in and a font to the group container.
3. Add a number of containers to the scene tree with a font on each.
4. Open the Text Link editor.
5. Drag and drop the containers with the fonts onto the Container 1-n drop zones.
6. Click **Initialize** to apply change.

See Also

- Image Link

12.15.25 Text Parameters

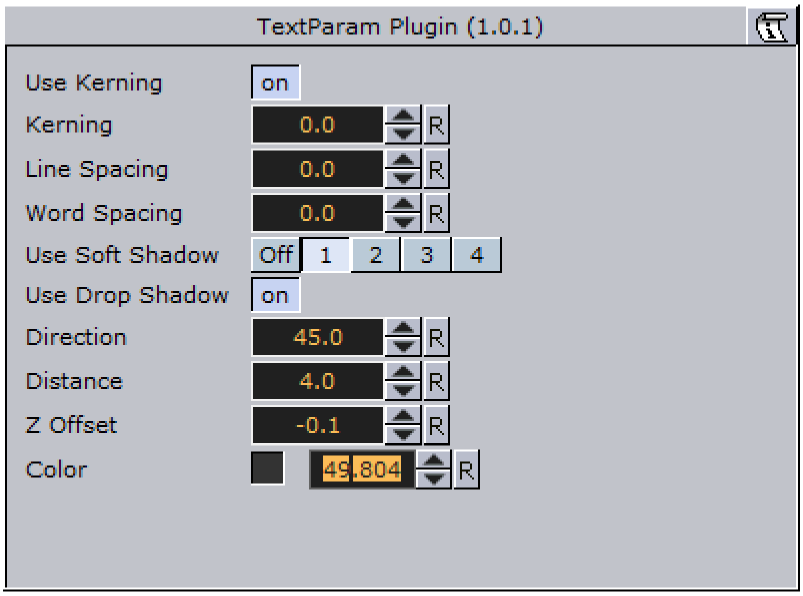
The Text Parameters plug-in gives the user the possibility to animate text parameters. The standard text parameters in the text editor can not be animated. The parameters can only be used for animation. The settings are not visible on the text until an animation update has been made.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\TextParam.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



TextParam Plugin (1.0.1)

Use Kerning: on
Kerning: 0.0
Line Spacing: 0.0
Word Spacing: 0.0
Use Soft Shadow: Off | 1 | 2 | 3 | 4
Use Drop Shadow: on
Direction: 45.0
Distance: 4.0
Z Offset: -0.1
Color: 49.804

- **Use Kerning**: Enables/disables use of Kerning.
- **Kerning**: Sets the kerning.
- **Line Spacing**: Sets the line spacing.
- **Word Spacing**: Sets the word spacing.
- **Use Soft Shadow**: Enables the use of soft shadow. Level 4 is the most blurred/ softest shadow. The drop shadow option must be enabled for the soft shadow to be visible.
- **Use Drop Shadow**: Enables/disables drop shadow.
- **Direction**: Sets the direction of the drop shadow in degrees.
- **Distance**: Sets the distance of the drop shadow.
- **Z Offset**: Sets the offset of the drop shadow in the Z-axis.
- **Color**: Sets the color of the shadow.
- **Alpha**: Sets the alpha value of the color.

12.16 Transformation

The following container plugins are housed in the Transformation folder:

- Justifier

12.16.1 Justifier

The Justifier plug-in enables you to animate the justification of an object using the object’s height, width or depth as its coordinate system rather than using the Viz coordinate system.

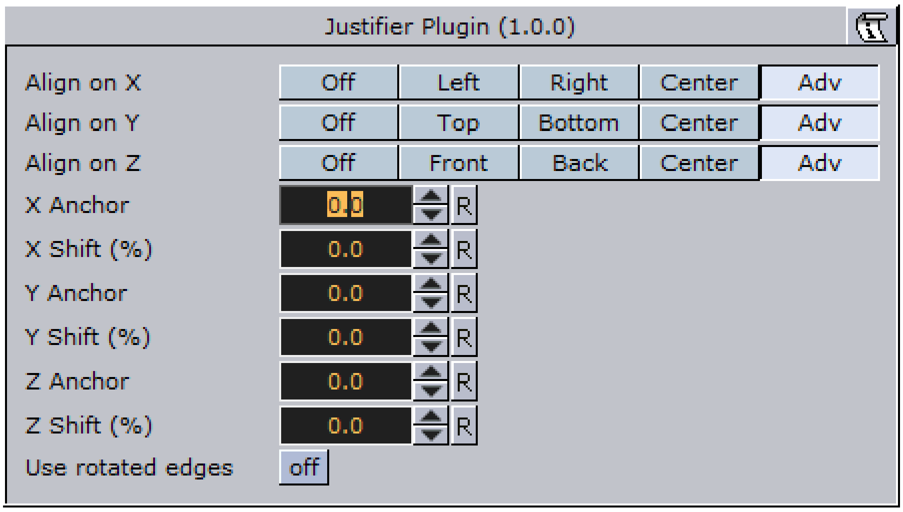
This plug-in acts on the local space of the container it is attached to. So a parent container can be used to rotate the container for use of arbitrary (world space) axes.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\Justifier.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



Justifier Plugin (1.0.0)

Align on X	Off	Left	Right	Center	Adv
Align on Y	Off	Top	Bottom	Center	Adv
Align on Z	Off	Front	Back	Center	Adv
X Anchor	0.0	R			
X Shift (%)	0.0	R			
Y Anchor	0.0	R			
Y Shift (%)	0.0	R			
Z Anchor	0.0	R			
Z Shift (%)	0.0	R			
Use rotated edges	off				

- **Align on X, Y and Z**: Aligns the position. Left, Right and Center enables the Anchor option(s). Adv (advanced) enables the Anchor and Shift option(s).
  - **X, Y and Z Anchor**: Sets the offset for the alignment.
  - **X, Y and Z Shift (%)**: Sets the shift of the position values to specify alignment between left/right, top/bottom and front/back (e.g. an Anchor at 50.0, Shift at -10%, gives an Anchor at 45.0.). Range is -100 to 100.

- **Use rotated edges**: If this is set to true and the container has been rotated, a new axis aligned bounding box will be used to computed the alignment. (I.e. a non rotated bounding box completely containing the rotated container).

To justify an object

1. Create a container that can only move up and down.
2. Add the Justify plugin on the container.
3. Set Align on X to Center.
4. Set Align on Z to Center.
    - Now the container can only be moved along its y axis.
5. Use the X Anchor and Z Anchor to move it along the respective axis.

12.17 Vizual Data Tools

The following container plugins are housed in the Vizual Data Tools folder:

- Data Fit
- Data Import
- Data Label
- Data Storage

See Also

- Vizual Data Tools Overview
- Area Chart
- Bar Chart
- Line Chart
- Pie Chart
- Scatter Chart
- Stock Chart
- Tutorial on www.vizrt.com under the Training section

12.17.1 Data Fit

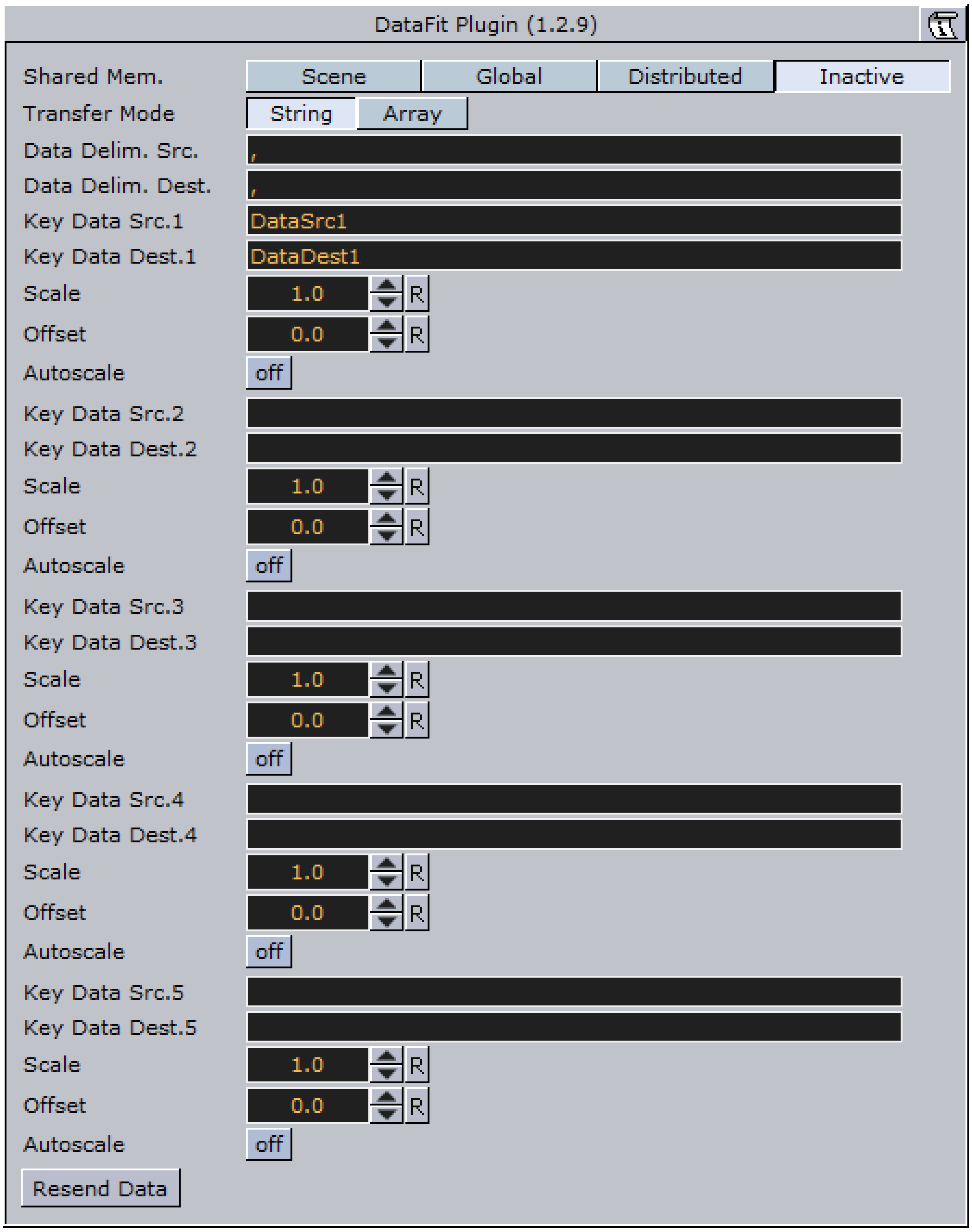
Data Fit listens for incoming data and modifies and redistributes it to another Shared Memory key.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\DataFit.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a scene using data fit

Properties and Parameters



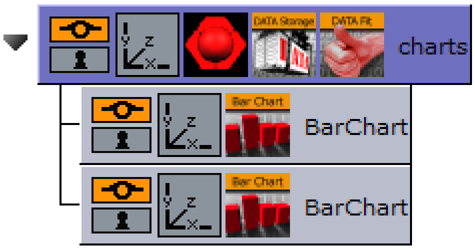
DataFit Plugin (1.2.9)

Shared Mem. Scene Global Distributed Inactive
Transfer Mode String Array
Data Delim. Src. ,
Data Delim. Dest. ,
Key Data Src.1 DataSrc1
Key Data Dest.1 DataDest1
Scale 1.0
Offset 0.0
Autoscale off
Key Data Src.2
Key Data Dest.2
Scale 1.0
Offset 0.0
Autoscale off
Key Data Src.3
Key Data Dest.3
Scale 1.0
Offset 0.0
Autoscale off
Key Data Src.4
Key Data Dest.4
Scale 1.0
Offset 0.0
Autoscale off
Key Data Src.5
Key Data Dest.5
Scale 1.0
Offset 0.0
Autoscale off
Resend Data

- **Shared Mem**: Changes between Scene-, Global- and Distributed-Shared Memory. Use Inactive memory to not forward any values via Shared memory.
- **Transfer Mode**: Sets string- or array-based data transfer.
- **Data Delim. Src.**: Defines the value separator sign(s) for all data sources.
- **Data Delim. Dest.**: Defines the value separator sign(s) for all data destinations.
- **Key Data Src.**1-5: Shared Mem. key name for input 1-5.
- **Key Data Dest.**1-5: Shared Mem. key name for output 1-5.
- **Scale**: Scales each incoming value by that factor.
- **Offset**: Adds an Offset to each incoming value.

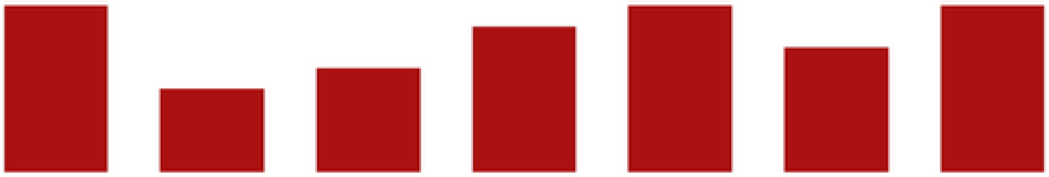
- **Autoscale**: Activates automatic scaling mode. Now all data is scaled to a certain range defined by the following 2 parameters:
  - **Start**: Autoscale starting range.
  - **Stop**: Autoscale stopping range.

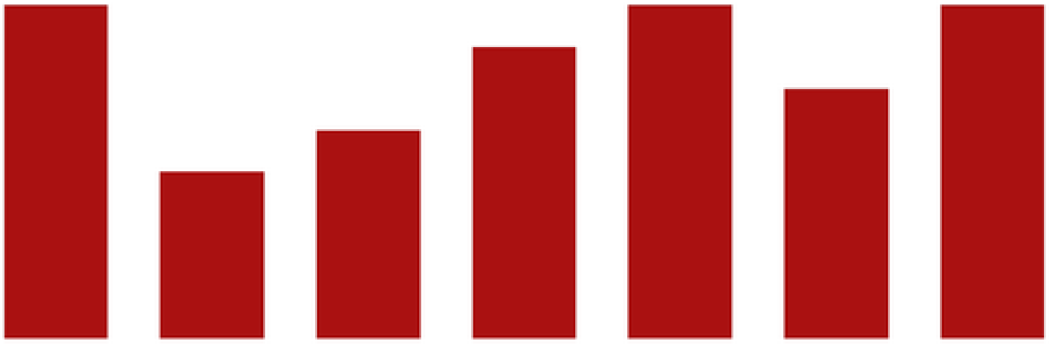
To create a scene using data fit



This procedure will feed two charts with data. One with the original and the other with the modified data from Data Fit.

1. Create a new container.
2. Drag and drop a Data Storage and a Data Fit plugin into this container.
3. Drag and drop two Bar Chart plugins to become children of this container.
4. Open the plugin editor of the first Bar Chart and set Shared Mem. to Scene and KeyData to “MyData”.
5. Do the same for the second Bar Chart but use “MyDataFit” as the key name.
6. Go to the DataFit plugin and set its Shared Mem. type also to Scene
7. Set Key Data Src.1 to “MyData” and Key Data Dest.1 to “MyDataFit” > set Scale to 2.0.
   - The input Shared Mem key is specified with Key Data Src. and the output with Key Data Dest.
8. Now edit Data Storage for the input values: set Shared Mem. to Scene, Key Data1 to MyData and type in some example values for Data1: 80,40,50,70,80,60,80.
   - As soon as you start typing you see both charts building up with the different values from Data Fit.





12.17.2 Data Import

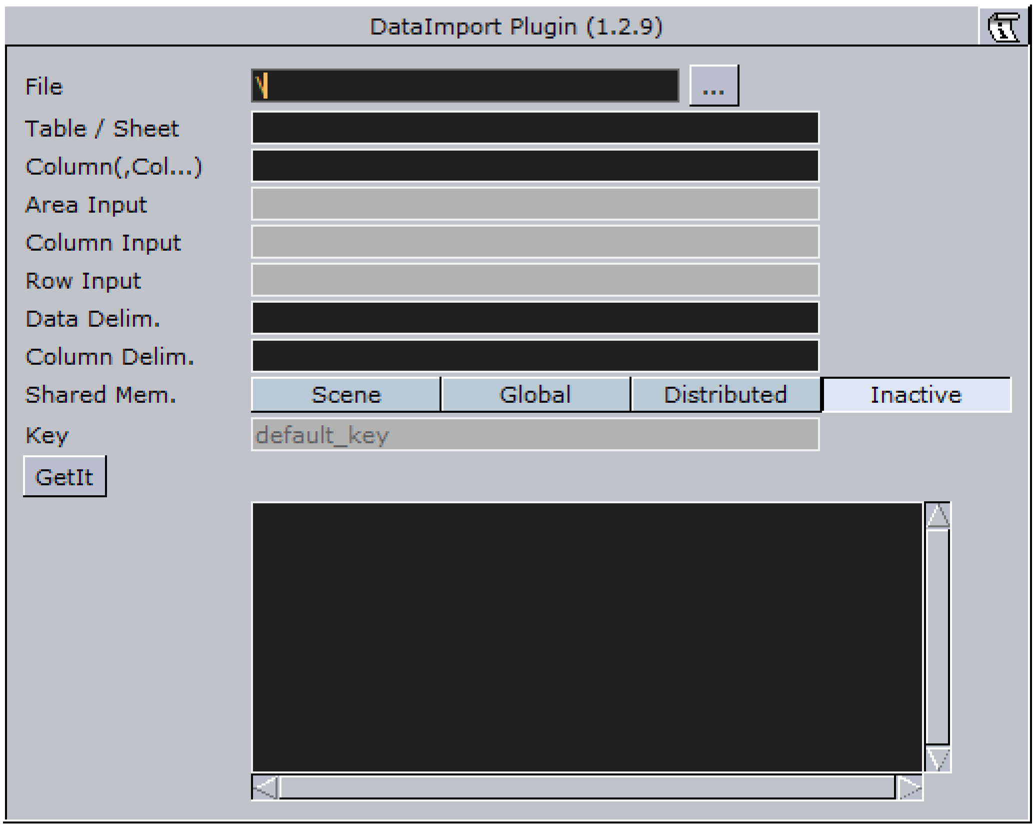
This plug-in enables Excel (.xls) and Access (.mdb) file import via ADO) and distributes it to a text field or a shared memory map.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\DataImport.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a scene using data import

Properties and Parameters



DataImport Plugin (1.2.9)

File

Table / Sheet

Column(,Col...)

Area Input

Column Input

Row Input

Data Delim.

Column Delim.

Shared Mem.: Scene | Global | Distributed | Inactive

Key: default_key

GetIt

- **File**: Choose your desired document.
- **Table / Sheet**: Define the Table's or Sheet's name.
- **Column(,Col...)**: Set one ore more column names to use (comma separated).
- **Area Input**: Excel also accepts areas.
- **Column Input**: Used column.
- **Row Input**: Used row.
- **Data Delim.**: Delimiter after each row.
- **Column Delim.**: Column separator sign(s).
- **Shared Mem.**: Set Scene-, Global- or Distributed-Shared Memory for data transport. Use Inactive memory to not forward any values via Shared memory.
- **Key**: Shared Memory key name.
- **GetIt**: Reads the document, sends the desired data and displays it in the text box.

To create a scene using data import

This procedure will import data from an Excel Sheet and feed a Line Chart with data from it through Shared Memory.

1. Open Microsoft Excel.
2. Enter ExcelDataY into the cell A1
3. Add some example values in the cells below (A2-A8: 80, 40, 50, 70, 80, 60, 80)

4. Rename this first sheet to MyTable (can be done with a double click on the sheet name at the bottom)
5. Save the Excel file as ExcelData.xls at C:\
6. Start Viz Artist and create a new scene.
7. Drag and drop a Line Chart and a Data Import plugin into the scene tree.
8. Setup a Scene Shared Mem. with “MyDataY” as key name in both plugins.
9. Set the parameter Data Delim. to # in each plugin.
10. In the LineChart set ChartWidth to 500.0. Activate DataY Fit, DataY Auto Scale and DataY Detect limits. Adjust DataY Stop to 200.0.
11. Activate Const. Thickness to give the chart a constant line width.
12. Open the DataImport plugin editor and choose the stored Excel file for the parameter File.
13. Enter MyTable for the Table / Sheet parameter and ExcelDataY for Column(,Col...)
14. Click the GetIt button and the chart appears.

12.17.3 Data Label

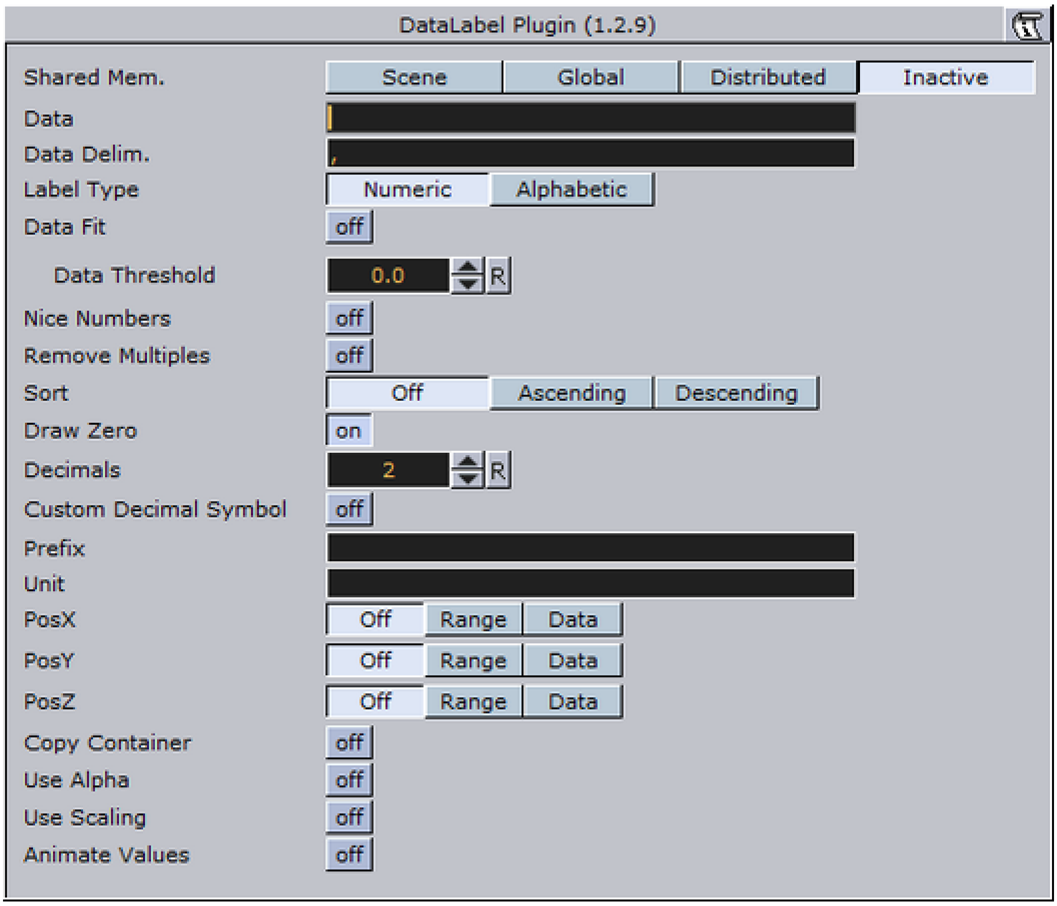
This plug-in enables graph labeling for all Financial Plug-ins. All containers beneath the current container get labeled with the values from your Shared Memory variables.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\DataLabel.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a scene using data label

Properties and Parameters



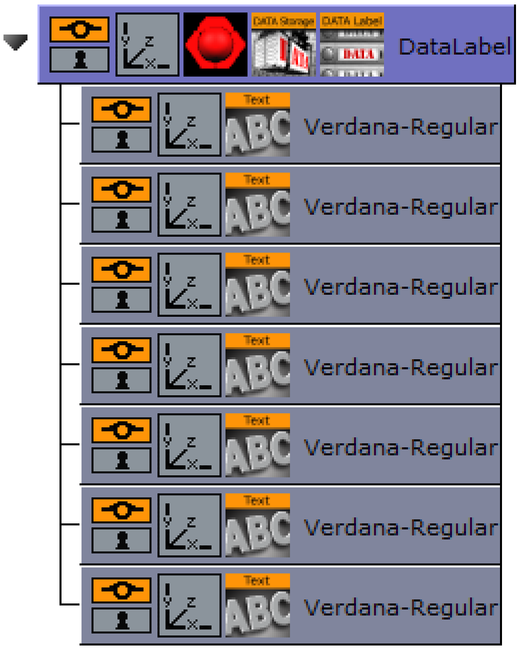
DataLabel Plugin (1.2.9)

Shared Mem. Scene | Global | Distributed | Inactive
Data
Data Delim. ,
Label Type Numeric | Alphabetic
Data Fit off
Data Threshold 0.0 R
Nice Numbers off
Remove Multiples off
Sort Off | Ascending | Descending
Draw Zero on
Decimals 2 R
Custom Decimal Symbol off
Prefix
Unit
PosX Off | Range | Data
PosY Off | Range | Data
PosZ Off | Range | Data
Copy Container off
Use Alpha off
Use Scaling off
Animate Values off

- **Shared Mem.**: Changes between Scene-, Global- and Distributed-Shared Memory. Use Inactive memory to not forward any values via Shared memory.
- **Key Data**: Shared Mem. key name
- **Data**: Input Parameter for data
- **Transfer Mode**: Sets string- or array-based data transfer.
- **Data Delim.**: Defines the value separator sign(s)
- **Label Type**: Sets the label type to numeric values (default) or alphabetic labels (for displaying weekdays, months, etc.).
- **Data Fit**: Enables data normalization.
  - **Data Threshold**: Adds a definable offset to the detected limit.
- **Data Scale**: Scales input by the selected factor
- **Data Offset**: Adds an offset to the incoming data
- **Data Auto Scale**: Enables automatic data normalization
- **Data Detect Limits**: Detects minimum and maximum of all values and scales them to adjusted Start and Stop.
- **Data Start**: Lower Auto Scale edge
- **Data Stop**: Upper Auto Scale edge
- **Nice Numbers**: Rounds to the best fitting next number.
- **Remove Multiples**: Removes multiple values.
- **Sort**: Sorts your data elements ascending or descending.
- **Draw Zero**: Show or hide 0 values.
- **Decimals**: Adjust decimal digits of all your labels.

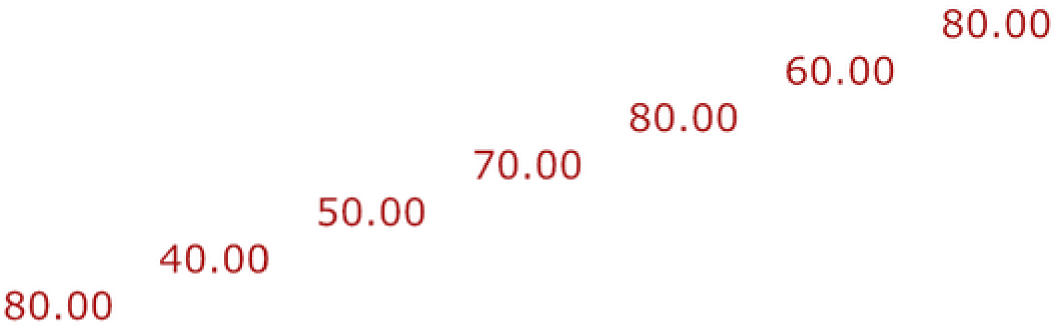
- **Custom Decimal Symbol**: Enables you to set a custom decimal symbol.
  - **Decimal Symbol**: Specifies the decimal symbol.
- **Prefix**: Adds a label prefix.
- **Unit**: Define a custom unit for your dat values.
- **PosX, Y, Z**: Positions container along the adjusted distance or uses numeric data value input to translate the label containers.
- **OffsetX, Y, Z**: Translates the label container on the current axis.
- **DistanceX, Y, Z**: Sets whole positioning distance for the containers on the current axis.
- **Copy Container**: If there are less containers than values, then new containers will be added.
- **Use Alpha**: Use input data for alpha scaling.
- **Use Scaling**: Use input data for alpha scaling.
- **Animate Values**: Let the current number run from 0 to the defined end value.
- **Relative Length**: If activated, each label will have it's own 100% (e.g. 7 labels equal 700%).
- **Const. Speed**: Sets the same animation duration for each label.
- **Total Length[%]**: Sets the accumulated value of all labels in percent.
- **Alpha max.**: Limits maximum alpha value.
- **Copy Container**: Limits maximum scaling value

To create a scene using data label



1. Add an empty container to the scene.
2. Start with an empty scene and drag and drop a DataLabel and a DataStorage plugin on a container in the scene-tree.
3. Add a Text child container
4. Go to the DataLabel plugin, set Shared Mem. to Scene and KeyData to “MyData”.
5. In DataStorage also set Shared Mem. to Scene, Key Data1 to “MyData” and Data1 to e.g.: 80,40,50,70,80,60,80

6. Go back to the DataLabel plugin and activate Copy Container to create multiple instances of the text containers automatically.
    - You see all labels located on the same position now. In the previous examples always the chart plugin was responsible for the label positioning. In this example we want to move that responsibility to the DataLabel plugin itself.
7. Switch PosX to Range, DistanceX to 500.0 and OffsetX to -250.0.
    - We want to position the labels over a certain X, Y or Z range in space.
8. Set PosY to Range, OffsetY to -75.0 and DistanceY to 150.0
9. You can also do a value dependent translation by setting PosX, Y, or Z to Data



10. Set Sort to Ascending to sort the labels.
    - You see that there are labels with the same values. We want to get rid of them and activate Remove Multiples for that. You can also try the Nice Numbers option now which searches for the next logarithmic style labels. But we continue with deactivated Nice Numbers parameter now.
11. Activate Animate Values and turn down Total Length[%] to 0.0. Create a keyframe, set Total Length[%] back to 100.0 and create another keyframe.
    - This will create an animation where the label values should count up to their final value.
    - Start the animation and see the labels counting and distributing over the specified XY range.

12.17.4 Data Storage

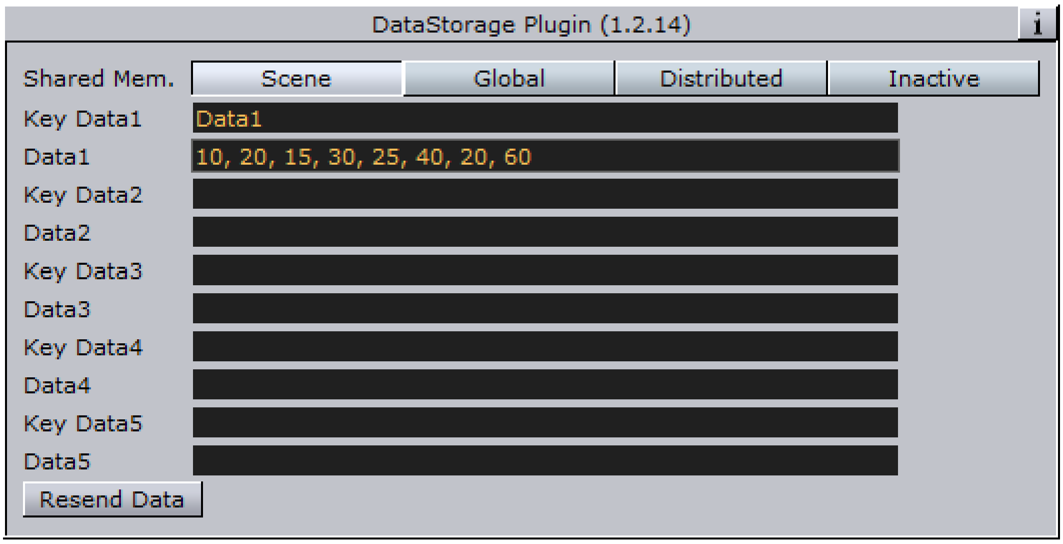
This plug-in stores data in the current scene and sends it to the other plugins on startup. This plugin helps artists during scene design if there is no data source available, and to hold data within e.g. the scene (see Data Sharing).

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\DataStorage.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To store and send data
- To enable data input from external sources

Properties and Parameters



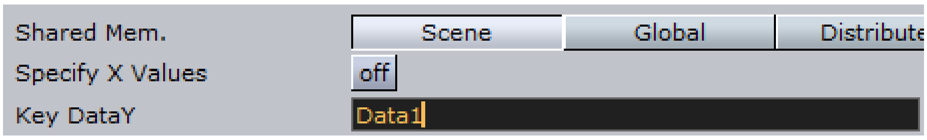
DataStorage Plugin (1.2.14)

Shared Mem.	Scene / Global / Distributed / Inactive
Key Data1	Data1
Data1	10, 20, 15, 30, 25, 40, 20, 60
Key Data2	
Data2	
Key Data3	
Data3	
Key Data4	
Data4	
Key Data5	
Data5	

- **Shared Mem.**: Changes between Scene-, Global- and Distributed-Shared Memory. Use Inactive memory to not forward any values via Shared memory.
- **Key Data 1-5**: Shared Mem. key name.
- **Data 1-5**: Contains (separated) data.
- **Resend Data**: Resends the actual values.

To store and send data

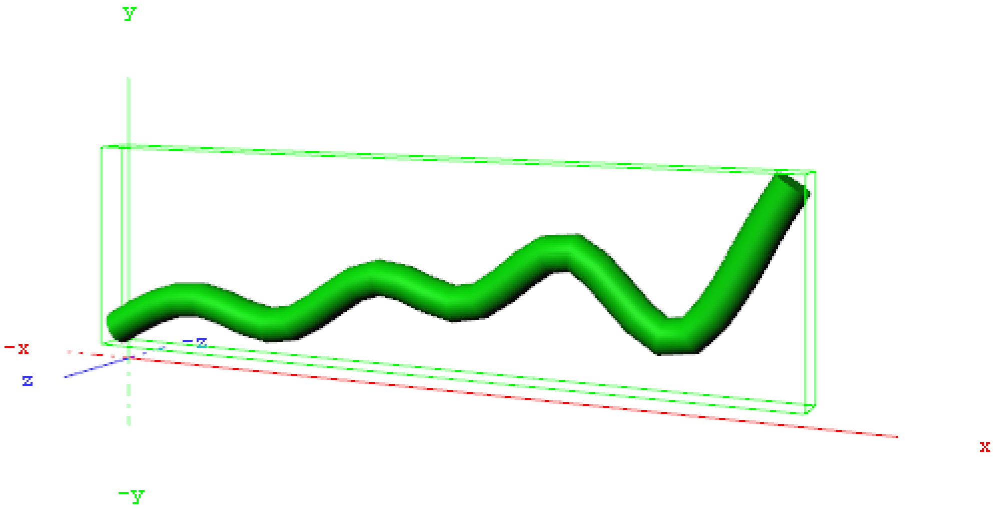
1. Create a chart scene, e.g. using the Line Chart plugin and set **Shared Mem.** to **Scene**
2. Add the Data Storage plugin to the scene and set **Shared Mem.** to **Scene**
3. Open the DataStorage editor and set the following parameters:
   - **Shared Mem.** to **Scene**
   - **Key Data1** to `Data1`
   - **Data1** to `10, 20, 15, 30, 25, 40, 20`



Shared Mem.	Scene / Global / Distribute
Specify X Values	off
Key DataY	Data1

4. Open the Chart plugin editor and set the **Key DataY** value (e.g. `Data1`) using the Key Data1 value you set for the DataStorage plugin

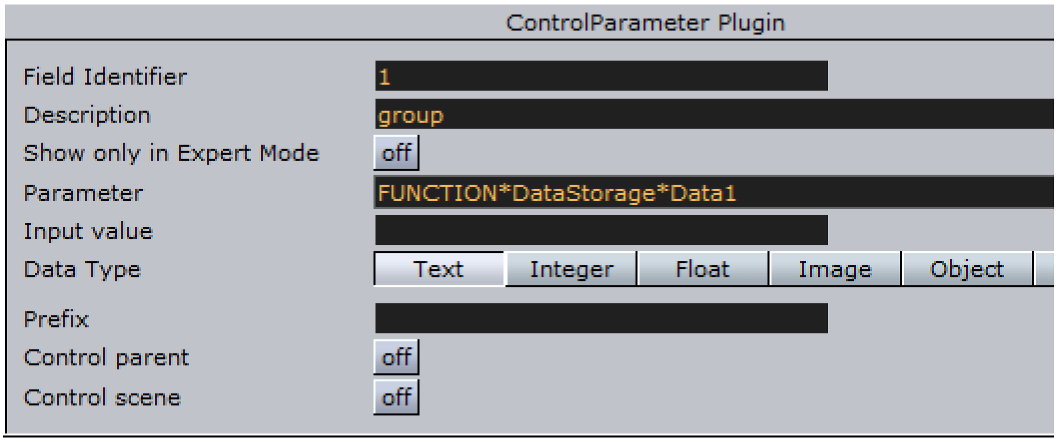
Note: The chart plugins have different property names (e.g. Data and Key Data). Also, some have X, Y or Z appended at the end, defining the axis.



5. Open the DataStorage editor and click the **Resend Data** button

Note: For more information on use of shared data and Data plugins, please read the Data Sharing section.

To enable data input from external sources


ControlParameter Plugin

Field Identifier	1
Description	group
Show only in Expert Mode	off
Parameter	FUNCTION*DataStorage*Data1
Input value	
Data Type	Text | Integer | Float | Image | Object
Prefix	
Control parent	off
Control scene	off

The following example builds on the previous example on how To store and send data.

6. Add the Control Parameter to the scene tree
   - The Control Object plugin is automatically added
7. Open the Control Parameter editor
8. Enter `FUNCTION*DataStorage*Data1` into the **Parameter** field
9. Set **Data Type** to **Text**
   - This will allow you to pass a string of numbers to the Control Parameter plugin that will be stored in the shared memory (see Data Sharing).
   - It will also allow your template designer or operator to further extend/ enhance the input methods through e.g. scripting.

See Also

- To create a scene using data fit
- To create a scene using data import
- To create a scene using data label
- To create a scene using area chart and data storage
- Data Sharing

13 Shader Plugins

Shader plugs are housed in the following folders:

- Default
- Effects
- Filter
- Material
- RTT Advanced Materials
- Texture

13.1 Default

This folder is empty and reserved for your custom shader plugins.

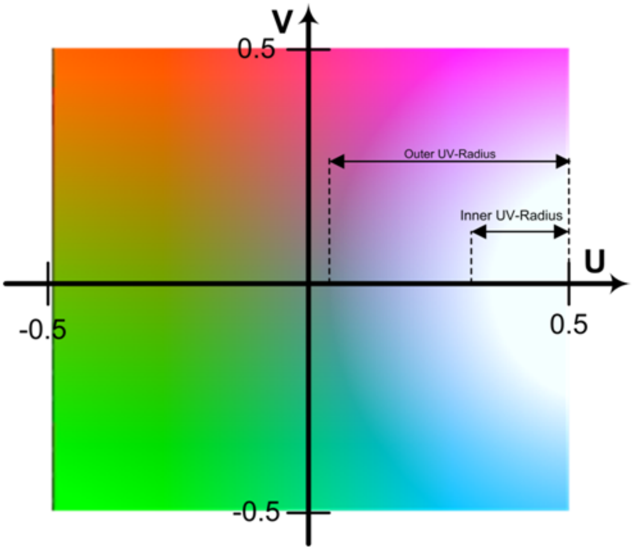
13.2 Effects

The following shader plugins are housed in the Effects folder:

- Chroma Keyer
- Fluid
- Frame Mask
- Image Mask
- Soft Mask
- Water Shader

13.2.1 Chroma Keyer

This shader clear regions in a video defined by a color. The colorspace of this shader is YUV which is defined as following:



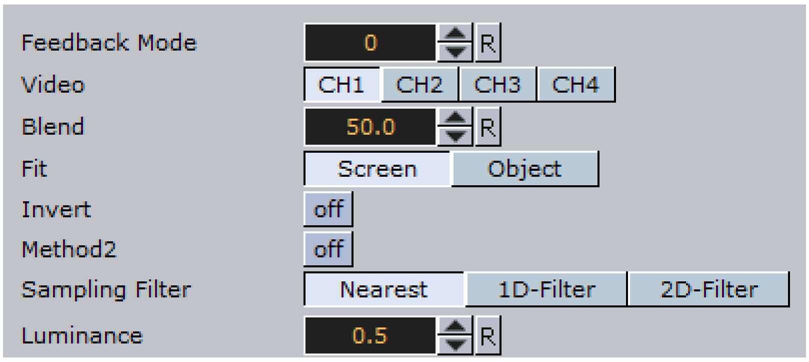
The luminance (Y) component determines the brightness of the color, while the U and V components determine the color itself (the chroma). Luminance (Y) ranges from 0.0 (dark) to 1.0 (bright). U and V ranges from -0.5 to 0.5.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ChromaKeyer.vip`.

This section contains information on the following topics:

- Properties and Parameters

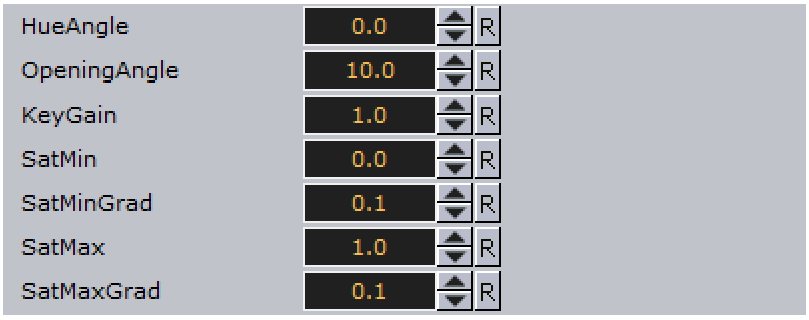
Properties and Parameters



Feedback Mode	0 R
Video	CH1 CH2 CH3 CH4
Blend	50.0 R
Fit	Screen Object
Invert	off
Method2	off
Sampling Filter	Nearest 1D-Filter 2D-Filter
Luminance	0.5 R

- **Video**: Sets the input video channel for keying. Be sure that for the set video channel the “Keying Mode” has been set to M-Zone (see Configuring the Keying Mode for Video).
- **Blend**: Sets the alpha for the key signal
- **Invert**: Flips the key signal.
- **Sampling Filter**
  - **Nearest**: Color information is taken from one pixel.
  - **1D-Filter**: Color information is taken after applying a 1D-linear filter.
  - **2D-Filter**: Color information is taken after applying a 2D-linear filter.
- **Luminance**: Sets the luminance.

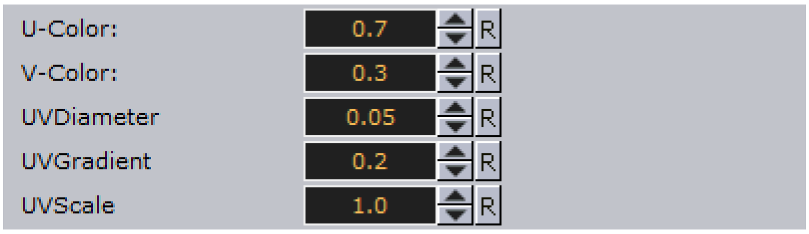
Basic parameters for first method:



HueAngle	0.0
OpeningAngle	10.0
KeyGain	1.0
SatMin	0.0
SatMinGrad	0.1
SatMax	1.0
SatMaxGrad	0.1

- **HueAngle**: Selects the color to key in YUV space.
- **OpeningAngle**: Tolerance for the color in YUV space.
- **KeyGain**: Gradient for borders to neighboring colors (1=soft, 10=sharp).
- **SatMin**: Lower saturation limit for keyed colors.
- **SatMinGrad**: Gradient for lower saturation limit (0=sharp, 1=soft).
- **SatMax**: Upper saturation limit for keyed colors.
- **SatMaxGrad**: Gradient for upper saturation limit (0=sharp, 1=soft).

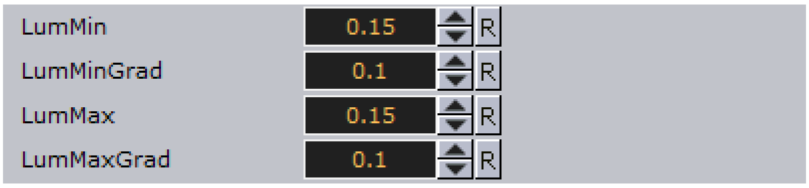
Basic parameters for second method:



U-Color:	0.7
V-Color:	0.3
UVDiameter	0.05
UVGradient	0.2
UVScale	1.0

- **U-Color**: U-value for color keying.
- **V-Color**: V-value for color keying.
- **UV-Diameter**: All colors between the reference UV and this diameter will be removed.
- **UVGradient**: Gradient for fade out at diameter border (0=sharp, 1=soft).

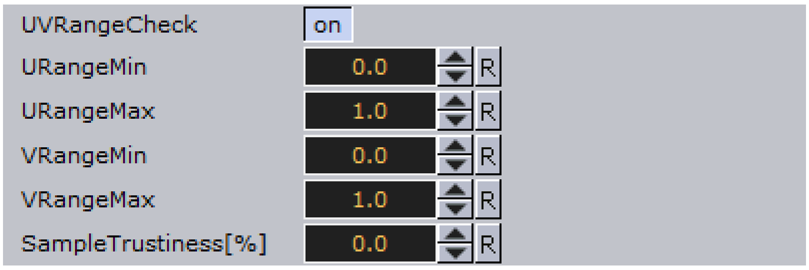
Removing keying from Highlights and Shadows:



LumMin	0.15
LumMinGrad	0.1
LumMax	0.15
LumMaxGrad	0.1

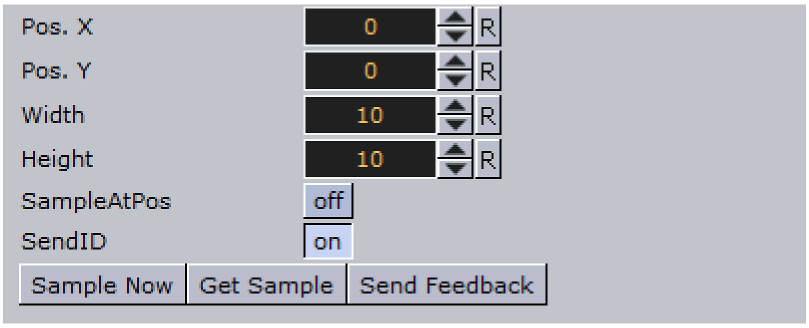
- **LumMin**: Lower luminance limit for keyed colors.
- **LumMinGrad**: Gradient for lower luminance limit (0=sharp, 1=soft).
- **LumMax**: Upper luminance limit for keyed colors.
- **LumMaxGrad**: Gradient for upper luminance limit (0=sharp, 1=soft).

Sampling reference point:



UVRangeCheck	on
URangeMin	0.0
URangeMax	1.0
VRangeMin	0.0
VRangeMax	1.0
SampleTrustiness[%]	0.0

- **UVRangeCheck**: This option can be used to get an information of the sampled color
- **URangeMin**: Defines the minimum UColor that should be accepted
- **URangeMax**: Defines the minimum UColor that should be accepted
- **VRangeMin**: Defines the minimum UColor that should be accepted
- **VRangeMax**: Defines the minimum UColor that should be accepted
- **SampleTrustiness**: Percentage of samples in the limited UV-Color space



Pos. X	0
Pos. Y	0
Width	10
Height	10
SampleAtPos	off
SendID	on

- **Pos. X**: X-coordinate for sampling.
- **Pos. Y**: Y-coordinate for sampling.

Note: The origin of the coordinate system is in the upper left corner.

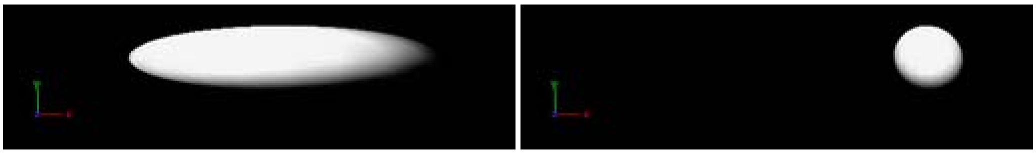
- **Width**: Sets number of pixels in x-direction.
- **Height**: Sets number of pixels in y-direction.
- **Sample at Pos**: If enabled Pos. X and Pos. Y will be replaced by XY-coordinates of the object center. If disabled the XY-coordinates will be set by values in the interface.
- **Get Sample**: Sample another time the reference point.

See Also

- Configuring the Keying Mode for Video

13.2.2 Fluid

With Fluid you can get a geometry mesh to be elastic during animation.

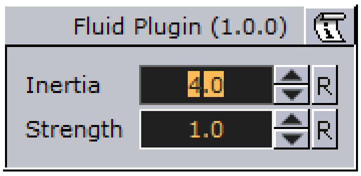


The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Fluid.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



Fluid Plugin (1.0.0)

Inertia 4.0 R

Strength 1.0 R

- **Inertia**: Sets the inertia of the motion.
- **Strength**: Sets the strength of the motion.

13.2.3 Frame Mask

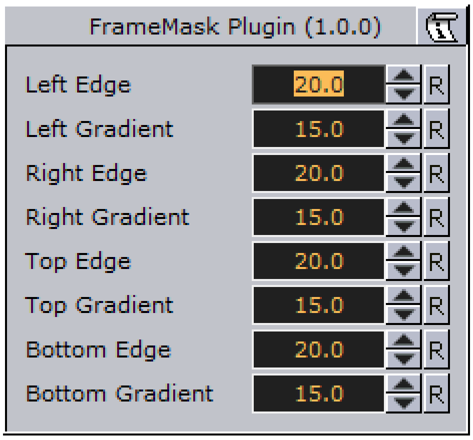
This plug-in adds a framed mask around the object.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\FrameMask.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



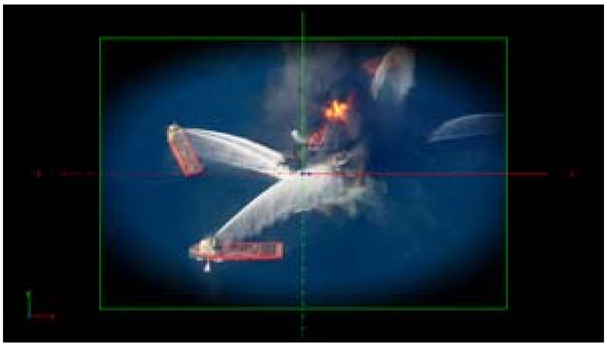
FrameMask Plugin (1.0.0)

Left Edge	20.0
Left Gradient	15.0
Right Edge	20.0
Right Gradient	15.0
Top Edge	20.0
Top Gradient	15.0
Bottom Edge	20.0
Bottom Gradient	15.0

- **Left, Right, Top and Bottom Edge**: Sets the edge of the frame.
- **Left, Right, Top and Bottom Gradient**: Sets the gradient from the edge of the frame to the edge of the image.

13.2.4 Image Mask

This plug-in applies a mask to an image using other images (e.g. an alpha image).

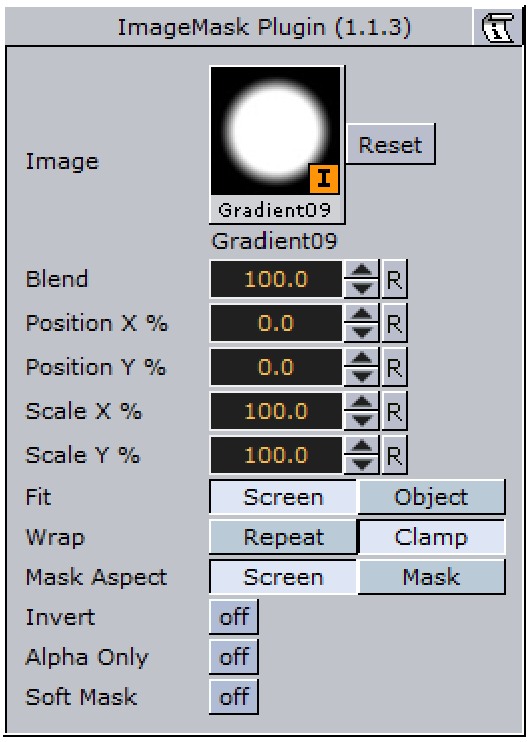


The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\ImageMask.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



ImageMask Plugin (1.1.3)

Image: Gradient09 (Reset)

Blend: 100.0

Position X %: 0.0

Position Y %: 0.0

Scale X %: 100.0

Scale Y %: 100.0

Fit: Screen | Object

Wrap: Repeat | Clamp

Mask Aspect: Screen | Mask

Invert: off

Alpha Only: off

Soft Mask: off

- **Image**: Placeholder for image.
- **Blend**: Blends the image with the object.
- **Position X, Y %**: Sets the position of the mask in percentage according to the screen or object.
- **Scale X, Y %**: Sets the size of the mask in percentage according to the screen or object.
- **Fit**: Fits the mask to the screen or object. Moving an object with fit to object enabled will also move the image used in the mask, and conversely.
- **Wrap**: Repeats or clamps the image used by the mask plug-in. Repeating the wrap and scaling down the image will cause the image to be applied multiple times on the screen or object, and conversely.
- **Invert**: Inverts the image.
- **Alpha only**: Enables the alpha channel for the image (e.g. an RGBA image).
- **Soft Mask**: Applies a soft mask.

13.2.5 Soft Mask

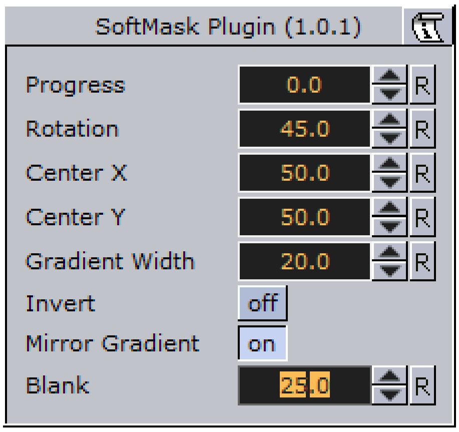
Applies a soft mask to an object (e.g. image or geometry).

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Softmask.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



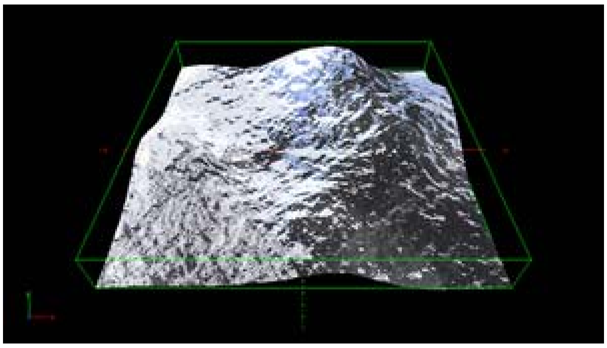
SoftMask Plugin (1.0.1)

Progress	0.0
Rotation	45.0
Center X	50.0
Center Y	50.0
Gradient Width	20.0
Invert	off
Mirror Gradient	on
Blank	25.0

- **Progress**: Moves the mask perpendicular to the mask direction. Default is 0.0 and is based on the configured center position.
- **Rotation**: Rotates the mask.
- **Center X and Y**: Defines the X and Y position of the center the mask will have.
- **Gradient Width**: Sets the size of the gradient.
- **Invert**: Inverts the mask.
- **Mirror Gradient**: Mirrors the mask on both axis'.
- **Blank**: Offsets the area that should not be affected by the mask.

13.2.6 Water Shader

This shader simulates a water surface.

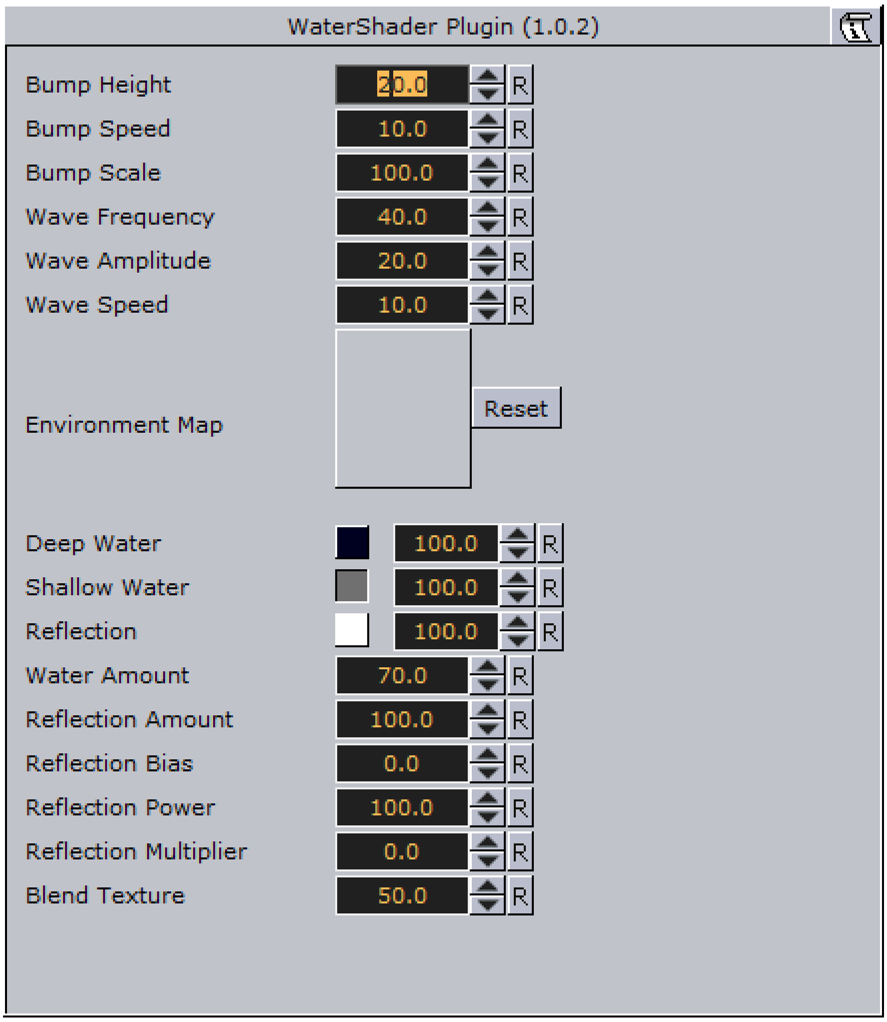


The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\WaterShader.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



WaterShader Plugin (1.0.2)

Bump Height 20.0
Bump Speed 10.0
Bump Scale 100.0
Wave Frequency 40.0
Wave Amplitude 20.0
Wave Speed 10.0
Environment Map
Deep Water 100.0
Shallow Water 100.0
Reflection 100.0
Water Amount 70.0
Reflection Amount 100.0
Reflection Bias 0.0
Reflection Power 100.0
Reflection Multiplier 0.0
Blend Texture 50.0

- **Bump Height**: Sets the height of the bump map.

- **Bump Speed**: Moves the bump map of the surface. Useful to simulate a current.
- **Bump Scale**: Scales the bump map.
- **Wave Frequency**: Sets the frequency of the wave.
- **Wave Amplitude**: Sets the amplitude of the wave.
- **Wave Speed**: Defines the big waves of the water. You need geometry with many vertices to get smooth waves.
- **Environment Map**: Defines the texture which is reflected in the water. If no texture is applied the shader will use a default texture. The size of the environment map should be kept low (512x512 max), because it can consume a lot of texture memory.
- **Deep Water**: Sets the color of deep water.
- **Shallow Water**: Sets the color of shallow water.
- **Reflection**: Sets the color of the reflections.

Note: The blending of deep and shallow water depends on the angle of the camera and how it looks at the water.

- **Water Amount:** Controls how the deep and shallow color affects the look of the water.
- **Reflection Amount, Bias, Power and Multiplier**: Controls the reflection
- **Blend Texture:** Allows you to blend the texture of the container into the water.

13.3 Filter

The following shader plugins are housed in the Filter folder:

- Blend Image
- Blur
- Color Balance
- Radial Blur
- Sepia
- Sharpen

13.3.1 Blend Image

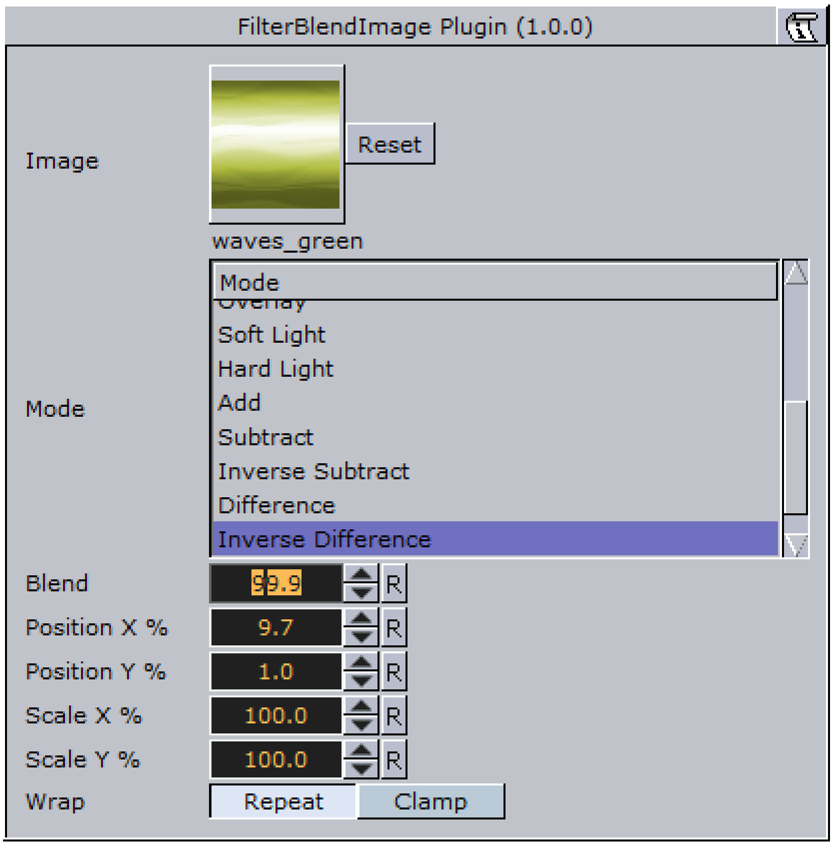
Use this plug-in to blend two images in different ways.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\FilterBlendImage.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



FilterBlendImage Plugin (1.0.0)

Image: waves_green

Mode: Inverse Difference

Blend: 99.9

Position X %: 9.7

Position Y %: 1.0

Scale X %: 100.0

Scale Y %: 100.0

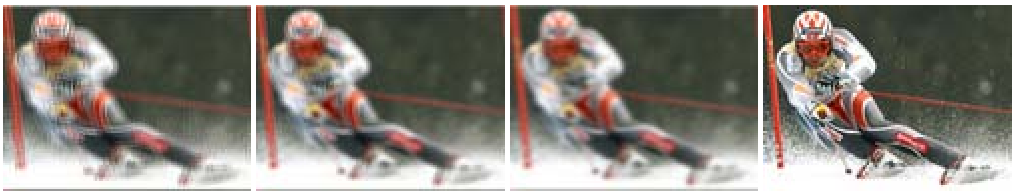
Wrap: Repeat

- **Image**: Image placeholder. Reset removes the image.
- **Mode**: Available blend modes are:
  - **Blend**: Ordinary picture blending.
  - **Darken**: Uses the base or blend color, whichever is darker, as the destination color.
  - **Lighten**: Uses the base or blend color, whichever is lighter, as the destination color.

    - **Multiply**: Multiplies the base color with the blend color. The result color will be always a darker color.
    - **Screen**: Multiplies the inverse of the base and blend color. The result color will be always a lighter color.
    - **Color Burn**: Darkens the base color to reflect the blend color by increasing the contrast of the base and blend color.
    - **Color Dodge**: Brightens the base color to reflect the blend color by decreasing the contrast of the base and blend color.
    - **Overlay**: Screens or multiplies the colors depending of the base color.
    - **Soft Light**: Darkens or lightens the colors, depending on the blend color.
    - **Hard Light**: Multiplies or screens the colors, depending on the blend color.
    - **Add**: Brightens the base color to reflect the blend color by increasing the brightness.
    - **Subtract**: Subtracts the base and blend colors.
    - **Inverse Subtract**: Same as subtract but inverses the base and blend colors.
    - **Difference**: Subtracts either the base color from the blend color or the blend color from the base color depending on which color has the greater brightness value.
    - **Inverse Difference**: Same as difference but inverses the base and blend color.
    - **Exclusion**: Similar to the “Difference” mode but lower in contrast.
- **Blend**: Blends the image with the object.
- **Position X and Y %**: Positions the image on the object.
- **Scale X and Y %**: Scales the image.
- **Wrap**: Wraps the object with the image. Available options are:
    - **Repeat**: Repeats the image.
    - **Clamp**: Clamps the image (no repeat).

13.3.2 Blur

Use this plug-in to blur an object with an image texture and/or a material on it. Values can be animated. You also have a blur plug-in directly on the image editor but that will only work on one image, using the shader all images within a group can be blurred.



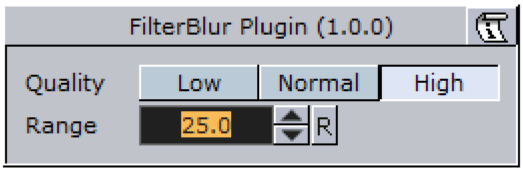
The samples above depict the same image with blur quality set to Low, Normal and High with a range of 100.0. The last is the original (reference) with no blur filter.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\FilterBlur.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



FilterBlur Plugin (1.0.0)

Quality: Low | Normal | High

Range: 25.0 R

- **Quality**: Sets the quality of the blur filter. Low will be more pixelated, whereas high will be smooth. Available options are Low, Normal and High.
- **Range**: Sets the range of the blur. Range is from 0.0 to 100.0.

See Also

- Radial Blur

13.3.3 Color Balance

With Color balance you can decide brightness, contrast, saturation and hue.

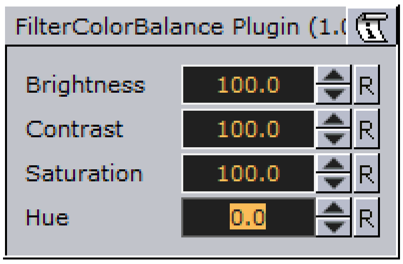
Tip: Most shaders also work on video clips.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\FilterColorBalance.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



FilterColorBalance Plugin (1.(

Brightness 100.0 R

Contrast 100.0 R

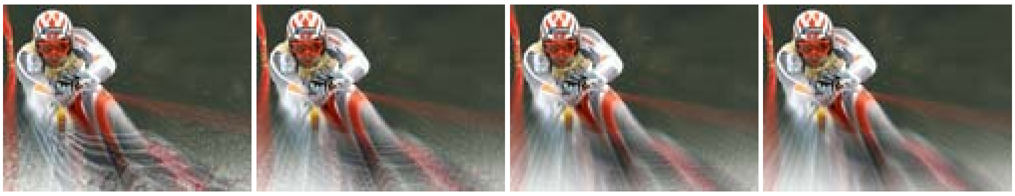
Saturation 100.0 R

Hue 0.0 R

- **Brightness**: Sets the brightness of the object. Range is 0.0 to 1000.0.
- **Contrast**: Sets the contrast of the object. Range is -1000.0 to 1000.0.
- **Saturation**: Sets the saturation of the object. Range is -1000.0 to 1000.0.
- **Hue**: Sets the hue of the object. Range is 0.0 to 360.0.

13.3.4 Radial Blur

Use this plug-in if you want to blur the object in different ways.



The samples above depict the same image with radial blur quality set to Low, Normal, High and Very High with Center X and Y set to -25.0 and 25.0, Inner Range and Scale set to 25.0 and 50.0, respectively.

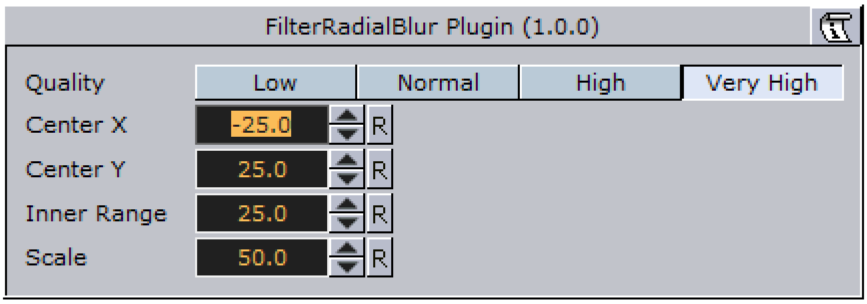
Note: Radial blur does not clamp the object.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\FilterRadialBlur.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters


FilterRadialBlur Plugin (1.0.0)

Quality: Low | Normal | High | [✓ Very High]

Center X: -25.0

Center Y: 25.0

Inner Range: 25.0

Scale: 50.0

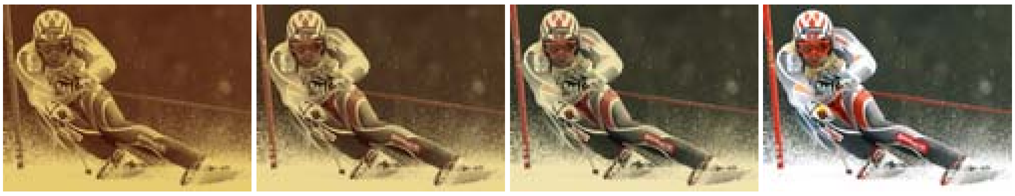
- **Quality**: Sets the quality of the blur filter. Low will be more pixelated, whereas very high will be smooth. Available options are Low, Normal, High and Very High.
- **Center X and Y**: Sets the position of the object's radial blur.
- **Inner Range**: Sets the inner range of the blur.
- **Scale**: Sets the scale of the object's blur.

See Also

- Blur

13.3.5 Sepia

Sepia can be used to generate a sepia effect using two colors, and to adjust the desaturation and tone of the color for example in order to blend the image with the color scheme of the overall scene. Sepia is similar to what is known as duo tone in most photo editing suites.



The samples above depict the same image with tone levels set to 100.0, 50.0 and 0.0. All other values are set to default. The last image is the original (reference) without the sepia plug-in.

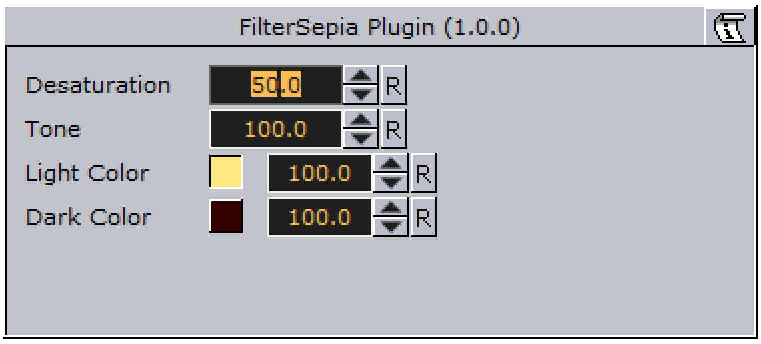
Note: The color range is from 0.0 - 100.0 divided by 255 (0.392, 0.784 etc.).

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\FilterSepia.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters


FilterSepia Plugin (1.0.0)

Desaturation 50.0 R

Tone 100.0 R

Light Color 100.0 R

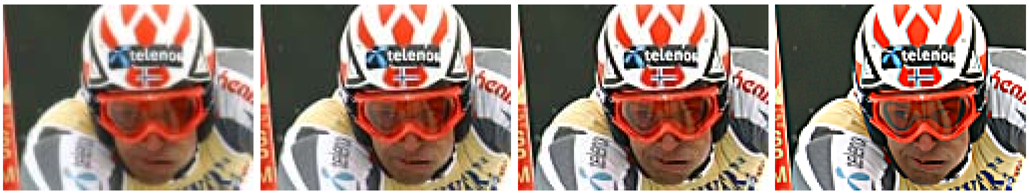
Dark Color 100.0 R

- **Desaturation**: Sets the saturation level. Range is 0.000 (color) to 100.000 (black & white).
- **Tone**: Sets the color tone. Range is 0.000 to 100.000.
- **Light Color**: Specifies the light color and the strength of it. Range is 0.000 to 100.000.
- **Dark Color**: Specifies the dark color and the strength of it. Range is 0.000 to 100.000.

13.3.6 Sharpen

Sharpen

Use this plug-in for sharpening the given images on objects.



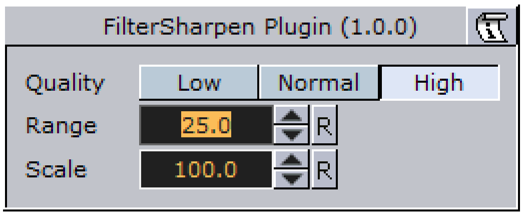
The samples above depict the same image with Quality set to High, and Scale values at 0.0, 100.0, 200.0 and 300.0, respectively. Range is set to 25.0 (default). The first image is the original (reference) with Scale set to 0.0.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\FilterSharpen.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



FilterSharpen Plugin (1.0.0)

Quality: Low | Normal | High

Range: 25.0

Scale: 100.0

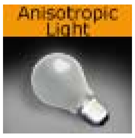
- **Quality**: Sets the quality of the sharpen filter. Low will be more pixelated, whereas high will be smooth. Available options are Low, Normal and High.
- **Range**: Increase this value to prepare more pixels to be included in the operation.
- **Scale**: Increase this value to sharpen.

13.4 Material

The following shader plugins are housed in the Material folder:

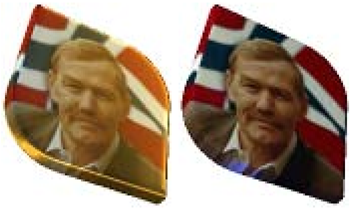
- Anisotropic Light
- Bump Map
- Cartoon
- Gooch
- Normal Map
- Simple Bump Map

13.4.1 Anisotropic Light



This shader uses a lookup texture to compute the distribution of the specular component. All the shader properties are defined by the material of the container. Only Light number 1 is used and the light is always treated as a local light source.

Figure 174: With anisotropic light and regular material



The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\AnisotropicLight.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters

AnisotropicLight Plugin (1.0.0)

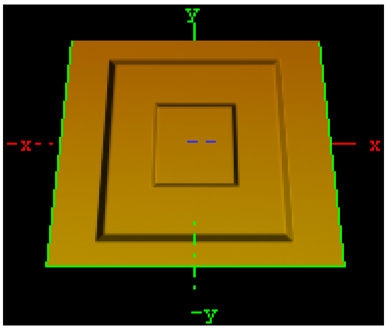
This plug-in does not have any properties or parameters.

13.4.2 Bump Map

With this plug-in you have the opportunity to simulate complex surface structures. Instead of using many triangles to render a 3D object with many structures on its surface, you can use bump mapping to get a similar result.

Use plug-ins for Adobe PhotoShop, Gimp, built-in functions of your modelling package or NVIDIA's tool called Melody.

Note: Make sure the geometry supports tangent space vectors. Otherwise use the Simple Bump Map shader.

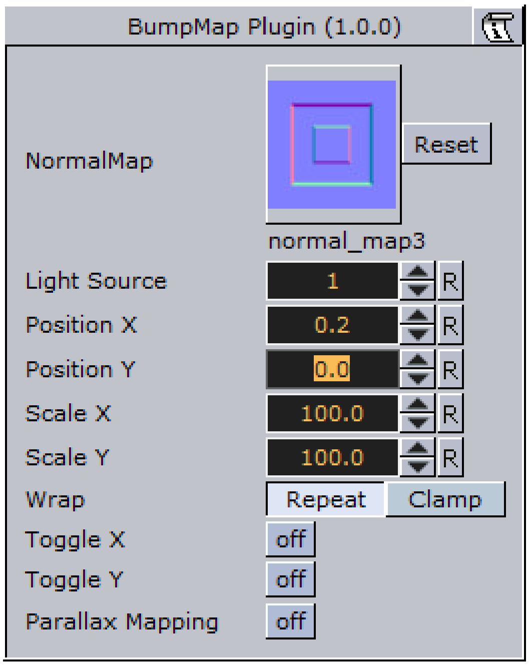


The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\BumpMap.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



BumpMap Plugin (1.0.0)

NormalMap: normal_map3

Light Source: 1

Position X: 0.2

Position Y: 0.0

Scale X: 100.0

Scale Y: 100.0

Wrap: Repeat | Clamp

Toggle X: off

Toggle Y: off

Parallax Mapping: off

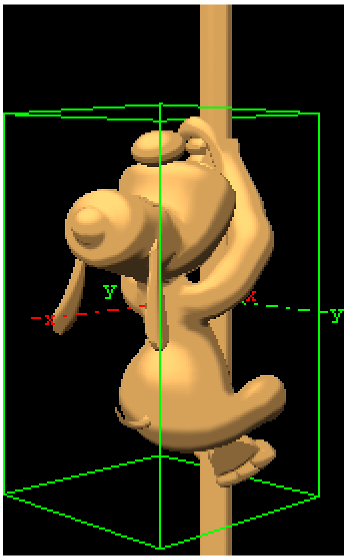
- **NormalMap**: Move an normal map via drag and drop onto this field. Normal maps are images which store a direction of normals directly in the RGB data.
- **HeightMap**: Move an height map via drag and drop onto this field. The Height Map file is only visible and used when parallax mapping is enabled.
- **Light Source**: Choose the light source in the range 1-8. Please consider that only one light source is possible. By default the first light source is used.
- **Position X and Y**: The position of the normal map in X-direction and Y-direction.
- **Scale X and Y**: Here you can scale the normal map, respectively change the width and height size of the image.
- **Wrap**: Repeats or clamps the bump map for the image.
- **Toggle X and Y**: Changes the direction of the normals stored in the normal map image.
- **Scale Height**: Changes the intensity of the bump map effect. Scale Height is visible when parallax mapping is enabled.
- **Parallax Mapping**: With this additional map the texture will have more apparent depth and the object is more realism.

See Also

- Simple Bump Map

13.4.3 Cartoon

This shader renders objects in a cartoon like manner. Shading is done in steps and you can define number of steps, the brightest color and the darkest color. Edge width allows you to blend between the steps. Material and texture of the container are ignored. Only Light number 1 is used and the light is always treated as local light source.

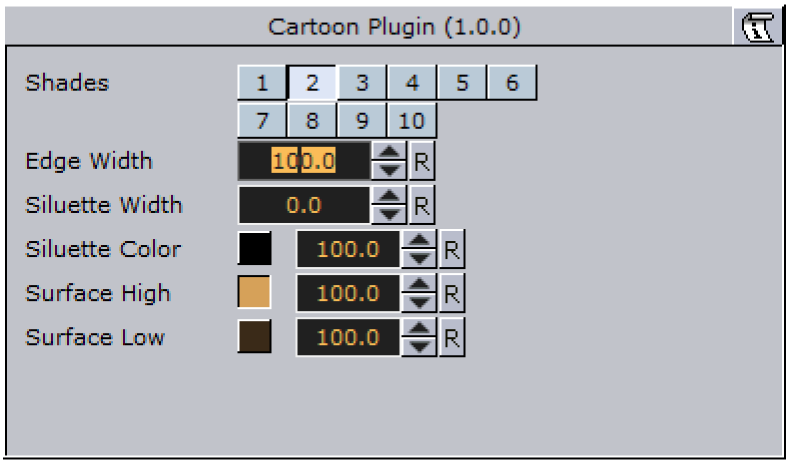


The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Cartoon.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



Cartoon Plugin (1.0.0)

Shades: 1 2 3 4 5 6 7 8 9 10

Edge Width: 100.0 R

Siluette Width: 0.0 R

Siluette Color: ■ 100.0 R

Surface High: ■ 100.0 R

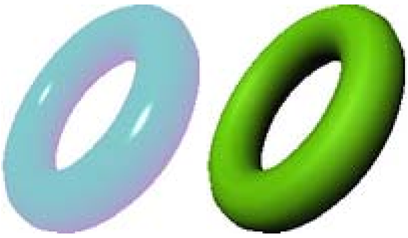
Surface Low: ■ 100.0 R

- **Shades**: Number of shade levels.
- **Edge Width**: Create a smooth effect. The transitions are more hard or more soft.
- **Silhouette Width**: The width of the contours around the rendered objects.
- **Silhouette Color**: The color of the contours around the rendered objects. Additionally you can select the alpha value if you want transparent areas. 0 (Invisible) – (Visible) 100%.
- **Surface High**: The color in the broad range. Additionally you can select the alpha value if you want transparent areas. 0 (Invisible) – (Visible) 100%.
- **Surface Low**: The color in the dark range. Additionally you can select the alpha value if you want transparent areas. 0 (Invisible) – (Visible) 100%.

13.4.4 Gooch

Gooch is a per pixel light with a reflective high light. Material of the container is ignored, only the shader parameters are used to define the surface color. Texture mapping works as usual, but the texture can be blended with the surface color using the Texture Alpha parameter. Only Light number 1 is used and the light is always treated as local light source.

Figure 175: With gooch and regular material

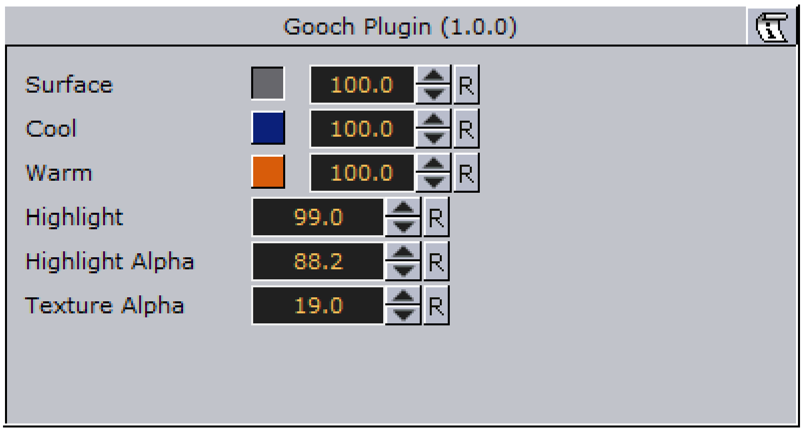


The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Gooch.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



Gooch Plugin (1.0.0)

Surface	■	100.0
Cool	■	100.0
Warm	■	100.0
Highlight		99.0
Highlight Alpha		88.2
Texture Alpha		19.0

- **Surface**: Describes the intensity of color like the ambient color of a material.
- **Cool**: Defines the cool color. Both colors cool and warm will be merged. The result is a color gradient.
- **Warm**: Defines the warm color.
- **Highlight**: The diameter of the highlight.
- **Highlight Alpha**: The alpha or transparent value of the highlight.
- **Texture Alpha**: Describes the visible value of the texture image in percent. 0 (Invisible) - (Visible) 100%.

13.4.5 Normal Map

The Normal Map Shader adds a color to each face according to the direction of the face’s normal. RGB values change between 100-200 while the normal XYZ rotates.

If the Normal is pointing along the Y-axis, this will give the RGB value (100,200,100). A normal along the X-axis give the RGB value (200,100,100). 45° between x and Y-axis will give RGB value (150,150,100).

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Normal.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters

Normal Plugin (1.0.0)

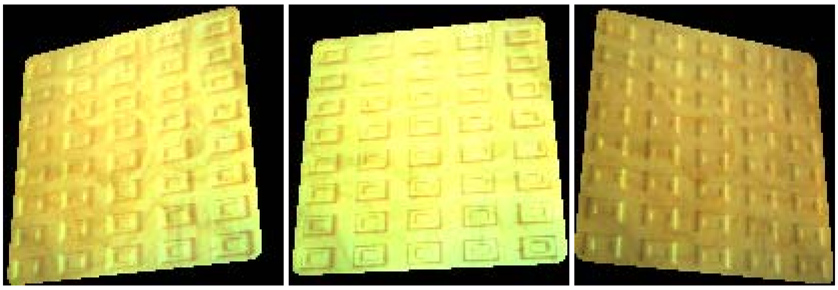
This plug-in does not have any properties or parameters.

13.4.6 Simple Bump Map

With this plug-in you have the opportunity to simulate complex surface structures. Instead of many triangles to render a 3D object with many structures on its surface you can use bump mapping to get a similar result.

You can only use a gray scaled image as normal map. If you want a more accurate result and better effects please use the Bump Map plug-in instead which is using an RGB normal map image.

To generate a Simple Bump Map you can use the plug-ins for PhotoShop, Gimp, the built-in functions of your modelling package or Nvidia's tool Melody. The Height Map (used for parallax mapping) should be a grayscaled image (white -> highest, black -> lowest bumps).



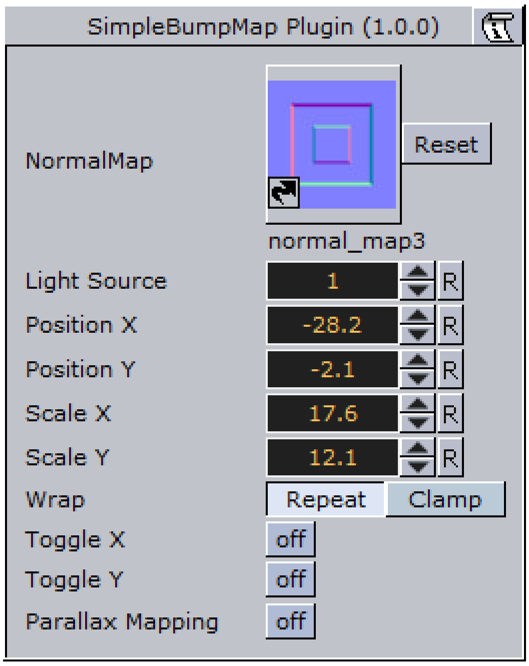
Note: All calculations are done in object space.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\SimpleBumpMap.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



SimpleBumpMap Plugin (1.0.0)

NormalMap: normal_map3 (Reset)

Light Source: 1

Position X: -28.2

Position Y: -2.1

Scale X: 17.6

Scale Y: 12.1

Wrap: Repeat | Clamp

Toggle X: off

Toggle Y: off

Parallax Mapping: off

- **NormalMap**: Move an normal map via drag and drop onto this field. Normal maps are images which store a direction of normals directly in the RGB data.
- **HeightMap**: Move an height map via drag and drop onto this field. The Height Map file is only visible and used when parallax mapping is enabled.
- **Light Source**: Choose the light source in the range 1-8. Please consider that only one light source is possible. By default the first light source is used.
- **Position X and Y**: The position of the normal map in X-direction and Y-direction.
- **Scale X and Y**: Here you can scale the normal map, respectively change the width and height size of the image.
- **Wrap**: Repeats or clamps the bump map for the image.
- **Toggle X and Y**: Changes the direction of the normals stored in the normal map image.
- **Scale Height**: Changes the intensity of the bump map effect. Scale Height is visible when parallax mapping is enabled.
- **Parallax Mapping**: With this additional map the texture will have more apparent depth and the object is more realism.

See Also

- Bump Map

13.5 RTT Advanced Materials

The following RTT shader plugins are housed in the *RTTAdvancedMaterials* folder:

- Anisotropic Light Shader
- Brushed Metal Shader
- Bump Shader

- Bump Optimized Shader
- Fabric Shader
- Glass Shader
- Gooch Shader
- Lacquered Surfaces Shader
- Metal Reflection Shader
- Microstructure Shader
- Monitor Shader
- Velvet Shader

RTT Advanced Materials plug-ins allow you to create a number of high-quality materials using state-of-the art shader technology. Using the plug-ins is very easy. Simply drag the shader plug-in of your choice onto selected geometry, assign a Viz material and set up the additional parameters as displayed by the shader in its own editor. Some plug-ins remain inactive until you have applied all necessary textures, depending on the shader plug-in. This may include a basic Viz texture on your container and/or additional textures to be dropped in the plug-in interface. You can modify all parameters of the applied Viz material on your object, such as Ambient Color, Diffuse, Specular and Emission, Shininess and Alpha. All these parameters will also affect the shader.

All RTT Advanced Materials plug-ins (except the Bump Optimized Shader) allow manipulating the texture mapping of each texture map individually. This means you can use a different position, rotation and scaling for each texture. As for the mapping of additional textures inside the Shader, the same mapping method as being assigned to the texture on your Viz container is used. It is also possible to change the alignment of the environment textures to achieve proper reflections on the respective object surface.

If you do not want to use individual texture transformations, you may switch the option off. Each additional texture (except the environment texture) offers the button “Enable Individual Transform”. With the button turned “off “, the texture mapping coordinates of the Viz base texture will be used.

All Shaders support up to 5 light sources from Viz. By enabling more than 5 lights, the last activated light will be disabled. The supported Viz light types are infinite and local lights; also the light color of these lights is supported.

If you import the Viz archive “*RTT_ShaderPresets_V1*” you will obtain about 60 sample scenes which are meant to give you an idea how to work with the Shaders. These presets are free to use, but in general, they are thought as a guideline for how to work with textures and the Shaders. Correctly working with the textures will dramatically improve the look of your Shader. The Shaders use color maps, bump maps (in the normal map format), specular maps and cubic environment maps in HDR and LDR formats.

Creating color maps is straightforward. They are regular RGB textures that you can use either on the Viz container or inside the Shader, depending on the Shader you are using.

Creating specular maps is also straightforward. They are grayscale textures where you define, through a gradient from white to black, where light is reflected or not.

Creating normal maps for the bump inside your Shader is not complicated either; just go to NVIDIA's corporate web page and download the relevant NVIDIA Photoshop plug-ins, or use this link: http://developer.nvidia.com/object/photoshop_dds_plugins.html. After installing the filter in question, you have a new section in the filter category in Adobe Photoshop. By applying this filter, you will be shown a window with various settings (Please read the corresponding NVIDIA user manual); yet there is not much work involved. Just have a look at the settings in the NVIDIA Photoshop Plug-in Image below and start exploring them on your own.

Creating cubic environment maps (HDR or LDR) is a well known technique for 3D experts. This manual will not explain the creation of these environment maps here in detail as there are numerous of tutorials on the web for this texture creation procedure.

If you do not have a license for the RTT Advanced Materials plug-ins, you may contact either one of your local Vizrt Support people or you can send an e-mail to license@vizrt.com to obtain either a time-limited demo license or a full license. Without a license, you will be able to load the preset scenes but you cannot modify them; only a few parameters will be adjustable.

This section also contains information on the following topics:

- Cube Map
- HDR
- Normal Map
- Fresnel
- RTT

Cube Map

Cube map textures are typically used for approximated environmental reflection and refraction. They consist of a set of six two-dimensional textures that form a textured cube centered at the origin. Unlike 2D sphere maps, they grant higher details for the whole environment. The RTT Advanced Materials plug-ins accept only cube maps as reflection textures and require them to be stored in a vertical cross layout. Cube maps can soon be created easily by using Vizrt's "Dynamic Texture" plug-in. To achieve the best results with the Shaders we strongly recommend to create your cube maps in HDR format. Please note that the vertical cross cubemap size is required to be one of the following: 192 by 256, 384 by 512, 768 by 1024, 1536 by 2048 or 3072 by 4096 pixels, where a higher resolution indicates higher image quality. The general rule of thumb is that the contained six images need to have power-of-2 resolution.

HDR

High Dynamic Range (HDR) images can store a much wider range of brightness values than common (i.e. low dynamic range) images. Therefore, it is possible to create more brilliant highlights and reflections when using HDR images. HDR images can be shot by using special spherical cameras, or created with the aid of HDR image processing tools. Furthermore ready-to-use HDR cube maps are available as DVD collections. All plug-ins with environmental reflections support HDR cube maps. ".hdr" format images are supported.

Normal Map

Normal maps are textures used to calculate bump mapping. They encode surface details - the normals of a surface - as RGB color values and therefore appear usually in some shades of blue. They can be created easily from (grayscale) height maps with plug-ins for Adobe Photoshop or Gimp.

Fresnel

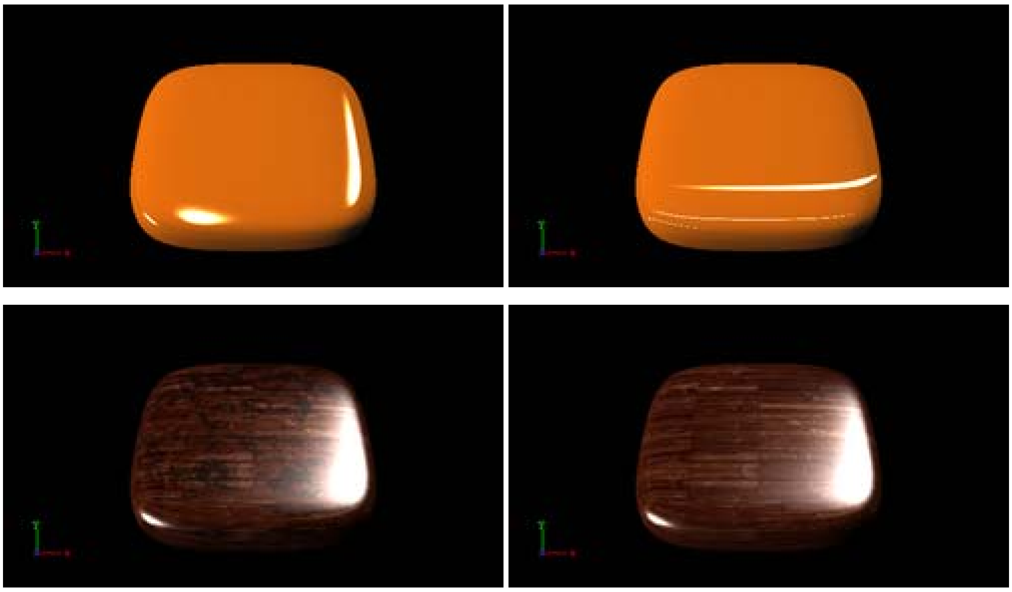
The Fresnel effect describes how much light is reflected on a surface and how much light is refracted through this surface. At shallow angles the reflection is strong while there is almost no refraction.

RTT

All RTT Advanced Materials are developed by Realtime Technology AG. Realtime Technology AG is a worldwide leading supplier of 3D real-time visualization technologies and services for industrial applications in the automotive, aircraft and consumer goods industries.

13.5.1 Anisotropic Light Shader

An anisotropic highlight is calculated to simulate advanced surfaces. The highlight can be controlled in using two parameters. In addition, a color texture can be used to add more details. This shader is particularly useful when no surrounding reflection is required.



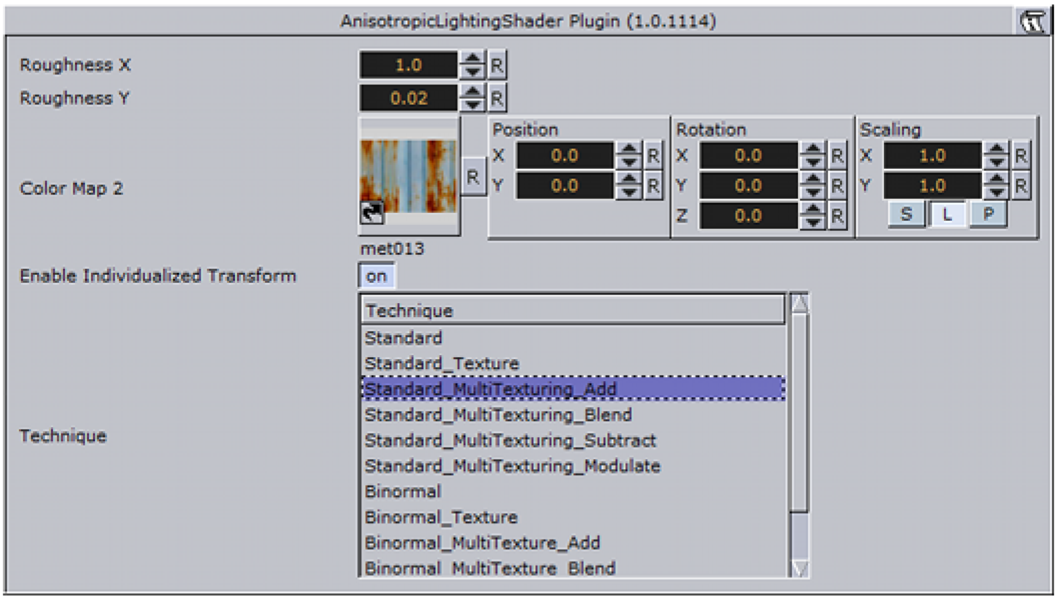
Note: All RTT shaders can be uninstalled from the Viz program menu.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\RttAnisotropicShader.vip`.

This section contains information on the following topics:

- Properties and Parameters
- Best Practices

Properties and Parameters



AnisotropicLightingShader Plugin (1.0.1114)

Roughness X 1.0

Roughness Y 0.02

Color Map 2 met013

Position X 0.0 Y 0.0

Rotation X 0.0 Y 0.0 Z 0.0

Scaling X 1.0 Y 1.0

Enable Individualized Transform on

Technique

- Standard
- Standard_Texture
- Standard_MultiTexturing_Add
- Standard_MultiTexturing_Blend
- Standard_MultiTexturing_Subtract
- Standard_MultiTexturing_Modulate
- Binormal
- Binormal_Texture
- Binormal_MultiTexture_Add
- Binormal_MultiTexture_Blend

- **Roughness X and Y**: Controls the width and height of the anisotropic highlight.
- **Color Map 2**: Offers the possibility to assign a second diffuse color texture that is blended with the first color texture (according to the chosen technique)
- **Enable Individualized Transform**: Activates the option to define an independent transformation (Position, Rotation and Scale values next to the Image Parameter) for Normal- and/or Bumpmaps instead of using the regular texture coordinates.
- **Technique**: Displays a list of available techniques.
  - **Standard**: Is a technique that creates an anisotropic highlight (to be used with geometry that has no texture coordinates and normals).
  - **Standard_Texture**: Is a technique that mixes a diffuse texture color into the material color.
  - **Standard_MultiTexture_Add**: Is a technique to additively blend Color Map 2 with the Viz texture.
  - **Standard_MultiTexture_Blend**: Is a technique to blend Color Map 2 with the Viz texture according to their alpha values.
  - **Standard_MultiTexture_Subtract**: Is a technique to subtractively blend Color Map 2 with the Viz texture.
  - **Standard_MultiTexture_Modulate**: Is a technique to blend Color Map 2 by multiplying it by the Viz texture.
  - **Binormal**: Is a technique to create an anisotropic highlight using binormals and tangents of the geometry (to be used with geometry having texture coordinates and normals).

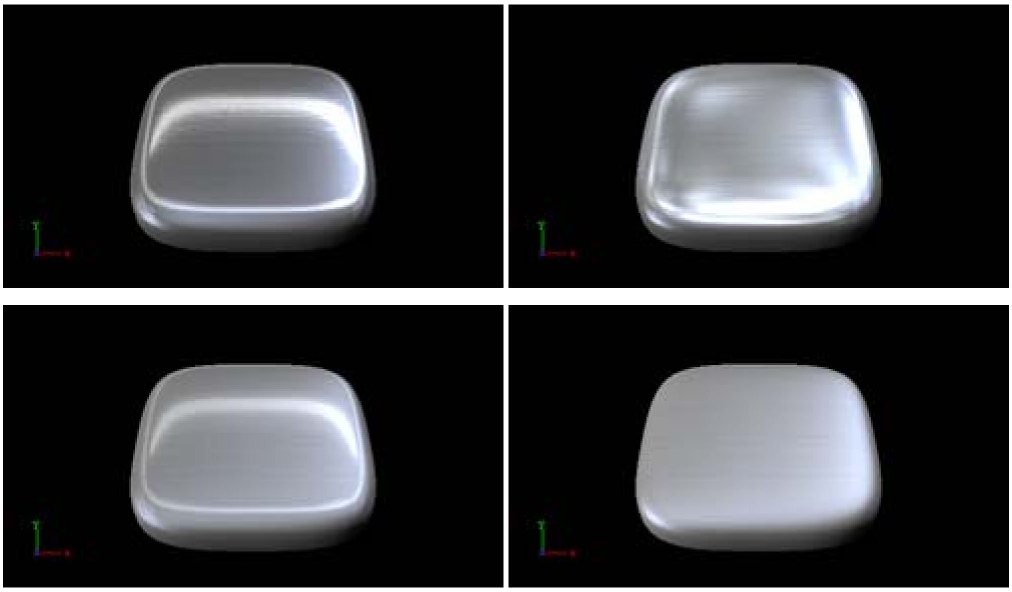
- **Binormal_Texture**: Is a technique to mix a diffuse texture color into the material color.
- **Binormal_MultiTexture_Add**: Is a technique additively blend Color Map 2 with the Viz texture.
- **Binormal_MultiTexture_Blend**: Is a technique to blend Color Map 2 with the Viz texture according to their alpha values.
- **Binormal_MultiTexture_Subtract**: Is a technique to subtractively blend Color Map 2 with the Viz texture.
- **Binormal_MultiTexture_Modulate**: Is a technique to blend Color Map 2 by multiplying it by the Viz texture.

Best Practices

Apart from the parameters in the plug-in container, it is necessary to assign a Viz material. When applying a texture technique it is compulsory to assign a Viz texture as a color/basic texture. In case of multi texturing an additional second texture must be assigned in the corresponding Shader rollout. Therefore, do not forget to adjust Viz material parameters in addition, such as ambient, diffuse, specular color and shininess.

13.5.2 Brushed Metal Shader

By using bump mapping and reflections from a cube map, the impression of more sophisticated metal surfaces, such as brushed metals, is created. The surface is lit by an anisotropic highlight. To describe the surface structure, a normal map is used.



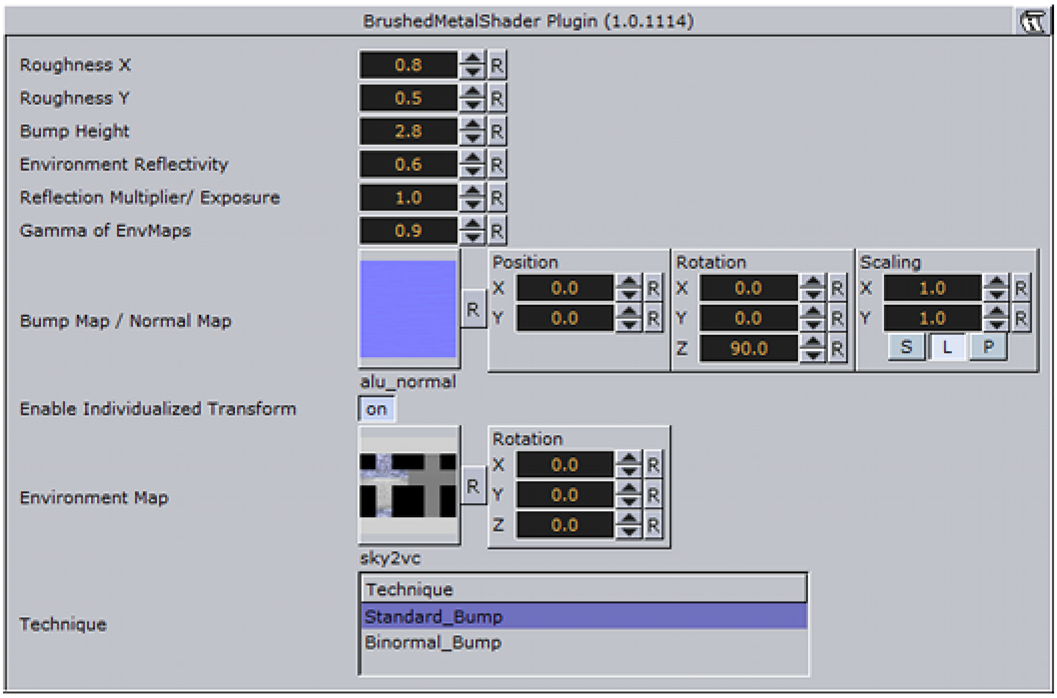
Note: All RTT shaders can be uninstalled from the Viz program menu.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\RttBrushedMetalShader.vip`.

This section contains information on the following topics:

- Properties and Parameters
- Best Practices

Properties and Parameters



BrushedMetalShader Plugin (1.0.1114)

Roughness X 0.8
Roughness Y 0.5
Bump Height 2.8
Environment Reflectivity 0.6
Reflection Multiplier/ Exposure 1.0
Gamma of EnvMaps 0.9
Bump Map / Normal Map alu_normal — Position X 0.0, Y 0.0; Rotation X 0.0, Y 0.0, Z 90.0; Scaling X 1.0, Y 1.0
Enable Individualized Transform on
Environment Map sky2vc — Rotation X 0.0, Y 0.0, Z 0.0
Technique Standard_Bump

- **Roughness X and Y**: Controls width and height of the anisotropic highlight.
- **Bump Height**: Affects the height of the created bumps and wrinkles on the surface.
- **Environment Reflectivity**: Manipulates the influence of the reflection color from the environment map on the resulting color.
- **Reflection Multiplier / Exposure**: Affects the exposure of the environment map. This is especially helpful when working with HDR images.
- **Gamma of EnvMaps**: Controls the gamma correction of the environment map. This is once again particularly helpful when working with HDR images.
- **Bump Map / Normal Map**: Is the texture that defines the surface structure with encoded normals.
- **Enable Individualized Transform**: Activates the option to define an independent transformation (Position, Rotation and Scale values next to the Image Parameter) for Normal- and/or Bumpmaps instead of using the regular texture coordinates.
- **Environment Map**: Is a vertical cross cube map that describes the environmental surrounding used to calculate reflections.
- **Technique**: Displays a list of available techniques.
  - **Standard Bump**: Is a technique that should be used with geometry that has no standard UV texture coordinates and normals.

- **Binormal Bump**: Is a technique that should be used with geometry having texture coordinates and normals.

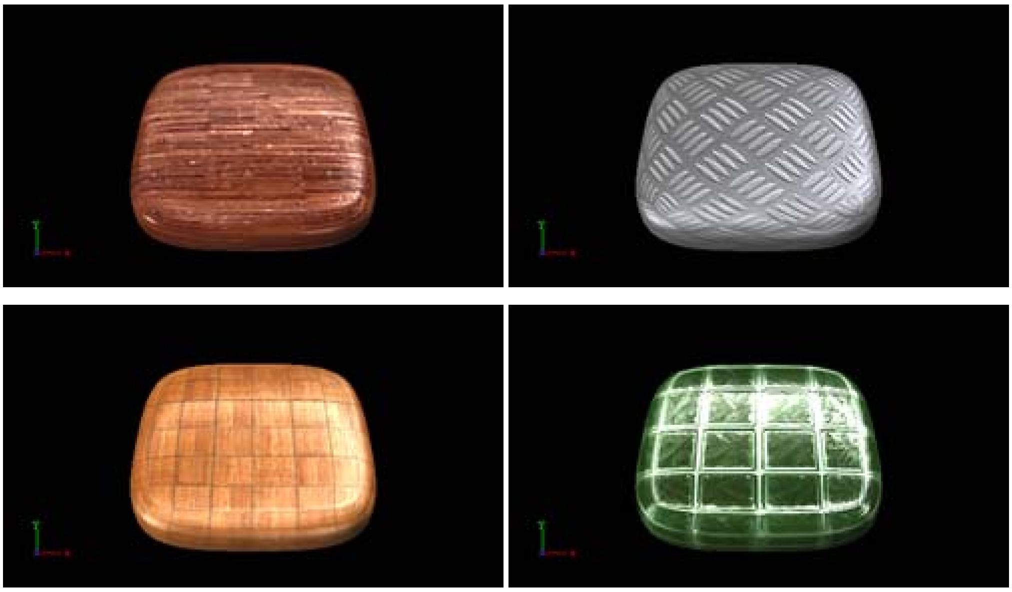
Best Practices

The Brushed Metal Shader should be preferred to Anisotropic Light Shader whenever you want to create highly detailed metal surfaces. However, the shader is more complex and requires more rendering time. Therefore, it may be unsuitable, depending on the complexity of the other parts of the scene and also on the graphics hardware available.

Apart from the parameters in the plug-in container, it is necessary to assign a Viz texture as a color/basic texture and a Viz material. Do not forget to adjust Viz material parameters in addition, such as ambient, diffuse, specular color and shininess, in order to achieve the desired look for your surface.

13.5.3 Bump Shader

By using bump mapping, the impression of detailed surface structures is created while the structures are defined by normal maps. In addition to a highlight, the reflection of the surrounding is calculated (from a cube map). It is possible to add blurriness to the reflection in order to create rougher surfaces. A parameter is available to modulate whether the reflections follow the surface structures or behave like a clear coat layer on the top of it. The specular map describes where on the surface highlight and reflections are displayed and where the surface shows only diffuse lighting.



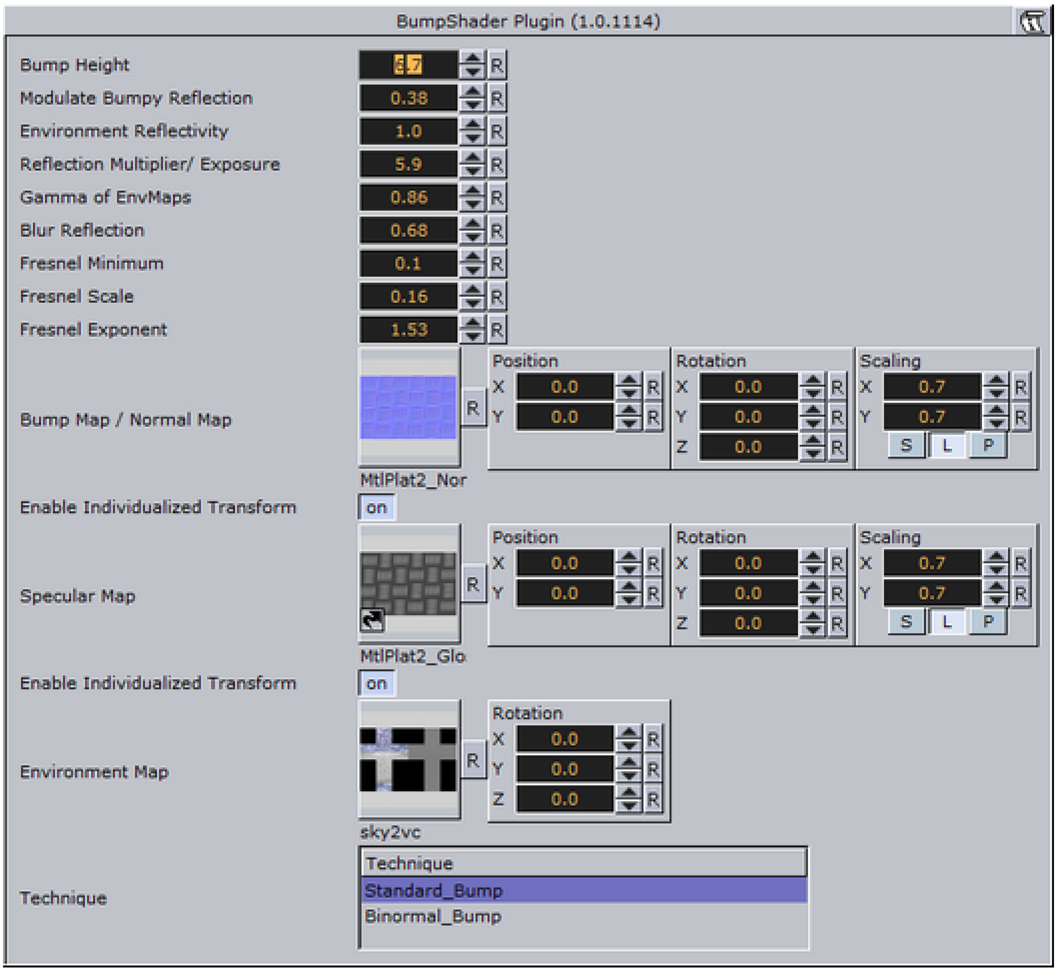
Note: All RTT shaders can be uninstalled from the Viz program menu.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\RttBumpGlossyShader.vip`.

This section contains information on the following topics:

- Properties and Parameters
- Best Practices

Properties and Parameters



BumpShader Plugin (1.0.1114)

Bump Height 6.7
Modulate Bumpy Reflection 0.38
Environment Reflectivity 1.0
Reflection Multiplier/ Exposure 5.9
Gamma of EnvMaps 0.86
Blur Reflection 0.68
Fresnel Minimum 0.1
Fresnel Scale 0.16
Fresnel Exponent 1.53
Bump Map / Normal Map MtlPlat2_Nor
Position X 0.0 Y 0.0
Rotation X 0.0 Y 0.0 Z 0.0
Scaling X 0.7 Y 0.7
Enable Individualized Transform on
Specular Map MtlPlat2_Glo
Position X 0.0 Y 0.0
Rotation X 0.0 Y 0.0 Z 0.0
Scaling X 0.7 Y 0.7
Enable Individualized Transform on
Environment Map sky2vc
Rotation X 0.0 Y 0.0 Z 0.0
Technique
Technique
Standard_Bump
Binormal_Bump

- **Bump Height**: Manipulates the height of the created bump on the surface.
- **Modulate Bumpy Reflection**: Controls whether the environment reflection is smooth or influenced by surface details. To calculate reflections on a smooth clear coat layer, value 0.0 is set by default. On the other hand, value 1.0 is used to incorporate surface details.
- **Environment Reflectivity**: Manipulates the influence of the reflection color from the environment map on the resulting color.
- **Reflection Multiplier/Exposure**: Affects the exposure of the environment map. This is especially helpful when working with HDR images.
- **Gamma of EnvMaps**: Controls the gamma correction of the environment map. This is once again particularly helpful when working with HDR images.
- **Blur Reflection**: States how much the environment reflection is blurred to create the impression of a rougher surface.

- **Fresnel Minimum, Fresnel Scale and Fresnel Exponent**: Affect the Fresnel effect upon the reflection.
- **Bump Map/Normal Map**: Is the texture that defines the surface structure/ bump with encoded normals.
- **Specular Map**: Is the texture that defines where the surface is lit by specular highlight and reflection.
- **Environment Map**: Is a cube map that describes the environmental surrounding used to calculate reflections.
- **Technique**: Displays a list of available techniques.
    - **Standard Bump**: Is a technique to be used with geometry that has no standard UV texture coordinates and normals.
    - **Binormal Bump**: Is a technique to be used with geometry having texture coordinates and normals.

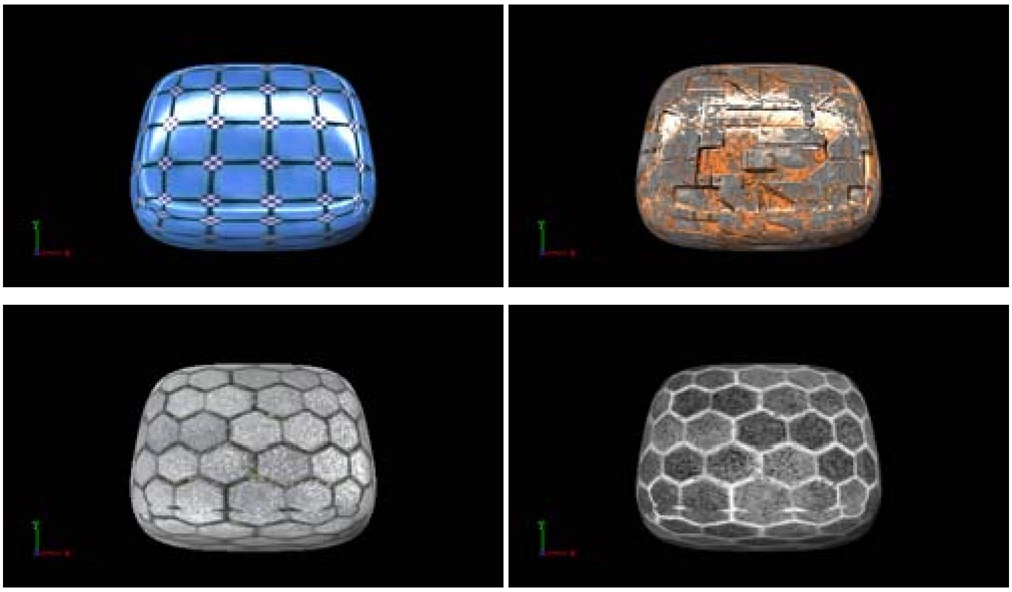
Best Practices

If you use this shader in complex scenes, you should favor the Bump Optimized Shader. The difference between the Bump Shader and the Bump Optimized Shader is that you can modify the texture coordinates inside the Bump Shader, which, however, needs more rendering performance. If you do not have to adjust the texture coordinates individually, or animate them separately, you should use the BumpOptimizedShader to save rendering performance.

Apart from the parameters in the plug-in container, it is necessary to assign a Viz texture as a color texture and a Viz material.

13.5.4 Bump Optimized Shader

By using bump mapping, the impression of detailed surface structures is created while the structures are defined by normal maps. In addition to a highlight, the reflection of the surrounding is calculated (from a cube map). It is possible to add blurriness to the reflection in order to create rougher surfaces. A parameter is available to modulate whether the reflections follow the surface structures or behave like a clear coat layer on the top of it. The specular map describes where on the surface highlight and reflections are displayed and where the surface shows only diffuse lighting.



This plug-in offers the same functionality as the Bump Shader plug-in, but as an optimization, it does not allow separate texture mapping for bump and specular maps. It consequently requires less rendering time and should be used whenever additional parameters are not required, i.e. you have matching sets of textures that can be used with the shader or you do not need to animate the textures. This shader should be preferred especially in complex scenes.

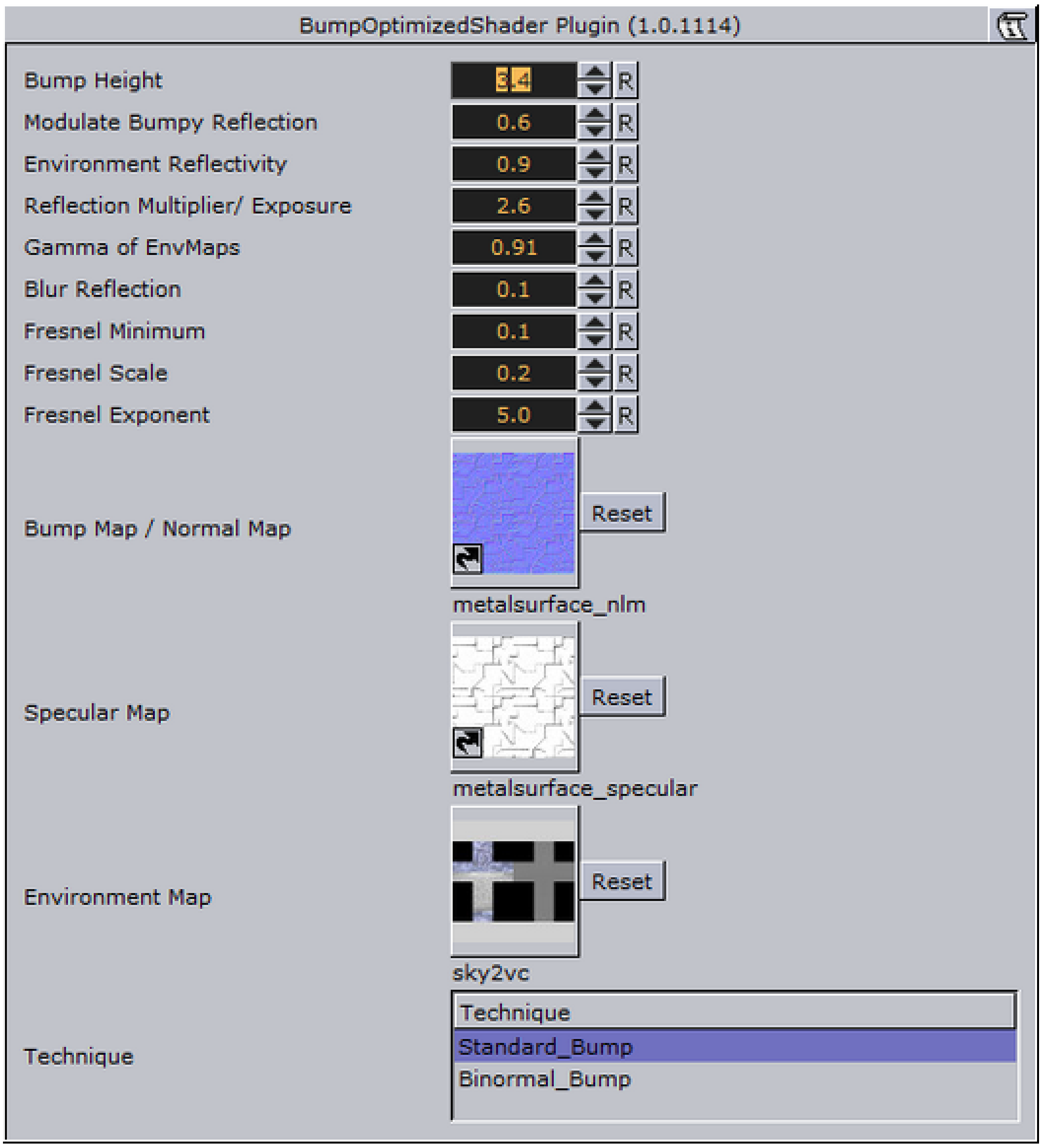
Note: All RTT shaders can be uninstalled from the Viz program menu.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\RttBumpOptimizedShader.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters



BumpOptimizedShader Plugin (1.0.1114)

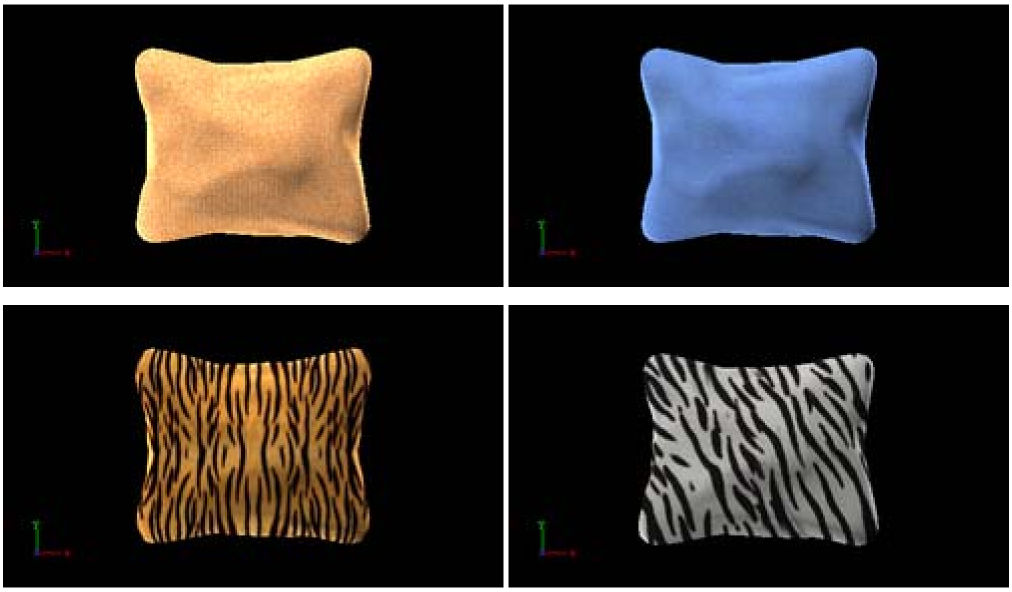
Bump Height 3.4
Modulate Bumpy Reflection 0.6
Environment Reflectivity 0.9
Reflection Multiplier/ Exposure 2.6
Gamma of EnvMaps 0.91
Blur Reflection 0.1
Fresnel Minimum 0.1
Fresnel Scale 0.2
Fresnel Exponent 5.0
Bump Map / Normal Map metalsurface_nlm
Specular Map metalsurface_specular
Environment Map sky2vc
Technique Standard_Bump

- **Bump Height**: Manipulates the height of the created bump on the surface.
- **Modulate Bumpy Reflection**: Controls whether the environment reflection is smooth or influenced by surface details. To calculate reflections on a smooth clear coat layer, value 0.0 is set by default. On the other hand, value 1.0 is used to incorporate surface details.
- **Environment Reflectivity**: Manipulates the influence of the reflection color from the environment map on the resulting color.
- **Reflection Multiplier/Exposure**: Affects the exposure of the environment map. This is especially helpful when working with HDR images.
- **Gamma of EnvMaps**: Controls the gamma correction of the environment map. This is once again particularly helpful when working with HDR images.
- **Blur Reflection**: States how much the environment reflection is blurred to create the impression of a rougher surface.
- **Fresnel Minimum, Fresnel Scale and Fresnel Exponent**: Affect the Fresnel effect upon the reflection.

- **Bump Map/Normal Map**: Is the texture that defines the surface structure/ bump with encoded normals.
- **Specular Map**: Is the texture that defines where the surface is lit by specular highlight and reflection.
- **Environment Map**: Is a cube map that describes the environmental surrounding used to calculate reflections.
- **Technique**: Displays a list of available techniques.
  - **Standard Bump**: Is a technique to be used with geometry that has no standard UV texture coordinates and normals.
  - **Binormal Bump**: Is a technique to be used with geometry having texture coordinates and normals.

13.5.5 Fabric Shader

On a dull surface, the impression of detailed surface structure is created by using bump mapping. The structure is defined by a normal map. A highlight is placed along the edges of the object to increase the impression of fabric surfaces.



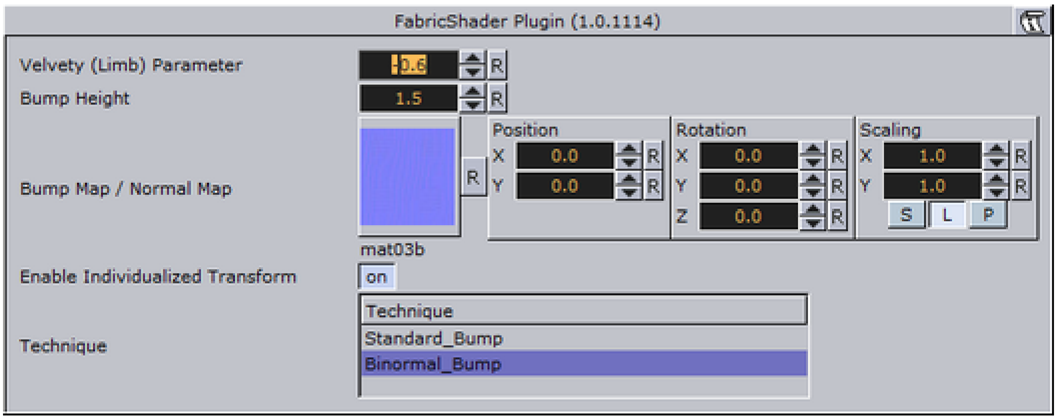
Note: All RTT shaders can be uninstalled from the Viz program menu.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\RttFabricShader.vip`.

This section contains information on the following topics:

- Properties and Parameters
- Best Practices

Properties and Parameters



FabricShader Plugin (1.0.1114)

Velvety (Limb) Parameter 0.6

Bump Height 1.5

Bump Map / Normal Map mat03b

Position X 0.0 Y 0.0

Rotation X 0.0 Y 0.0 Z 0.0

Scaling X 1.0 Y 1.0

Enable Individualized Transform on

Technique: Technique, Standard_Bump, Binormal_Bump

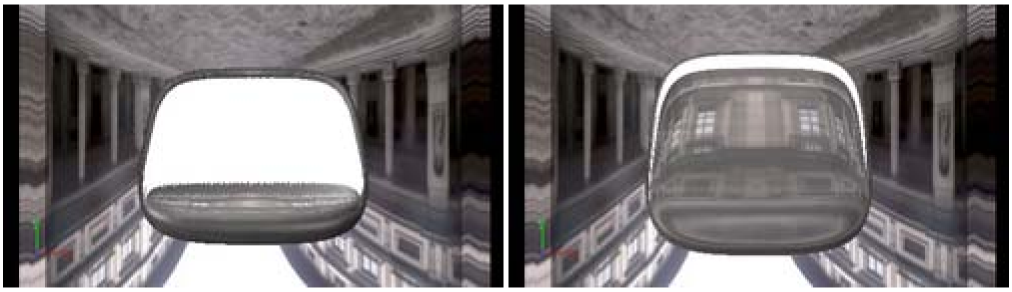
- **Velvety (Limb) Parameter**: Controls the strength of the velvety effect at grazing angles of view.
- **Bump Height**: Manipulates the height of the created bumps and wrinkles on the surface.
- **Bump Map/Normal Map**: Is the texture that defines the surface structure with encoded normals.
- **Technique**: Displays a list of available techniques.
  - **Standard_Bump**: Is a technique has to be used with geometry that has no standard UV texture coordinates and normals.
  - **Binormal_Bump**: Is a technique should be used with geometry having texture coordinates and normals.

Best Practices

13.5.6 Glass Shader

Apart from the parameters in the plug-in container, it is necessary to assign a Viz texture as a color/basic texture and a Viz material. Do not forget to adjust Viz material parameters in addition, such as ambient, diffuse, specular color and shininess, in order to achieve the desired look for your surface.

Both refraction and reflection are taken from the surrounding (from a cube map) and are blended to create the dielectric effect of glass. In addition, lighting by a highlight and a color texture are used to describe the material in detail. The opacity of the object can be controlled to modulate between the solidly lit surface and the transparent impression of glass.



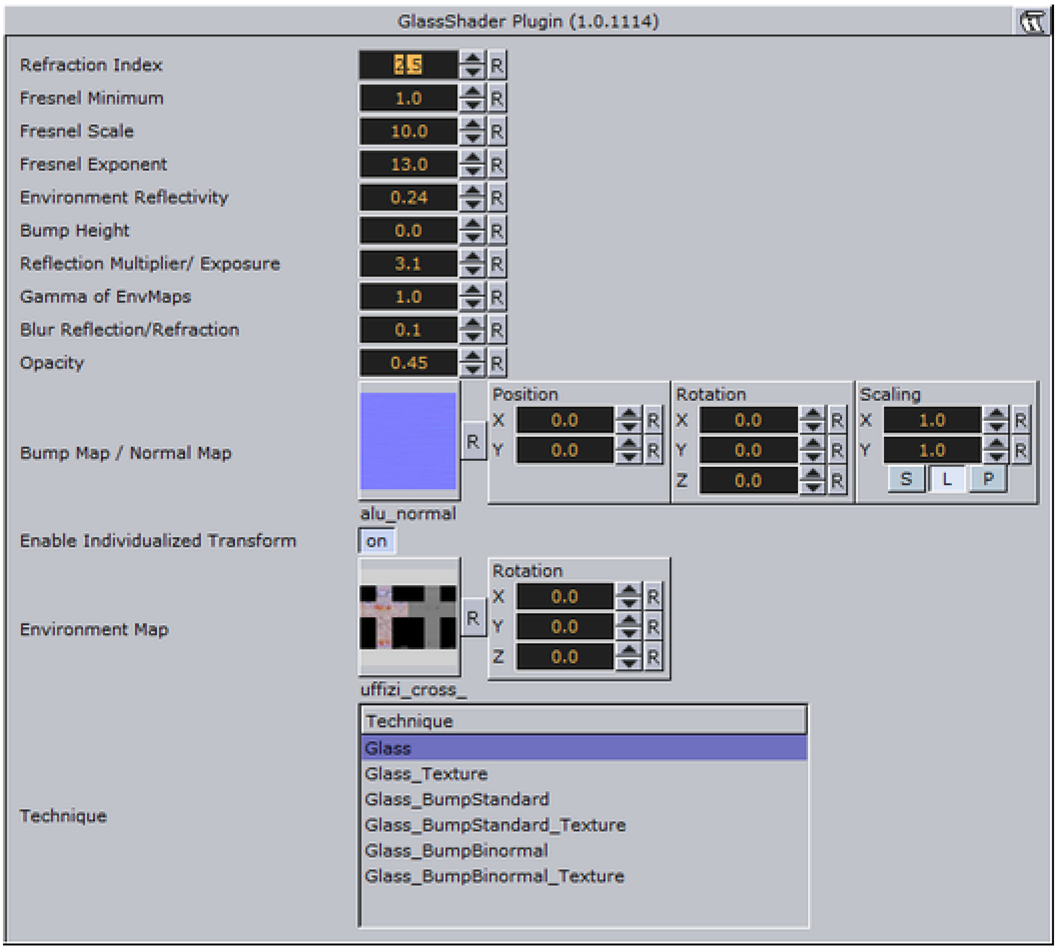
Note: All RTT shaders can be uninstalled from the Viz program menu.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\RttGlassShader.vip`.

This section contains information on the following topics:

- Properties and Parameters
- Best Practices

Properties and Parameters



GlassShader Plugin (1.0.1114)

Refraction Index 2.5

Fresnel Minimum 1.0

Fresnel Scale 10.0

Fresnel Exponent 13.0

Environment Reflectivity 0.24

Bump Height 0.0

Reflection Multiplier/ Exposure 3.1

Gamma of EnvMaps 1.0

Blur Reflection/Refraction 0.1

Opacity 0.45

Bump Map / Normal Map alu_normal

Position X 0.0 Y 0.0

Rotation X 0.0 Y 0.0 Z 0.0

Scaling X 1.0 Y 1.0

Enable Individualized Transform on

Environment Map uffizi_cross_

Rotation X 0.0 Y 0.0 Z 0.0

Technique

Technique: Glass (selected), Glass_Texture, Glass_BumpStandard, Glass_BumpStandard_Texture, Glass_BumpBinormal, Glass_BumpBinormal_Texture

- **Refraction Index**: Controls the refraction index of the material, and accordingly the strength of the refraction.
- **Fresnel Minimum, Fresnel Scale and Fresnel Exponent**: Affects the Fresnel effect upon the reflection.
- **Environment Reflectivity**: Manipulates the influence of the reflection color from the environment map on the resulting color.
- **Bump Height**: Manipulates the height of the created bumps on the surface.
- **Reflection Multiplier/Exposure**: Affects the exposure of the environment map.
- **Gamma of EnvMaps**: Controls the gamma correction of the environment map. This is particularly helpful when working with HDR images.
- **Blur Reflection**: States to which extent the environment reflection and refraction are blurred to create the impression of a rougher surface.
- **Opacity**: Influences the mixing of the solid material color and the dielectric glass effect (reflection and refraction).
- **Bump Map/Normal Map**: Is the texture that defines the surface structure with encoded normals.
- **Environment Map**: Is a cube map that describes the environmental surrounding used to calculate reflections and refractions.
- **Technique**: Displays a list of available techniques.
  - **Glass**: Is a technique to be used for plain glass.

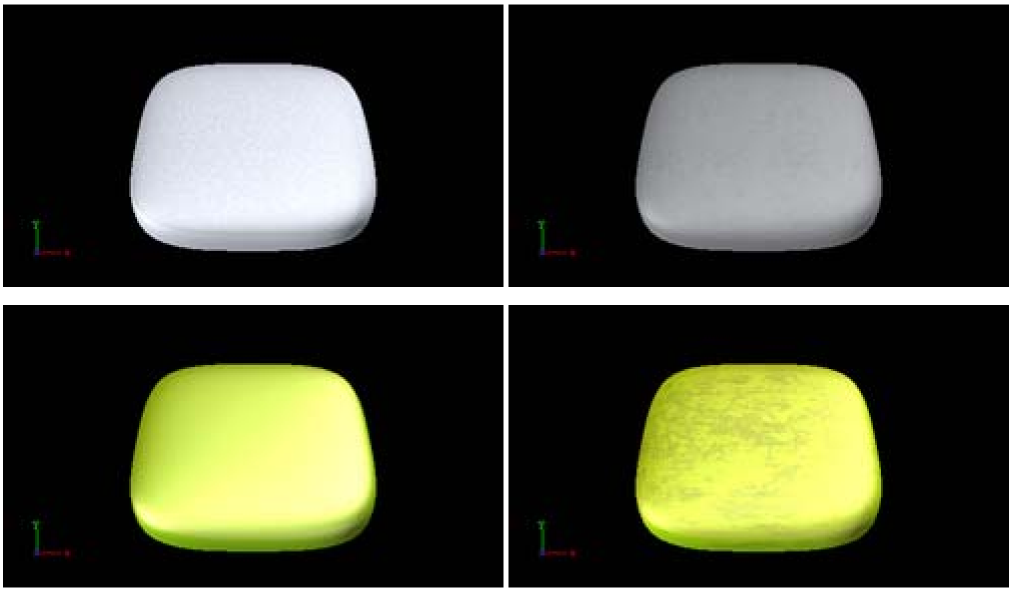
- **Glass_Texture**: Is a technique to mix a diffuse texture color into the material color.
- **Glass_BumpStandard**: Is a technique to create bump mapping with the given normal map on the surface of the geometry that has no standard UV texture coordinates and normals.
- **Glass_BumpStandard_Texture**: Is a technique to use a diffuse color texture and bump mapping on the surface of the geometry that has no texture coordinates and normals.
- **Glass_BumpBinormal**: Is a technique to create bump mapping for geometry with texture coordinates and normals.
- **Glass_BumpBinormal_Texture**: Is a technique to use a diffuse color texture and bump mapping on the surface of geometry with texture coordinates and normals.

Best Practices

Apart from the parameters in the plug-in container, it is necessary to assign a Viz material and a texture as a color/basic texture. Do not forget to adjust Viz material parameters in addition, such as ambient, diffuse, specular color and shininess, in order to achieve the desired look for your surface.

13.5.7 Gooch Shader

Gooch shading is used to calculate the lighting of the surface. Accordingly, a warm color is displayed in lit areas while a cold color is used in unlit areas. A highlight is further added. Besides, a color texture may be used to add more details. This shader is particularly useful when no surrounding reflection is required.



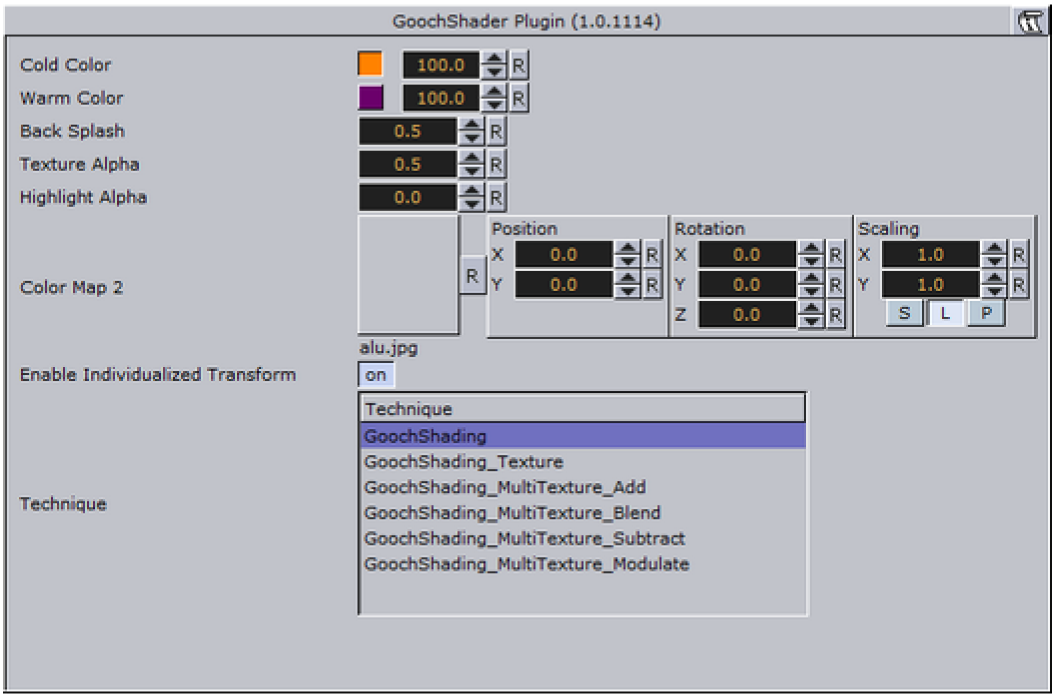
Note: All RTT shaders can be uninstalled from the Viz program menu.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\RttGoochShader.vip`.

This section contains information on the following topics:

- Properties and Parameters
- Best Practices

Properties and Parameters



GoochShader Plugin (1.0.1114)

Cold Color	100.0
Warm Color	100.0
Back Splash	0.5
Texture Alpha	0.5
Highlight Alpha	0.0
Color Map 2	alu.jpg Position X 0.0, Y 0.0 Rotation X 0.0, Y 0.0, Z 0.0 Scaling X 1.0, Y 1.0
Enable Individualized Transform	on
Technique	GoochShading

Technique
GoochShading
GoochShading_Texture
GoochShading_MultiTexture_Add
GoochShading_MultiTexture_Blend
GoochShading_MultiTexture_Subtract
GoochShading_MultiTexture_Modulate

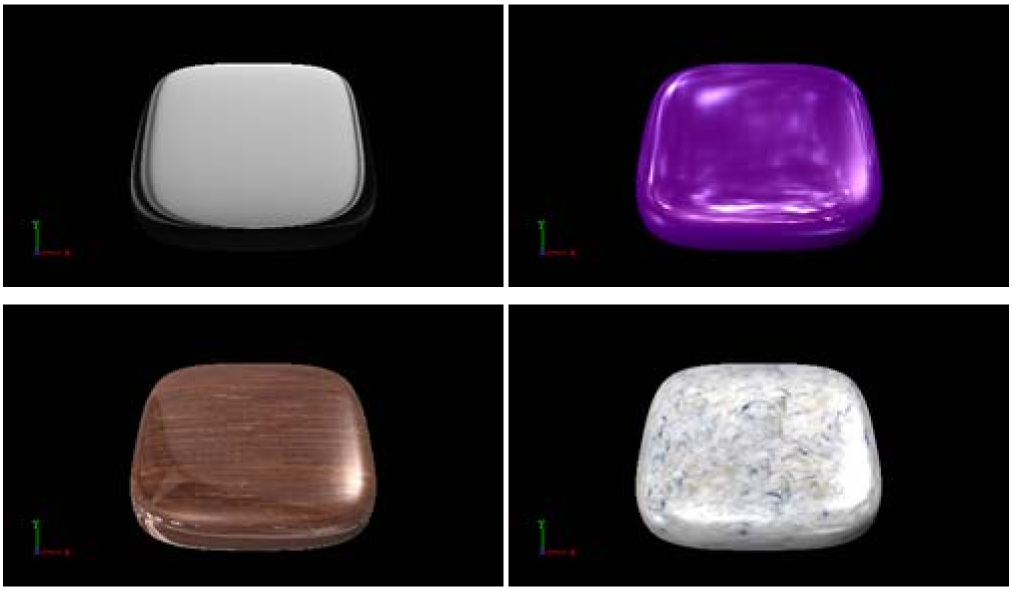
- Cold Color: Assigns the color used for unlit surfaces.
- Warm Color: Assigns the color used for lit surfaces.
- Back Splash: Controls a factor to increase the perception of curvature on grazing angles. The resulting color is weighted according to the back splash factor.
- Texture Alpha: Defines to which amount the diffuse color texture is blended into the resulting color.
- Highlight Alpha: Defines to which extent the calculated highlight is blended into the resulting color.
- Color Map 2: Offers the possibility to assign a second diffuse color texture that is blended with the first color texture (according to the chosen technique).
- Technique: Displays a list of available techniques.
  - GoochShading: Is a technique to use Gooch shading.
  - GoochShading_Texture: Is a technique to mix a diffuse texture color into the material color.
  - GoochShading_MultiTexture_Add: Is a technique to additively blend Color Map 2 with the Viz texture.
  - GoochShading_MultiTexture_Blend: Is a technique to blend Color Map 2 with the Viz texture according to their alpha values.
  - GoochShading_MultiTexture_Subtract: Is a technique to subtractively blend Color Map 2 with the Viz texture.
  - GoochShading_MultiTexture_Modulate: Is a technique to blend Color Map 2 by multiplying it by the Viz texture.

Best Practices

Apart from the parameters in the plug-in container, it is necessary to assign a Viz material. Therefore, do not forget to adjust Viz material parameters in addition, such as ambient, diffuse, specular color and shininess. When applying a texture technique it is compulsory to assign an Viz texture as a color/basic texture. In case of multi texturing an additional, second texture must be assigned in the corresponding Shader rollout.

13.5.8 Lacquered Surfaces Shader

The RTT Lacquered Surface shader simulates smooth lacquered surfaces lit by a highlight. On the clear coat layer, the surrounding is reflected (from a cube map). The reflection blurriness can be controlled to create the appearance of a rougher surface.



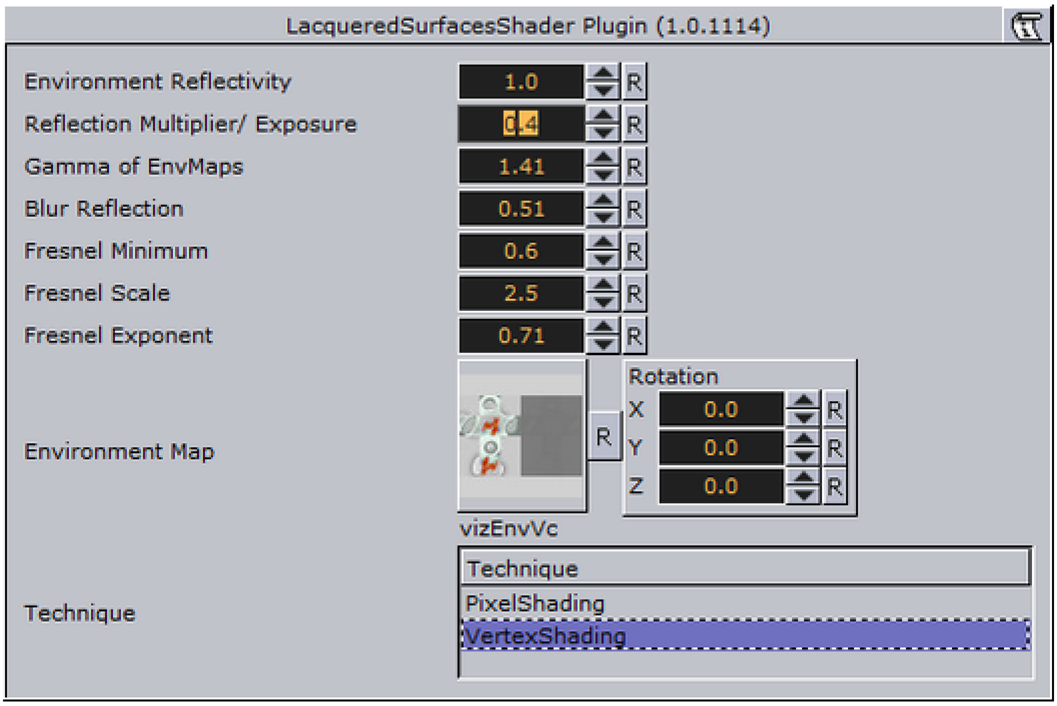
Note: All RTT shaders can be uninstalled from the Viz program menu.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\RttWoodGlossyShader.vip`.

This section contains information on the following topics:

- Properties and Parameters
- Best Practices

Properties and Parameters



LacqueredSurfacesShader Plugin (1.0.1114)

Environment Reflectivity 1.0 R
Reflection Multiplier/ Exposure 0.4 R
Gamma of EnvMaps 1.41 R
Blur Reflection 0.51 R
Fresnel Minimum 0.6 R
Fresnel Scale 2.5 R
Fresnel Exponent 0.71 R
Environment Map vizEnvVc R
Rotation
X 0.0 R
Y 0.0 R
Z 0.0 R
Technique
Technique
PixelShading
VertexShading

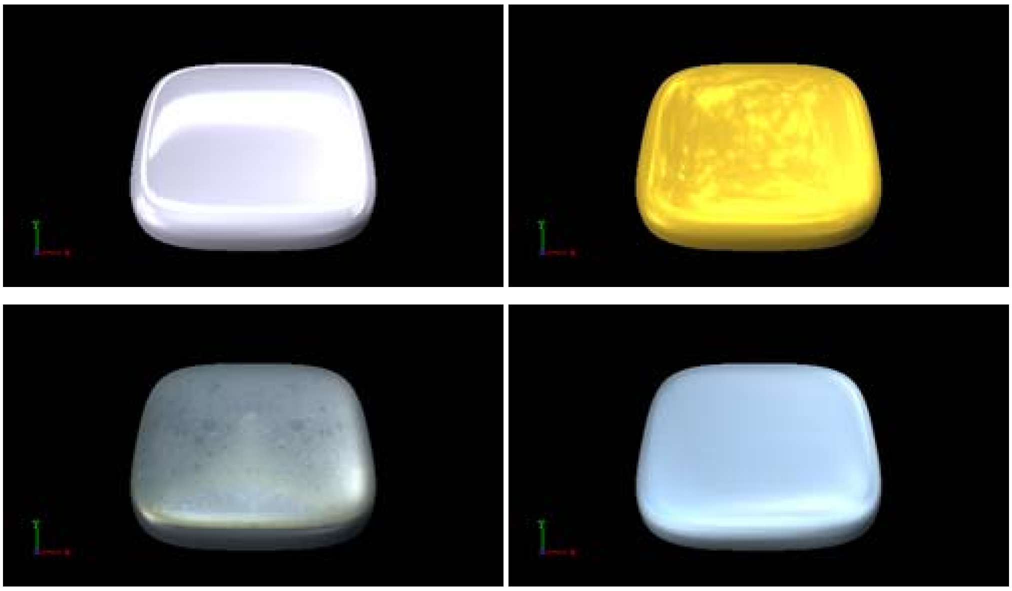
- **Environment Reflectivity**: Manipulates the influence of the reflection color from the environment map on the resulting color.
- **Reflection Multiplier/Exposure**: Affects the exposure of the environment map. This is especially helpful when working with HDR images.
- **Gamma of EnvMaps**: Controls the gamma correction of the environment map. This is once again particularly helpful when working with HDR images.
- **Blur Reflection**: States how much the environment reflection is blurred to create the impression of a rougher surface.
- **Fresnel Minimum, Fresnel Scale and Fresnel Exponent**: Affects the Fresnel effect upon the reflection.
- **Environment Map**: Is a cube map that describes the environmental surrounding used to calculate reflections.
- **Technique**: Displays a list of available techniques.
  - **PixelShading**: Is a technique to be used for finer details on the surface.
  - **VertexShading**: Is a technique to be used for objects in the distance or for highly tessellated objects.

Best Practices

Apart from the parameters in the plug-in container, it is necessary to assign a Viz material and a texture as a color/basic texture. Do not forget to adjust Viz material parameters in addition, such as ambient, diffuse, specular color and shininess, in order to achieve the desired look for your surface.

13.5.9 Metal Reflection Shader

The object is lit by a highlight and by the reflection of the surrounding (from a cube map). Both highlight and reflection take into account the material color of the object to generate a realistic metal effect.



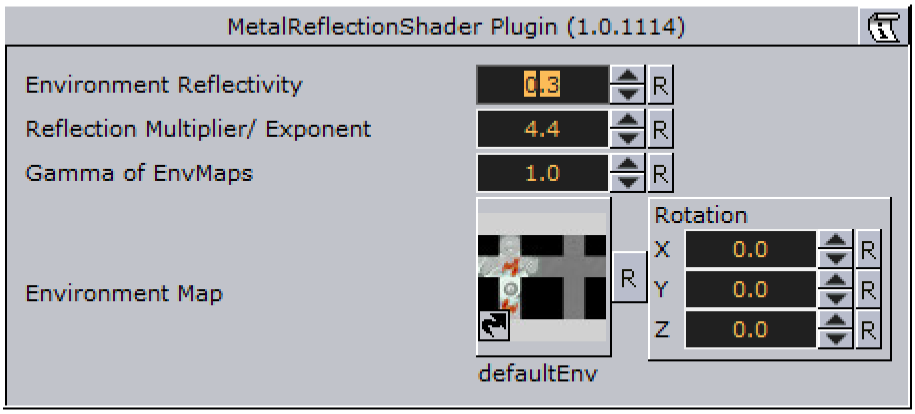
Note: All RTT shaders can be uninstalled from the Viz program menu.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\RttMetalShader.vip`.

This section contains information on the following topics:

- Properties and Parameters
- Best Practices

Properties and Parameters



MetalReflectionShader Plugin (1.0.1114)

Environment Reflectivity 0.3 R

Reflection Multiplier/ Exponent 4.4 R

Gamma of EnvMaps 1.0 R

Environment Map R defaultEnv

Rotation

X 0.0 R

Y 0.0 R

Z 0.0 R

- **Environment Reflectivity**: Manipulates the influence of the reflection color from the environment map on the resulting color.
- **Reflection Multiplier/Exposure**: Affects the exposure of the environment map. This is especially helpful when working with HDR images.
- **Gamma of EnvMaps**: Controls the gamma correction of the environment map. This is once again particularly helpful when working with HDR images.
- **Environment Map**: Is a cube map that describes the environmental surrounding used to calculate reflections.

Best Practices

Apart from the parameters in the plug-in container, it is necessary to assign a Viz material and a texture as a color/basic texture. Do not forget to adjust Viz material parameters in addition, such as ambient, diffuse, specular color and shininess, in order to achieve the desired look for your surface. To achieve a photorealistic, metallic look, it is recommended to assign any color but white as a specular color.

13.5.10 Microstructure Shader

A detailed surface structure is simulated by bump mapping and lit with the aid of a highlight. Additionally, a clear coat layer reflects the surrounding (from a cube map). A normal map is used to define the surface structure.

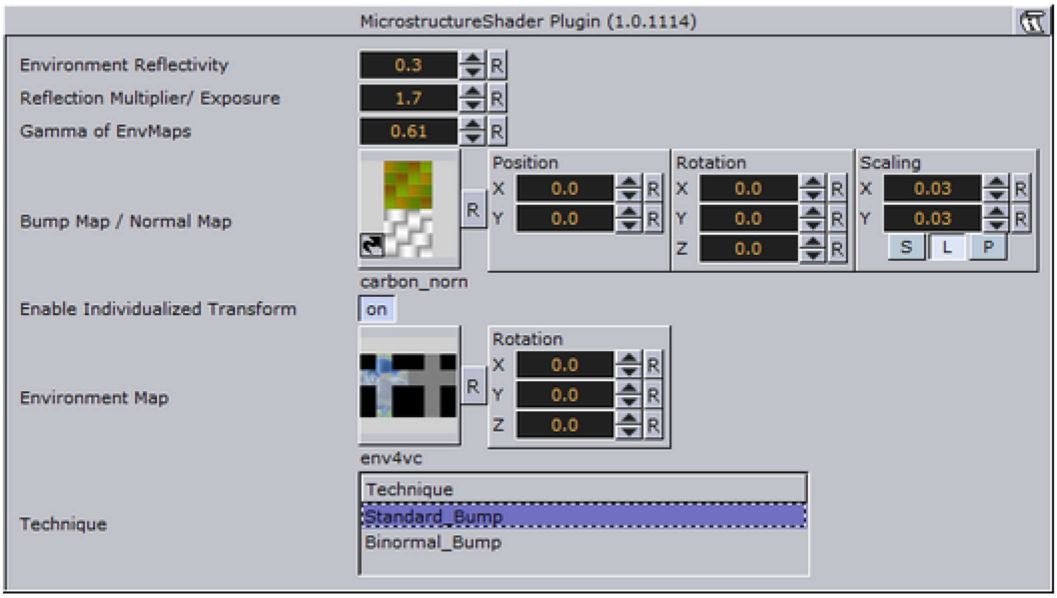
Note: All RTT shaders can be uninstalled from the Viz program menu.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\RttMicrostructureShader.vip`.

This section contains information on the following topics:

- Properties and Parameters
- Best Practices

Properties and Parameters



MicrostructureShader Plugin (1.0.1114)

Environment Reflectivity 0.3

Reflection Multiplier/ Exposure 1.7

Gamma of EnvMaps 0.61

Bump Map / Normal Map carbon_norn — Position X 0.0, Y 0.0; Rotation X 0.0, Y 0.0, Z 0.0; Scaling X 0.03, Y 0.03

Enable Individualized Transform on

Environment Map env4vc — Rotation X 0.0, Y 0.0, Z 0.0

Technique: Standard_Bump, Binormal_Bump

- **Environment Reflectivity**: Manipulates the influence of the reflection color from the environment map on the resulting color.
- **Reflection Multiplier/Exposure**: Affects the exposure of the environment map. This is especially helpful when working with HDR images.
- **Gamma of EnvMaps**: Controls the gamma correction of the environment map. This is once again particularly helpful when working with HDR images.
- **Bump Map / Normal Map**: Is the texture that defines the surface structure with encoded normals.
- **Environment Map**: Is a cube map that describes the environmental surrounding used to calculate reflections.
- **Technique**: Displays a list of available techniques.

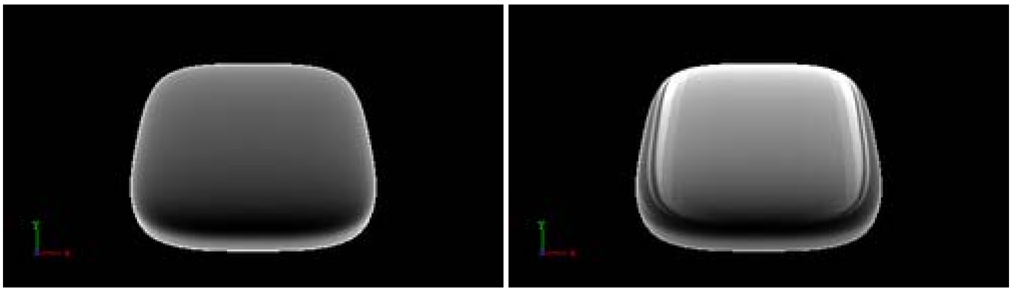
- **Standard_Bump**: Is a technique to be used with geometry that has no standard UV texture coordinates and normals.
- **Binormal_Bump**: Is a technique to be used with geometry having texture coordinates and normals.

Best Practices

Apart from the parameters in the plug-in container, it is necessary to assign a Viz material and a texture as a color/basic texture. Do not forget to adjust also the Viz material parameters like ambient, diffuse, specular color and shininess, to achieve the desired look for your surface

13.5.11 Monitor Shader

The RTT Monitor shader simulates the appearance of a flat screen display where saturation decreases at grazing angles of view. Therefore, the displayed color texture blends into the diffuse color of the material. In addition, a reflection from the surrounding (from a cube map) is added, increasing in strength towards more acute angles of view.



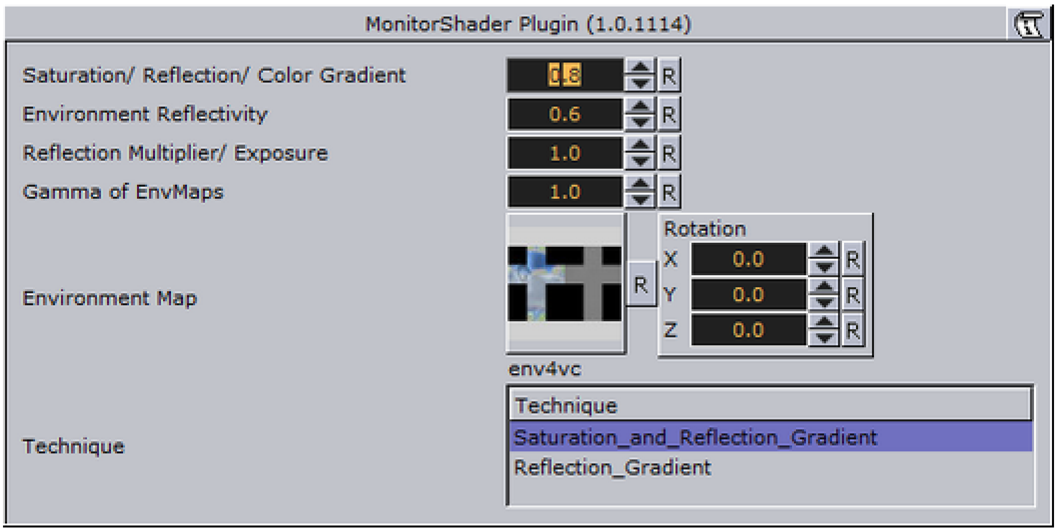
Note: All RTT shaders can be uninstalled from the Viz program menu.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\RttScreenShader.vip`.

This section contains information on the following topics:

- Properties and Parameters
- Best Practices

Properties and Parameters



MonitorShader Plugin (1.0.1114)

Saturation/ Reflection/ Color Gradient 0.8 R

Environment Reflectivity 0.6 R

Reflection Multiplier/ Exposure 1.0 R

Gamma of EnvMaps 1.0 R

Environment Map R env4vc

Rotation

X 0.0 R

Y 0.0 R

Z 0.0 R

Technique

Technique

Saturation_and_Reflection_Gradient

Reflection_Gradient

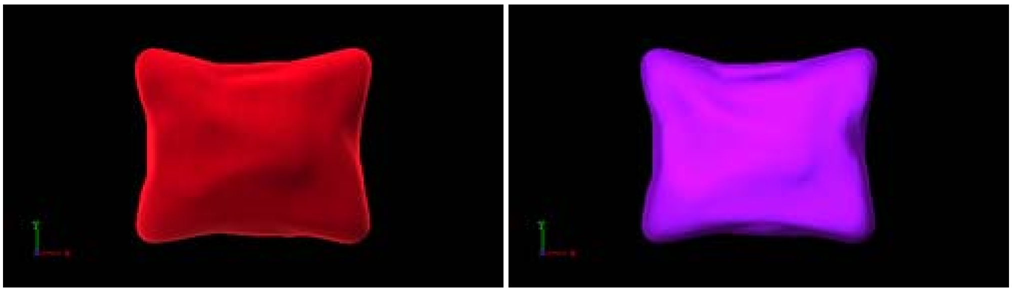
- **Saturation/Reflection/Color Gradient**: Defines to which amount the material's diffuse color is mixed into the resulting color at grazing angles.
- **Environment Reflectivity**: Manipulates the influence of the reflection color from the environment map on the resulting color.
- **Reflection Multiplier/Exposure**: Affects the exposure of the environment map. This is especially helpful when working with HDR images.
- **Gamma of EnvMaps**: Controls the gamma correction of the environment map. This is once again particularly helpful when working with HDR images.
- **Environment Map**: is a cube map that describes the environmental surrounding used to calculate reflections.
- **Technique**: Displays a list of available techniques.
  - **Saturation_and_Reflection_Gradient**: Is a technique to influence saturation and reflection appearance alike.
  - **Reflection_Gradient**: Is a technique to influence only the reflection appearance.

Best Practices

Apart from the parameters in the plug-in container, it is necessary to assign a Viz material and a texture as a color/basic texture. For this shader, a background image should be selected as the basic texture to achieve a highly realistic monitor effect. Do not forget to adjust Viz material parameters in addition, such as ambient, diffuse, specular color and shininess, in order to achieve the desired look for your surface.

13.5.12 Velvet Shader

A highlight along the edges of the surface is created to generate velvety impression. The highlight is controlled in using two parameters that describe the transition towards diffuse lighting. Furthermore, a surface color can be added to brighten the barely lit areas. A color texture may be used to add more surface details. This shader is particularly useful when no surrounding reflection is required.



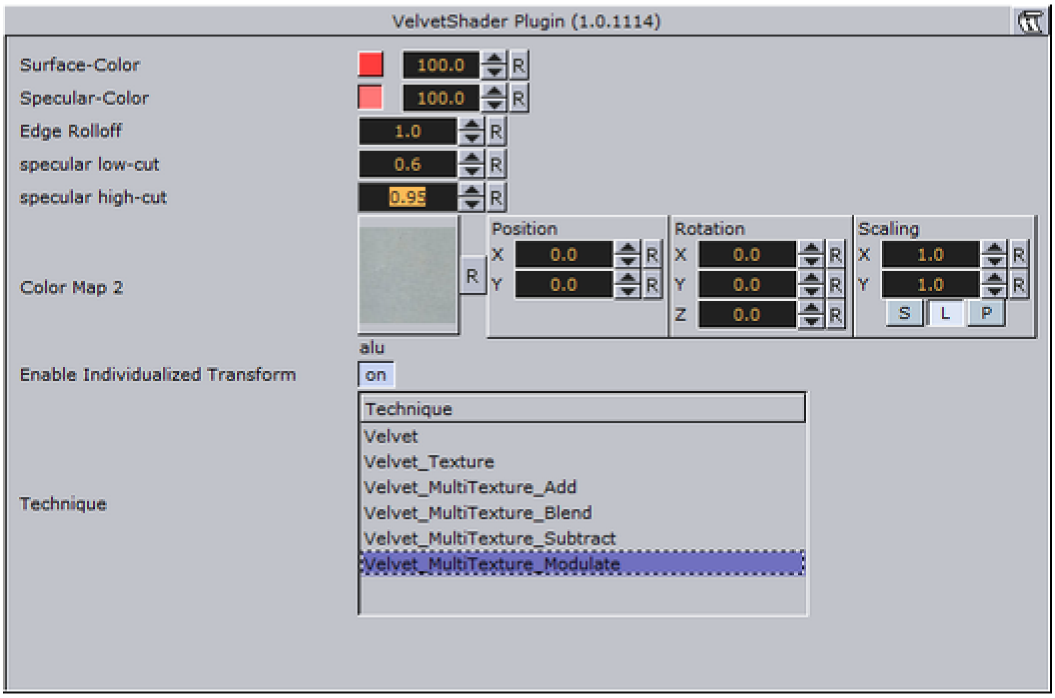
Note: All RTT shaders can be uninstalled from the Viz program menu.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\RttVelvetShader.vip`.

This section contains information on the following topics:

- Properties and Parameters
- Best Practices

Properties and Parameters



VelvetShader Plugin (1.0.1114)

Surface-Color 100.0

Specular-Color 100.0

Edge Rolloff 1.0

specular low-cut 0.6

specular high-cut 0.95

Color Map 2: alu

Position X 0.0, Y 0.0

Rotation X 0.0, Y 0.0, Z 0.0

Scaling X 1.0, Y 1.0

Enable Individualized Transform: on

Technique:
Velvet
Velvet_Texture
Velvet_MultiTexture_Add
Velvet_MultiTexture_Blend
Velvet_MultiTexture_Subtract
Velvet_MultiTexture_Modulate

- **Surface-Color**: Assigns the color of the surface (underneath the velvety hairs).
- **Specular-Color**: Assigns the specular color of the material.
- **Edge Rolloff**: Defines how much of the surface color is to be seen in barely lit areas.

- **Specular low-cut and specular high-cut**: Defines the sharpness of the velvety highlight.
- **Color Map 2**: Offers the possibility to assign a second diffuse color texture that is blended with the first color texture.
- **Technique**: Displays a list of available techniques.
  - **Velvet**: Is a technique to use velvety shading,
  - **Velvet_Texture**: Is a technique to mix a diffuse texture color into the material color.
  - **Velvet_MultiTexture_Add**: Is a technique to additively blend Color Map 2 with the Viz texture.
  - **Velvet_MultiTexture_Blend**: Is a technique to blend Color Map 2 with the Viz texture according to their alpha values.
  - **Velvet_MultiTexture_Subtract**: Is a technique to subtractively blend Color Map 2 with the Viz texture.
  - **Velvet_MultiTexture_Modulate**: Is a technique to blend Color Map 2 by multiplying it by the Viz texture.

Best Practices

Apart from the parameters in the plug-in container, it is necessary to assign a Viz material. Therefore, do not forget to adjust Viz material parameters in addition, such as ambient, diffuse, specular color and shininess. When applying a texture technique it is compulsory to assign an Viz texture as a color/basic texture. In case of multi texturing an additional, second texture must be assigned in the corresponding Shader rollout.

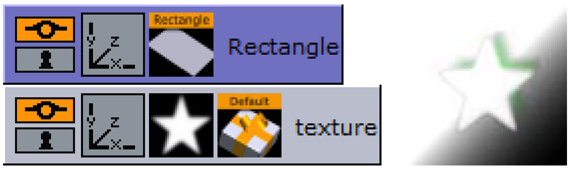
13.6 Texture

The following shader plugins are housed in the Texture folder:

- Drop Shadow
- Emboss

13.6.1 Drop Shadow

Drop Shadow is a shader plug-in that generates 2D shadow of a texture.

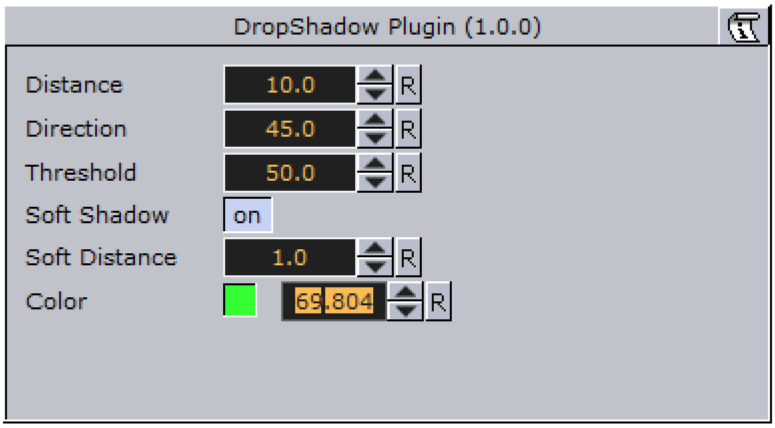


The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\DropShadow.vip`.

This section contains information on the following topics:

- Properties and Parameters

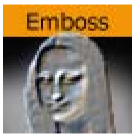
Properties and Parameters


DropShadow Plugin (1.0.0)

Distance 10.0
Direction 45.0
Threshold 50.0
Soft Shadow on
Soft Distance 1.0
Color 🟩 69.804

- **Distance**: Sets the distance of the shadow.
- **Direction**: Sets the direction of the shadow in degrees.
- **Threshold**: Sets the shadow threshold.
- **Soft Shadow**: Enables the Soft Distance parameter for applying a soft shadow.
  - **Soft Distance**: Sets the distance of the soft shadow.
- **Color**: Sets the color of the shadow.

13.6.2 Emboss



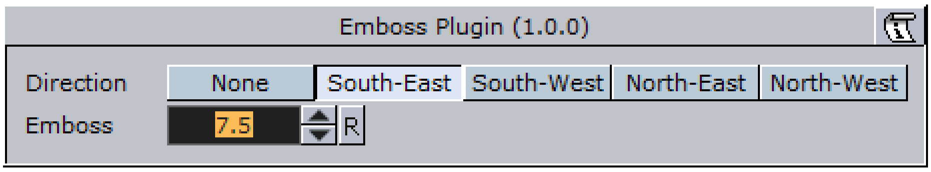
Emboss is a simple shader plug-in that allows you to raise highlighted surfaces and lower shadows of textures and images creating an embossed look.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Emboss.vip`.

This section contains information on the following topics:

- Properties and Parameters

Properties and Parameters


Emboss Plugin (1.0.0)

Direction: None | South-East | South-West | North-East | North-West

Emboss: 7.5 R

- **Direction**: Sets the direction. Available options are: None, South-East, South-West, North-East and North-West.
- **Emboss**: Sets the level in percentage.

14 Scene Plugins

Scene plugs are housed in the following folders:

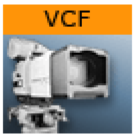
- Default
- Image
- Lineup
- Script Plugins
- Texture
- Tools

14.1 Default

The following scene plugins are housed in the Default folder:

- VCF

14.1.1 VCF



The VCF scene plug-in works in conjunction with the VCF Parameter plug-in that will allow you to create a seamless interpolated transition from a virtual camera flight (VCF) to a real camera - and conversely. This is only a relevant plug-in to set up if you have purchased the virtual set expansion components.

You must have a real camera with data tracking enabled which is set in remote mode in the camera editor. In case you have several tracked cameras, the virtual camera will interpolate its position to the real camera that is selected on air.

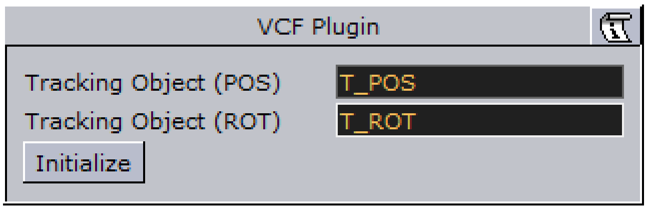
Note: Scene plug-ins are added under the Scene Settings' Plugin tab.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\VCF.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To animate a virtual camera flight

Properties and Parameters



VCF Plugin

Tracking Object (POS) T_POS

Tracking Object (ROT) T_ROT

- **Tracking Object (POS)**: Defines the position or eye-point of the virtual camera flight. Use the name of the container in which center the camera should be placed.
- **Tracking Object (ROT)**: Defines the direction of the virtual camera flight. Use the name of the container the camera will be looking at.
- **Initialize (button)**: Places the camera in position and sets the direction.

To animate a virtual camera flight

1. Start by adding the VCF scene plug-in under the Scene Settings' Plugin tab.
2. Create a new group in your scene and add two new containers under it.
   - These are to be the objects that will define the virtual camera flight. One will define the position, the other the direction.
3. Name the containers according to the names you entered in the VCF scene plug-in (for example T_POS and T_ROT).
4. Click Initialize to finish.
5. Add the VCF Parameter plug-in onto the container that holds the position object.
6. Animate your virtual camera flight using the two objects to define position and direction.
   - Do **not** switch the whole container invisible because the animation will not run.
   - You may switch the objects to be invisible at any time.
   - It is not possible to animate the roll of the camera.

See Also

- VCF Parameter

14.2 Image

The following scene plugins are housed in the Image folder:

- Background Clip

14.2.1 Background Clip

This plug-in is designed to play back a sequence of still images (tga, tiff, etc.) in the background of your scene rather than playing an AVI-file.

Since it plays the sequence from memory, the scene has a higher loading time and the drawback is that it consumes system memory which can influence the system stability. So keep track of how much memory each sequence needs so you don't run out of memory on the render engine! Scene plug-ins are added under the Scene Settings' Plugin tab.

Since all images are loaded into memory a large number of images or a large image size would require large amounts of memory. Memory can be calculated as follows:

```
number of images * image width * image height * 3 (or 4 for alpha).
```

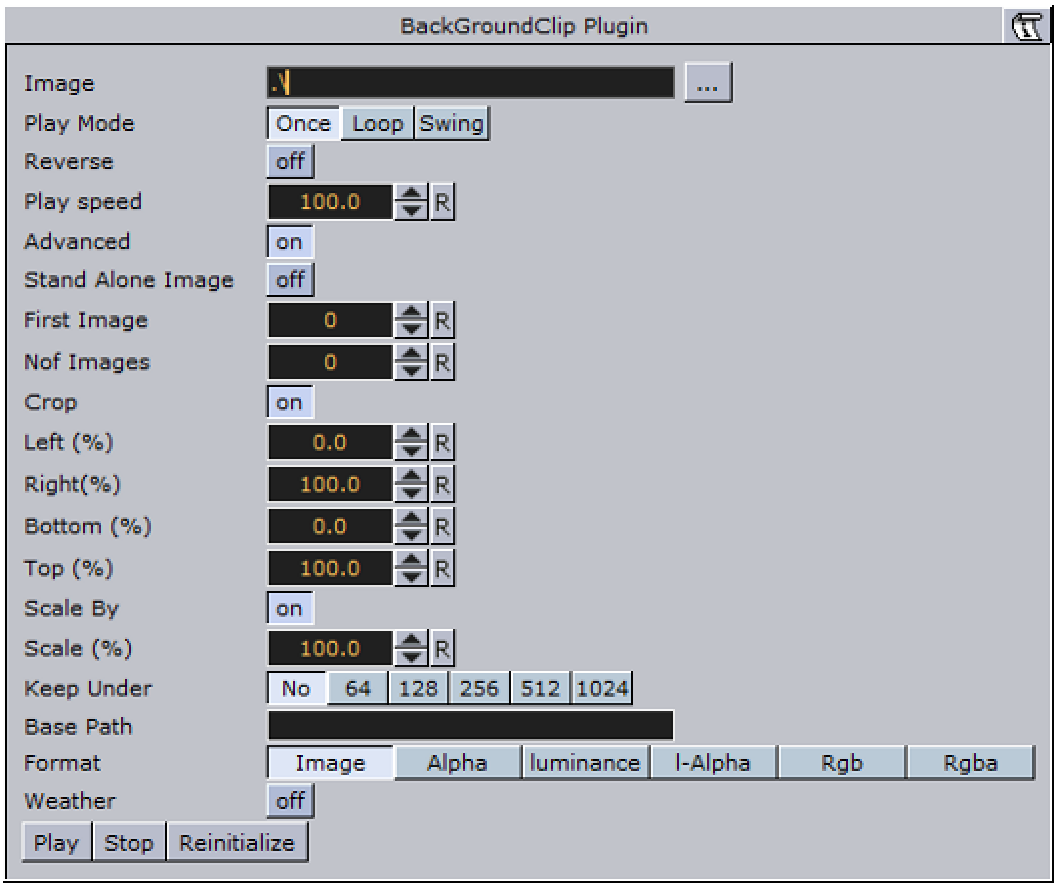
Note: Scene plug-ins are added under the Scene Settings' Plugin tab.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\BackgroundClip.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To animate a virtual camera flight

Properties and Parameters



BackGroundClip Plugin

Image: .\ ...
Play Mode: Once | Loop | Swing
Reverse: off
Play speed: 100.0 R
Advanced: on
Stand Alone Image: off
First Image: 0 R
Nof Images: 0 R
Crop: on
Left (%): 0.0 R
Right(%): 100.0 R
Bottom (%): 0.0 R
Top (%): 100.0 R
Scale By: on
Scale (%): 100.0 R
Keep Under: No | 64 | 128 | 256 | 512 | 1024
Base Path:
Format: Image | Alpha | luminance | I-Alpha | Rgb | Rgba
Weather: off
Play | Stop | Reinitialize

- **Image**: Allows you to enter the name of the image sequence or browse for it. Select the first image you would like to use in your clip. Do not use clip names with numbers (except the counter).
- **Play Mode**: Can be set to Once, Loop (over and over again) and Swing (equal to loop but plays the clip reverse too).
- **Reverse**: Switch from off to on if you want the clip to played from end to start.
- **Play speed**: Lets you alter the speed in percent.
- **Advanced**: Opens advanced parameters if set.
  - **Stand alone image**: Needed when you want to play the same clip in different speeds.
  - **First image**: Select the number of the image which you want to start with.
  - **Nof images**: It stands for “number of images”. Please type in here the number of images you want play as a clip, otherwise all images will be played back.
  - **Crop**: Crops the image in percent from the left, right, bottom and top side.
  - **Scale**: Will scale all images to the closest power of 2. When not selected an automatic textures coordinates will be applied. So the image will fit the texture.
  - **Keep under**: It forces the image size. For example if you have selected 64, the image will be trimmed to the size 64 x 64.
  - **Format**: If the “image” format is selected, the file will be loaded in original condition. Please select another format, if you want load a black and white image (“Alpha”) or full color image (“RGB”, “RGBA”: with alpha value).

- **Play, Stop and Reinitialize**: Enables you to play, stop and reinitialize the clip.

To add a background image clip

1. Create a directory which is including pictures like “abc001.png”, “abc002.png” ... "abc389.png".
   - Please consider that the images will be played as a sequence of the filename including the numbers.
   - After you have created the images, load the first image of the sequence and the clip is now visible in the render output.
2. Another possibility is to create a file with a "*.vln" extension. This file includes the base path and also the names of the images to load. In this case the images must not have a counter number in their filename. You can handle this file as an ordinary text file.
3. Load the "vln - file" instead of loading an image file located in a directory.

Example:

```
BASE_PATH 'C:/clip/images'
{
   'radar_200504110800.png' 2005_04_11_10:00
   'radar_200504110815.png' 2005_04_11_10:15
   'radar_200504110830.png' 2005_04_11_10:30
   'radar_200504110845.png' 2005_04_11_10:45
   'radar_200504110900.png' 2005_04_11_11:00
   'radar_200504110915.png' 2005_04_11_11:15
   'radar_200504110930.png' 2005_04_11_11:30
   'radar_200504110945.png' 2005_04_11_11:45
}
```

See Also

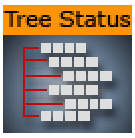
- Implementing Video
- Image Clip

14.3 Lineup

The following scene plugins are housed in the Lineup folder:

- Tree Status

14.3.1 Tree Status



The Tree Status plug-in is needed by the Lineup template in Viz Content Pilot in order to collect information about the scene hierarchy.

Consider that the Tree Props plug-in is required for using the Tree Status plug-in. Move the Tree Props plug-in onto the group which is holding the desired transformation to control. Viz Content Pilot is then able to build its own internal tree properties list.

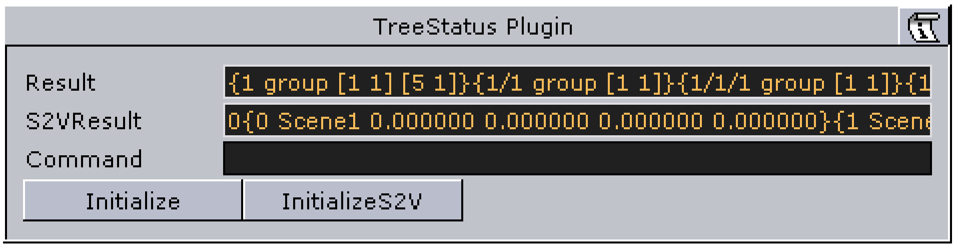
Note: Scene plug-ins are added under the Scene Settings' Plugin tab.

The default path for this plugin is `C:\Program Files\vizrt\Viz3\plugin\TreeStatus.vip`.

This section also contains information on the following topics:

- Properties and Parameters

Properties and Parameters



Result: {1 group [1 1] [5 1]}{1/1 group [1 1]}{1/1/1 group [1 1]}{1

S2VResult: 0{0 Scene1 0.000000 0.000000 0.000000 0.000000}{1 Scene

- **Result**: This field holds the entire information about the complete scene tree.
- **S2VResult**: S2V stands for *scene to virtual*. This parameter holds the information X-, Y-, Z-translation and the Y-rotation given in the card file *virtual set*.
- **Command**: Any Viz command is allowed here to take place.
- **Initialize (button)**: This button initializes the *Result* field.
- **InitializeS2V (button)**: This button initializes the *S2VResult* field.

14.4 Script Plugins

This is an empty folder for you to save your own script plug-ins into.

14.5 Texture

The following scene plug-ins are housed in the Texture folder:

- Graffiti

14.5.1 Graffiti

The Graffiti scene plug-in allows telestration on top of scenes in the screen space. Telestration is done by drawing with a brush shape using a mouse or a 6DOF interface. The plug-in can also recognize some rendered shapes and replace the hand-drawn elements with the recognized shapes (e.g. circle, ellipse, cross, arrow).

The graffiti scene plug-in is not intended for use on video output. To use graffiti textures on video output, use the Graffiti container plug-in.

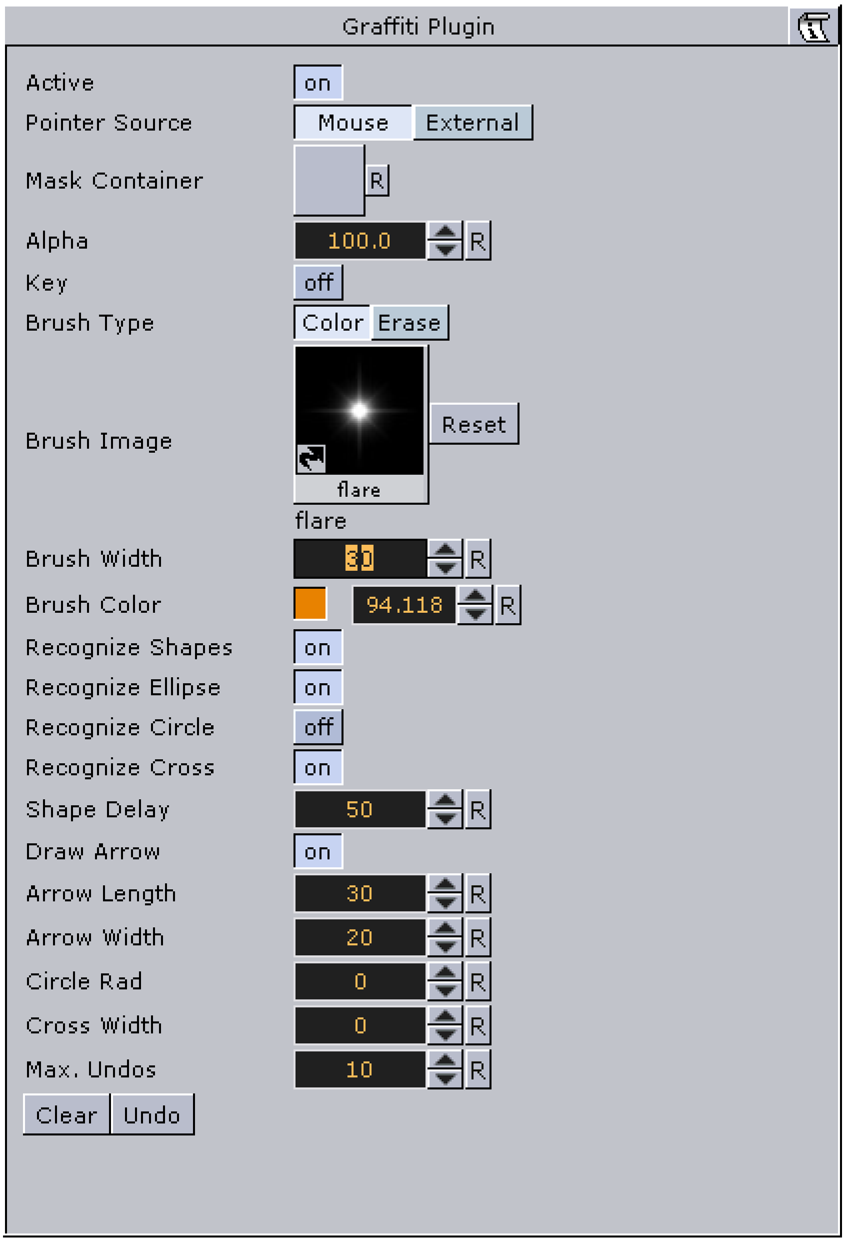
Note: Scene plug-ins are added under the Scene Settings' Plugin tab.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\Graffiti.vip`.

This section contains information on the following topics:

- Properties and Parameters
- To create a scene level graffiti effect

Properties and Parameters



Graffiti Plugin

Active: on

Pointer Source: Mouse | External

Mask Container

Alpha: 100.0

Key: off

Brush Type: Color | Erase

Brush Image: flare (Reset)

flare

Brush Width: 30

Brush Color: 94.118

Recognize Shapes: on

Recognize Ellipse: on

Recognize Circle: off

Recognize Cross: on

Shape Delay: 50

Draw Arrow: on

Arrow Length: 30

Arrow Width: 20

Circle Rad: 0

Cross Width: 0

Max. Undos: 10

Clear | Undo

- **Active**: Enable/disable drawing.
- **Pointer Source**: The plugin supports input from the mouse, or from another plugin via a dispatcher interface. This button selects whether the plugin will listen to the mouse or the dispatcher for input.
- **Mask Container**: This points to either a container or a group of containers. These containers mask the area over which the brush is rendered.
- **Alpha**: Transparency of the brush.
- **Key**: Selects whether the brush is rendered in the key signal
- **Brush Type**: Color or eraser brush.
- **Brush Image**: Optional image that will determine the shape of the brush. If empty, a round brush will be used.
- **Brush Width**: Width of the brush in pixels. Visible only if the color brush is selected.

- **Eraser Brush Width**: Width of the eraser in pixels. Visible only if the eraser brush is selected.
- **Brush Color**: Color of the brush.
- **Recognize Shapes**: Shape recognition mode (on/off).
- **Recognize Ellipse**: Specifies whether shape recognition will try to recognize ellipse shape.
- **Recognize Circle**: Specifies whether shape recognition will try to recognize circle shape.
- **Recognize Cross**: Specifies whether shape recognition will try to recognize cross shape.
- **Shape Delay**: Number of frames to wait from mouse up before trying to recognize shapes.
- **Draw Arrow**: Specifies whether non-recognizable shapes will be converted to an arrow.
- **Arrow Length**: Length of arrow head.
- **Arrow Width**: Width of arrow head.
- **Circle Rad**: Radius of the circle replacing a recognized circle. If zero, the radius of the recognized circle will be used.
- **Cross Width**: Width of the cross replacing the recognized cross shape. If zero, the width of the recognized cross shape will be used.
- **Max. Undos**: Max number of undo operations.
- **Clear (button)**: Clears the canvas.
- **Undo (button)**: Undo.

To create a scene level graffiti effect

- Add the plug-in to the Plug-in pane under Scene Settings, set the plug-in Properties and Parameters, set Viz in On Air Mode and start drawing.

See Also

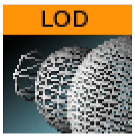
- Plug-in
- Graffiti (container plug-in)

14.6 Tools

The following scene plugins are housed in the Tools folder:

- Level Of Detail (LOD) Manager

14.6.1 Level Of Detail (LOD) Manager



The LOD Manager scene plug-in works in conjunction with the Level Of Detail (LOD) container plug-in which is rendering objects with different details depending on the camera range; however, an integration with the LOD Manager is not required as the LOD plug-in is capable of initializing the parameter values by itself. The LOD Manager plug-in is used to change the visual view of the complete scene during rendering depending on the camera and scale settings.

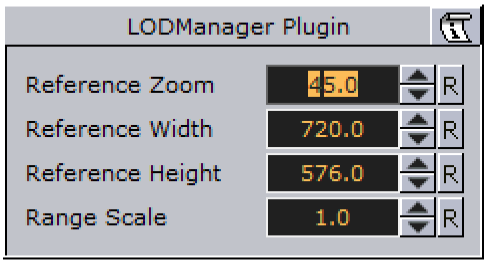
Note: Scene plug-ins are added under the Scene Settings' Plugin tab.

The default path for this plugin is `C:\Program Files\Vizrt\Viz3\plugins\LODManager.vip`.

This section contains information on the following topics:

- Properties and Parameters

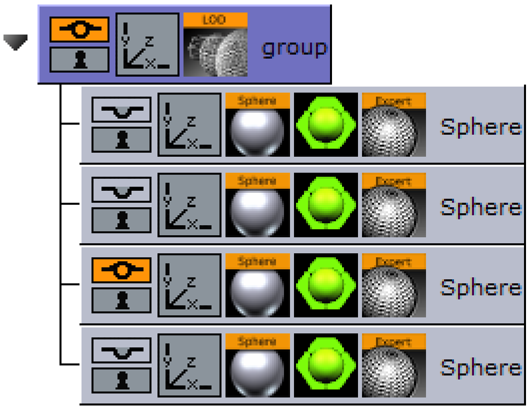
Properties and Parameters



Reference Zoom 45.0
Reference Width 720.0
Reference Height 576.0
Range Scale 1.0

- **Reference Zoom**: Sets the reference zoom that is used to correlate switching ranges based on the actual camera zoom value.
- **Range Scale**: Sets the global range scale which is used to scale all LOD ranges on all LOD plug-ins in the scene.
- **Reference Width**: Not used.
- **Reference Height**: Not used.

To use the LOD Manager plug-in



1. Start by adding the LOD Manager scene plug-in under the Scene Settings' Plugin tab.
2. Add a group container to your scene tree and add at least two new sub containers of the group container.
3. Add a sphere geometry to the sub containers as it has many triangles.
4. Add a material to each sub container.
5. Open the Sphere editor for the all sub containers and set different Tesselation values for each to change the number of triangles and therefore the level of detail of an object.

Note: All spheres should, for the sake of example, be placed in the same position.

6. Add the Level Of Detail (LOD) plug-in to the group container.
7. Open the LOD editor and change the Range parameter values provided by the LOD plug-in. For example set Range 0 to 500, Range 1 to 1000 and so on.
   - These settings affects the instant of time when an object 1, object 2 will be rendered, which you have placed in step 3.
   - The high resolution object is rendered at first and if the camera to object distance is increasing the next object will be rendered because now a low-resolution object has near the same visual quality despite fewer triangles. Consider that the visual quality depends on your settings and object graduation.
8. Finally change the LOD Manager parameters Reference Zoom and Range Scale to adjust the scene view additionally.

15 RealFX Plugins

The RealFX plugin set enables you to create particle effects in Viz Artist.

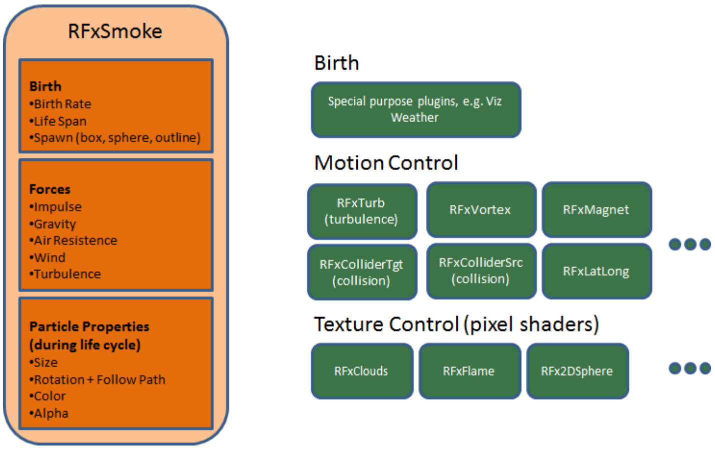
Particle systems are a computer graphics technique to simulate certain physics-based effects, which are otherwise very hard to reproduce with conventional rendering techniques. Examples of such effects which are commonly replicated using particle systems include fire, explosions, smoke, weather effects, sparks, falling leaves, dust, meteor tails, or abstract visual effects like glowing trails, magic spells, etc.

The particle effects in Viz run in real-time, meaning that there are a few inherent constraints that must be taken into account when considering best practices for employing this plugin set. For example, there is a trade-off between the number of particles and performance optimization; more generally there needs to be a considered balance between performance and visual quality.

RFxSmoke is the baseline plugin within the RealFX plugin set. The remaining plugins in this set are applied on top of RFxSmoke in any given container. RFxSmoke includes built-in functionality and the ability to host the additional functionality contained in the other plugins in this set. Part of the built-in functionality, e.g. turbulence, is kept for compatibility with previous version of Viz Artist.

There are three categories of additional plugins. Birth plugins refer to where the particles are spawned. Motion control plugins govern the position, direction, velocity, size and color of each particle. Texture control plugins affect the texture mapping and the “look” of each particle by using pixel shader technology.

Figure 176: RealFX Architecture



RealFX plugins are located in the **Built-ins** menu, in the **RealFX** folders in groups under the **Container Plugins** and **Shaders** tabs.

This section contains information on the following topics:

- RFxSmoke
- RFxTurb

- RFxVortex
- RFxMagnet
- RFxLatLong
- RFxVortex
- RFxColliderSrc
- RFxColliderTgt
- RFxClouds
- RFxFlame
- RFx2DSphere

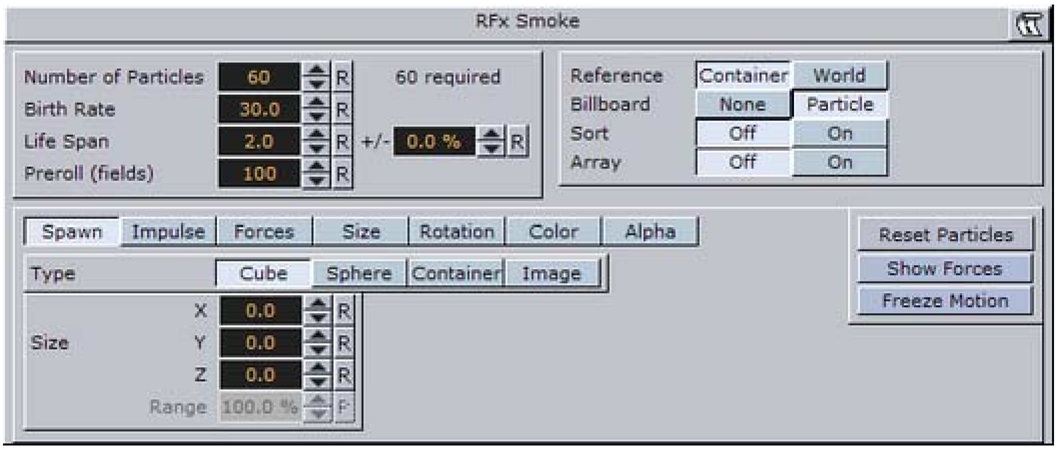
15.1 RFxSmoke

RFxSmoke is the baseline plugin within the RealFX plugin set. It allows you to create realistic visualization of smoke, fire, explosions and much more (the level of creative skill is the only limitation). The basic function of the object is that it emits particles from a point and moves them in some direction until they die or fade away. By using different texture mappings and settings many effects can be achieved.

This section contains information on the following topics:

- Properties and Parameters
- Spawn
- Impulse
- Forces
- Size
- Rotation
- Color
- Alpha

15.1.1 Properties and Parameters



RFx Smoke

Number of Particles 60 R 60 required
Birth Rate 30.0 R
Life Span 2.0 R +/- 0.0 % R
Preroll (fields) 100 R

Reference: Container | World
Billboard: None | Particle
Sort: Off | On
Array: Off | On

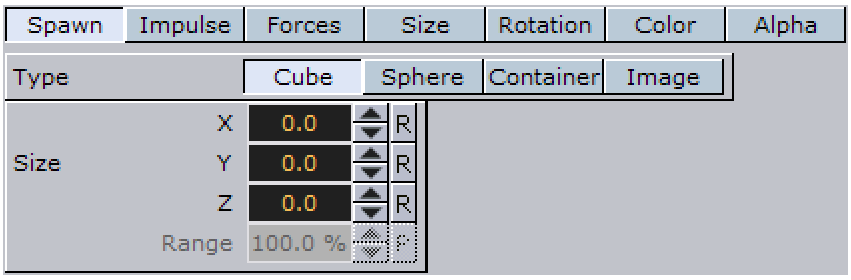
Spawn | Impulse | Forces | Size | Rotation | Color | Alpha

Type: Cube | Sphere | Container | Image

Size
X 0.0 R
Y 0.0 R
Z 0.0 R
Range 100.0 % F

- **Number of particles**: Sets the number of particles to be active at the same time. The number of particles is the product of the **Birth Rate** and the **Life Span**, so the value should normally not be set higher than that. To the right of the input value field, you see “required” which is Birth Rate*Life Span. If the value is set much lower than the required, the particles will create puffballs instead of continuous smoke.
- **Birth Rate**: Sets the birth rate of particles per second.
- **Life Span**: Sets the life span of the objects in seconds. The field to the right of Life Span, allows you to set a percentage, over which the life span of the particles will vary. If you have set **Life Span** to 3 and the variation to 50%, the life span of particles will vary randomly between 1,5 and 4,5 seconds. This will help create more realism, since particles in real smoke or fire does not have equal life span.
- **Reference**: Allows you to choose the particles reference to either **Container** or **World**. If **Container** is chosen, after being emitted the particles will relate their position to the containers position. If you move the emitter object, the particles will alter their position correspondingly. If **World** is chosen, the particles will maintain their position and path even if the emitter is moved, they have their own world coordinates.
- **Billboard**: If you enable billboarding for the particles, the particles will maintain a frontal position against the camera when it moves.
- **Sort**: Lets you switch sorting on/off.
- **Array**: Enables/disables the use of array.
- **Reset Particles**: Clicking this button removes all the particles already emitted.
- **Show Forces**: Enabling this option displays “help lines” that shows the forces settings of some of the parameters.
- **Freeze Motion**: Halts the emitting process.

15.1.2 Spawn



Opens a section of parameters related to the position of the smoke emitter. You can define the shape and size of the area in which the emitter should emit the particles.

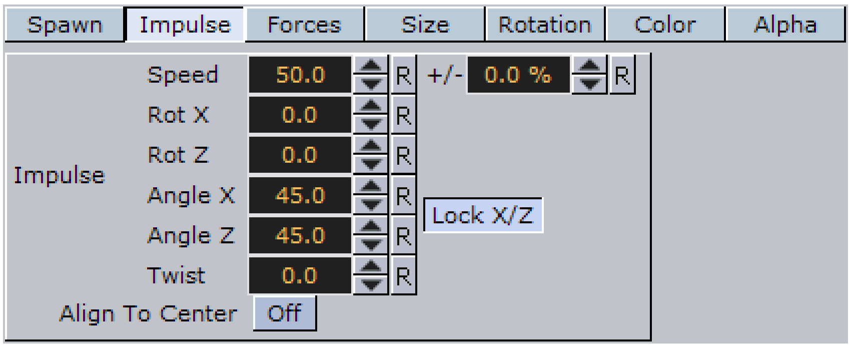
- **Type**: The buttons **Cube**, **Sphere**, **Container** and **Image** allows you to set whether the spawn area should be shaped as a **Cube** or a **Sphere** or if a container or image should be used.
- **Size**: The **X**-, **Y**- and **Z**-values sets the size of the selected form. If you have selected **Sphere** and you set only the X-value, the emitter will create particles while it moves randomly along a line. If you set an Y-value in addition, the emitter will move within a circle, and finally if you set a Z-value, within a sphere. If you have selected a **Cube** the same applies but the stages would then be a line, a rectangle and a Cube.
- **Spawn Range**: Sets the range within the sphere or cube, the emitter should use for distributing the particles. Default is 100% where it uses the whole spawn area. If you reduce the value, a section starting from the middle of the spawn area, where no particles are being emitted, will increase correspondingly. If the value is set to 1%, the emitter only emits particles at the outer edge or surface of the cube or sphere.

Container is especially valuable when animating a scene as the smoke will be emitted from the outline of an animated container. To set the container you want the smoke to be emitted from, drag it from the scene tree onto the drop zone.

Note: A container must be 2D, such as a rectangle, circle, square, etc.

To learn more about animation see Creating Animations.

15.1.3 Impulse

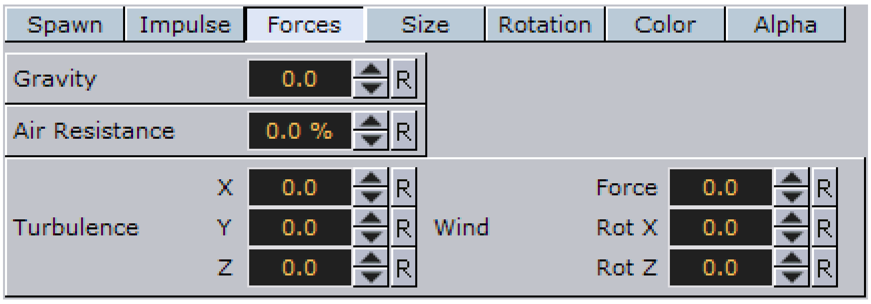


Impulse
Speed 50.0 +/- 0.0 %
Rot X 0.0
Rot Z 0.0
Angle X 45.0
Angle Z 45.0
Twist 0.0
Align To Center Off

Opens a sub section where you can set parameters on how particles are emitted.

- **Speed**: Sets the speed the particles have when the are being emitted. The field to the right of **Speed**, allows you to set a percentage over which the particle speed will vary
- **Rot X**: Sets the X-axis angle of the “opening” in the emitter, the emitter hole.
- **Rot Z**: Sets the Z-axis angle of the “opening” in the emitter, the emitter hole.
- **Angle X**: Sets the X-axis opening angle of the emitter hole. It can be done locked with the Z-axis, or independently. The button to the right of the value field enables/disables locking of the axes.
- **Angle Z**: Sets the Z-axis opening angle of the emitter hole.
- **Twist** creates a twist of the opening angels of the emitter hole. It can also be described as a rotation of the emitter around the Y-axis.
- **Align to Center**: Disables all the above impulse parameters and aligns the particles above the emitter.

15.1.4 Forces



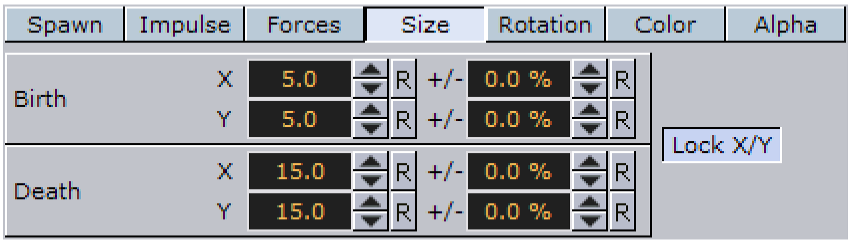
Gravity 0.0
Air Resistance 0.0 %
Turbulence X 0.0
Turbulence Y 0.0
Turbulence Z 0.0
Wind Force 0.0
Wind Rot X 0.0
Wind Rot Z 0.0

Opens a sub section with parameters for creating environmental effects, like wind, gravity and air resistance.

- **Gravity**: Sets the degree of gravity affecting the path of the particles. If it is set high, the smoke will go downwards. With a negative gravity the particles will rise faster.
- **Air Resistance**: Sets the degree of air resistance. It creates a force that, to a level you define, prevents the particles from rising.
- **Turbulence X**, **Y** and **Z**: Allows you to define a simulation of turbulence on the particles as they rise. The turbulence effect is achieved through a jittering of the particles. The **X**-, **Y**- and **Z**-values set the axis of the jittering movement.

- **Wind Force**: Sets a level cross wind. A positive value makes it blowing from the left to towards the right, a negative the other way around. Use the two next parameters to adjust the angle of the wind, if you do not want a horizontal wind force.
- **Wind Rot X**: Sets the rotation of the wind force on the X-axis. Altering this value and the value below, will change the direction of the wind defined in the **Wind Force**.
- **Wind Rot Z**: Sets the rotation of the wind force on the Z-axis

15.1.5 Size

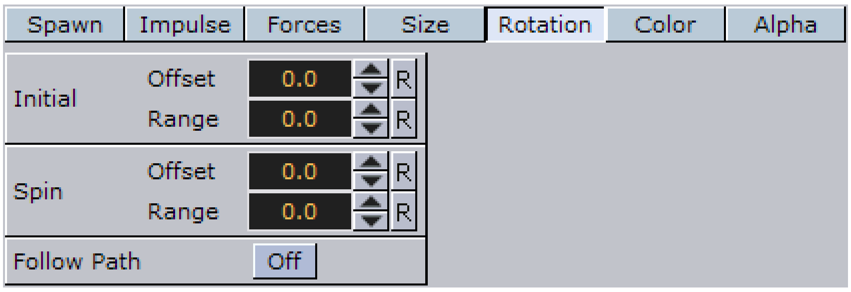


Birth X 5.0 +/- 0.0 %
Birth Y 5.0 +/- 0.0 %
Death X 15.0 +/- 0.0 %
Death Y 15.0 +/- 0.0 %

Opens a section with parameters to the size of the particles at the moment of their birth and death. Between the time of birth and death, the size is a product of a linear interpolation.

- **Size Birth X** and **Y**: Sets the size of each particle at the time it is being emitted.
- **Size Death X** and **Y**: Sets the size of each of the particle at the time it fades away and dies.

15.1.6 Rotation



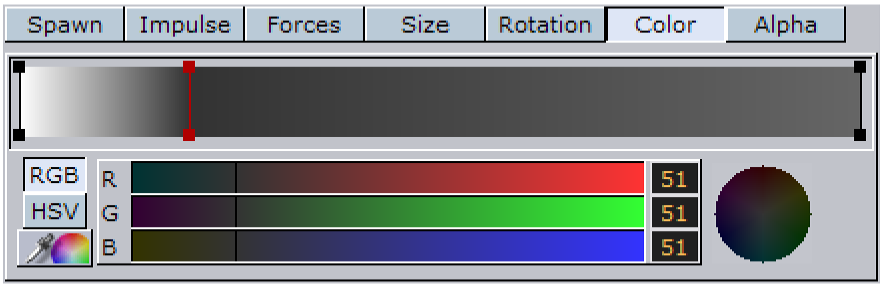
Initial Offset 0.0
Initial Range 0.0
Spin Offset 0.0
Spin Range 0.0
Follow Path Off

Opens a section where you can define a static rotated position or a spin of the particles. Both the static rotated position and the spin can either be set as an offset equal on all particles or as a range where the particles will be rotated or span randomly. To easily see the effect, set the number of particles to a very low value and alter the rotation parameters.

- **Initial Offset**: Sets the degree of rotation the particles should have. This rotated position is static and will remain throughout the life of each particle.
- **Initial Range**: Sets a degree range of rotation the particles should have. This rotated position is static and will remain throughout the life of each particle. If for example the value 30 is set, the static rotated position of the particles will vary randomly between 30° and -30°.

- **Spin Offset**: Sets the degree of spin each particle should have throughout its life time. The higher the value, the higher the number of spins. A positive value creates an counterclockwise rotation, a negative creates a clockwise.
- **Spin Range**: Sets a spin degree range for the particles. They will spin randomly between the parameter value and the same value mirrored through zero.
- **Follow Path**: If the RFxSmoke object is animated, enabling this option will make the particles align their position to the animation path in the same way as is the case with the follow path option in the transformation editor.

15.1.7 Color

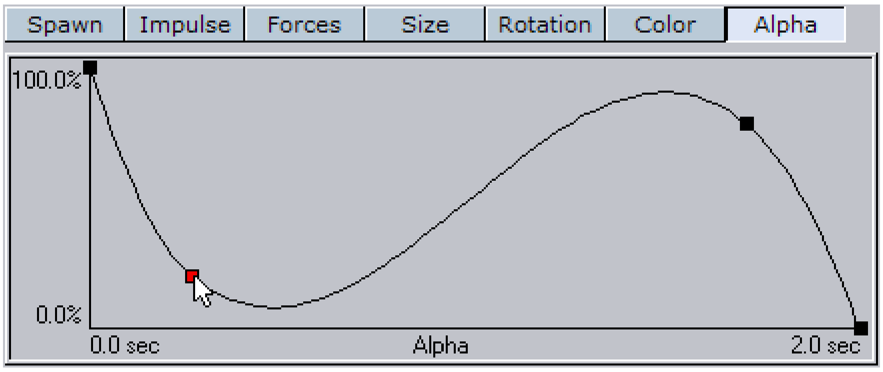


Allows you to set the color of the particles throughout their life cycle. The particles can be set to change their color through stages from their birth to their death by adding more keyframes (single click the color stage).

The left side of the color stage is at birth time and the right is at death time. By default there are two color keyframes at each end that you can click on to set their color. The color range between two keyframes will be an interpolated transformation. More keyframes can be added to create a more complex color transformation between particles birth and death.

Delete color keyframes by selecting them and pressing the DELETE or Backspace button on the keyboard.

15.1.8 Alpha



Allows you to set the alpha values of the particles throughout their life cycle. The alpha editor contains a curve with points that you can move in both X and Y directions to change the form of the curve. The X-axis is the life time of the particles and the Y-axis is the alpha value.

Select a point, or create a new one by left-clicking on the graph, and drag the point to the desired X and Y position. Delete a point by selecting it and pressing the DELETE button on the keyboard.

15.2 RFxTurb

Located under the Container Plugins tab, RFxTub applies turbulence-like forces to particles. It can be used to create the effect of random changes in wind force and direction.

Adjust the following parameters as required to achieve the desired effect:

- Amount
- Inertia
- Wavelength
- Update Rate

See Also

- RFxSmoke

15.3 RFxVortex

Located under the Container Plugins tab, RFxVortex applies vortex-like forces to particles. It can be used to create the effect of a tornado-like twister.

Adjust the following parameters as required to achieve the desired effect:

- Amount
- Inertia
- Center X
- Center Y
- Center Z
- Rot X
- Rot Y

See Also

- RFxSmoke

15.4 RFxMagnet

Located under the Container Plugins tab, RFxMagnet creates motion patterns that resemble a magnetic field.

Adjust the following parameters as required to achieve the desired effect:

- Amount
- Center X
- Center Y
- Center Z
- Power Law
- Min Distance Clamp

See Also

- RFxSmoke

15.5 RFxLatLong

Located under the Container Plugins tab, RFxLatLong fits any other RealFX motion onto a globe surface. The effects are mainly geographic and large-scale.

Adjust the following parameters as required to achieve the desired effect:

- Radius
- Spawn Min Long
- Spawn Max Long
- Spawn Min Lat
- Spawn Max Lat
- Update from Globe (on/off)
- Spawn Min Alt
- Spawn Max Alt
- Orientation (Horizontal/Radial)
- Trim Radius

See Also

- RealFX Plugins

15.6 RFxColliderSrc

Located under the Container Plugins tab, RFxColliderSrc is used in conjunction with RFxColliderTgt to create a collision and collision detection system. For example the source could be a notional wall or floor, whereas the target would be the particles themselves. The particle target can work with a number of sources.

Note: When RFxCollisionSrc is added to a container, the RFxCollisionManager plugin is automatically added at the scene level. This plugin is used internally and not configurable.

Adjust the following parameters as required to achieve the desired effect:

- Object Type (Auto, Sphere, Box)
- Bounce
- Random
- Proximity Fade and Fade Range: Allows you to fade particles based on their proximity to other objects (only spheres and boxes).
- Flag 1–8 (on/off)

See Also

- RFxSmoke
- RFxColliderTgt

15.7 RFxColliderTgt

Located under the Container Plugins tab, RFxColliderTgt is used in conjunction with RFxColliderSrc to create a collision and collision detection system. For example the source could be a notional wall or floor, whereas the target would be the particles themselves. The particle target can work with a number of sources.

Adjust the following parameters as required to achieve the desired effect:

- Flag 1–8 (on/off)

See Also

- RFxSmoke
- RFxColliderTgt

15.8 RFxClouds

Located under the Shader tab, RFxClouds create a dynamic plume-like texture on each particle, and changes randomly for each particle over time It can be used on clouds and smoke.

Adjust the following parameters as required to achieve the desired effect:

- Brightness
- Turb Scale X
- Turb Scale Y
- Turb Speed

See Also

- RFxSmoke

15.9 RFxFlame

RFxClouds create a dynamic flame-like texture on each particle, and changes randomly for each particle over time It can be used on water and fire.

See Also

- RFxSmoke

15.10 RFx2DSphere

RFxSphere creates a 3D sphere-like look on each particle. The sphere is affected by light sources.

Adjust the following parameters as required to achieve the desired effect:

- Color Mode (Material, Color, Color-Material)
- Radius

- Blend

See Also

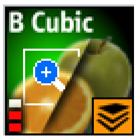
- RFxSmoke

16 PixelFX Plugins

Note: The PixelFX plugins are a separate package from the standard Viz Artist installation, but function within the Viz Artist environment.

PixelFX affects pixels and geometry, as well as produces various special effects. It does three main things: color correction, lens flares, and other pixel-based effects, such as noise, gradients, various distortion effects, transitions, and other such effects of that nature.

On each of the plugin icons, there are mini icons that indicate certain properties of the plugin in question. On the left, there is a performance bar, with three blocks, akin to a cellphone battery indicator. One block means the plugin won't have a drastic performance effect, and three means the plugin is taxing and should therefore be used with caution.



The orange stack one on the bottom right indicates whether the plugin is stackable, meaning whether it can be used in conjunction with color correction plugins.

Some plugin icons themselves are divided in half with one side showing the visual effect with the plugin and the other without it.

Some of the PixelFX plug-ins are functional only when applied directly on a container that has a texture on it. The rest however, function on the Renderer's pixel buffer. For example, you may apply color correction to the root container in your scene, therefore perform a color correction to the entire scene. Another example is pxNoise - apply it to the root level container of a scene or an entire tree branch to add a kind of film-grain effect.

One of the parameters that feature in a number of the PixeFX plugins is Seed. This defines the pseudo-random rule for changed parameters. Once a value is chosen, although the original state will be random, the effect will actually look the same on other computers as well.

The PixelFX plugins are divided into the following groups:

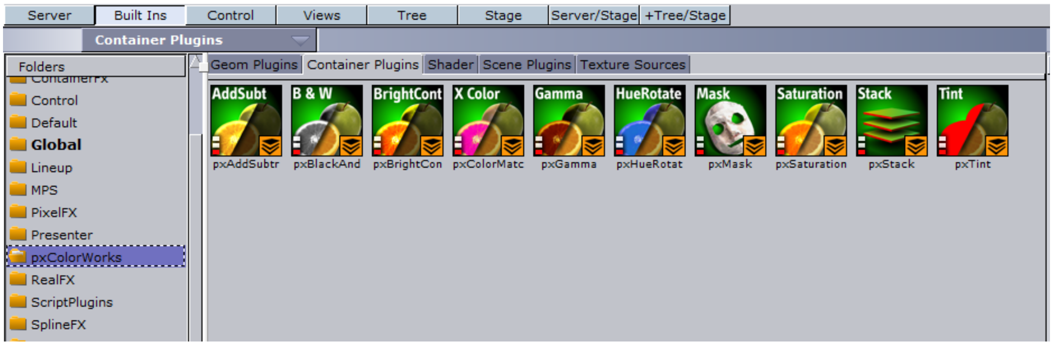
- Color Correction Plugins
- Lense Flare Plugins
- Shader Plugins

16.1 Color Correction Plugins

The color correction plugins are generally fast and efficient, however there is a certain performance penalty when animating them in a stacked scenario.

The PixelFX color correction plugins can be found through the following click sequence:

- **Built-ins** -> **Container Plugins** ->**pxColorWorks** folder.



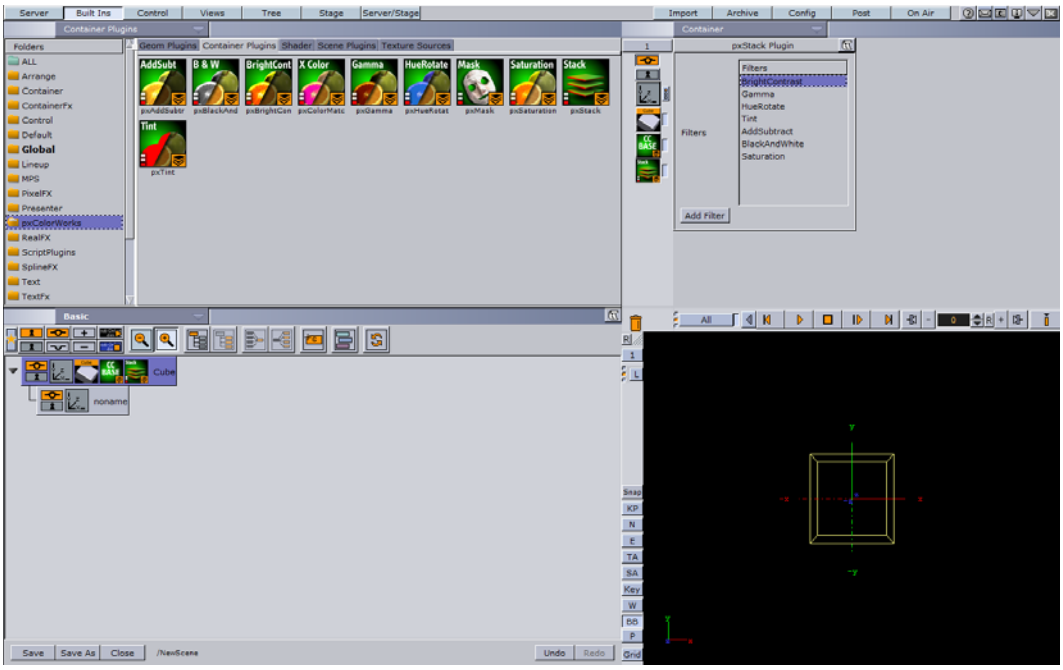
This section contains information on the following plugins:

- pxStack
- pxMask
- pxAddSubtract
- pxBlackAndWhite
- pxBrightContrast
- pxColorMatch
- pxGamma
- pxHueRotate
- pxSaturation
- pxTint

16.1.1 pxStack

To apply to the pxStack plugin

1. Add the pxStack plugin to the desired container in the scene tree.



Note: Upon dropping the stack plugin on a container, it automatically adds to the container the CC BASE plugin. Also, it automatically generate a child container to host the color correction nodes you will define later.

2. Click the pxStack icon in the container.

   The Color Correction Plugins will be displayed in the plugin editor as a list of color correction functions from which you can choose to achieve your desired result.
3. Stack up color correction functions you need to have by selecting them and then clicking **Add Filter**.

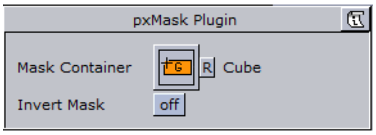
   Each color correction function you apply is added into as a container underneath the 'noname' container in the scene tree. We call it a 'Color Correction Node'.
4. Click on the desired color correction nodes and pop the editor to manipulate the effect properties.
5. Change the order of the stacked properties, as required.

IMPORTANT! Selected properties act in the sequence in which they are ordered in the scene tree.

- Disable/enable a node by simply turning it off/on (top left icon in container).
- To remove a color correction function, delete the plugin.

16.1.2 pxMask

pxMask allows you to supply a mask for the color correction operation. The mask is defined by a black and white image that is dragged into the pxMask plugin parameters. pxMask, as indicated by the stackable icon, can function in conjunction with a mask.



pxMask Plugin

Mask Container G R Cube

Invert Mask off

16.1.3 pxAddSubtract

pxAddSubtract allows you to add or subtract a constant color value to each pixel. It can be applied to Everything, Shadows, Midtones and Highlights.

16.1.4 pxBlackAndWhite

pxBlackAndWhite allows you to grayscale an RGB image or a tree branch holding geometry. It can be applied to Everything, Shadows, Midtones and Highlights.

16.1.5 pxBrightContrast

pXBrightContrast allows you to adjust the brightness and the contrast on each of R, G and B on a given image or a tree branch holding geometry. It can be applied to Everything, Shadows, Midtones and Highlights.

16.1.6 pxColorMatch

pxColorMatch lets you replace one color with another. It works on images and tree branches holding geometry alike.

Some important parameters:

- **Weight**: Defines the color range proximity to the selected color.
- **Inertia**: Defines the smoothness of the range proximity graph.
- **Count**: Defines the number of color replace pairs you wish to use.

16.1.7 pxGamma

pXGamma allows you to adjust the gamma correction on each of R, G and B on a given image. It can be applied to Everything, Shadows, Midtones and Highlights.

16.1.8 pxHueRotate

pxHueRotate allows you to adjust the hue rotation on an image. It can be applied to Everything, Shadows, Midtones and Highlights.

16.1.9 pxSaturation

pxSaturation allows you to adjust the color saturation on an image. It can be applied to Everything, Shadows, Midtones and Highlights. You also have the option to preserve highlights, or not.

16.1.10 pxTint

pxTint allows you to add a color tint and apply it to an image in the desired amount. It can be applied to Everything, Shadows, Midtones and Highlights. You also have the option to preserve highlights, or not.

16.2 Lense Flare Plugins

Lense flare plugins simulate the effect of streaks and spots of light, caused in real cameras when light enters and bounces inside a camera lense. This plugin set contains geometries that simulate different shapes, which can be used for lens flare or simple flare effects.

The flare draws itself as a combination of geometry shader and pixel shader. You will notice a non-editable pixel shaders appear on a container automatically once the flare container is created.

In addition, the Expert plugin is automatically added and configured in such a way as to cause the shape used to always be seen in front of all other objects, blending with the rest of the scene to resemble the effect of light bouncing off the lens.

Lens Flares

Note: Lens flare plugins reside in their own containers. Applying these plugins to a container with geometry will overwrite the geometry and disable any shaders in child containers.

The baseline plugin within this set is the pxLensMulti plugin, which is a container plugin, whose purpose is to manage other geometry plugins. The one exception in this set is the pxLensEnergyBolt, which does not simulate lens flare, but rather the effect of electricity or lightning and it is not managed by pxLensMulti.

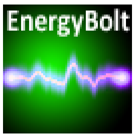
The PixelFX lens flare plugins can be found through the following click sequence:

- **Built-ins** -> **Shader** ->**PixelFXLensFlare** folder.

This section contains information on the following topics:

- pxLensEnergyBolt
- pxLensRays
- Lens Flare Geometries
- pxLensMulti

16.2.1 pxLensEnergyBolt



pxLensEnergyBolt produces the effect of electricity-like lightning between two points in space. These points may be defined manually by entering positions, or by tracking the positions of other containers. The parameters of Radius, Core, End Caps, Distortion and N define the effect's look and Speed defines change rate.

16.2.2 pxLensRays

pxLensRays can be used to produce the effect of rays. The rays can be static or move randomly based on the Speed parameter. The shape is defined with the following parameters:

- Angle
- Width
- Size
- Inner radius

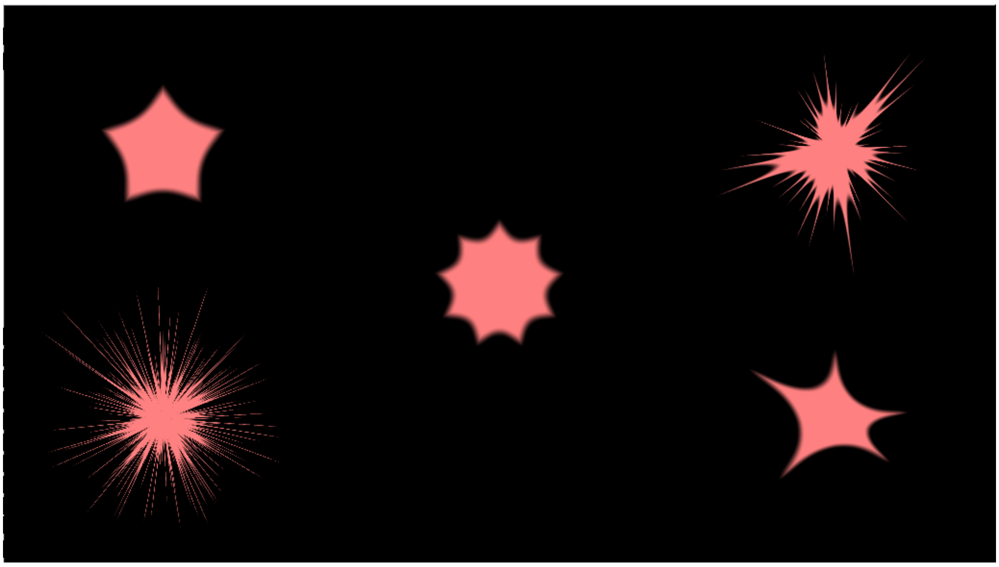
The number or density of the rays is defined by the Rays parameter. Rays can receive their color from a predefined or user defined gradient ramp, or from an image, by dragging the image into the Image box.

16.2.3 Lens Flare Geometries

The following lens flare plugins are simply different shapes for the composition of final lens flare effects. Each has its own parameters to define its appearance.

- pxLensGlowBall
- pxLensGlowSpikes
- pxLensPolyElement
- pxLensRandomPoly
- pxLensSpark
- pxLensStripes
- pxLensDistort

Each plugin can in its own right take on very different appearance depending on how the parameters are adjusted. The best advice for working with these plugins is to simply play around with the parameter configurations. The following example show varying manifestations of the pxLensRandomPoly.



16.2.4 pxLensMulti

pxLensMulti serves as a parent container for all the other lens flare plugins. It simplifies the process of animating lens flares so that they resemble realistic behavior. In the child containers any number of lens flare shapes can be placed. pxLensMulti positions, colors, governs the opacity of all according to the configurable parameters.

The lens flare shapes at origin are a flat geometry. pxLensMulti ensures that all the shapes it handles are constantly facing the camera.

Note: pxLensMulti creates a notional line along which all shapes are scattered.

Properties and Parameters

Under the Spread tab, the following parameters can be configured to handle shapes' position:

- **Mode**: In Polar mode, the notional line can be defined by angle and radius originating from the screen's center. In Cartesian mode, the line is created between the screen's center and given an x-y location in the screen's coordinates. In Container mode, you need to drag a container from the scene tree into the Container box; the container imitates the light position so that the notional line starts from that position and goes through the screen's center.

Note: In Container mode, one point of the notional line is in the Viz coordinates and the second point, through which the notional line passes, is defined in the screen coordinates. As such, camera movement in Container mode actually defines the shapes' movement. See Animating Cameras.

- **Distance**: The notional line passes through the screen's center. The distance of this center point from the camera is defined by the this parameter's value.
- **Count**: The quantity of each shape in each child container is multiplied by the value entered for this parameter.
- **Radius Scale**: Scales the notional line from the origin onward.
- **Radius Rand**: Scales the notional line's length, regardless of origin.
- **Single Seed**: When set to On, only one seed is used for each shape in the child containers. When set to Off, a different seed is used for each shape in the child containers.
- **Seed**: Defines the pseudo-randomality rule for randomly changed parameters. Once a value is chosen, although the original statement will be random, the effect will actually look the same on other computers as well.

Under the Attributes tab, the following can be configured to handle non-positional parameters:

- **Amount**: Determines the amount of light used in the shapes.
- **Amount Rand**: Randomizes the Amount parameter between different shapes.
- **Size Rand**: Randomizes the sizes between different shapes.
- **Color Rand**: Randomizes the color between different shapes.

16.3 Shader Plugins

Viz does not support more than one general shader on a container, or in a nested fashion. However, in some cases, PixelFX, by its nature, does support multiple shaders per container.

Some have high cautionary indications. Some are very light. Some are stackable, and others aren't.

The PixelFX shader plugins can be found through the following click sequence:

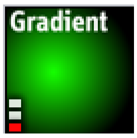
- **Built-ins** -> **Shader** ->**PixelFX** folder.

This section contains information on the following plugins:

- pxGradient
- pxTurbDissolve and pxTurbWipe
- pxEqualize
- pxInvert
- pxLensDistort
- pxMotionBlur
- pxNoise
- pxPixelate
- pxPosterize

- pxRecolor
- pxRipple
- pxSparkle
- pxTurbulence
- pxTwirl
- pxWaves

16.3.1 pxGradient



It can be applied on a container that holds, a geometry, image, or even an empty container.

The pxGradient plugin has a similar functionality to the gradient function in Photoshop. You can define as many gradient stops as you wish, as well as their origin points, angles, and extent. You can also define the gradient on the alpha channel (256 levels) and introduce turbulence noise that will comply with the defined gradient.

At the top of the plugin parameters, you see the controls that let you define the direction, radius and origin of the gradient. Below that you see a two radio button sets. The first lets you set the gradient type, followed by three Repeat types

Below that, you see the color slider, the color values and alpha ramp settings.

Note: Under the title Color Ramp and Alpha Ramp, you have an active/inactive option, should you want to have an RGB ramp and no alpha ramp, or vice versa.

Below the RGB and Alpha area of the editor, you find the option of adding Turbulence to the gradient. The turbulence has three simple parameters. Amount determines how aggressive the turbulence looks. Wavelength determines how big or small the turbulence increments are. Progress (similar to that in TextFX plugins) allows you to breathe life into the turbulence, providing interesting an effect when animated.

At the very bottom, it is important to ensure that you turn off the Show GUI button, whose default is On; if shown on air, the effect will be more embarrassing than dramatic.

Best Practices

It is important to understand that the Gradient plugin is actually drawing pixels. Therefore if this plugin is applied to an empty group or if it is applied to a geometry object that does not contain a texture, the gradient plugin will generate the texture into which it will draw its pixels. So if you have a cube and add the Gradient to it, it behaves exactly as a texture. However it can also work in conjunction with a texture. If you then take an image and drop it on the cube, you give a color wash to the image that works in a gradient-like fashion. The gradient-generated texture can also be edited via the texture editor as any other texture would be.

The biggest benefit of using the Gradient is conservation of texture memory. Instead of storing very many different alpha and color ramps in your image pool, you can simply achieve the overwhelming majority of these effects by well thought out and judicious use of the Gradient plugin. It has a minimal performance penalty and it can be animated. If you use ramps often in your designs, the Gradient is one of your handiest tools.

Tips and Tricks

Often you put reflection maps on a translucent geometry to create a glass-like reflection (think of a light box or an aquarium). For the reflection texture, one often uses an image of static noise or ramps of sorts. Since the texture is mapped on the object using a reflection mapping mode, it is very difficult to create the right intensity of noise in your texture so that the resulting reflection looks good. The Gradient turbulence functionality allows you to set the intensity of the noise of the texture “on the fly”, minimizing the round trips to Photoshop.

16.3.2 pxEqualize

pxEqualize allows you to adjust the black, mid-range and white color levels in an image.

16.3.3 pxInvert

Often you create an image with an alpha channel in Photoshop, import it to Viz and only then realize that your alpha channel is inversed. Or you find an image in the Viz pool that you want to use, but the way you want to use it is where the RGB is actually the alpha channel.

pxInvert allows you to perform these operations without adding yet another image to your image pool, or overloading your scenes with a few versions of the same image, where the difference is simple flips of the channels.

This plugin allows you to invert each of the color channels RGB and Alpha individually, and to use the RGB as alpha and the alpha as RGB.

16.3.4 pxLensDistort

pxLensDistort creates a “fish eye” lens distortion effect on an image. It can also be used to create the effect of an old television with a curved screen.

The amount of the effect can be configured using the following parameters:

- Amount
- 2nd Order
- Scale

The center around which the distortion goes can be configured by using the X Center and Y Center parameters.

16.3.5 pxMotionBlur

pxMotionBlur applies motion blur to the texture. Supports Color Correction Plugins and pxStack.

Adjust the following parameters as required to achieve the desired effect:

- X Center
- Y Center
- Rotate
- Zoom
- X Shift
- Y Shift
- Fix Edges (on/off)
- Samples (enter a value)

16.3.6 pxNoise

pxNoise is a white noise generator. It deploys the same turbulence as seen in other PixelFX plugins. Similarly to the pxGradient, it generates its own texture if applied to a container without an image, or blends with the existing texture if the container has an image applied.

Adjust the following parameters as required to achieve the desired effect:

- **Size**
- **Amplitude**: Intensity/contrast of grains
- **Color**
- **Speed**: Turbulence noise
- **Stretch**: Stretch the grain to achieve a rain look of noise
- **Stretch Angle**
- **Movement**: Transforms the generation of noise

- **Transform as Texture - On**: Conform to texture coordinates. **Off**: Ignore texture coordinates and behave as stencil mapping.
- **Apply to: RGB, RGBA, Alpha**: The channels in which the noise will be generated.

16.3.7 pxPixelate

Similar to the pixelate function in Photoshop, the pxPixelate effect appears as if you reduce the resolution of your image. You may work in a proportional (locked) or disproportional mode.

To achieve a nice animated transition, tweak the Smoothness parameter so that the change between the different pixelization levels will be smoother or harsher.

Adjust the following parameters as required to achieve the desired effect:

- Lock X/Y (on/off)
- X Size
- Y Size
- Smoothness

Supports Color Correction Plugins and pxStack.

16.3.8 pxPosterize

pxPosterize recolors an image using only a limited number of colors for a 'posterize' effect. You can adjust the number of levels and set the transition to one of Off, Fade or Glow.

16.3.9 pxRecolor

pxRecolor 'false-colors' an image by using either Red, Green, Blue or Luma as the input to a color ramp.

16.3.10 pxRipple

pxRipple creates a pond ripple effect. Supports Color Correction Plugins and pxStack.

Adjust the following parameters as required to achieve the desired effect:

- X Center
- Y Center
- Radius
- Amount
- Wavelength
- Progress

16.3.11 pxSparkle

The pxSparkle plugin generates a sparkle pattern as a ray emitting from a point. The plugin editor provides options to control the density of the rays, the x,y position from where they are emitted, the brightness (Amount) of rays, as well as other parameters, which will be covered specifically in context. Before delving into these parameters however, it is important to understand the pxSparkle, like other PixelFX shaders (pxGradient, pxNoise) generates its own pixels. As well, it has the ability to be applied on top of an image and can blend with the image’s pixels. When the pxSparkle plugin is applied to a container without an image, it creates sparkle results. However you can also apply it on a container with an image. In this case, you need to decide whether the sparkle pattern should blend on top of the image or whether it cuts the image. If the Cut Image option is on, the host image is seen only in areas that are brighter than zero. The brighter the sparkle, the more apparent the host image will be.

The pxSparkle plugin can be used as a flare to achieve its effect in a radius (the size of the hot spot). The inner Amount (the brightness of the hot spot) and the radius will crop or fade the sparkle radially. You can also so the Speed parameter to set the sparkle in motion.

- Transform as texture: This option forces the rendering of the sparkle to ignore the viewmatics – the sparkle will always be facing the eye-point. This option is handy when applying the sparkle on top of a hierarchy.
- Aspect: This option allows you to set the proportion of the sparkle independently of the texture transformation. Often you stretch textures to get what you want. However when applying the pxSparkle plugin to a distorted texture, you may want to maintain its roundness. Therefore you can use the Aspect parameter to compensate for the texture distortion.

- Angle: This lets you rotate the sparkle without manipulating the texture coordinates. This option is handy when you have a static image on top of which you want to apply a rotating sparkle.

16.3.12 pxTurbulence

pxTurbulence creates a distortion effect using a Perlin turbulence function. Supports Color Correction Plugins and pxStack. Adjust the wavelength, amplitude and speed to achieve the desired effect.

16.3.13 pxTurbDissolve and pxTurbWipe

Both pxTurbDissolve and pxTurbWipe utilize the native multi-texturing support of Viz Artist 3.x.

pxTurbWipe includes the parameters of Angle and Softness.

To apply TubDissolve and/or TurbWipe

1. Insert an image into the scene tree.
2. Go to the image's texture editor and ensure that the texture Unit is set to 1 and the Inheritable option is active.
   This will be your Image A in an A-B transition paradigm.
3. Place a child image underneath the first one.
   This will be Image B.
4. In Image B's texture editor, set the Unit to 2 and ensure that the Inheritable option is inactive.
5. Apply the Dissolve plugin to the Image A (the parent image).
6. In the plugin's editor, first adjust the Transition, followed by the other parameters, as per your liking.

See Also

- pxGradient

16.3.14 pxTwirl

pxTwirl creates a distortion effect that twists an image around a central point. Supports Color Correction Plugins and pxStack.

Adjust the following parameters as required to achieve the desired effect:

- X Center
- Y Center
- Radius
- Amount
- Edge Softness

16.3.15 pxWaves

pxWaves creates a distortion effect that makes an image look wavy. Supports Color Correction Plugins and pxStack.

Adjust the following parameters as required to achieve the desired effect:

- Type: Sine, Noise, Triangle, Square
- Angle
- Amount
- Wavelength
- Progress
- Seed

16.3.16 pxBCubic

When dealing with images that contain line art or lines of high contrast, you may want to protect yourself from the less than the desirable effect of pixelization when the image is over scaled. The B Cubic plugin introduces three bi-cubic algorithms. Each is useful in different contexts of its respective type. It is very easy to find the best adjustment to your specific scenario.

17 Texture Sources

Texture Sources are housed in the following folders:

- Default

17.1 Default

The following texture sources are housed in the Default folder:

- Dynamic Scene
- Dynamic Texture
- Clip Channel
- Video Channel
- Viewport Tile

17.1.1 Dynamic Scene

The Dynamic Scene is a texture source that is able to render a different scene, local scene or parts of a local scene into a texture and display the texture in the current scene being rendered.

Showing a different scene as “a window” in your current scene can be used in a virtual set scene where you would like to “open a window” (as a texture) and render a different scene (e.g. a weather forecast).

You can also use it to show parts of a local scene if you want to animate the camera and at the same time show static elements in the scene. The animating camera can be a different camera (two) and be used to display dynamic images in camera one’s view. This way you can show more then one camera at the same time.

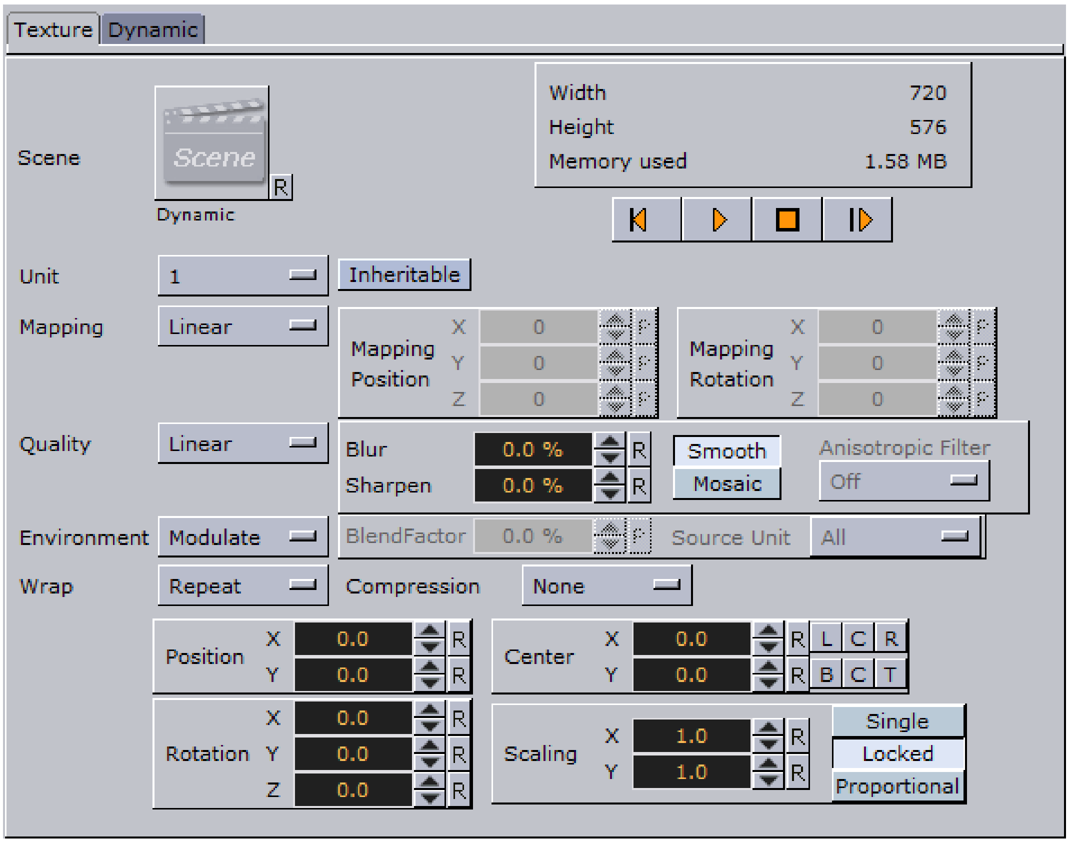
For visual effects, the entire scene or parts of it can be grabbed into an image, and have shaders applied to the entire scene or parts of it.

In order to transition from one scene (A) to another (B) both scenes are rendered into dynamic images and a third scene is used to hold both dynamic images to create a transition between them.

This section contains information on the following topics:

- Properties and Parameters

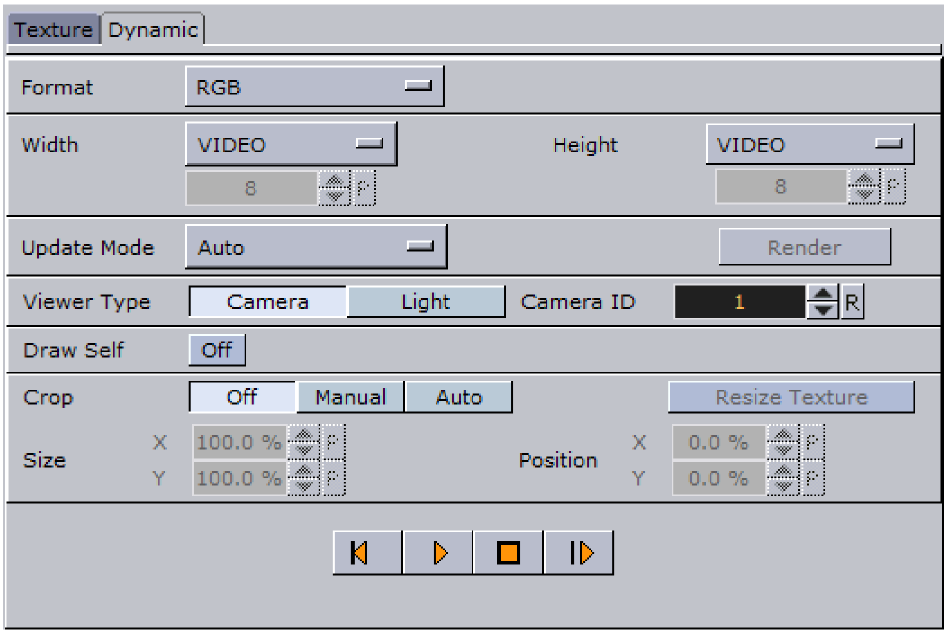
Properties and Parameters



Width 720
Height 576
Memory used 1.58 MB

Unit 1
Mapping Linear
Mapping Position X 0 Y 0 Z 0
Mapping Rotation X 0 Y 0 Z 0
Quality Linear
Blur 0.0 %
Sharpen 0.0 %
Anisotropic Filter Off
Environment Modulate
BlendFactor 0.0 %
Source Unit All
Wrap Repeat
Compression None
Position X 0.0 Y 0.0
Center X 0.0 Y 0.0
Rotation X 0.0 Y 0.0 Z 0.0
Scaling X 1.0 Y 1.0

- **Scene**: Placeholder for the rendered scene. If it is empty, the current scene will be rendered.



Format RGB
Width VIDEO 8
Height VIDEO 8
Update Mode Auto
Viewer Type Camera
Camera ID 1
Draw Self Off
Crop Off
Size X 100.0 % Y 100.0 %
Position X 0.0 % Y 0.0 %

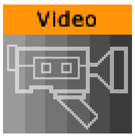
- **Format**: Sets the format to be used for the dynamic image. If the format has alpha key it must be used on the render scenes or it will appear as black.

- **Width** / **Height**: Sets the width and height of the image. The Video option will take the current output size. When selecting Custom you may set any size; however, it is always better (when possible) to use power of 2 images.
- **Update Mode**: Sets when the scene in the dynamic image will be rendered.
  - **Change**: When anything changes in the scenes it will be rendered.
  - **Always**: It will be render each frame regardless if there was any change.
  - **Command**: When the render command will be sent the scene will be rendered.
  - **Grab**: Renders the scene on command and can then be copied to another image.
- **View Type**: Sets the view port to Camera or Light. If Camera is selected and Camera ID is set to zero (0), it will take the current camera. If Light is selected it will use the light according to the Light ID.
- **Draw Self**: Creates an endless mirror effect.
- **Crop**: Enables different crop modes.
  - **Off**: The entire view port will be rendered.
  - **Manual**: Enables you to set the size and position of the rendered scenes.
  - **Auto**: The size and position of the rendered scene will be set by the bounding box holding the dynamic scene (acts as a mask).
- **Size** / **Position**: Sets the size and position of the dynamic image.
- **Resize Texture**: Should the image be resized automatically to reflect the size of the scene being rendered.
- **Back**/**Play**/**Stop**/**Forward**: Controls the playback of the scene.

See Also

- Texture Editor

17.1.2 Dynamic Texture



Dynamic Texture is a feature that generates dynamic textures on-the-fly. It can be used as an alternative to importing textures to save texture memory usage. Another advantage of using a dynamic texture is that you will have a texture that is not pixelated as you zoom in or out of it.

See Also

- Texture Editor
- Dynamic Scene
- Image FX and Noise options for the Trigger tab.
- Pixel FX plug-ins (separate plug-in installer)

17.1.3 Clip Channel

Clip channel plug-ins are useful to designers that want to control the clip channels at the container level as a texture.

See Also

- Implementing Video

17.1.4 Video Channel

Video channel plug-ins are useful to designers that want to control the video channels at the container level as a texture.

See Also

- Implementing Video

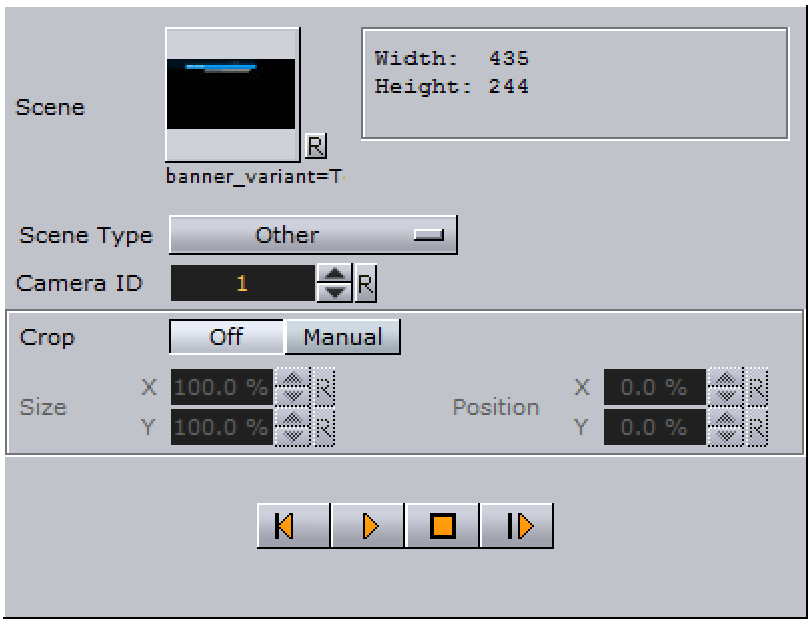
17.1.5 Viewport Tile

Viewport Tile renders a scene into a part of the current viewport without using a texture like Dynamic Scene. The benefit is that the rendering can be done faster but the disadvantage is that the Viewport Tile is always orthogonal to the camera.

This section contains information on the following topics:

- Properties and Parameters
- To render a scene in a viewport tile

Properties and Parameters



Scene

banner_variant=T

```
Width:  435
Height: 244
```

Scene Type: Other

Camera ID: 1

Crop: Off | Manual

Size X: 100.0 %

Size Y: 100.0 %

Position X: 0.0 %

Position Y: 0.0 %

- **Scene**: placeholder for the rendered scene.
- **Scene Type**:
  - **Local**: renders the current scene.
  - **Other**: renders the scene specified in the Scene parameter (just drag and drop a scene).
  - **Main Layer**: renders the scene of the main layer.

- **Camera ID**: select camera for the Viewport Tile scene.
- **Crop**:
  - **Off**: no cropping.
  - **Manual**: define the new viewport size and position with the parameters below.

- **Play controls**: can be used to play, stop, etc. the animations of the Viewport Tile scene.

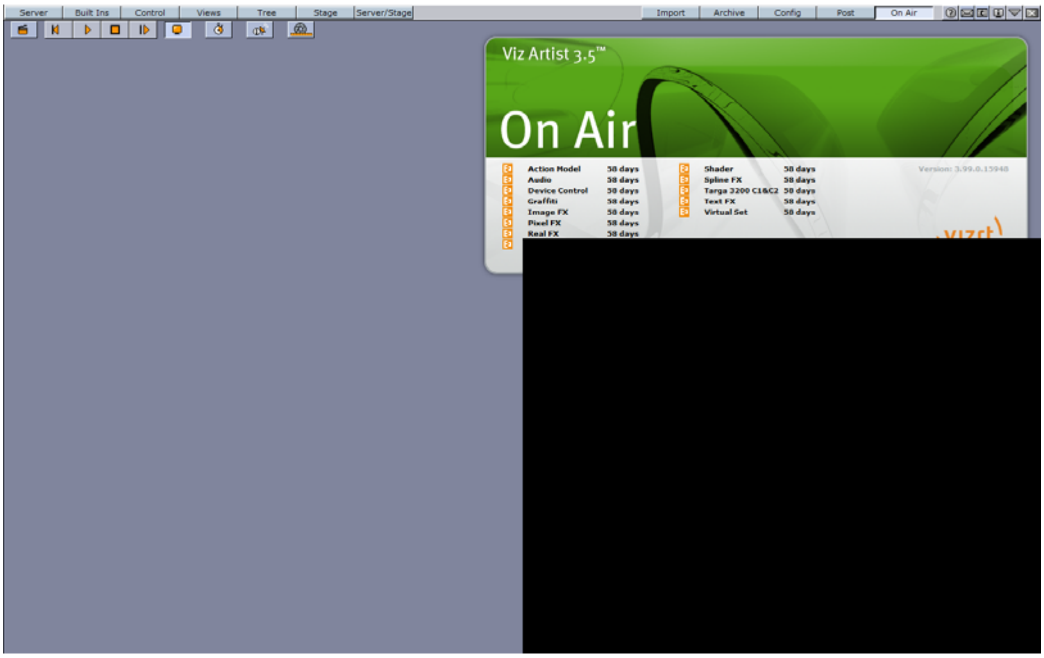
To render a scene in a viewport tile

1. Add the **Viewport Tile** plug-in to the scene tree
2. Open the Viewport Tile editor and add a scene to the **Scene** placeholder
3. Open the **transformation editor** and set the **width** and **height** of the tile
4. Open the Viewport Tile editor and click the **play** button

18 On Air Mode

The On Air interface may vary, depending on the software and hardware configuration used. In Viz Artist, designers can click the On Air button on the main menu to switch Viz Artist from a modeling tool to a render engine. The application will then wait for control commands; however, scene animations can also be rendered by the use of the Control Buttons (top-left corner).

Figure 177: On Air, Viz Engine VGA



Version: 3.99.0.15948

Action Model	58 days	Shader	58 days
Audio	58 days	Spline FX	58 days
Device Control	58 days	Targa 3200 C1&C2	58 days
Graffiti	58 days	Text FX	58 days
Image FX	58 days	Virtual Set	58 days
Pixel FX	58 days		
Real FX	58 days		

The top left of the On Air screen shows a set of Control Buttons, On Air Editor Buttons, as well as a Performance Bar button. All License Information is displayed at the top right of the screen.

Depending on the software and hardware settings, additional buttons and information is available. For example in design mode, the scene will be shown in an output window (lower right).

Note: Be sure to keep Viz Artist running in the foreground as not to disturb the broadcast. Furthermore make sure no window is displayed to overlap the render output as this would interfere with the broadcast.

All script events that are added to a scene or to single containers will be executed in On Air mode. To learn more about scripting see the script documentation.

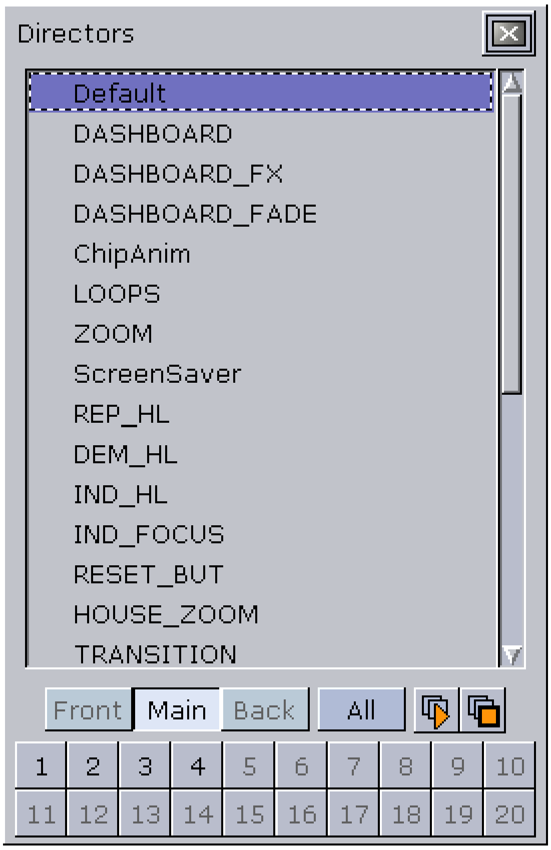
This section contains information on the following topics:

- Director Control Panel
- Control Buttons
- On Air Editor Buttons

- Performance Bar
- System Information
- License Information

18.1 Director Control Panel

Figure 178: Director Control Panel



The Standard-PC version of Viz displays a clapper board button when in On Air mode. This button opens the Director Control Panel window.

The Director Control Panel window can be used to select and animate one, multiple or all directors. In addition it can be used To set slots and to animate a combination of director(s).

This section contains information on the following topic and procedure:

- Properties and Parameters
- To set slots and to animate a combination of director(s)

Properties and Parameters

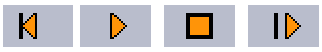
- **Front** - Shows all directors in the Front layer.
- **Main** - Shows all directors in the Main layer.
- **Back** - Shows all directors in the Back layer.
- **All** - Shows and selects all directors.

- **Play** - Plays the animation for the selected director(s).
- **Stop** - Stops the animation for all layers. Right-clicking stops only the selected director.
- **1-20** - Selects the selected director(s) configured for the slot. To set slots and to animate a combination of director(s).

To set slots and to animate a combination of director(s)

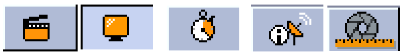
1. Select one, multiple or all directors
2. Drag and drop the director(s) onto a slot (1-20)
3. Click to select the newly configured slot
4. Click the Play button to animate the selected director(s)

18.2 Control Buttons



- **Back**: Jumps to beginning of your scene.
- **Play**: Will start to play your scene.
- **Stop**: Stops the scene.
- **Forward**: Continues to play after stopping.

18.3 On Air Editor Buttons



- **Clapper Board**: Opens the Director Control Panel window.
- **Screen**: Displays the VGA Preview window. VGA Preview must be activated in the Video Output section, and is only available on machines with video cards.

Note: VGA Preview is always enabled for the Standard-PC version of Viz Artist.

- **Clock**: Opens the Performance Bar.
- **System Information**: Opens the System Information screens.
- **Lense File**: Used for adjusting lens files for virtual sets. You can adjust field of view, lense deformation, mobile point and centership. It is useful in combination with lense calibration.

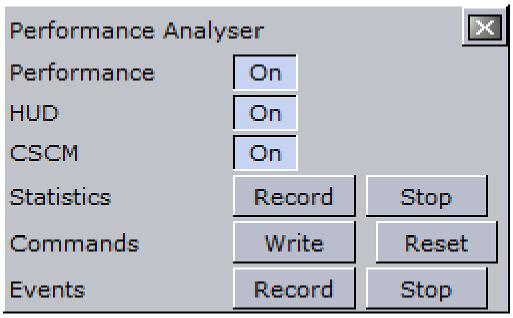
18.4 Performance

Analyzing the performance of Viz can be done using two tools; one is the Performance Bar that allows you to closely monitor a range of parameters for analyzing real-time performance of Viz, the other is the Performance Analyzer tool that allows you to monitor key performance and camera parameters as a head-up display in the renderer window and initiate logging of statistic, command and event information to the log file(s).

This section contains information on the following topics:

- Performance Analyzer
- Performance Bar

18.4.1 Performance Analyzer



The performance analyzer enables key performance and camera information to be displayed in the renderer view as a head-up display. Additionally the performance analyzer can be used to initiate writing of statistic, command and event information to the log files.

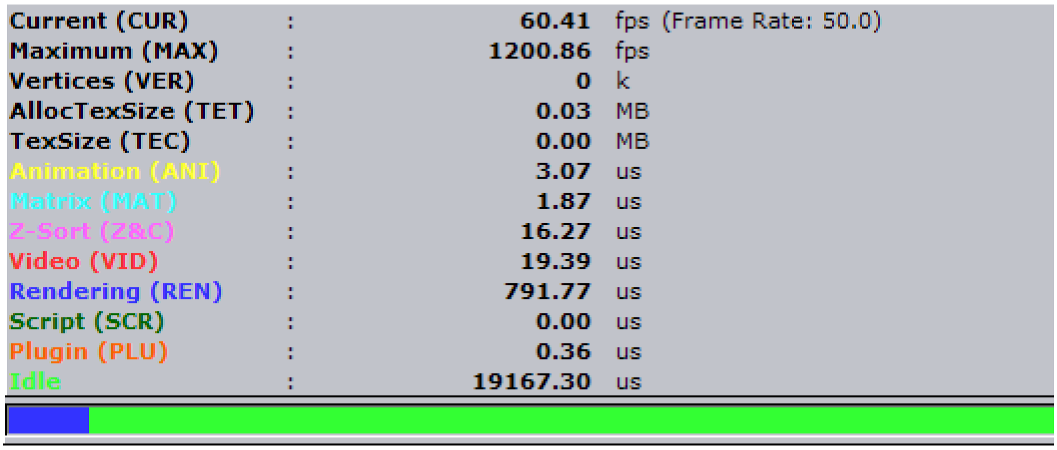
- **Performance**: Displays the current (CUR) and (MAX) parameters (see Performance Bar).
- **HUD**: Enables the head-up display (HUD) showing the following parameters in the renderer view:
  - **Camera 1-n**: Displays the currently selected camera.
  - **Position**: Displays the camera's X, Y and Z position.
  - **Pan/Tilt/Twist**: Displays the camera's pan, tilt and twist parameters.
  - **FovX/FovY**: Displays the camera's field of view (fov) for the horizontal (X) and vertical (Y) plane.
  - **Center Shift**: Displays the X and Y position of the camera's center shift.
- **CSCM**: Displays the center shift as a cross hair in the renderer.

Log files reside under the Viz program folder: `C:\Program Files\Vizrt\Viz3\`.

To open the performance analyzer

- Hold CTRL while clicking the right mouse button on the X (close) button in Viz.

18.4.2 Performance Bar



```
Current (CUR)        :       60.41  fps (Frame Rate: 50.0)
Maximum (MAX)        :     1200.86  fps
Vertices (VER)       :           0  k
AllocTexSize (TET)   :        0.03  MB
TexSize (TEC)        :        0.00  MB
Animation (ANI)      :        3.07  us
Matrix (MAT)         :        1.87  us
Z-Sort (Z&C)         :       16.27  us
Video (VID)          :       19.39  us
Rendering (REN)      :      791.77  us
Script (SCR)         :        0.00  us
Plugin (PLU)         :        0.36  us
Idle                 :    19167.30  us
```

The performance bar gives an idea of the current scene rendering performance (frames per second).

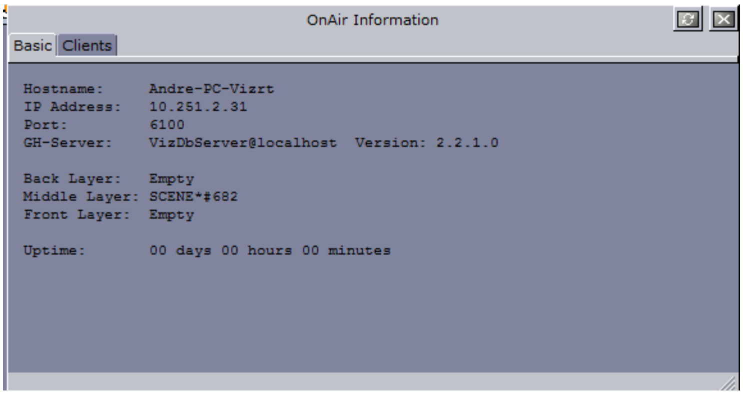
- Current (CUR): Shows how many frames per second the scene will render at in On Air mode. The number should be above 50 (PAL) or 60 (NTSC), according to the rate that has been specified in the Output Format section.
- Maximum (MAX): Shows how many frames per second the scene can render at without waiting for vertical retrace. The higher the maximum value, the more performance is left. If the maximum value is reduced to below 50 or 60, the scene is not rendering in real-time.
- Vertices (VER): Shows the number of vectors in the scene.
- AllocTexSize (TET): Shows the total allocated size of texture memory.
- TexSize (TEC): Shows the size of the currently used texture memory.
- Animation (ANI): Shows how many microseconds all active directors and animation channels take. This indicator is linked to the yellow bar.
- Matrix (MAT): Transforms each container in the scene into world coordinate space. This indicator is linked to the cyan bar.
- Z-Sort (Z&C): Refers to Z-sort and Culling, and sorts all containers for correct transparency drawing and determines if containers are visible in the current camera view. This indicator is linked to the pink bar.
- Video (VID): Shows how many microseconds video input (live video texture) and video output take. De-interlaced video inputs take longer time than progressive and interlaced. The only way to improve this value is to use a faster system. This indicator is linked to the red bar.
- Rendering (REN): Shows how many microseconds it takes to render all objects on the screen. A faster graphics card will improve this value. This indicator is linked to the blue bar.
- Script (SCR): Shows the consumed time in microseconds from all active scripts. This indicator is linked to the dark green bar.
- Plugin (PLU): Indicates how much time in microseconds all active plug-ins spend in each render cycle. This indicator is linked to the orange bar.
- Idle: Shows available resources in microseconds the renderer has available. This indicator is linked to the light green bar.

To open the performance bar

1. Click the performance bar button (timer icon).
2. To see all parameters, extend the view by clicking the Eject button.

18.5 System Information

Figure 179: System information - Basic tab

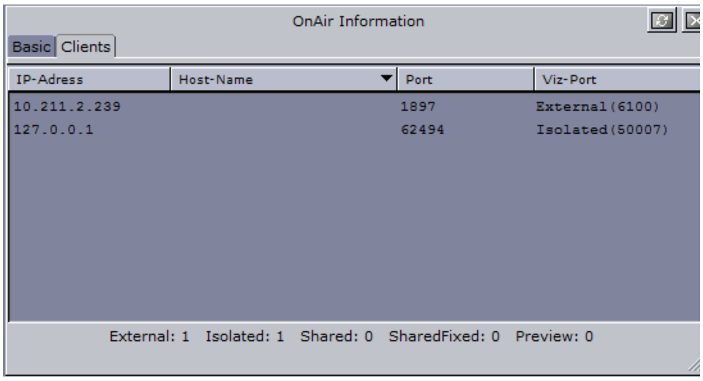


```
Hostname:      Andre-PC-Vizrt
IP Address:    10.251.2.31
Port:          6100
GH-Server:     VizDbServer@localhost  Version: 2.2.1.0

Back Layer:    Empty
Middle Layer:  SCENE*#682
Front Layer:   Empty

Uptime:        00 days 00 hours 00 minutes
```

In the Basic tab, the parameters you need to know for sending external control commands are displayed.

- **Refresh button**: Refreshes the status information.
- **Hostname**: Shows the name external control programs can use to communicate with Viz Artist.
- **IP Address**: Shows the IP address external control commands can communicate with Viz Artist.
- **Port**: Shows the port Viz Artist is using. Default port is 6100, but may be changed in Viz Config's Communication section.
- **GH-Server**: Shows the Viz Graphics Hub server Viz Artist is connected to.
- **Back Layer**: Shows the name of the scene that is defined to run in the background of the middle and front layer scene(s).
- **Middle Layer**: Shows the name of the scene that is defined to run in the middle between the back and front layer scene(s).
- **Front Layer**: Shows the name of the scene that is defined to run in the foreground of the back and middle layer scene(s).
- **Uptime**: Shows the time elapsed since Viz was started.

Figure 180: System information - Clients tab



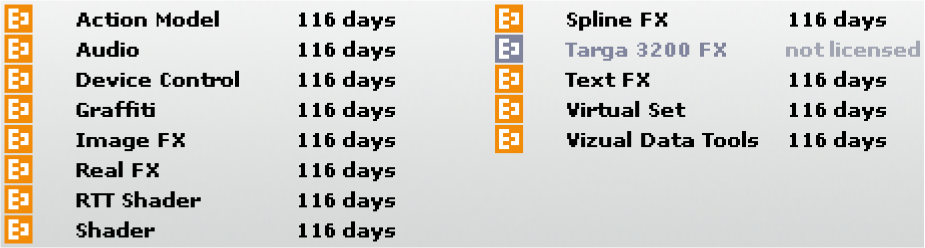
IP-Adress	Host-Name	Port	Viz-Port
10.211.2.239		1897	External(6100)
127.0.0.1		62494	Isolated(50007)

External: 1 Isolated: 1 Shared: 0 SharedFixed: 0 Preview: 0

In the clients tab, all connected clients are displayed with the IP address, host name and Viz Port.

18.6 License Information

Figure 181: Licensed features



Action Model	116 days	Spline FX	116 days
Audio	116 days	Targa 3200 FX	not licensed
Device Control	116 days	Text FX	116 days
Graffiti	116 days	Virtual Set	116 days
Image FX	116 days	Vizual Data Tools	116 days
Real FX	116 days		
RTT Shader	116 days		
Shader	116 days		

The license information listing displays the licensing information such as licensed features and how many days the license has left before it must be renewed.

19 Transition Logic

Transition Logic is first and foremost about:

- Having background graphics that are constantly on air in order to provide an artifact-free display of graphics and animations.
- Having morphing background graphics that are separate from the foreground graphics and animations.
- Having independently controlled multiple layers of graphics on air at the same time.
- Having graphics that will ease the operator’s job of handling multiple layers of graphics.

Transition Logic scenes are designed with a built in logic. A built in logic that can create morphing background graphics, handle multiple layers of graphics and transitioning between variants of each layer. The built in logic also minimizes the operator’s need to know how scenes affect each other as the scene by design controls each layer of graphics and how they affect each other which ultimately creates a look that is consistent and artifact-free.

In order to understand what Transition Logic is, one needs to understand a few basic facts:

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

Note that with Transition Logic scene design, *take in* and *take out* commands are still used as with standalone scene design; however, where standalone scene design demands that only a single scene can be on air at a time, Transition Logic allows for more than one scene to be on air at a time. For example, a graphics covering the lower third of the screen, and another covering the left and/or the right side of the screen (for over the shoulder graphics), can be on air at the same time.

Transition Logic can be played out by most of Vizrt’s control applications such as Viz Trio, Viz Content Pilot and Viz Multichannel.

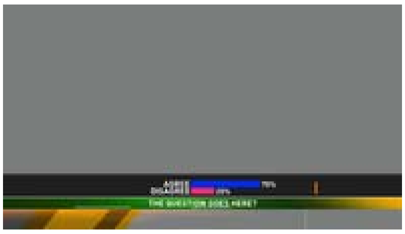
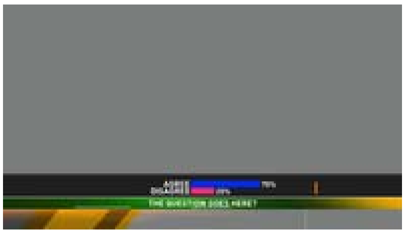
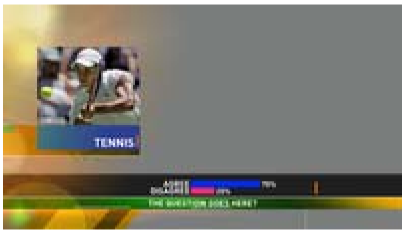
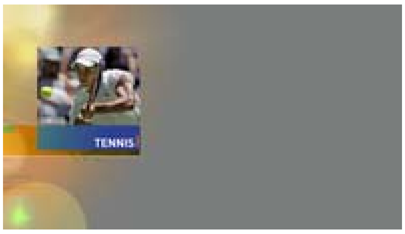
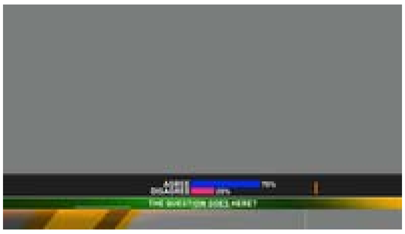
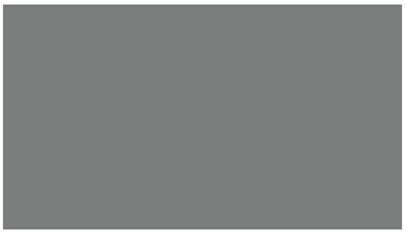
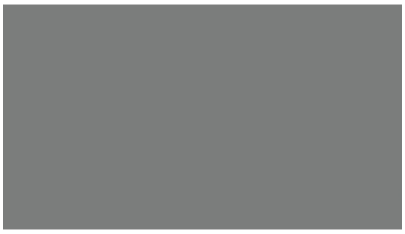
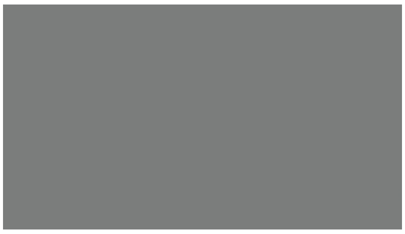
This section contains information on the following topics:

19.1 Standalone versus Transition Logic Scene Design

Unlike standalone scenes, a Transition Logic scene can have multiple layers defined that again can control the transitioning of multiple foreground scenes per layer (e.g. transitioning between variants of lower and top thirds). Having more than one layer will therefore enable a Transition Logic scene to control multiple variants of foreground scenes per layer (e.g. variants of over the shoulder, bug, full screen, lower and top third graphics).

The following table shows the difference in results when executing commands towards Transition Logic scenes compared to standalone scenes. The table shows a Transition Logic scene with two layers and two foreground scenes (over the shoulder and lower third) and two standalone scenes that are identical to the Transition Logic foreground scenes.

Table 27: Scene design comparison

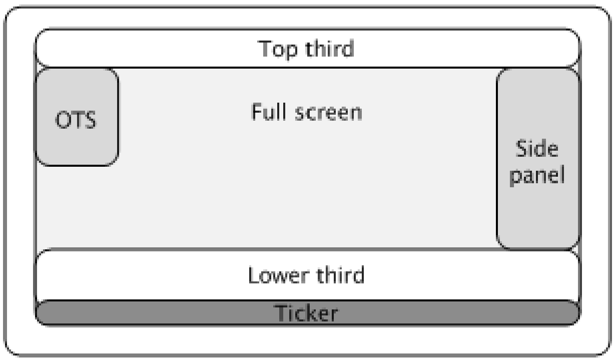
Action	Transition Logic Design	Standalone Scene Design
Take lower third		
Take OTS		
Take Out OTS		
Take Out lower third		

The first action, *take* lower third, results in a lower third graphics being placed on screen. The second action, *take* over the shoulder (OTS), places the OTS on screen; however, with standalone scene design this will also cause the lower third to be taken out (i.e. a hard cut).

The third action, *take out* OTS, will take out the OTS; however, with Transition Logic scene design the lower third will remain on screen until it is taken out (last action).

19.2 Toggle Layer

Figure 182: Layers



A toggle layer acts as a placeholder for one or several graphical elements. Having multiple versions of a lower third on the same layer, in order to change its appearance from one size to another is therefore possible.

Toggle layers offer the possibility to place graphical elements into several logical placeholders allowing multiple layers of graphics to be shown on screen at the same time.

Note: Layer, in Transition Logic terms, does not lend itself to how the term layer is understood with most 2D applications. Layers in Transition Logic refer to the areas on screen the layers reside.

A toggle layer needs to be created for each graphical element (i.e. logical group of foreground scenes) that needs to be on screen at the same time, and all toggle layers must reside in a common scene known as the Background Scene. In order to show multiple layers of graphics on screen at the same time, the background scene must have as many toggle layers as there are logical groups of graphics.

You may place graphics for a top third, lower third and a news ticker in the same logical layer; however, if the news ticker and the lower third needs to be on screen at the same time they have to be in different logical layers. Creating placeholders for logical layers of graphics is done using the built in Toggle plug-in. Hence, the term toggle layer.

Residing in different logical layers also means that their animations may affect each other. For example, having the lower third and news ticker on air at the same time may for instance push the lower third further up the screen to make room for the news ticker.

See Also

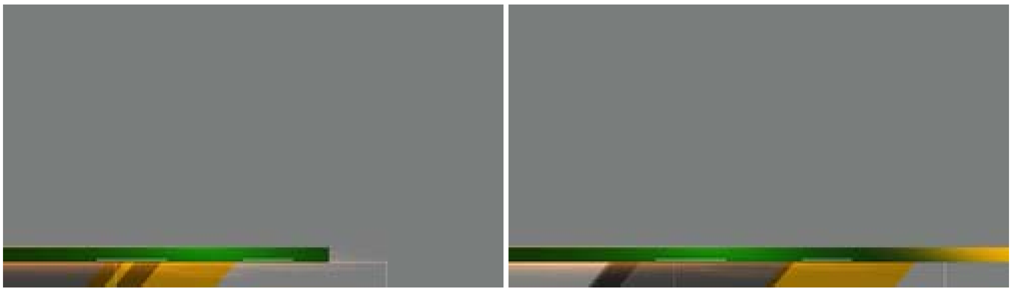
- Background Scene

- Toggle

19.3 State Transition Animation

When building a Transition Logic scene, it is the background scene’s Toggle Layers that are defined to be in different states. Each layer has its own director that defines the states each layer can be in, and each foreground scene refers to a single layer and state.

Figure 183: Toggle layer states



Animating between different toggle layer states is called state transition animation. The example above shows a lower third’s backplate being animated from a narrow version to a wide version (two states).

This section contains information on the following topics:

- Defining States
- Identifying States

19.3.1 Defining States

A toggle layer is required to have a default state named O. The O state is often referred to as the Out state as it is the state the toggle layer will be taken to when the operator issues a Take Out command. Hence, it is, in most cases, also the state where the toggle layer’s referring foreground scene is no longer visible on screen.

In addition to the O state, you may choose to add additional states in order to run animations when taking elements to air. A second state is often referred to as the in state, or visible state, because it most often refers to the sate when a referring foreground scene is shown on screen.

Use the following conventions when defining states:

- It is required to write all O states using upper case only.
- It is recommended to name states according to the referring foreground scene’s appearance.
- It is recommended to write all state names using upper case characters only.

States should be named according to the referring foreground scene’s appearance; hence, a graphics displayed in the lower third of the screen that varies in size from large to small will typically have two states named LARGE and SMALL. A graphics placed over the shoulder of a presenter, where the graphics should switch between the left and right side of the screen, would simply be named LEFT and RIGHT, or OTS_LEFT and OTS_RIGHT (OTS is short for Over The Shoulder).

O → SMALL

Defining two states, for example O and SMALL, will in practice allow the designer to create transition animation that will place a layer off or on screen. Hence, any layer placed off screen will first be in an O state (out), then transitioned to its SMALL state on screen when taking an element that demands the SMALL state, and back to its O state when taken out.

O → SMALL → O

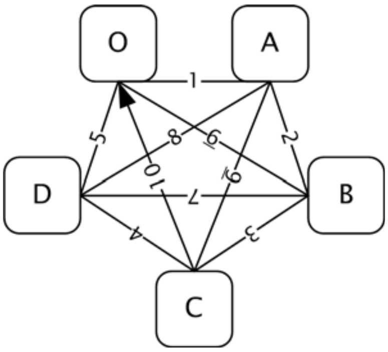
Transitioning from a SMALL to an O state is achieved by using the animation from the O to the SMALL state. In such a case the animation will be played in reverse. If this is not desired, you simply add another O state after the SMALL state, and create a new animation.

O → SMALL → LARGE → O

If the layer should transition from a SMALL to a LARGE state you simply add the state, and create a transition animation for the layer going from a small to a large sized graphics. This also requires that the LARGE state has an O state such that the graphics can transition and animate out from the LARGE state.

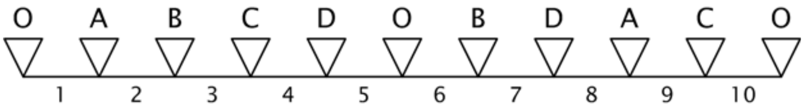
19.3.2 Identifying States

Figure 184: Euler walk



There are no hard limits to the number of states; however, increasing the number of states also increases the number of animations, and consequently the complexity of the scene. To decide the minimum number of animations required it is possible to do a Euler Walk. The Euler Walk, although not always possible, is defined as the optimal route taken in order to visit each point exactly once. However, the theory can be applied to Transition Logic in order to identify the minimum required states needed to connect each state to the others exactly once (see Figure 184: Euler walk).

Figure 185: Layer director



Straightening the Euler walk out onto the layer's director will show the required transition animations.

Note that when performing transition animations from one state to the other Viz will search from left to right on the layer’s director for the correct combination of states. If the operator wishes to go from state A to B, Viz will traverse the layer’s director for state A and look to the right and see if it finds state B. If a match is found, Viz will perform the transition animation.

Looking at the example director above (see Figure 185: Layer director) it becomes clear that not all combinations can be found by traversing from left to right. Hence, if a change from state A to D is not found Viz will, after traversing from left to right, start to backtrack and traverse from right to left looking for a D from A animation.

Having found the minimum required animations a designer will often start to add extra animations. It might be desirable to add a transition animation from state B to O in order to animate the layer out from state B. Adding new out states increases the complexity of a layer as it is still required to have a transition from B to C.

The following table shows the minimum required number of transition animations for a graphics layer with a given number of states.

Table 28: States, transitions and animations

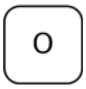
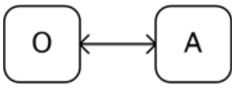
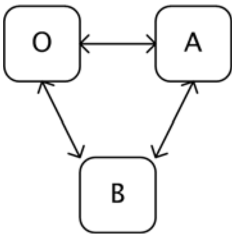
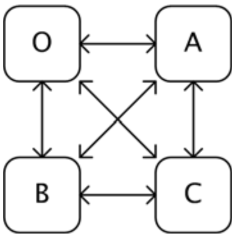
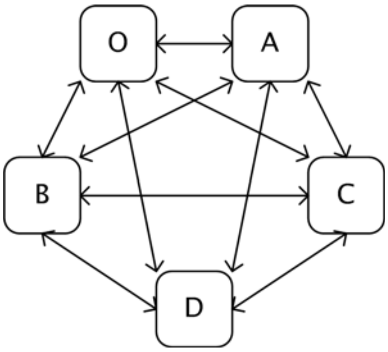
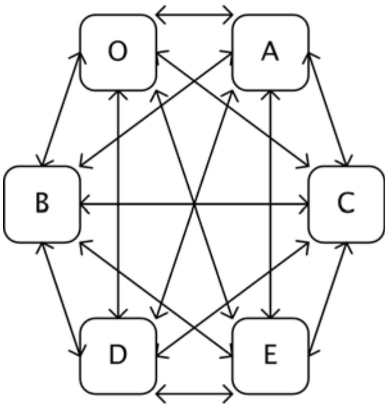
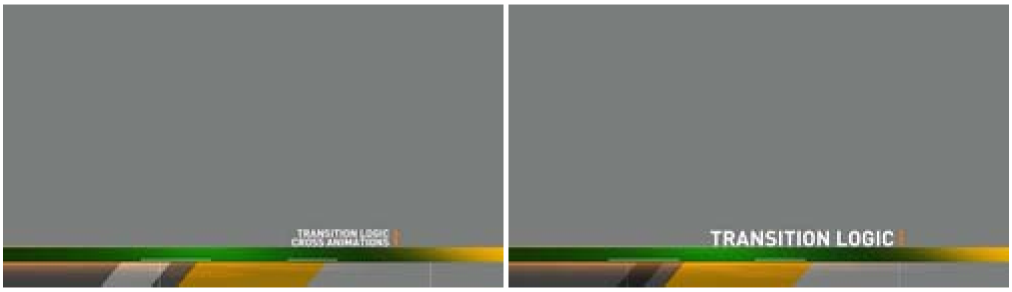
States	Transitions	Animations
1 state		Not possible.
2 states		Requires a minimum of 1 animation
3 states		Requires a minimum of 3 animations
4 states		Requires a minimum of 6 animations

Table 28: States, transitions and animations

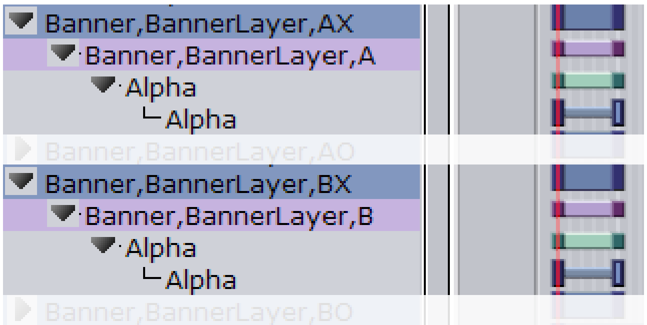
States	Transitions	Animations
5 states		Requires a minimum of 10 animations
6 states		Requires a minimum of 15 animations

19.4 Cross Animation



Cross animations are used in Transition Logic scene design in order to animate one object in and another out. The example above shows a lower third that has an alpha cross fade fading from a two lines of text to a one line of text.

Figure 186: AX and BX directors



Banner,BannerLayer,AX
Banner,BannerLayer,A
Alpha
Alpha
Banner,BannerLayer,AO
Banner,BannerLayer,BX
Banner,BannerLayer,B
Alpha
Alpha
Banner,BannerLayer,BO

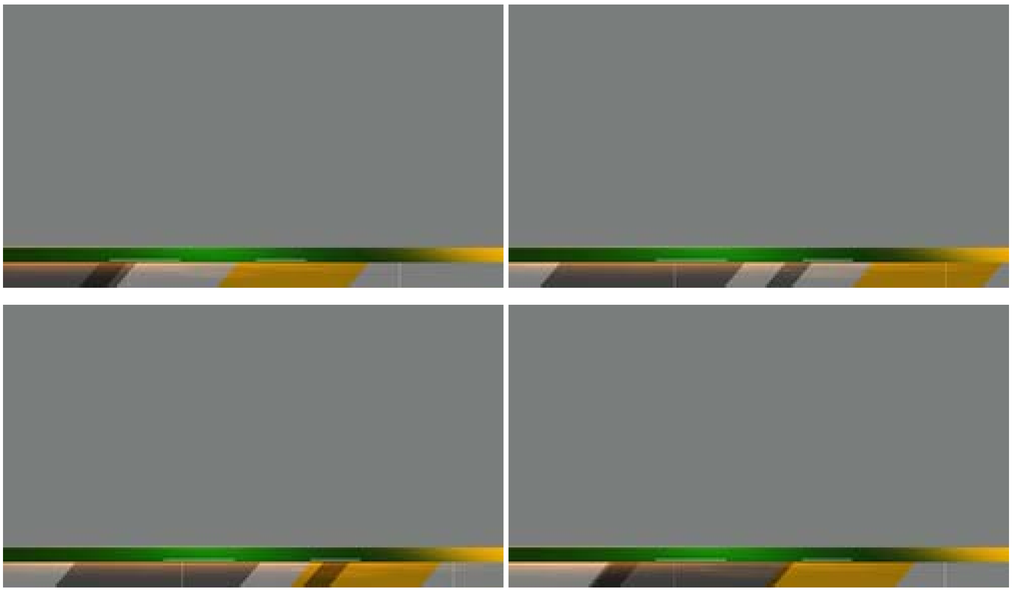
Cross animation directors are added by the Toggle plug-in. The directors are always tagged with the letters AX and BX. AX and BX holds the cross animation and effect used to animate from one object to another. The default cross animation is a 1 second cross fade of the alpha value.

Animation and effect length can be changed by adjusting the director's animation and effect(s) on the stage. In addition it is also possible to remove or add effects (e.g. material, position and scaling).

See Also

- Toggle

19.5 Geometry Animation



Geometry animations are used in foreground scenes, and are the same kind of animations created with standalone scene design. foreground scene animations are independent and not controlled by the background scene. The example above shows a lower third foreground scene with geometries that animate across the screen.

19.6 Background Scene

The background scene serves several important functions. Most importantly it is the controlling part of a Transition Logic scene, and is essentially what makes standalone scene design different from Transition Logic scene design.

A Transition Logic background scene is:

- Constantly on air, and is used for controlling state transition animations on backplate graphics in combination with the cross animations of foreground scenes (e.g. toggling from one line to two lines of text).
- Able to replace a layer's referring foreground scene without taking other layers off air.
- Able to have graphical elements such as backplates that separates the background for foreground scenes.

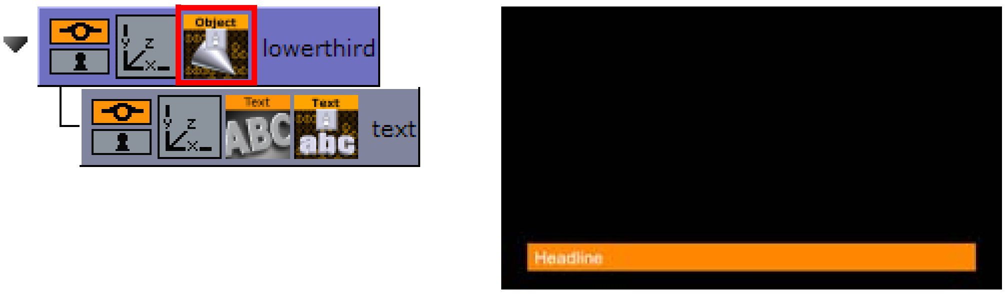
Having the background scene constantly on air means that when controlling the toggle layers it can smoothly morph the backplate graphics from one size to another providing an artifact-free backplate that will fit the changing requirements of the foreground scenes. With a one to two lines of text geometry, the text will, at a minimum, have a built in cross animation (e.g. fade out one line and fade in two lines). Additionally, as the background scene is already on air the transition animations on the background elements will have no artifacts (e.g. jittering) caused by graphics being loaded.

See Also

- State Transition Animation
- Cross Animation
- Foreground Scene

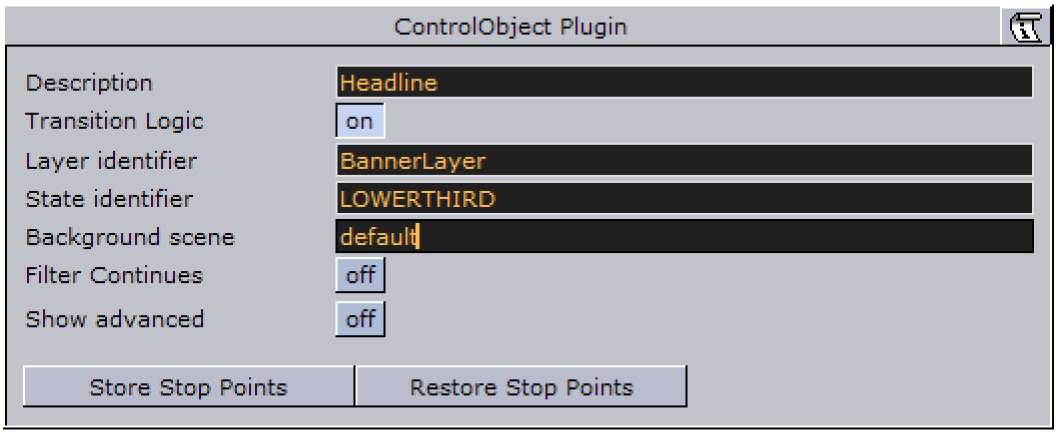
19.7 Foreground Scene

Figure 187: Simple lower third foreground scene



Transition Logic foreground scenes are similar to traditional standalone scenes that contain graphical elements such as geometrical shapes and textures (e.g. images and video clips), but are different in that they are capable of affecting the state of the background scene's toggle layer.

Figure 188: Transition Logic enabled Control Object plug-in


ControlObject Plugin

Description: Headline
Transition Logic: on
Layer identifier: BannerLayer
State identifier: LOWERTHIRD
Background scene: default
Filter Continues: off
Show advanced: off

In order for foreground scenes to have an effect on the background scene's toggle layer, it has to refer to the background scene, the layer and the layer's state. This is obtained by adding the Control Object plug-in to the foreground scene's scene tree at root level.

When Control Object is added to a foreground scene it must be Transition Logic enabled. This enables the foreground scene to set layer, state and background scene parameters.

These parameters are, when a foreground scene is taken on air by an operator, sent as commands to Viz requesting the background scene to change the toggle layer's current state to the new state the foreground scene is referring to.

If the foreground scene's Control Object parameters are not correct, Viz will not be able to trigger the correct state transition animation, and the foreground scene will not be able to animate in.

Note: Control Object also keeps track of other control plug-ins' properties that, when exposed, allows the operator to change their parameters (e.g. text size, color, geometry, texture, video clip and so on).

See Also

- Toggle Layer
- Geometry Animation
- Background Scene
- Control plug-ins

19.8 Tutorial

This section contains information on the following topics:

- Creating a Project
- Creating a Background Scene
- Creating a Foreground Scene

19.8.1 Creating a Project

It is good practice to create a project with a folder for each new design. This will ensure that a project with more than one design does not get mixed up, and potentially overwrite other files.

See Also

- Folders and Projects

19.8.2 Creating a Background Scene

Note that it is recommended to create the background scene before Creating a Foreground Scene as the background scene is needed to complete the foreground scene(s).

When creating a new Transition Logic scene design it is common practice to first define the Toggle Layers. After the toggle layers are defined you would typically move on to create the background scene, and add the defined toggle layers.

This section contains information on the following topics:

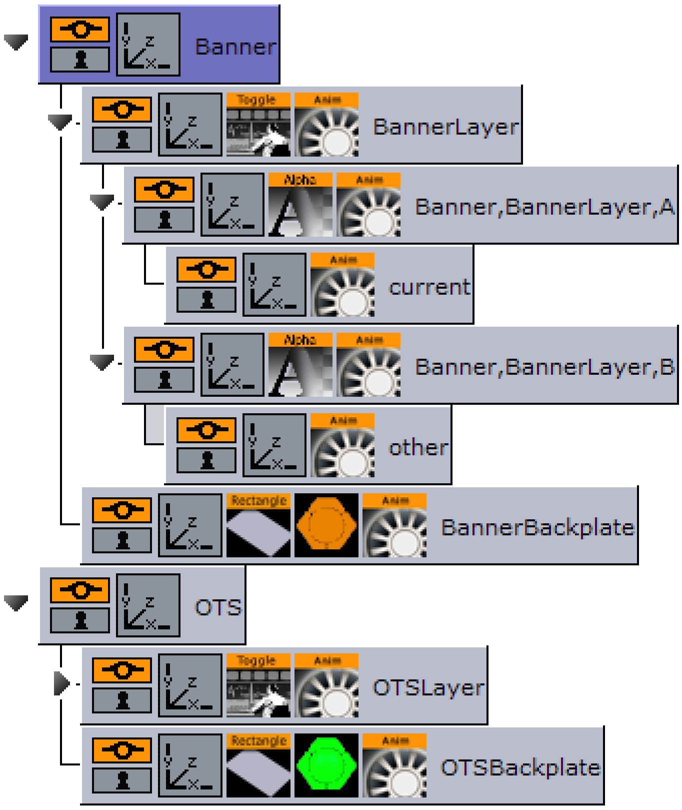
- To define the toggle layers
- To create a toggle layer
- To add key

To define the toggle layers

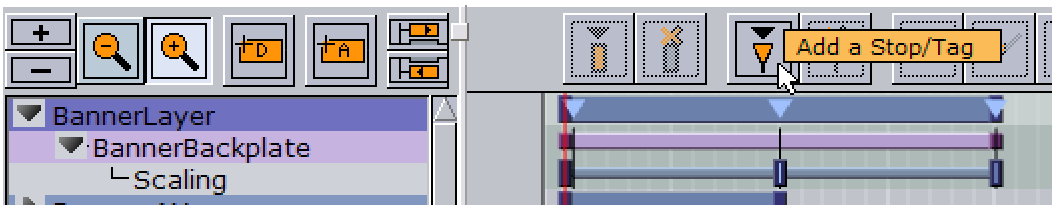
1. Create a list of all graphical elements to be shown. For example:
   - A large two lines lower third.
   - A small one line lower third.
   - A small one line top third.
   - A left side over the shoulder (OTS).
   - A right side OTS.
   - A logo bug.
   - A “coming up next” bug.
   - A news ticker.
   - A clock.
2. Create an operational scenario where you decide how elements will affect each other, and which elements can be on screen at the same time. For example:
   - Only one lower or top third can be on screen at a time (toggle layer 1).
   - Only one OTS can be on screen at a time (toggle layer 2).
   - Only one logo or “coming up next” bug can be on screen at a time (toggle layer 3).
   - The news ticker and clock should be possible to take in and out independently of all layers (toggle layers 4 and 5).

Note: All toggle layers can potentially be on screen at the same time, but only one foreground scene per layer can be shown at the same time.

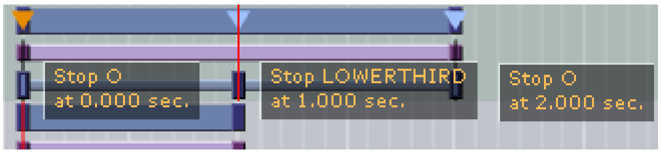
To create a toggle layer



1. Create a new scene, and name it **default.** This will be the background scene.
2. Add a group container to the scene tree, and name it **Banner**.
   - This will be the placeholder for the banner's toggle layer and backplate graphics for the banner foreground scenes.
3. Add a **sub-container** to the Banner container, and name it **BannerLayer**.
   - This will be the placeholder for the Toggle plug-in.
4. Add the **Toggle plug-in** to the BannerLayer container.
   - This will generate the default object placeholders for the toggle layer' s current and other state.
5. Add a **sub-container** to the Banner container, and name it **BannerBackplate.**
   - This will be the placeholder for the banner's backplate(s).
6. Add a **rectangle** and **material** to the BannerBackplate container, and adjust the Transformation Editor properties such that it creates the look and feel of a simple lower third graphics.
   - Axis Center: Set X to left (L) and Y to bottom (B). The X and Y values are set to -50.
   - Position: X: -330, Y: -160, Z: 0
   - Scaling (single): X: 1.0, Y: 0.4, Z: 1.0



7. Open the **Stage** by clicking the Stage button (main menu).
8. Add a **new director**, and name it **BannerLayer**. This will be the placeholder for the backplate animations.
9. **Animate** the banner backplate.
    - Scaling X: 0 (60/50 fps)
    - Scaling X: 6.6 (120/100 fps)
    - Scaling Y: 0 (180/150 fps)
10. Open the **Stage** and check that the animation resides under the BannerLayer director. If not, place it (drag and drop) under the BannerLayer director.



11. Select the Banner director, and the director's Time Marker to 0.000 seconds, and **add a tag**.
12. Select the tag and name it **O** (uppercase O, not zero).
13. Add a new tag for the animation's in point (e.g. LOWERTHIRD) and a second out point (O). The tags represent the basic states the graphics can be in.
14. Open the Toggle plug-in editor, and click the **Default keyframes** button.
    - Adding keyframes adds a 1 second default alpha cross fade effect to objects placed on that layer.

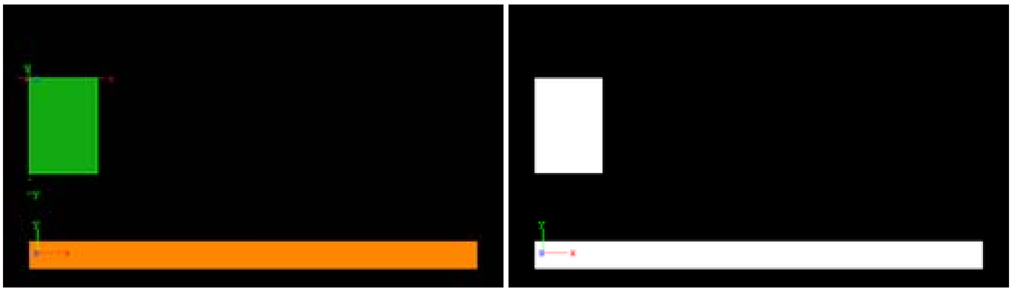
Note: Adding default keyframes is not possible until the layer's director has at least two states defined where one has to be the O state.

15. **Repeat** steps 2-14 for the OTS layer with the following transformation properties and animations.
    - Axis Center: Set X to left (L) and Y to top (T). The X value is set to -50 and the Y value to 50.
    - Position: X: -330, Y: 100, Z: 0
    - Scaling (single): X: 1.0, Y: 0.05, Z: 1.0

    - Scaling X: 0.0 (60/50 fps)
    - Scaling X: 1.0 (120/100 fps)
    - Scaling Y: 1.4 (180/150 fps)
    - Scaling X: 0.05 (240/200 fps)
    - Scaling X: 0.0 (300/250 fps)
16. **Save** the scene with all directors in their O state.

IMPORTANT! Set all directors to an O state because backplate graphics saved in a visible state will be shown on screen the first time any toggle layer is taken on air.

To add key



- Open Scene Settings and select the Auto Key option under Global Settings.

Tip: Click the Key button in the scene editor to toggle the key.

See Also

- Toggle Layer
- Creating a Project
- Creating a Foreground Scene
- Toggle

19.8.3 Creating a Foreground Scene

Before creating foreground scenes you should start by Creating a Background Scene. After the background scene has been created, the next step is to create the foreground scene(s) that reside on the different toggle layers defined in the background scene.

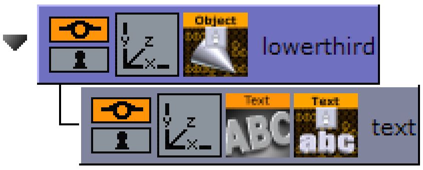
An important part of foreground scene design is to add the Control Object plug-in. The Control Object plug-in will enable the foreground scene to “tell” the Viz Engine which layer it resides on and the state the background scene should be in when taken on air by the operator.

Additionally the foreground scene can expose other parameters to the operator by adding control plug-ins (e.g. Control Text and Control Image). Additionally the foreground scene can be used for adding preview tags such that the scene can be properly previewed by newsroom users that work with the Viz Content Pilot Newsroom Component.

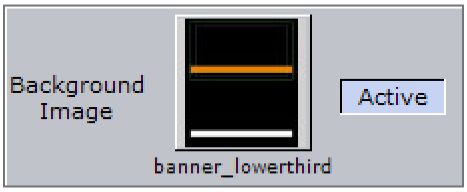
This section contains information on the following procedures:

- To create foreground scenes
- To create a background image for the foreground scene
- To create a preview point

To create foreground scenes

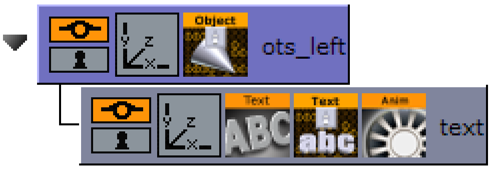


1. Create a new scene, and name it **1000**.
2. Add container to the scene tree, and name it **lowerthird**.
3. Add **Control Object** plug-in to the lowerthird container.
4. Add a **sub-container** to the lowerthird container, and name it **text**.
5. Add a **font** to the text container.
6. Add a **Control Text** plug-in to the text container.
7. *Optional:* Enable the Server view, and select the image (I) tab.



8. *Optional*: Open Scene Settings and select the Global Settings tab.
9. *Optional*: Drag and drop the snapshot image onto the **Background Image** placeholder (see how To create a background image for the foreground scene).
10. Open the Control Object's editor, and set the following parameters:
    - Description: Headline
    - Transition Logic: On
    - Layer Identifier: BannerLayer
    - State Identifier: LOWERTHIRD
    - Background scene: default
11. Open the text container's transformation editor, and set the following transformation properties and animations:
    - Position: X: -320, Y: -150, Z: 1
    - Scaling (locked): 0.5

    - Scaling (locked): 0.0 (60/50 fps)
    - Scaling (locked): 0.5 (120/100 fps)
12. **Save** the lowerthird scene.



13. **Repeat** steps 2-11 for the OTS foreground scene with the following parameters, transformation properties and animations.
    - Scene name: 2000

- Description: Text
- Transition Logic: On
- Layer Identifier: OTSLayer
- State Identifier: OTS_LEFT
- Background scene: default

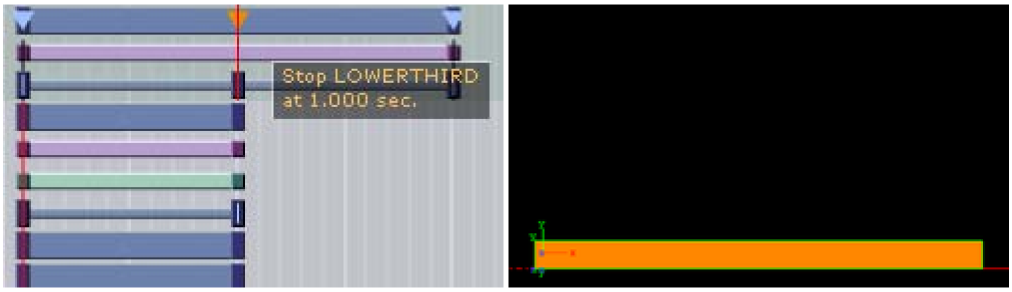
- Position: X: -330, Y: 120, Z: 1
- Scaling (locked): 0.3

- Scaling X: 0.0 (60/50 fps)
- Scaling X: 0.3 (120/100 fps)

Tip: Open the Control Text editor and disable the Single line parameter.

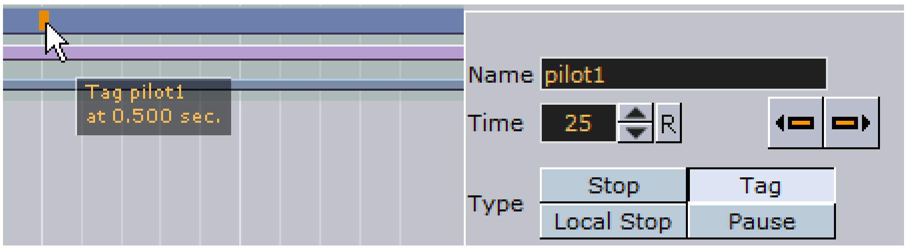
14. **Save** the OTS scene.

To create a background image for the foreground scene



1. Open the background scene.
2. Open the Stage view, and set the layer’s state where it matches the state of the foreground scene (e.g. LOWERTHIRD).
3. In the Scene Editor, click the Snap button and save the snapshot (e.g. as banner_lowerthird).
   - The snapshots can be used as “placeholders” for the foreground scenes.

To create a preview point



Name: pilot1
Time: 25

1. Open the Stage, and select the **Default** director.
2. **Position** the **Time Marker** from where you would like the scene to be previewed.
3. Click the **Add a Stop/Tag** button.
4. In the appearing editor enter Name as **pilot1** (case sensitive), and set Type to Tag.

- When previewing using a remote (external) Viz preview, either from a newsroom system, or when connected to a Media Sequencer with a control application, the **pilot1** tag will be the default preview point for the foreground scene.

See Also

- Foreground Scene
- Container Plugins
- Viz Content Pilot User’s Guide on remote Viz preview.

20 Scripting

Before starting to learn the computer language *Viz 3 Script*, you should learn the basic principles of programming which is explained in the following sections:

- How Programming Works
- Program Structure
- Variables
- Procedures
- Operators
- Conditions
- Loops
- Script Editor

In addition to the basic building blocks of script programming, you will also find a quick introduction on how to create and run scripts, how to create reusable plugins of your scripts and other example scripts that will help you script your own functionality:

- Creating and Running Scripts
- Creating Script-based Plugins
- Controlling 3D Stereoscopic Clip Playback
- Program Examples

20.1 How Programming Works

Technically, a computer works on a simple level processing the two states 0 and 1. The computer hardware only knows those two conditions and sets its “switches” according to them. Because a modern computer has a nearly uncountable number of those switches, it does not make any sense for a user to control them all by setting everyone to 0 or 1 by hand.

Programming language

Programming languages got developed for easy controlling the computer’s hardware by the user in a manageable way. With those languages, it is possible to develop readable commands which then will be translated into 0 and 1 for the computer. So these languages work as a translator between the user and the computer.

Syntax

This translation is processed by a compiler who converts text commands into machine code, but like every human language, a programming language has orthography and grammar rules. The user needs to follow those rules in order to make sure that the translation into machine code can be done without errors by the compiler.

The so called syntax describes those rules of spelling and constellation. Every programming language has its own syntax, so the script examples in this document will be written in the Viz 3 Script - syntax.

20.2 Program Structure

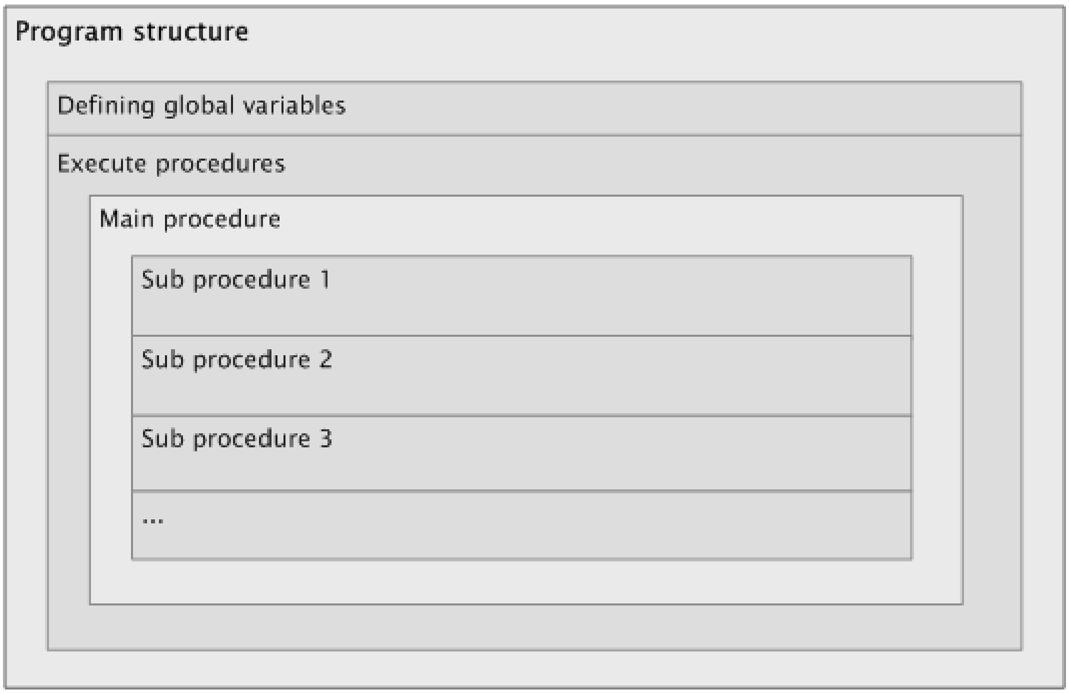
The certain text commands, which are written within the defined programing syntax, will lead to specific actions, done by the computer. Those commands and definitions are written with an adequate text editor and are called “source code”. This code will then be send to the compiler for processing.

Layout

The structure of a script program is similar to every other common language syntax. Before writing the actual commands and routines for the computer, it is necessary to define the variables which will be used within those commands so that the compiler knows how to handle them.

After defining variables (see Variables), the commands and routines can be defined in order to send them to the computer. A sequence of various commands will be held in so called procedures (see Procedures). Procedures can be seen as a container which includes those commands. If required, the user can call this container in order to execute the commands inside. This will avoid writing same code structures several times.

When the used commands got executed, the corresponding values of the variables got changed and translated into machine code, which will lead to the desired result of changing states between 0 and 1.



20.3 Variables

Basically, programs work with variables which work as placeholders for values and used for handling those values inside the procedures. Variables can contain different types of data (e.g. numbers, text, etc.) and allows the user to work with values even if they are not known at this time.

Example: A kitchen shelf contains 132 wine glasses. Instead of remembering the number `132` it makes more sense to store this value into a placeholder. So you create a variable `wineglass` and store the value `132` into it. You then use `wineglass` instead of remembering the exact value.

Variables are not part of the programing spelling or syntax, so the compiler will not recognize the word. So before using such a variable, it has to be defined in order for the compiler to use it in a command. If not the program will not be executable.

As mentioned before, variables can replace values of different types. So it is possible to store simple number values, text strings, colors or whole containers in it.

In order for the compiler to understand the type of variable it has to process, it is also necessary to declare what kind of data type the variable represents.

In programing a distinction is drawn between two types of variables:

- **Global Variables** are declared at the beginning of the program outside of procedures. These types can be used and modified in every procedure.
- **Local Variables** get declared within a certain procedure and can only be accessed by commands processed there. These variables cannot be seen by

other procedures, so modifying them from external procedures is not possible.

Definition and declaration of variables in Viz 3 Script are done by using the command `dim` which is short for *dimension*.

Note: All code examples described in this document are written within the syntax of Viz 3 Script. Therefore these examples can only be processed by an appropriate compiler.

Example

```
'Declaration and variable assignment in two operations
Dim wineglass as Integer
wineglass = 132

'Declaration and variable assignment in one operation
Dim wineglass as Integer = 132
```

20.3.1 Basic Data Types

This section describes and gives examples of the basic data types:

- Integer
- Double
- String
- Boolean
- Array
- Structure

Table 29:

Type	Description	Examples
Integer	Integer numbers	0,1,2,3 ...
Double	Floating point numbers	4,213415
String	Text	“Hello”, “Florian”
Boolean	State 1/0	true, false

Example

```
'String values are enclosed in double quotes
Dim name as String
name = "Florian"

Dim state as Boolean
state = true
```

Array

An Array is a Type of storage, which can be filled with an amount of variables of a certain data type. When defining an array, the data type of the containing variables has to be declared. By using an index, the variables can be addressed within the array.

In principle, an array can be imagined like a bookshelf which gets filled with single books (variables). Every place on the shelf has its own designated identifiable Number, so it is possible to detect which book is standing in a certain place.

By calling this identifier number, the array index, the array will respond with the assigned variable. The numeration starts at 0, so the first variable in the array is called by using the index “0”.

An Array containing names/text gets created and filled with text-strings. The command *push* allows it to load certain strings into the array.

```
Dim memory as Array[String]
   memory.push[“Florian”]
   memory.push[“Vizrt”]
   memory.push[“Austria”]
   memory.push[“Hello”]
```

By calling the index of the array, the assigned String gets returned. So "memory[2]" will return “Austria“. Other data types differ in most programing languages and will not be explained any further.

Special Viz 3 Script data types are referred to in the Scripting Data Types and Procedures chapter.

Structure

Structures are Containers that can store an amount of variables with different data types. Other than in arrays, those variables are not stored in a row with an index. You can define how many variables and different data types you want to use.

With structures, it is possible to work with a variable which can contain multiple values (stored in sub-variables).

In order to describe a soccer player with programing code, a new structure `soccer_player` with multiple variables can be declared. Those certain sub variables can be declared with any data type (Integer, Double, String, etc.).

```
structure soccer_player
   name as String
   team as String
   number as Integer
   age as Integer
end structure
```

The variable’s values can be defined with relevant values by referencing `soccer_player`.

```
soccer_player.name = “Lionel Andrés Messi”
soccer_player.team = “FC Barcelona”
soccer_player.number = 10
soccer_player.age = 24
```

See Also

- Array[Type]

- Boolean
- Double
- Integer
- String

20.4 Procedures

Like variables, procedures get defined with an appropriate name to manage them more easily in the code.

In order to double the amount of wine glasses, the procedure `double_wineglass()` can be called. Inside this procedure, the variable gets multiplied with the factor `2`. Every time `wineglass` gets called in the program, the actual amount of wine glasses is doubled.

```
sub double_wineglass()
   wineglass = wineglass * 2
end sub
```

Like in most programing languages, there are two types of procedures in Viz 3 Script:

- Sub-routines (process certain defined commands)
- Functions (sub-routines that return a variable)

20.4.1 Sub-routines

Both types, sub-routines and functions are basically a capsuled order of calculating processes (commands) that get stored under a specific name to make it possible to call these commands at any given time without rewriting it. The name of the procedure works as a reference to these commands.

Procedures can be imagined as sub-programs inside the main program which process calculations, modifies the variables and can call other sub programs.

Transfer Variables

To make procedures be more dynamic, it is not efficient to write a sub-routine for every single variable. By calling the sub-routine it makes sense to pass the variable to a sub-routine.

Inside the procedure, a new independent variable is used to process the calculations and commands.

By writing the sub-routine (sub - end sub), a new variable with an appropriate data type gets declared. The procedure itself now doubles every integer variable that is sent when called by the other procedures.

```
sub double(input as Integer)
   input = input * 2
end sub
```

By calling `double(wineglass)` the variable `wineglass` is sent to the procedure and doubled without using the exact variable name `wineglass` inside the procedure.

20.4.2 Functions

The only difference between sub-routines and functions is that functions are able to return a value after getting called. It makes it possible to use a function inside the code as a normal variable.

By creating the function, the data type which should get returned must be declared. In this case the function `multiply` returns a variable (same name multiply) as integer.

```
function multiply(input as Integer) as Integer
   multiply = input * input
end function
```

Now it's possible to handle the function as a standard variable.

```
sub tmp()
   new_glass = multiply(wineglass) + 10
end sub
```

20.5 Operators

To use and modify the various variables inside the program, a method to access and change the values is necessary.

```
dim number as Integer
number = 10
```

Operator is `=`

There is a range of different types of operators for working with variables:

- Assignment operator(s)
- Comparison operator(s)
- Arithmetic operator(s)
- Logical operator(s)
- String operator(s)
- Combined operator(s)

Assignment operator(s)

Assignment operators are used for defining variables and assigning values to variables.

Table 30:

Operator	Description
=	Assigning integer, double, string, etc.
==	Boolean assignment (true/false)

Comparison operator(s)

To compare certain variables in order to execute different commands (see Conditions).

Arithmetic operator(s)

Arithmetic operators are used for executing basic mathematical calculations.

Table 31:

Operator	Description
+	Addition
-	Substraction
*	Multiplication
/	Division

Logical operator(s)

Allows it to combine two Boolean states.

Table 32:

Operator	Description
Not	Returns "true" if conditions "false"
And	Returns "true" if both operands are "true"
Or	Returns "true" if at least one of the operands is "true"
Xor	Returns "true" if the two operands differ from each other

String operator(s)

Table 33:

Operator	Description
&	To combine single strings

```
dim a as String
dim b as String
dim c as String

a = "Hello"
b = "World"

c = a & b
```

The string value `c` now contains `Hello World`.

Combined operator(s)

A combination of assignment and arithmetical operators.

Table 34:

Operator	Description
+=	Addition of a value to the actual variable
-=	Subtraction of a value from the actual variable
*=	Multiplication of a value of the actual variable
/=	Division of a value on the actual variable
&=	Extension of a string with another one

20.6 Conditions

To make decisions in order to further processes other commands, it is often necessary to be able to request the status of certain variables or procedures and react accordingly.

Such conditions can be proofed by using **if-then/Else** or **case** statements. Variables or functions can be compared with reference values or states and trigger different actions when the condition is fulfilled or not.

For example, in an office building, the temperature gets controlled by an air condition system. If the temperature reaches a value below 25°, the air condition should stop. Else, if the temperature is above 25° the air condition should run and cool the building.

```
if temperature < 25 then
   Stop_Conditioning()
else
   Start_Conditioning()
end if
```

Note: The declaration of variables and procedures are not shown

To implement different conditions, the following comparison operators are available:

Table 35:

Operator	Literal	Description
=	equals	Identical values
<	smaller than	Value is smaller than the referenced value
>	greater than	Value is greater than the referenced value
<=	less than or equal	Value is less than or equal to the referenced value
>=	greater than or equal	Value is greater than or equal to the referenced value

Table 35:

Operator	Literal	Description
<>	unequal	Value is different from that of the referenced value

20.7 Loops

In order to execute commands, which need to be started several times, without typing them explicit in the code, loops are used to repeat those commands for a specified number of times.

Those loops are depending to an abortion condition and repeat the defined commands till this condition gets fulfilled (or not, depending on the condition commands). Inside the loop, a user defined sequence of commands and procedures get called.

In programing a distinction is drawn between two basic types of loops:

- Condition-controlled Loops
- Count-controlled Loops

20.7.1 Condition-controlled Loops

Thereby the loop is controlled by a specified abortion condition. Those loops basically get defined with the commands `Do` and `Loop`. The command sequence inside the loop will be executed as long as the condition is fulfilled. Additional command words are `While` and `Until` to specify the abortion condition more precise.

For example, the variable `a` gets checked if it is less than or equal to the value `10`. While the variable does fulfill the condition, the command `Println` gets called and afterwards the variable gets increased by 1. `Println` is used to output the variable's current value to the console.

```
Do While a <= 10
   Println a
   a = a + 1
Loop
```

Another way of defining the loop condition is to check the result after the commands are executed.

```
Do Println a
   a = a + 1
Loop Until a > 10
```

20.7.2 Count-controlled Loops

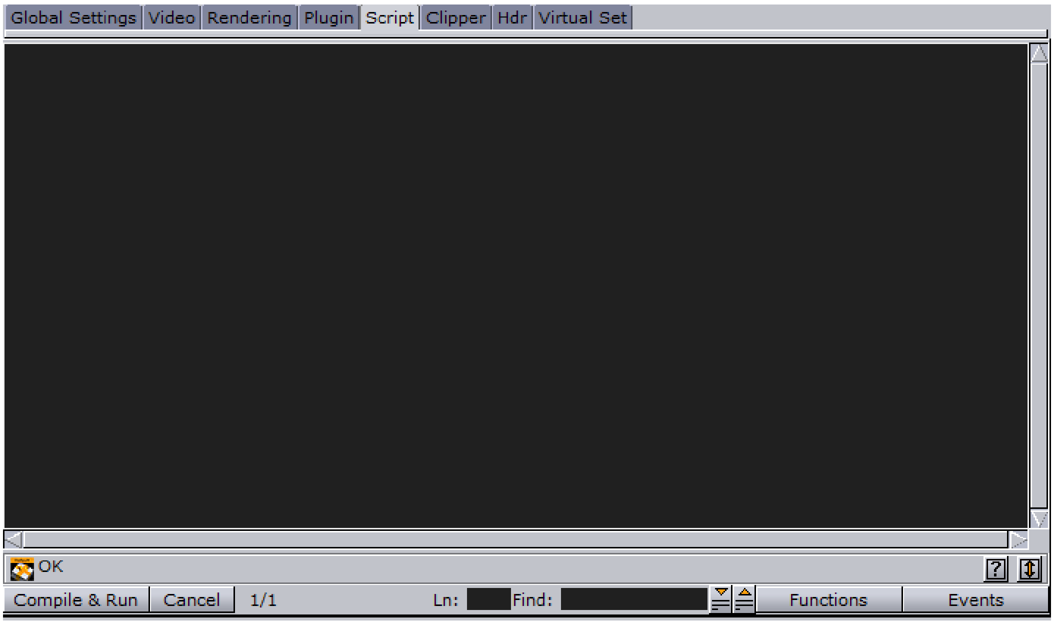
This type of loop controls its abortion condition using a count value. A variable with a start value gets defined and is iterated to end at a specified value. While the variable has not reached the defined value, the commands in the *loop body* are executed.

In the following example the variable i works as the counted variable that gets increased by 1 every time as it counts from 0 to 10. `Println` is used to output the variable's current value to the console.

```
For i = 0 to 10
   Println i
Next
```

The output should be the following: 1 2 3 4 5 6 7 8 9 10 (one number per line)

20.8 Script Editor



- **OK**: Shows that all parts of the script entered are correct. If there is an error, it will indicate where.
- **Compile & Run**: Compiles and runs the script.
- The **Cancel**: Cancels all changes.
- **Ln**: Jump to the line number entered in the field.
- **Find**: Free text search. The up-arrow will search above the cursor's location. The down-arrow will search below the cursor's location.
- **Functions**: List of all the functions defined.
- **Events**: List of all built-in events, such as mouse-click, mouse-move, keyboard button press, etc.
- ?: Launches the script documentation.
- **Up/Down arrows**: Pops-out/docks script editor.

To launch the script editor

1. Add a Script plug-in to your scene, or
2. In the main menu, click **Server**
3. In the right pane, select **Scene Settings**
4. Click the **Script** tab

To add scripts to the editor

- Copy (CTRL+C) your script and paste (CTRL+V) it into the Script Editor.

To add script plugins to the editor

- Drag and drop your compiled script plugin onto the Script Editor and select to replace the existing script, add it to the end or cancel the operation.

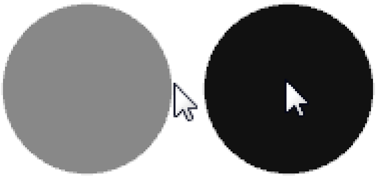
See Also

- Script plug-in
- Creating Script-based Plugins

20.9 Creating and Running Scripts

This section contains two examples; one showing how to write a simple script that controls the alpha value of an object (see Example I), the second how to write a simple script that scales a container’s x-scaling value based on another container’s x-scaling value (see Example II).

Example I



As an example, start with a simple event-driven script. First, create a script object by dragging the built-in Script plugin from the Container Plugins’ Global folder onto one of your containers. Then, after clicking on the container's new script object, copy and paste the following code into the the Script Editor.

```
Sub OnEnter()
   Alpha.Value = 100.0
End Sub

Sub OnLeave()
   Alpha.Value = 50.0
End Sub
```

This script consists of two event procedures, **OnEnter** and **OnLeave**, which are recognized by the system and invoked whenever the mouse cursor enters or leaves the area occupied by the container in the output window. Take a closer look at what happens inside **OnEnter**:

```
Alpha.Value = 100.0
```

This code sets the alpha value of the container to 100 (the maximum value). If there is no Alpha object, no action occurs, so make sure that the container has an alpha object by adding the Alpha plugin. **Alpha** refers to the container's alpha object, **Value** is a member (in this case: the only member) of the alpha object, representing its value. Changes made to **Alpha.Value** are immediatly made visible in the output window.

Similarly, the code in the body of **OnLeave** sets the alpha value back to 50.

To execute the script, click the **Compile & Run** button. Click **Edit** to return to editing mode.

Example II

The following example defines the event procedure **OnExecPerField**, which is called periodically for each field.

```
Sub OnExecPerField()
   Dim cube As Container
   cube = Scene.FindContainer("Circle")
   If scaling.x > 1.5 Then
      cube.Scaling.x = 2.0
   Else
      cube.Scaling.x = 1.0
   End If
End Sub
```

In the first line, a variable named **cube**, of type Container, is defined:

```
Dim cube As Container
```

Variables of type Container can be initialized to point to a container in the scene tree. In the next line, we call the built-in function FindContainer (which is a member of the scene class, accessible via the container's Scene property) to initialize cube to point to a container named cube. If there is no such container, cube is set to Null, the consequence being that any further operations on cube will have no effect.

```
cube = Scene.FindContainer("Cube")
```

Now, we check if the x-scaling of the current container (that is, the container holding the script) is greater than 1.5, and, depending on the result, we set the x-scaling of the container named cube to 2.0 or 1.0:

```
If Scaling.x > 1.5 Then
   cube.Scaling.x = 2.0
Else
   cube.Scaling.x = 1.0
End If
```

Note: The expression `Scaling.x` refers to the current container while `cube.Scaling.x` refers to the container referenced by the variable cube.

See Also

- Alpha
- Container
- Scaling

20.10 Creating Script-based Plugins

If you want to reuse a container script on other containers, you can create a plugin of that script.

- Select a folder other than Global in the Function Container pool and drag the container's script icon in the pool
  - Plugins created this way are stored in the plugin directory under the name **pluginname.vsl**
  - They can be used like ordinary plugins and, in addition, allow the following operations.

Script-based plugins can be deleted from the plugin pool by dragging and dropping the onto the trash can.

The source code of a script-based plugin can be copied into a script editor by dragging the plugin from the pool to the Script Editor.

20.11 Controlling 3D Stereoscopic Clip Playback

If you want to playout 3D Stereo Clips, you need to start Viz Engine with the -3D parameter. This sets Viz Engine in a special mode where a side by side clip will be split and played out on channel A and B on the Matrox board. Note that this is for clips only!

The playout on each output channel can be controlled with several shared memory variables. Each channel has its own set of variables indicated by the A and B in the names. The meaning of each of them is the same for both output channels.

- **HScale** / **VScale**: Scales the output image relative to the image anchor (see XAnchor and YAnchor) in the respective direction.
- **HSort** / **VSoft**: Softness is applied to the scaled region of the source image only, and is defined as the percentage of pixels to be softened from the border to the center.
- **XTrans** / **YTrans**: Translates the output image in the respective direction.
- **XAnchor** / **YAnchor**: Defines the position of the anchor on the input image. The axis center equals the center of the input image.
- **Sharpness**: Sets the amount of additional sharpness to apply to the output image.

To control the behavior of clip playback, the following shared memory variables exist:

```
vizrt.com.StereoClip.changed
vizrt.com.StereoClip.A.HScale
vizrt.com.StereoClip.A.VScale
vizrt.com.StereoClip.A.HSoft
vizrt.com.StereoClip.A.VSoft
vizrt.com.StereoClip.A.XTrans
vizrt.com.StereoClip.A.YTrans
vizrt.com.StereoClip.A.XAnchor
vizrt.com.StereoClip.A.YAnchor
vizrt.com.StereoClip.A.Sharp
vizrt.com.StereoClip.B.HScale
vizrt.com.StereoClip.B.VScale
vizrt.com.StereoClip.B.HSoft
vizrt.com.StereoClip.B.VSoft
vizrt.com.StereoClip.B.XTrans
vizrt.com.StereoClip.B.YTrans
vizrt.com.StereoClip.B.XAnchor
vizrt.com.StereoClip.B.YAnchor
vizrt.com.StereoClip.B.Sharp
```

See Also

- Camera configuration for setting stereo modes
- Clip Channel – Stereo script example

20.12 Program Examples

To explain the described terms and definitions more comprehensible, this chapter will show some source code examples and explains the important steps and commands.

Most of the following code samples are written in an easy and understandable form for programing beginners. Some are basic, while others are more advanced and specific for their use case. More efficient code structures and improvements are of course possible.

The following sub-sections contain examples on scripting:

- Initializing Scripts
- Calculating Values
- Registering Parameters
- Registering Parameters and Moving Objects
- Mouse Events I (Set)
- Mouse Events II (Start)
- Mouse Events III (Start, Stop and Reverse)
- Mouse Events IV (Grow and Shrink)
- Mouse Events V (Advanced)
- Planets
- Tetris
- Clip Channel
- Clip Channel – Stereo

20.12.1 Initializing Scripts

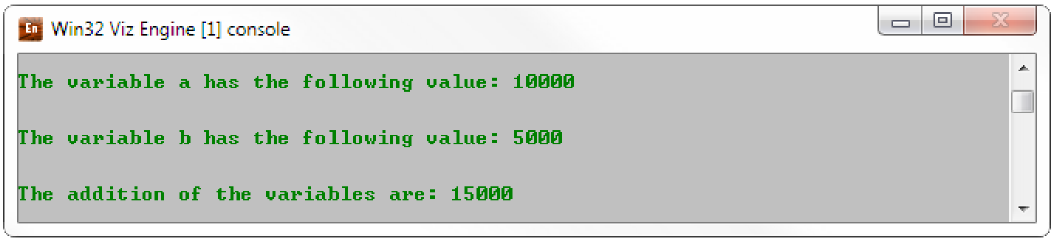
The following example declares and defines two variables with a value. In the main procedure `OnInit()` those two variables and their addition are output to the console using Println.

Callback procedures like `OnInit()` are used to initialize scripts (see Callback Procedures).

```
'Declaration of the variables
dim a as Integer
a = 10000

dim b as Integer
```

```
b = 5000

'Start of the main procedure
sub OnInit()

'Executing the commands
   println "\n"
   println "The variable a has the following value: " & a
   println "\n"
   println "The variable b has the following value: " & b
   println "\n"
   println "The addition of the variables are: " & a + b

'End of the main procedure
end sub
```

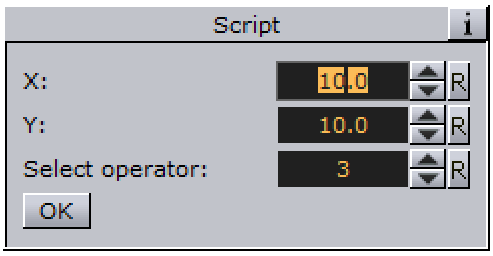
Result output to the console:



```
The variable a has the following value: 10000

The variable b has the following value: 5000

The addition of the variables are: 15000
```

20.12.2 Calculating Values

The following example shows a calculator that is able to calculate with 2 values using basic arithmetic operators + , – , * , / (see Arithmetic operator(s)).



X: 10.0
Y: 10.0
Select operator: 3

The values and the operators are defined using a text input form (1=”+”, 2=”–”, 3=”*” and 4=”/”).

Global variables used in every procedure are declared at the beginning of the program. For defining the operator, a structure data type is used to store the type of operator as an integer and its name as a string (in order to print the according text symbol).

```
'Declaring operator as a container of type structure
structure operator
   op as integer
   name as string
end structure

'Declaring sign as a variable of type operator
dim sign as operator

'Declaring valid as a variable of type boolean
dim valid as boolean = true

'Declaring result as a variable of type double
dim result as double
```

The variable's values (in *Main procedure* further down) are read in using a Viz 3 Script text input with the command `GetParameterDouble`. The procedure `OnInitParameters()` is not relevant for the case as such, but it creates an interface for the designer to easily input the values that are fetched by `GetParameterDouble`.

```
'Calculator interface
sub OnInitParameters()
   'Getting the values
   RegisterParameterDouble("x","X: ",0,0.0,1000000)
   RegisterParameterDouble("y","Y: ",0,0.0,1000000)
   RegisterParameterInt("operator","Select operator: ",1,1,4)
   RegisterPushButton("action", " OK ",1)
end sub
```

At the beginning of the main procedure `OnExecAction()`, the two input variables for the calculation are declared as double data types because calculations may result in values with a floating point. These variables are local because they are only used within this one procedure.

By using the operator = the values are stored in the variable placeholder `x` and `y`. The mathematic operator is stored as an integer value into the structure. Following, the function `calculator` gets called, the variables `x` and `y` and the type of arithmetic operator get sent to it. The result of `calculator` gets stored in the variable `result`.

At the end the function "print" prints the results on the screen.

```
'Main procedure
sub OnExecAction(action as Integer)
   dim x as double = GetParameterDouble("x")
   dim y as double = GetParameterDouble("x")

   sign.op = GetParameterInt("operator")

   'Starting the calculation process
   result = calculator(x,y,sign.op)

   'Output to the console
   print(x,y,result)
end sub
```

In function `calculator`, the mathematic calculation process gets executed. By using an *If* condition the operator gets checked and the resulting operation started. The correct text symbol gets stored into a string for the further printing function and a Boolean flag (`valid`) gets set to true if the calculation was correct (division by 0 is not possible).

After calculation the operator's name gets stored into the structure as string (`sign.name`) and calculator returns the result back to the main procedure.

```
'Calculation function
function calculator(x as double, y as double, operator as integer) as
double

   'Declaring local variables
   dim tmp as string

      if operator = 1 then
         result = x+y
         tmp = "+"
         valid = true

      elseif operator = 2 then
```

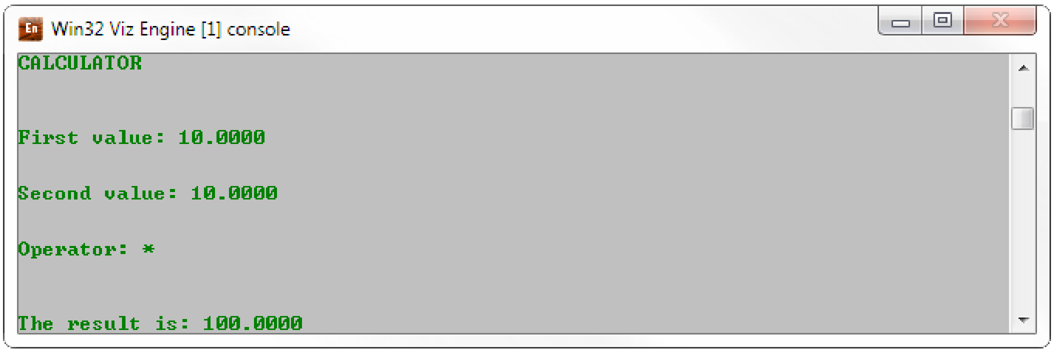
```
            result = x-y
            tmp = "-"
            valid = true

         elseif operator = 3 then
            result = x*y
            tmp = "*"
            valid = true

         elseif operator = 4 then
            if y=0 then
               valid = false
            else
               result = x/y
               tmp = "/"
               valid = true
            end if
         end if

         sign.name = tmp
         calculator = result
   end function
```

To avoid a division by 0, the `y` variable gets checked if it is equal to `0` when the operator is set to *division*. If it equals `0`, the calculation will be set to invalid (`valid = false`) and an error message is output to the console.

The final printing function takes the 2 variables “x” and “y” and the result and prints it formatted on the screen. If the Boolean variable “valid” is set to false, the error message will be shown.

```
   ‘Console output
   sub print(x as double, y as double, result as double)
      'Check if calculation is valid
      if valid = true then
         println "\n\n\n\n"
         println "CALCULATOR"
         println "\n\n"
         println "First value: " &x
         println "\n"
         println "Second value: " &y
         println "\n"
         println "Operator: " &sign.name
         println "\n\n"
         println "The result is: " &result
         println "\n\n\n\n"
      else
         println "\n\n\n\n"
         println "Calculation was not possible"
         println "\n\n\n\n"
      end if
   end sub
```

Result printed on the screen:



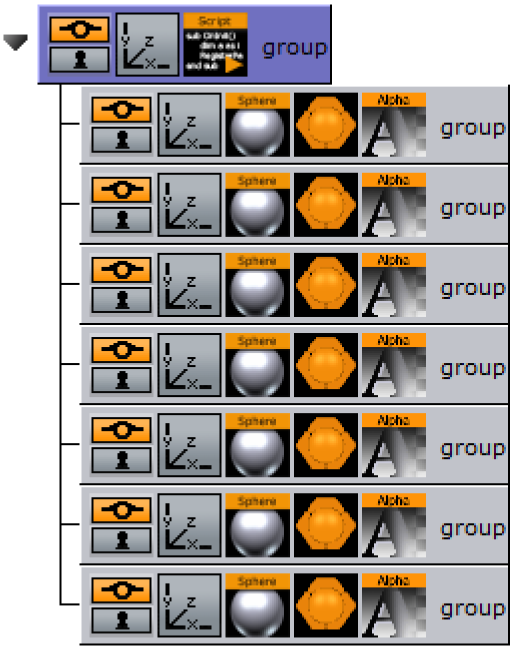
```
CALCULATOR

First value: 10.0000

Second value: 10.0000

Operator: *

The result is: 100.0000
```

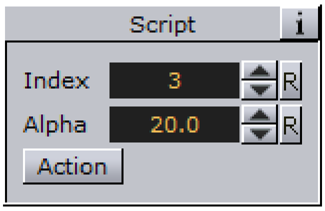
See Also

- Callback Procedures
- Global Procedures
- Boolean
- Double
- Integer
- String
- Structure

20.12.3 Registering Parameters

This script example creates a custom editor for filling in index and alpha values. The index value decides which container that will not receive the alpha value wheras the alpha value is set for all other containers.

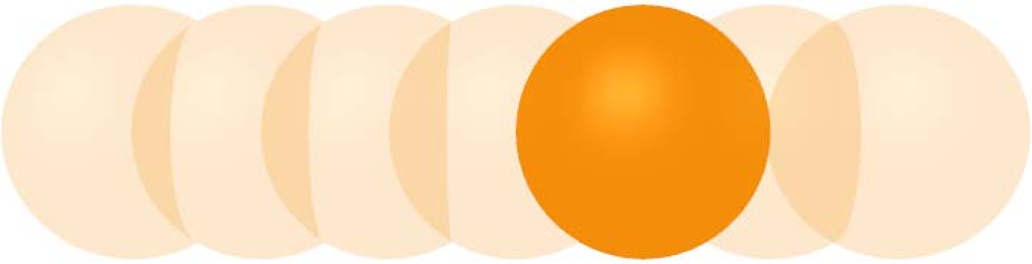




Index 3

Alpha 20.0

```
sub OnInitParameters()
   RegisterParameterInt("index", "Index", 1, 1, 7)
   RegisterParameterDouble("alpha", "Alpha", 50.0, 0.0, 100.0)
   RegisterPushButton("action", "Action", 1)
end sub
```



```
sub OnExecAction(buttonId as Integer)
   dim c as Container
   dim i, index as Integer
   dim a as Double
   index = GetParameterInt("index")
   a = GetParameterDouble("alpha")
   i = 1
   c = ChildContainer
   do while c <> null
      if i = index then
         c.alpha.value = 100.0
      else
         c.alpha.value = a
      end if
   i = i + 1
   c = c.NextContainer
   loop
end sub
```

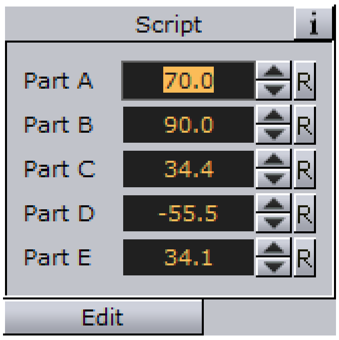
See Also

- Callback Procedures
- Global Procedures
- Boolean
- Container
- Double
- Integer

20.12.4 Registering Parameters and Moving Objects

The following example initializes a set of registered parameters that enables the `OnParameterChanged` procedure to rotate the objects accordingly.

Figure 189: Registered parameters



Script

Part A 70.0
Part B 90.0
Part C 34.4
Part D -55.5
Part E 34.1

```
dim part_a as container=scene.FindContainer("move1")
dim part_b as container=scene.FindContainer("move2")
dim part_c as container=scene.FindContainer("move3")
dim part_d as container=scene.FindContainer("move4")
dim part_e as container=scene.FindContainer("move5")
```

```
sub OnInitParameters()
   RegisterParameterDouble("move_a","Part A",0,0.0,70)
   RegisterParameterDouble("move_b","Part B",0,-360,360)
   RegisterParameterDouble("move_Gec","Part C",0,-360,360)
   RegisterParameterDouble("move_d","Part D",0,-90,45)
   RegisterParameterDouble("move_e","Part E",0,-360,360)
end sub
```

Figure 190: Moving parts

```
sub OnParameterChanged(parameterName as String)
   if parameterName="move_a" then
      part_a.rotation.z=getParameterDouble("move_a")
   end if

   if parameterName="move_b" then
      part_b.rotation.x=getParameterDouble("move_b")
   end if
   if parameterName="move_c" then
      part_c.rotation.y=getParameterDouble("move_c")
   end if

   if parameterName="move_d" then
      part_d.rotation.y=getParameterDouble("move_d")
   end if

   if parameterName="move_e" then
      part_e.rotation.z=getParameterDouble("move_e")
   end if
end sub
```

See Also

- Callback Procedures
- Global Procedures
- Container
- Rotation
- Scene

20.12.5 Mouse Events I (Set)

This script example shows how you can use a “mouse event” to set an object’s alpha value using OnEnter and OnLeave.

```
sub OnEnter()
   alpha.value = 100.0
end sub

sub OnLeave()
   alpha.value = 50.0
end sub

sub OnExecPerField()
```

```
      dim cube as Container
      cube = Scene.FindContainer("Cube")
      if scaling.x > 1.5 then
         cube.material.diffuse.red = 0.0
      else
         cube.material.diffuse.red = 100.0
      end if
   end sub
```

20.12.6 Mouse Events II (Start)

This script example shows how you can use a “mouse event” to start an animation using OnMButtonDown and StartAnimation.

```
sub OnInit()
   OnEnter()
end sub

sub OnEnter()
   alpha.value = 100.0
end sub

sub OnLeave()
   alpha.value = 50.0
end sub

sub OnMButtonDown()
   dim dir as Director
   if ContainsMouseCursor() then
      dir = Stage.FindDirector("Default")
      dir.StartAnimation
   end if
end sub
```

20.12.7 Mouse Events III (Start, Stop and Reverse)

When adding this script to a container you can start the animation by pressing the mouse button over an object (e.g. rectangle). Press again and it will stop. Press again and it wil continue from the point it stopped.

```
dim my_dir = stage.FindDirector("Default")
sub OnLButtonDown()
   if my_dir.IsAnimationRunning() then
      my_dir.StopAnimation()
   else
      my_dir.ContinueAnimation()
   end if
end sub
```

With the below script you can start an animation and let it run until a stop point. Click again and it will run in reverse.

```
dim my_dir = stage.FindDirector("Defult")
sub OnLButtonDown()
   menu.ContinueAnimation()
   if my_dir.reverse == true then
      my_dir.reverse = false
   else
      my_dir.reverse = true
   end if
end sub
```

See Also

- Callback Procedures
- Director

- Stage

20.12.8 Mouse Events IV (Grow and Shrink)

When adding the below script to a container the object will grow when the mouse is hovering over the object and conversely. In this case the centre of the object is set to the lower right corner in order to let it grow out from that centre.

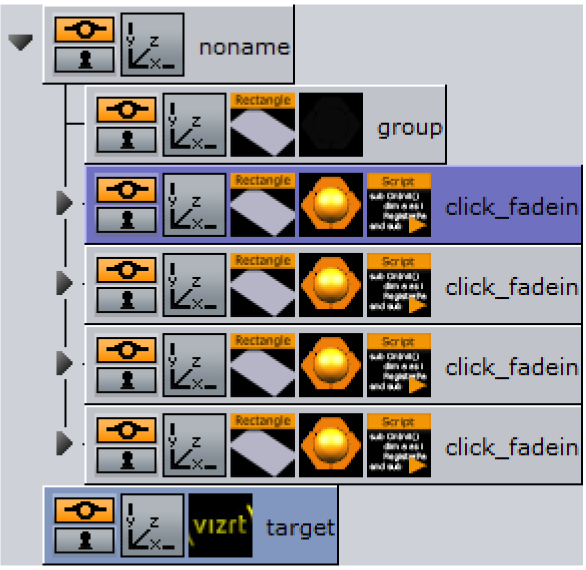
```
dim growing as Boolean = false

sub OnInitParameters()
   RegisterParameterDouble("min", "MinScale", 1.0, 0.0, 10.0)
   RegisterParameterDouble("max", "MaxScale", 1.5, 0.0, 10.0)
   RegisterParameterDouble("fac", "GrowFactor", 1.1, 1.0, 10.0)
end sub

sub OnEnter()
   growing = true
end sub

sub OnLeave()
   growing = false
end sub

sub OnExecPerField()
   dim min = GetParameterDouble("min")
   dim max = GetParameterDouble("max")
   dim fac = GetParameterDouble("fac")

   if growing then
      scaling.xyz *= fac
      if scaling.x > max then scaling.xyz = max
   else
      scaling.xyz /= fac
      if scaling.x < min then scaling.xyz = min
   end if
end sub
```

See Also

- Callback Procedures
- Global Procedures
- Boolean
- Scaling

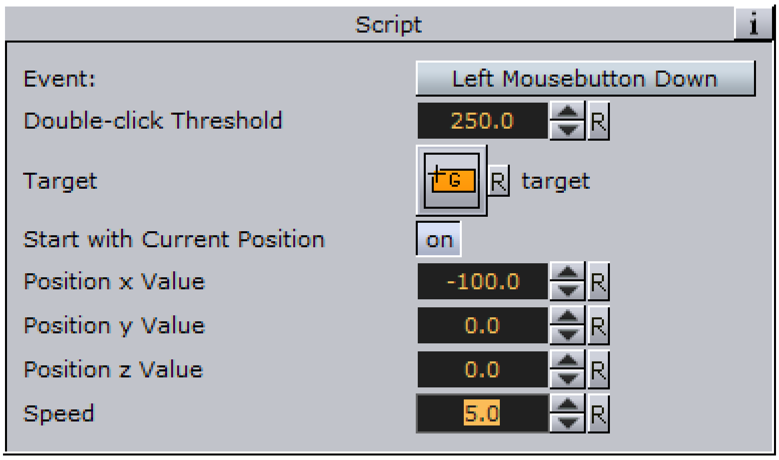
20.12.9 Mouse Events V (Advanced)

This script shows how you can use “mouse events” to click buttons that will trigger pre-defined actions. This script example also shows how to input parameters by scripting your own custom GUI.



The same script can be added to multiple containers by creating several buttons that perform different animations.

```
dim events as Array[String]
dim action as Array[String]
dim str_action as string
dim str_event as string
dim doubleclick_tracker as Integer
dim dbl_threshhold as Double
dim target_container as container
dim p_anim as Boolean = false

dim p_step as Vertex
dim p_dest as Vertex
dim p_speed,p_time as Double
dim animP0, animP1, animP2,animP3 as Vertex


'show/hide GUI buttons
events.clear
events.push("Left Mousebutton Down")
'events.push("Left Mousebutton Up")
'events.push("On Mouse Enter")
'events.push("On Mouse Leave")
'events.push("Left Mousebutton Double-click")
'events.push("Right Mousebutton Down")
'events.push("Right Mousebutton Up")
'events.push("Right Mousebutton Double-click")
'events.push("Middle Mousebutton Down")
'events.push("Middle Mousebutton Up")

sub OnInit()
   'initialize GUI
   str_event=events[GetParameterInt("do_event")]
   target_container=GetParameterContainer("target_con")
   dbl_threshhold=GetParameterDouble("dbl_threshhold")
end sub
```



Script

Event: Left Mousebutton Down

Double-click Threshold 250.0

Target target

Start with Current Position on

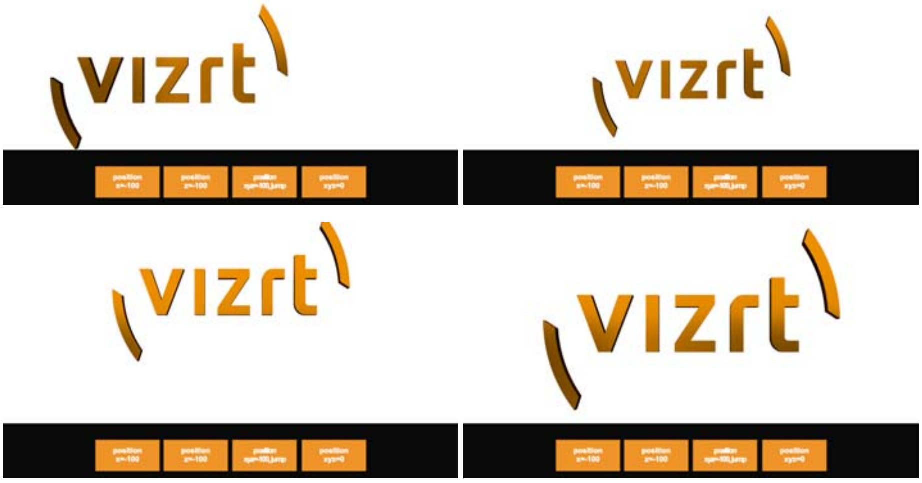
Position x Value -100.0

Position y Value 0.0

Position z Value 0.0

Speed 5.0

```
sub OnInitParameters()
   'create GUI
   RegisterRadioButton("do_event", "Event:",2,events)
   RegisterRadioButton("do_action", "Action:",0,action)
   RegisterParameterDouble("dbl_threshhold","Double-click
Threshold",250,10,1000)
   RegisterParameterContainer("target_con","Target")
   RegisterParameterBool("start_current","Start with Current
Position",false)
   RegisterParameterDouble("position_x","Position x
Value",0.0,1000.0,1000.0)
   RegisterParameterDouble("position_y","Position y
Value",0.0,1000.0,1000.0)
   RegisterParameterDouble("position_z","Position z
Value",0.0,1000.0,1000.0)
   RegisterParameterDouble("speed","Speed",1.0,0.1,10.0)
end sub

'react to mouse events
sub OnEnter()
   if str_event="On Mouse Enter" then
      animate()
   end if
end sub

sub OnLeave()
   if str_event="On Mouse Leave" then
      animate()
   end if
end sub

sub OnLButtonDown()
   if str_event="Left Mousebutton Down" then
      animate()
   end if

   doubleclick_tracker=GetTickCount()
end sub

sub OnLButtonUp()
   if str_event="Left Mousebutton Up" then
      animate()
   end if
   if str_event="Left Mousebutton Double-click" then
      'check for double-click events
      if (GetTickCount()-doubleclick_tracker<dbl_threshhold)  then
         animate()
      end if
   end if
end sub

sub OnMButtonDown()
   doubleclick_tracker=GetTickCount()
   if str_event="Middle Mousebutton Down" then
      animate()
   end if
end sub

sub OnMButtonUp()
   if str_event="Middle Mousebutton Up" then
      animate()
   end if
end sub

sub OnRButtonDown()
   if str_event="Right Mousebutton Down" then
      animate()
   end if
   doubleclick_tracker=GetTickCount()
end sub

sub OnRButtonUp()
   if str_event="Right Mousebutton Up" then
      animate()
   end if

   'check for double-click events
   if str_event="Right Mousebutton Double-click" then
      if (GetTickCount()-doubleclick_tracker<dbl_threshhold) then
         animate()
      end if
   end if
end sub
```



```
sub animate()
   'Check if target container exists, if not, use current container

p_dest=cVertex(GetParameterDouble("position_x"),GetParameterDouble("po
sition_y"),getParameterDouble("position_z"))

   p_speed=getParameterDouble("speed")
   p_speed=p_speed/100

   if GetParameterBool("start_current")=false then
      target_container=GetParameterContainer("target_con")
      target_container.position.xyz=p_dest
   else
      animP1=target_container.position.xyz
      animP0=animP1
      animP2=p_dest
      animP3=animp2
      p_time=0.0
      p_anim=true
   end if
   'enable the animation
'  p_anim=true
end sub

sub OnParameterChanged(parameterName as String)
   'react to changes in the GUI
   if parameterName="target_con" then
      target_container=GetParameterContainer("target_con")
   end if

   if parameterName="do_event" then
      str_event=events[GetParameterInt("do_event")]
   end if

   if parameterName="dbl_threshhold" then
      dbl_threshhold=GetParameterDouble("dbl_threshhold")
   end if
end sub

sub OnExecPerField()
   if p_anim then
      p_time+=p_speed
      p_time=min(1,p_time)
      println p_time
      dim newpos as Vertex=spline(animP0, animP1, animP2, animP3,
p_time)
      target_container.position.xyz=newpos
      if p_time=1 then
         target_container.position.xyz=p_dest
         p_anim=false
      end if
   end if
end sub

function spline(P0 as Vertex,P1 as Vertex,P2 as Vertex,P3 as
Vertex,time as Double) as Vertex
   dim pos as Vertex
   dim final_term as Vertex
   dim term1 as Vertex
   dim term2 as Vertex
   dim term3 as Vertex

   term1=multiply_vertex(P1,2.00)+(-P0+P2)*time
   term2=(multiply_vertex(P0,2.00)-
multiply_vertex(P1,5.00)+multiply_vertex(P2,4.00)-P3)*time*time
   term3=(-P0+multiply_vertex(P1,3.00)-
multiply_vertex(P3,3.00)+P3)*time*time*time

   final_term=term1+term2+term3
   spline=multiply_vertex(final_term,0.5)
end function

'----------------------------------------------------------------------
'helpers
function multiply_vertex(P as Vertex,multiplier as Double) as Vertex
   multiply_vertex.x=P.x*multiplier
   multiply_vertex.y=P.y*multiplier
   multiply_vertex.z=P.z*multiplier
end function
```

See Also

- Callback Procedures
- Global Procedures
- Array[Type]
- Boolean
- Container
- Double
- Integer

- String
- Vertex

20.12.10 Planets

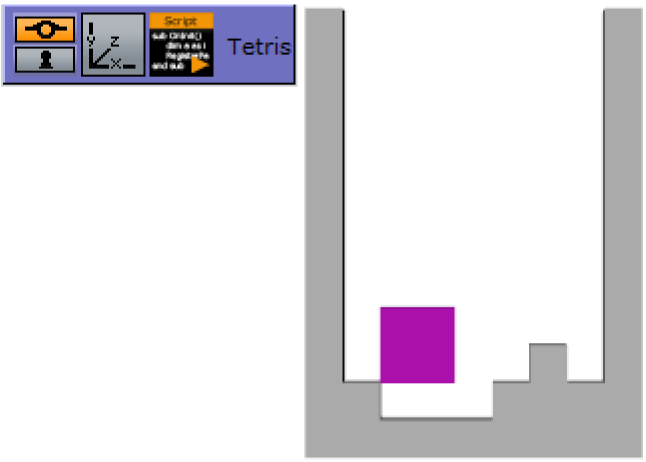
```
dim MAXNUM, NUM as Integer
dim MAXMASS, G, POS1X, POS1Z as Double
MAXNUM = 40
NUM = 9
MAXMASS = 1000000
G = 0.006674
POS1X = 0
POS1Z = 0
structure Planet
   rx, ry, vx, vy, mass as Double
end structure

dim planetContainers as Array[Container]
dim masses as Array[Planet]
dim lastPlanetIndex as Integer

sub OnInit()
   dim c as Container
   dim i as Integer
   dim dx, dy, r, v as Double

   c = ChildContainer

   do while c <> null
      planetContainers.push c
      c = c.NextContainer
   loop

   masses.size = planetContainers.size
   lastPlanetIndex = planetContainers.size - 1

   if lastPlanetIndex < 0 then exit sub

   ' the sun
   masses[0].rx = 0.0
   masses[0].ry = 0.0
   masses[0].vx = 2.0
   masses[0].vy = 0.0
   masses[0].mass = 300000000

   ' the planets
   for i = 1 to lastPlanetIndex
      dx = random() * 20000.0 - 10000.0
      dy = random() * 20000.0 - 10000.0

      if dx <= 0.0 and dx > -500.0 then dx = -500.0
      if dx >  0.0 and dx <  500.0 then dx =  500.0
      if dy <= 0.0 and dy > -500.0 then dy = -500.0
      if dy >  0.0 and dy <  500.0 then dy = 500.0

      masses[i].rx = POS1X + dx
      masses[i].ry = POS1Z + dy

      r = sqrt(dx * dx + dy * dy)
      masses[i].mass = random() * MAXMASS
      v = sqrt(G * masses[0].mass / r)
      masses[i].vx = dy / r * v
      masses[i].vy = -dx / r * v
   next
end sub

sub OnExecPerField()
   dim ax, az, ar, r2, dx, dy as Double
   dim i, j as Integer

   for i = 0 to lastPlanetIndex
```

```
      planetContainers[i].position.x = masses[i].rx * 0.01
      planetContainers[i].position.y = masses[i].ry * 0.01
   next

   ' calculate new velocities
   for i = 0 to lastPlanetIndex
      ax = 0
      az = 0
      for j = 0 to lastPlanetIndex
         if j <> i then
            dx = masses[j].rx - masses[i].rx
            dy = masses[j].ry - masses[i].ry
            r2 = dx*dx + dy*dy
            ar = masses[j].mass / (r2 * sqrt(r2))
            ax += ar * dx
            az += ar * dy
         end if
      next

      masses[i].vx += ax * G
      masses[i].vy += az * G
   next

   ' calculate new positions

   for i = 0 to lastPlanetIndex
      masses[i].rx += masses[i].vx
      masses[i].ry += masses[i].vy
   next
end sub
```

See Also

- Callback Procedures
- Global Procedures
- Array[Type]
- Boolean
- Container
- Integer

20.12.11 Tetris



This script example creates the classical game of Tetris using Viz Engine scripting and graphics.

```
dim HEIGHT, WIDTH, TOTAL, NUM_BRICKS as Integer
dim currentBrick, currentRot, counter as Integer
dim xpos, ypos as Integer ' position of the falling brick
dim squares as Array[integer]
dim squareContainers as Array[Container]
dim bricks as Array[String]

HEIGHT = 15
WIDTH = 9
TOTAL = HEIGHT * WIDTH

squares.size = TOTAL

sub OnInit()
   dim i, j as Integer

   ' 0100
   ' 0100
   ' 0100
   ' 0100

   bricks.push("0100010001000100")
   bricks.push("0000111100000000")
   bricks.push("0100010001000100")
   bricks.push("0000111100000000")

   ' 0000
   ' 0000
   ' 0110
   ' 0110

   bricks.push("0000000001100110")
   bricks.push("0000000001100110")
   bricks.push("0000000001100110")
   bricks.push("0000000001100110")

   ' 0000
   ' 0100
   ' 0110
   ' 0010

   bricks.push("0000010001100010")
   bricks.push("0000000001101100")
   bricks.push("0000010001100010")
   bricks.push("0000000001101100")

   ' 0000
   ' 0010
   ' 0110
   ' 0100

   bricks.push("0000001001100100")
   bricks.push("0000000011000110")
   bricks.push("0000001001100100")
   bricks.push("0000000011000110")

   ' 0000
   ' 0110
   ' 0010
   ' 0010

   bricks.push("0000011000100010")
   bricks.push("0000001011100000")
   bricks.push("0100010001100000")
   bricks.push("0000011101000000")

   ' 0000
   ' 0110
   ' 0100
   ' 0100

   bricks.push("0000011001000100")
   bricks.push("0000111000100000")
   bricks.push("0010001001100000")
   bricks.push("0000010001110000")
   ' 0000
   ' 0000
   ' 1110
   ' 0100

   bricks.push("0000000011100100")
   bricks.push("0000010011000100")
   bricks.push("0000010011100000")
   bricks.push("0000010001100100")

   NUM_BRICKS = bricks.size / 4

   deleteGeometry
   update
   deleteChildren

   dim blocksGroup as Container
   blocksGroup = addContainer(TL_DOWN)

   open = false

   for i=0 to HEIGHT-1
      for j=0 to WIDTH-1
         dim b as Container
         if i >= 3 then
            b = blocksGroup.addContainer(TL_DOWN)
            b.createMaterial
            b.name = "f_" & j & "_" & i
            b.createGeometry("BUILT_IN*GEOM*Cube")
            b.scaling.xyz = 0.1
            b.position.x = (j-WIDTH/2.0+0.5) * 10.0
            b.position.y = - (i-HEIGHT/2.0-1.0) * 10.0
         end if
         squareContainers.push(b)
      next
   next

   initGame
end sub

sub display()
   dim i, j, index, fb as Integer
   dim fc as Container

   for i=3 to HEIGHT-1
      for j=0 to WIDTH-1
         index = i * WIDTH + j
         fb = squares[index]
         fc = squareContainers[index]
         if fb = 0 then
            fc.active = false
         else
            fc.active = true
            if fb = 1 then
               fc.material.diffuse = CColor(60.0, 60.0, 60.0)
            elseif fb = 3 then
               fc.material.diffuse = CColor(30.0, 30.0, 30.0)
            else
               fc.material.diffuse = CColor(40.0, 0.0, 80.0)
            end if
         end if
      next
   next
end sub

sub initGame()
   dim i as Integer

   for i=0 to TOTAL-1
      squares[i] = 0
   next

   for i=0 to HEIGHT-1
      squares[WIDTH*i] = 3
      squares[WIDTH*i + WIDTH-1] = 3
   next

   for i=TOTAL-WIDTH to TOTAL-1
      squares[i] = 3
   next

   newBrick
   display
end sub

sub newBrick()
   xpos = WIDTH / 2 - 2
   ypos = -1
   counter = 0
   currentBrick = random(NUM_BRICKS)
   currentRot = 0
end sub

sub clear2()
   ' clear the squares occupied by the falling brick
   dim i as Integer
   for i=0 to TOTAL-1
      if squares[i] = 2 then
         squares[i] = 0
      end if
   next
end sub

function drawBrick(x as Integer, y as Integer, rot as Integer, fill as
Integer) as Boolean
   dim b as string
   dim index, i, xb, yb, x2, y2 as Integer

   clear2
   b = bricks[currentBrick * 4 + rot]
   for i=0 to 15
      xb = i mod 4
      yb = i / 4
      if b.getChar(i) = "1" then
         x2 = x + xb
         y2 = y + yb
         if y2 >= 0 then
            index = y2 * WIDTH + x2
            if x2 < 0 or x2 >= WIDTH or y2 >= HEIGHT or squares[index] <>
0 then
               drawBrick = false ' collision
               exit function
            elseif y2 >= 3 or fill <> 1 then
               squares[index] = fill
            end if
         end if
      end if
   next

   drawBrick = true ' no collision
end function

function isLineFilled(line as Integer) as Boolean
   dim i, index as Integer

   index = WIDTH * line
   for i=0 to WIDTH-1
      if squares[index+i] = 0 then
         isLineFilled = false
         exit function
      end if
   next

   isLineFilled = true
end function

sub eraseFullLines()
   dim line, l2, r as Integer
   for line = HEIGHT - 2 to 0 step -1
      if isLineFilled(line) then
         ' move lines one line down
         for l2 = line to 0 step -1
            for r=0 to WIDTH-1
               squares[l2*WIDTH+r] = squares[(l2-1)*WIDTH+r]
            next
         next
         ' clear top line
         for r=1 to WIDTH-2
            squares[r] = 0
         next
         line++ ' test the same line again
      end if
   next
end sub

sub OnKeyDown(keyCode as Integer)
   dim x, y, r as Integer
   x = xpos
   y = ypos
   r = currentRot

   if keyCode = KEY_RIGHT then
      x++
   elseif keyCode = KEY_LEFT then
      x--
   elseif keyCode = KEY_DOWN then
      y++
   elseif keyCode = KEY_S then
      initGame
      exit sub
   elseif keyCode = KEY_CONTROL then
      r = (r + 1) mod 4
   end if

   if r <> currentRot then
      if drawBrick(x, y, r, 2) then
         currentRot = r
      elseif drawBrick(x-1, y, r, 2) then
         xpos = x - 1
         currentRot = r
      elseif drawBrick(x+1, y, r, 2) then
         xpos = x + 1
         currentRot = r
      else
         drawBrick(xpos, ypos, currentRot, 2)
      end if
   elseif x <> xpos or y <> ypos then
      if drawBrick(x, y, r, 2) then
         xpos = x
         ypos = y
      else
         drawBrick(xpos, ypos, currentRot, 2)
      end if
   end if

   display
end sub

sub OnExecPerField()
   counter++
   if counter > 30 then
      counter = 0
      if drawBrick(xpos, ypos + 1, currentRot, 2) then
         ypos++
      else
         ' we hit the bottom
         drawBrick xpos, ypos, currentRot, 1
         eraseFullLines
         newBrick
      end if
      display
   end if
end sub
```

See Also

- Callback Procedures
- Global Procedures
- Array[Type]
- Boolean
- Container
- Integer
- String

20.12.12 Clip Channel

```
' ------------------------------------------------------------------
'
' ------------------------------------------------------------------
structure SClipInstance
   oClipChannel               as ClipChannel
   iClipChannelID             as Integer
   bLiveUpdate                as Boolean
   szShmKeyModeChanged        as String
   szShmKeyFrameChanged       as String
   bAutoRun                   as Boolean
   bLoopMode                  as Boolean
   bReverseFields             as Boolean
   iRepeatMode                as Integer
   iModeOnLoadError           as Integer
   dPlaybackSpeed             as Double
   iMode                      as Integer
   szClipName                 as String
   szClipNameToLoad           as String
   iDuration                  as Integer
   iCurrentFrame              as Integer
   iFrameIn                   as Integer
   iFrameOut                  as Integer
end structure

dim oClip    as Array[SClipInstance]
dim iClip    as Integer = 0

' ------------------------------------------------------------------
'
' ------------------------------------------------------------------
sub OnInit()
   println( "OnInit()" )
   println( "channel=" & GetParameterInt( "channel" ) )
   dim iNumberOfClipChannels = system.GetNumberOfClipChannel()

   for i = 0 to iNumberOfClipChannels
      ClipPlayerInitializeClipChannel( i )
   next
   iClip = 0
end sub


dim kButtonStop     as Integer = 0
dim kButtonPlay     as Integer = 1
dim kButtonPause    as Integer = 2
dim kButtonContinue as Integer = 3
dim kButtonFlush    as Integer = 4
dim kButtonLoad     as Integer = 5

' ------------------------------------------------------------------
'
' ------------------------------------------------------------------
sub OnInitParameters()
```

```
  println( "OnInitParameters()" )

  dim clip_channel_entries as Array[String]
  clip_channel_entries.push( "Clip Channel 1" )
  clip_channel_entries.push( "Clip Channel 2" )
  RegisterParameterDropDown("channel","Clip Channel",
0,clip_channel_entries, 20, 2)

  RegisterFileSelector( "file", "Browse Filename", "D:/clips", "",
"*.avi *.dv *.mov")
  RegisterPushButton( "cmd_load",     "Load Clip", kButtonLoad )

  RegisterPushButton( "cmd_stop",     "Stop",     kButtonStop )
  RegisterPushButton( "cmd_play",     "Play",     kButtonPlay )
  RegisterPushButton( "cmd_pause",    "Pause",    kButtonPause )
  RegisterPushButton( "cmd_continue", "Continue", kButtonContinue )
  RegisterPushButton( "cmd_flush",    "Flush",    kButtonFlush )

  RegisterParameterBool  ( "live_update",     "Live Update",  false )
  RegisterParameterString( "status_mode",     "Current Mode", "", 20,
20, "?" )
  RegisterParameterInt   ( "status_duration", "Duration (frames)", 0,
0, 2000000000 )
  RegisterParameterBool( "auto_run",       "Auto Run",       false )
  RegisterParameterBool( "loop_mode",      "Loop Mode",      true  )
  RegisterParameterBool( "reverse_fields", "Reverse Fields", false )

  dim repeat_mode_entries as Array[String]
  repeat_mode_entries.push( "none" )
  repeat_mode_entries.push( "field" )
  repeat_mode_entries.push( "frame" )
  RegisterParameterDropDown("repeat_mode","Repeat Mode", 1,
repeat_mode_entries, 20, 3 )

  dim mode_on_load_error_entries as Array[String]
  mode_on_load_error_entries.push( "none" )
  mode_on_load_error_entries.push( "pause" )
  mode_on_load_error_entries.push( "stop" )
  mode_on_load_error_entries.push( "flush" )
  RegisterParameterDropDown("mode_on_load_error","Mode on Load Error",
0, mode_on_load_error_entries, 20, 4 )

  RegisterParameterDouble("speed", "Speed", 1,-2.00,2.00)
  RegisterParameterInt("currentframe", "Current Frame", 0, 0,
2000000000)

  RegisterParameterInt("framein", "Frame In", 0, 0, 2000000000)
  RegisterParameterInt("frameout", "Frame Out", 0, 0, 2000000000)
end sub

' -------------------------------------------------------------------
'
' -------------------------------------------------------------------
sub OnParameterChanged( parameterName as String )
  println( "OnParameterChanged: " & parameterName )

  if parametername.StartsWith( "status_" ) then exit sub

  if oClip[iClip].iMode <> 2 then     println( "OnParameterChanged: "
& parameterName )

  dim cmd as string
  if "channel" == parameterName  then
    ClipPlayerChangeClipChannel( GetParameterInt( parameterName ))
  elseif "file" == parameterName then
    oClip[iClip].szClipNameToLoad = GetParameterString( parameterName
)
  elseif "auto_run" == parameterName then
    ClipPlayerSetAutoRun( GetParameterBool( parameterName ))
  elseif "loop_mode" == parameterName then
    ClipPlayerSetLoopMode( GetParameterBool( parameterName ))
  elseif "reverse_fields" == parameterName then
    ClipPlayerSetReverseFields( GetParameterBool( parameterName ))
  elseif "repeat_mode" == parameterName then
```

```
      ClipPlayerSetRepeatMode( GetParameterInt( parameterName ))
   elseif "mode_on_load_error" == parameterName then
      ClipPlayerSetModeOnLoadError( GetParameterInt( parameterName ))
   elseif "speed" == parameterName then
      ClipPlayerSetPlaybackSpeed( GetParameterDouble( parameterName ))
   elseif "currentframe" == parameterName then
      ClipPlayerScrub( GetParameterInt( parameterName ))
   elseif "framein" == parameterName then
      ClipPlayerSetFrameIn( GetParameterInt( parameterName ))
   elseif "frameout" == parameterName then
      ClipPlayerSetFrameOut( GetParameterInt( parameterName ))
   elseif "live_update" == parameterName then
      ClipPlayerSetLiveUpdate( GetParameterBool( parameterName ))
   end if
end sub

' ------------------------------------------------------------------
'
' ------------------------------------------------------------------
sub OnExecAction( buttonId as Integer )
   println("Pressed button: "&buttonId )
   if kButtonLoad == buttonId then
      ClipPlayerSetClipName( oClip[iClip].szClipNameToLoad )
   elseif kButtonStop == buttonId then
      ClipPlayerStop()
   elseif kButtonPlay == buttonId then
      ClipPlayerPlay()
   elseif kButtonPause == buttonId then
      ClipPlayerPause()
   elseif kButtonContinue == buttonId then
      ClipPlayerContinue()
   elseif kButtonFlush == buttonId then
      ClipPlayerFlush()
   end if
end sub

' ------------------------------------------------------------------
'
' ------------------------------------------------------------------
sub OnSharedMemoryVariableChanged( map as SharedMemory, mapKey as
String )
   ' Remove this as soon as shared memory is working.
   println( "shared memory variable: " & mapKey & " changed" )
   'exit sub

   if mapKey == oClip[iClip].szShmKeyModeChanged then
      oClip[iClip].iMode = CInt( map[ mapKey ] )
      UpdateAllFromClipPlayer( false )
   elseif mapKey == oClip[iClip].szShmKeyFrameChanged then
      oClip[iClip].iCurrentFrame = CInt( map[ mapKey ] )
      ClipPlayerChangedFrame()
   else
      println( "shared memory variable: " & mapKey & " changed to:" &
map[ mapKey ] )
   end if
end sub

' ------------------------------------------------------------------
'
' ------------------------------------------------------------------
function ModeToString(mode as Integer) as string
   if mode == 0 then
      ModeToString = "none"
   elseif mode == 1 then
      ModeToString = "stopped"
   elseif mode == 2 then
      ModeToString = "playing"
   elseif mode == 3 then
      ModeToString = "scrubbing"
   elseif mode == 4 then
      ModeToString = "ERROR"
   elseif mode == 5 then
      ModeToString = "playcont"
   else
```

```
      ModeToString = IntToString(mode, 1)
   end if
end function

' ----------------------------------------------------------------
'
' ----------------------------------------------------------------
sub UpdateAllFromClipPlayer( bDoGuiAlways as Boolean )
   if oClip[iClip].oClipChannel.VizId <= 0 then exit sub

   if ClipPlayerGetAutoRun() OR bDoGuiAlways then
      this.scriptPlugininstance.SetParameterBool("auto_run",
oClip[iClip].bAutoRun )
   end if

   if ClipPlayerGetLoopMode() OR bDoGuiAlways then
      this.scriptPlugininstance.SetParameterBool( "loop_mode",
oClip[iClip].bLoopMode )
   end if

   if ClipPlayerGetReverseFields() OR bDoGuiAlways then
      this.scriptPlugininstance.SetParameterBool( "reverse_fields",
oClip[iClip].bReverseFields )
   end if

   if ClipPlayerGetRepeatMode() OR bDoGuiAlways then
      this.scriptPlugininstance.SetParameterInt( "repeat_mode",
oClip[iClip].iRepeatMode )
   end if

   if ClipPlayerGetModeOnLoadError() OR bDoGuiAlways then
      this.scriptPlugininstance.SetParameterInt( "mode_on_load_error",
oClip[iClip].iModeOnLoadError )
   end if

   if ClipPlayerGetPlayBackSpeed() OR bDoGuiAlways then
      this.scriptPlugininstance.SetParameterDouble( "speed",
oClip[iClip].dPlayBackSpeed )
   end if

   if ClipPlayerGetFrameIn() OR bDoGuiAlways then
      this.scriptPlugininstance.SetParameterInt( "framein",
oClip[iClip].iFrameIn )
   end if

   if ClipPlayerGetFrameOut() OR bDoGuiAlways then
      this.scriptPlugininstance.SetParameterInt( "frameout",
oClip[iClip].iFrameOut )
   end if

   if ClipPlayerGetClipName() OR bDoGuiAlways then
      this.scriptPlugininstance.SetParameterString( "file",
oClip[iClip].szClipName )
   end if

   this.scriptPlugininstance.SetParameterString( "status_mode",
ModeToString( oClip[iClip].iMode ))

   oClip[iClip].iDuration = oClip[iClip].oClipChannel.Duration

   this.scriptPlugininstance.SetParameterInt( "status_duration",
oClip[iClip].iDuration )

   if NOT oClip[iClip].bLiveUpdate then
      dim iCurrentFrame as Integer =
oClip[iClip].oClipChannel.CurrentFrame
      if iCurrentFrame <> oClip[iClip].iCurrentFrame then
         oClip[iClip].iCurrentFrame =
oClip[iClip].oClipChannel.CurrentFrame
         this.scriptPlugininstance.SetParameterInt( "currentframe",
oClip[iClip].iCurrentFrame )
      end if
   end if
   SendGuiRefresh()
end sub

' ----------------------------------------------------------------------
'
' ----------------------------------------------------------------------
sub ClipPlayerChangedFrame()
   if oClip[iClip].oClipChannel.VizId <= 0 then exit sub

   if oClip[iClip].bLiveUpdate then
      this.scriptPlugininstance.SetParameterInt( "currentframe",
oClip[iClip].iCurrentFrame )
      SendGuiRefresh()
   end if
end sub

' ----------------------------------------------------------------------
'
' ----------------------------------------------------------------------
sub ClipPlayerInitializeClipChannel( iClipChannel as Integer )
   dim oClipInst   as SClipInstance
   iClip = iClipChannel

   oClipInst.oClipChannel = system.GetClipChannel( iClip )
   oClipInst.iClipChannelID = oClipInst.oClipChannel.ClipChannelID
   oClipInst.bLiveUpdate = false
   println( "[" & iClip & "]iClipChannelID=" & oClipInst.iClipChannelID
)
   println( "[" & iClip & "]object id:=" & oClipInst.oClipChannel.VizId
)
   oClip.push( oClipInst )

   ClipPlayerGetRegisteredChangedClipModeKey()
   ClipPlayerCreateChangedClipModeKey()

   ClipPlayerGetRegisteredChangedClipFrameKey()
   ClipPlayerCreateChangedClipFrameKey()

   ClipPlayerGetCurrentMode()
   ClipPlayerGetCurrentFrame()
   UpdateAllFromClipPlayer( true )
end sub

' ----------------------------------------------------------------------
'
' ----------------------------------------------------------------------
sub ClipPlayerChangeClipChannel( iClipChannel as Integer )
   if oClip[iClip].bLiveUpdate then
      ClipPlayerUnRegisterChanges()
   end if

   iClip = iClipChannel
   ClipPlayerGetCurrentMode()
   ClipPlayerGetCurrentFrame()

   UpdateAllFromClipPlayer( true )

   if oClip[iClip].bLiveUpdate then
      ClipPlayerRegisterChanges()
   end if
end sub

' ----------------------------------------------------------------------
'
' ----------------------------------------------------------------------
sub ClipPlayerGetRegisteredChangedClipModeKey()
   if oClip[iClip].oClipChannel.VizId <= 0 then exit sub

   oClip[iClip].szShmKeyModeChanged =
oClip[iClip].oClipChannel.GetRegisterChangedClipMode()

   println( "[" & iClip & "]GetRegisterChangedClipMode()=" &
oClip[iClip].szShmKeyModeChanged )
end sub
```

```
' ---------------------------------------------------------------
'
' ---------------------------------------------------------------
sub ClipPlayerGetRegisteredChangedClipFrameKey()
   if oClip[iClip].oClipChannel.VizId == 0 then exit sub

   oClip[iClip].szShmKeyFrameChanged =
oClip[iClip].oClipChannel.GetRegisterChangedClipFrame()
   println( "[" & iClip & "]GetRegisterChangedClipFrame()=" &
oClip[iClip].szShmKeyFrameChanged )
end sub

' ---------------------------------------------------------------
'
' ---------------------------------------------------------------
sub ClipPlayerCreateChangedClipModeKey()
   if oClip[iClip].szShmKeyModeChanged.length == 0 then
      oClip[iClip].szShmKeyModeChanged = "com.vizrt.clip" &
IntToString( oClip[iClip].iClipChannelID , 1 ) & ".mode"
   end if
   dim oShmMap as SharedMemory
   if Scene.Map.ContainsKey( oClip[iClip].szShmKeyModeChanged ) then
      oShmMap = Scene.Map
   elseif System.Map.ContainsKey( oClip[iClip].szShmKeyModeChanged )
then
      oShmMap = System.Map
   else
      oShmMap = Scene.Map
      oShmMap.CreateKey( oClip[iClip].szShmKeyModeChanged )
   end if

   ' oClip.oClipChannel.RegisterChangedClipMode(
oClip[iClip].szShmKeyModeChanged )
   oShmMap.RegisterChangedCallback( oClip[iClip].szShmKeyModeChanged )
end sub

' ---------------------------------------------------------------
'
' ---------------------------------------------------------------
sub ClipPlayerCreateChangedClipFrameKey()
   if oClip[iClip].szShmKeyFrameChanged.length == 0 then
      oClip[iClip].szShmKeyFrameChanged = "com.vizrt.clip" &
IntToString( oClip[iClip].iClipChannelID , 1 ) & ".frame"
   end if

   dim oShmMap as SharedMemory
   if Scene.Map.ContainsKey( oClip[iClip].szShmKeyFrameChanged ) then
      oShmMap = Scene.Map
   elseif System.Map.ContainsKey( oClip[iClip].szShmKeyFrameChanged )
then
      oShmMap = System.Map
   else
      oShmMap = Scene.Map
      oShmMap.CreateKey( oClip[iClip].szShmKeyFrameChanged )
   end if

   ' oClip[iClip].oClipChannel.RegisterChangedClipFrame(
oClip[iClip].szShmKeyFrameChanged )
   oShmMap.RegisterChangedCallback( oClip[iClip].szShmKeyFrameChanged )
end sub

' ---------------------------------------------------------------
'
' ---------------------------------------------------------------
function ClipPlayerSetClipName( szClipName as String ) as Boolean
   ' if oClip[iClip].szClipName <> szClipName then
      oClip[iClip].szClipName = szClipName
      oClip[iClip].oClipChannel.SetClipName( oClip[iClip].szClipName )
      ClipPlayerSetClipName = true
      println( "[" & iClip & "]SetClipName(" & oClip[iClip].szClipName
& ")" )
   ' else
      ' ClipPlayerSetClipName = false
   ' end if
end function

' --------------------------------------------------------------------
'
' --------------------------------------------------------------------
function ClipPlayerGetClipName() as Boolean
   dim szClipName as String = oClip[iClip].oClipChannel.GetClipName()
   if oClip[iClip].szClipName <> szClipName then
      oClip[iClip].szClipName = szClipName
      ClipPlayerGetClipName = true
      println( "[" & iClip & "]GetClipName=" & oClip[iClip].szClipName
   else
      ClipPlayerGetClipName = false
   end if
end function

' --------------------------------------------------------------------
'
' --------------------------------------------------------------------
function ClipPlayerSetAutoRun( bAutoRun as Boolean ) as Boolean
   if oClip[iClip].bAutoRun <> bAutoRun then
      oClip[iClip].bAutoRun = bAutoRun
      oClip[iClip].oClipChannel.AutoRun = oClip[iClip].bAutoRun
      ClipPlayerSetAutoRun = true
      println( "[" & iClip & "]SetAutoRun(" & oClip[iClip].bAutoRun &
")" )
   else
      ClipPlayerSetAutoRun = false
   end if
end function

' --------------------------------------------------------------------
'
' --------------------------------------------------------------------
function ClipPlayerGetAutoRun() as Boolean
   dim bAutoRun as Boolean = oClip[iClip].oClipChannel.AutoRun
   if oClip[iClip].bAutoRun <> bAutoRun then
      oClip[iClip].bAutoRun = bAutoRun
      ClipPlayerGetAutoRun = true
      println( "[" & iClip & "]GetAutoRun=" & oClip[iClip].bAutoRun )
   else
      ClipPlayerGetAutoRun = false
   end if
   end function

' --------------------------------------------------------------------
'
' --------------------------------------------------------------------
function ClipPlayerSetLoopMode( bLoopMode as Boolean ) as Boolean
   if oClip[iClip].bLoopMode <> bLoopMode then
      oClip[iClip].bLoopMode = bLoopMode
      oClip[iClip].oClipChannel.LoopMode = oClip[iClip].bLoopMode
      ClipPlayerSetLoopMode = true
      println( "[" & iClip & "]SetLoopMode(" & oClip[iClip].bLoopMode &
")" )
   else
      ClipPlayerSetLoopMode = false
   end if
end function

' --------------------------------------------------------------------
'
' --------------------------------------------------------------------
function ClipPlayerGetLoopMode() as Boolean
   dim bLoopMode as Boolean = oClip[iClip].oClipChannel.LoopMode
   if oClip[iClip].bLoopMode <> bLoopMode then
      oClip[iClip].bLoopMode = bLoopMode
      ClipPlayerGetLoopMode = true
      println( "[" & iClip & "]GetLoopMode=" & oClip[iClip].bLoopMode )
   else
      ClipPlayerGetLoopMode = false
   end if
end function
```

```
' ----------------------------------------------------------------
'
' ----------------------------------------------------------------
function ClipPlayerSetReverseFields( bReverseFields as Boolean ) as
Boolean
   if oClip[iClip].bReverseFields <> bReverseFields then
      oClip[iClip].bReverseFields = bReverseFields
      oClip[iClip].oClipChannel.ReverseFields =
oClip[iClip].bReverseFields
      ClipPlayerSetReverseFields = true
      println( "[" & iClip & "]SetReverseFields(" &
oClip[iClip].bReverseFields & ")" )
   else
      ClipPlayerSetReverseFields = false
   end if
end function

' ----------------------------------------------------------------
'
' ----------------------------------------------------------------
function ClipPlayerGetReverseFields() as Boolean
   dim bReverseFields as Boolean =
oClip[iClip].oClipChannel.ReverseFields
   if oClip[iClip].bReverseFields <> bReverseFields then
      oClip[iClip].bReverseFields = bReverseFields
      ClipPlayerGetReverseFields = true
      println( "[" & iClip & "]GetReverseFields=" &
oClip[iClip].bReverseFields )
   else
      ClipPlayerGetReverseFields = false
   end if
end function

' ----------------------------------------------------------------
'
' ----------------------------------------------------------------
function ClipPlayerSetRepeatMode( iRepeatMode as Integer ) as Boolean
   if oClip[iClip].iRepeatMode <> iRepeatMode then
      oClip[iClip].iRepeatMode = iRepeatMode
      oClip[iClip].oClipChannel.RepeatMode = oClip[iClip].iRepeatMode
      ClipPlayerSetRepeatMode = true
      println( "[" & iClip & "]SetRepeatMode(" &
oClip[iClip].iRepeatMode & ")" )
   else
      ClipPlayerSetRepeatMode = false
   end if
end function

' ----------------------------------------------------------------
'
' ----------------------------------------------------------------
function ClipPlayerGetRepeatMode() as Boolean
   dim iRepeatMode as Integer = oClip[iClip].oClipChannel.RepeatMode
   if oClip[iClip].iRepeatMode <> iRepeatMode then
      oClip[iClip].iRepeatMode = iRepeatMode
      ClipPlayerGetRepeatMode = true
      println( "[" & iClip & "]GetRepeatMode=" &
oClip[iClip].iRepeatMode )
   else
      ClipPlayerGetRepeatMode = false
   end if
end function

' ----------------------------------------------------------------
'
' ----------------------------------------------------------------
function ClipPlayerSetModeOnLoadError( iModeOnLoadError as Integer ) as
Boolean
   if oClip[iClip].iModeOnLoadError <> iModeOnLoadError then
      oClip[iClip].iModeOnLoadError = iModeOnLoadError
      oClip[iClip].oClipChannel.ModeOnLoadError =
oClip[iClip].iModeOnLoadError
      ClipPlayerSetModeOnLoadError = true
      println( "[" & iClip & "]SetModeOnLoadError(" &
oClip[iClip].iModeOnLoadError & ")" )
   else
      ClipPlayerSetModeOnLoadError = false
   end if
end function

' ------------------------------------------------------------------
'
' ------------------------------------------------------------------
function ClipPlayerGetModeOnLoadError() as Boolean
   dim iModeOnLoadError as Integer =
oClip[iClip].oClipChannel.ModeOnLoadError
   if oClip[iClip].iModeOnLoadError <> iModeOnLoadError then
      oClip[iClip].iModeOnLoadError = iModeOnLoadError
      ClipPlayerGetModeOnLoadError = true
      println( "[" & iClip & "]GetModeOnLoadError=" &
oClip[iClip].iModeOnLoadError )
   else
      ClipPlayerGetModeOnLoadError = false
   end if
end function

' ------------------------------------------------------------------
'
' ------------------------------------------------------------------
function ClipPlayerSetPlayBackSpeed( dPlaybackSpeed as Double ) as
Boolean
   if oClip[iClip].dPlaybackSpeed <> dPlaybackSpeed then
      oClip[iClip].dPlaybackSpeed = dPlaybackSpeed
      oClip[iClip].oClipChannel.PlayBackSpeed =
oClip[iClip].dPlaybackSpeed
      ClipPlayerSetPlayBackSpeed = true
   println( "[" & iClip & "]SetPlayBackSpeed(" &
oClip[iClip].dPlaybackSpeed & ")" )
   else
      ClipPlayerSetPlayBackSpeed = false
   end if
end function

' ------------------------------------------------------------------
'
' ------------------------------------------------------------------
function ClipPlayerGetPlayBackSpeed() as Boolean
   dim dPlaybackSpeed as Double =
oClip[iClip].oClipChannel.PlayBackSpeed
   if oClip[iClip].dPlaybackSpeed <> dPlaybackSpeed then
      oClip[iClip].dPlaybackSpeed = dPlaybackSpeed
      ClipPlayerGetPlayBackSpeed = true
      println( "[" & iClip & "]GetPlayBackSpeed=" &
oClip[iClip].dPlaybackSpeed )
   else
      ClipPlayerGetPlayBackSpeed = false
   end if
end function

' ------------------------------------------------------------------
'
' ------------------------------------------------------------------
function ClipPlayerSetFrameIn( iFrameIn as Integer ) as Boolean
   if oClip[iClip].iFrameIn <> iFrameIn then
      oClip[iClip].iFrameIn = iFrameIn
      oClip[iClip].oClipChannel.FrameIn = oClip[iClip].iFrameIn
      ClipPlayerSetFrameIn = true
      println( "[" & iClip & "]SetFrameIn(" & oClip[iClip].iFrameIn &
")" )
   else
      ClipPlayerSetFrameIn = false
   end if
end function

' ------------------------------------------------------------------
'
' ------------------------------------------------------------------
```

```
function ClipPlayerGetFrameIn() as Boolean
   dim iFrameIn as Integer = oClip[iClip].oClipChannel.FrameIn
   if oClip[iClip].iFrameIn <> iFrameIn then
      oClip[iClip].iFrameIn = iFrameIn
      ClipPlayerGetFrameIn = true
      println( "[" & iClip & "]GetFrameIn=" & oClip[iClip].iFrameIn )
   else
      ClipPlayerGetFrameIn = false
   end if
end function

' -----------------------------------------------------------------
'
' -----------------------------------------------------------------
function ClipPlayerSetFrameOut( iFrameOut as Integer ) as Boolean
   if oClip[iClip].iFrameOut <> iFrameOut then
      oClip[iClip].iFrameOut = iFrameOut
      oClip[iClip].oClipChannel.FrameOut = oClip[iClip].iFrameOut
      ClipPlayerSetFrameOut = true
      println( "[" & iClip & "]SetFrameOut(" & oClip[iClip].iFrameOut &
")" )
   else
      ClipPlayerSetFrameOut = false
   end if
end function

' -----------------------------------------------------------------
'
' -----------------------------------------------------------------
function ClipPlayerGetFrameOut() as Boolean
   dim iFrameOut as Integer = oClip[iClip].oClipChannel.FrameOut
   if oClip[iClip].iFrameOut <> iFrameOut then
      oClip[iClip].iFrameOut = iFrameOut
      ClipPlayerGetFrameOut = true
      println( "[" & iClip & "]GetFrameOut=" & oClip[iClip].iFrameOut )
   else
      ClipPlayerGetFrameOut = false
   end if
end function

' -----------------------------------------------------------------
'
' -----------------------------------------------------------------
function ClipPlayerGetCurrentMode() as Boolean
   dim iMode as Integer = oClip[iClip].oClipChannel.PlayMode
   if oClip[iClip].iMode <> iMode then
      oClip[iClip].iMode = iMode
      ClipPlayerGetCurrentMode = true
      println( "[" & iClip & "]GetCurrentMode=" & oClip[iClip].iMode &
")" )
   else
      ClipPlayerGetCurrentMode = false
   end if
end function

' -----------------------------------------------------------------
'
' -----------------------------------------------------------------
function ClipPlayerGetCurrentFrame() as Boolean
   dim iCurrentFrame as Integer =
oClip[iClip].oClipChannel.CurrentFrame
   if oClip[iClip].iCurrentFrame <> iCurrentFrame then
      oClip[iClip].iCurrentFrame = iCurrentFrame
      ClipPlayerGetCurrentFrame = true
      println( "[" & iClip & "]GetCurrentFrame=" &
oClip[iClip].iCurrentFrame & ")" )
   else
      ClipPlayerGetCurrentFrame = false
   end if
end function

' -----------------------------------------------------------------
'
' -----------------------------------------------------------------
```

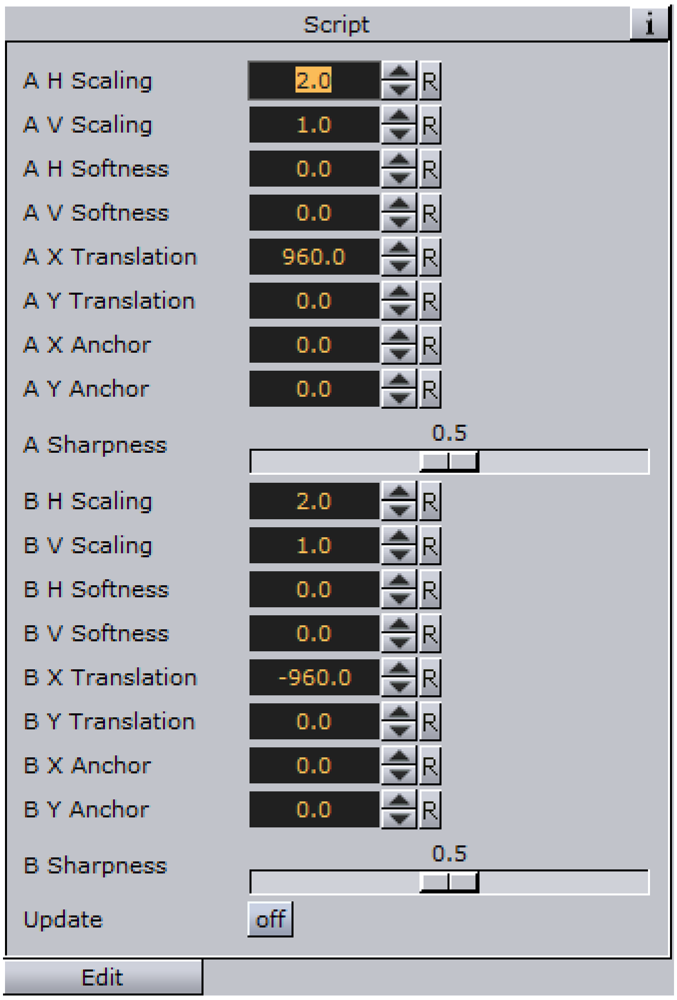
```
sub ClipPlayerPlay()
   oClip[iClip].oClipChannel.Play(0)
end sub

' ---------------------------------------------------------------------
'
' ---------------------------------------------------------------------
sub ClipPlayerPause()
   oClip[iClip].oClipChannel.Pause()
end sub

' ---------------------------------------------------------------------
'
' ---------------------------------------------------------------------
sub ClipPlayerContinue()
   oClip[iClip].oClipChannel.Continue()
end sub

' ---------------------------------------------------------------------
'
' ---------------------------------------------------------------------
sub ClipPlayerStop()
   oClip[iClip].oClipChannel.Stop()
end sub

' ---------------------------------------------------------------------
'
' ---------------------------------------------------------------------
sub ClipPlayerFlush()
   oClip[iClip].oClipChannel.Flush()
end sub

' ---------------------------------------------------------------------
'
' ---------------------------------------------------------------------
function ClipPlayerScrub( iScrubFrame as Integer ) as Boolean
   if oClip[iClip].iMode == 3 AND oClip[iClip].iCurrentFrame <>
iScrubFrame then
      oClip[iClip].iCurrentFrame = iScrubFrame
      oClip[iClip].oClipChannel.scrub( oClip[iClip].iCurrentFrame )
      ClipPlayerScrub = true
      println( "[" & iClip & "]ClipPlayerScrub" &
oClip[iClip].iCurrentFrame & ")" )
   else
      ClipPlayerScrub = false
   end if
end function

' ---------------------------------------------------------------------
'
' ---------------------------------------------------------------------
sub ClipPlayerSetLiveUpdate( bLiveUpdate as Boolean )
   oClip[iClip].bLiveUpdate = bLiveUpdate
   if oClip[iClip].bLiveUpdate then
      ClipPlayerRegisterChanges()
   else
      ClipPlayerUnRegisterChanges()
   end if
end sub

' ---------------------------------------------------------------------
'
' ---------------------------------------------------------------------
sub ClipPlayerRegisterChanges()
   oClip[iClip].oClipChannel.SetRegisterChangedClipMode (
oClip[iClip].szShmKeyModeChanged  )
   oClip[iClip].oClipChannel.SetRegisterChangedClipFrame(
oClip[iClip].szShmKeyFrameChanged )
end sub

' ---------------------------------------------------------------------
'
' ---------------------------------------------------------------------
sub ClipPlayerUnRegisterChanges()
```

```
      oClip[iClip].oClipChannel.SetRegisterChangedClipMode ("")
      oClip[iClip].oClipChannel.SetRegisterChangedClipFrame("")
   end sub
```

See Also

- Callback Procedures
- Global Procedures
- Array[Type]
- Boolean
- ClipChannel
- Double
- Integer
- String

20.12.13 Clip Channel - Stereo



Script

A H Scaling	2.0
A V Scaling	1.0
A H Softness	0.0
A V Softness	0.0
A X Translation	960.0
A Y Translation	0.0
A X Anchor	0.0
A Y Anchor	0.0
A Sharpness	0.5
B H Scaling	2.0
B V Scaling	1.0
B H Softness	0.0
B V Softness	0.0
B X Translation	-960.0
B Y Translation	0.0
B X Anchor	0.0
B Y Anchor	0.0
B Sharpness	0.5
Update	off

The following script example shows a scene that has a clip channel activated. The scene script shows the influence of each of the parameters. For details on the parameters, see the Controlling 3D Stereoscopic Clip Playback section.

```
   sub OnInitParameters()
      Dim Output_Width As Double = System.VideoWidth
      Dim Output_Height As Double = System.VideoHeight
      if System.OutputInterlaced then Output_Height *= 2

      println( "VideoWidth=" & Output_Width  )
      println( "VideoHeight=" & Output_Height  )


      RegisterParameterDouble( "vizrt.com.StereoClip.A.HScale",  "A H
   Scaling",     2.0, 0.0, 10.0 )
      RegisterParameterDouble( "vizrt.com.StereoClip.A.VScale",  "A V
   Scaling",     1.0, 0.0, 10.0 )
      RegisterParameterDouble( "vizrt.com.StereoClip.A.HSoft",   "A H
   Softness",    0.0, 0.0, 1.0 )
      RegisterParameterDouble( "vizrt.com.StereoClip.A.VSoft",   "A V
   Softness",    0.0, 0.0, 1.0 )
      RegisterParameterDouble( "vizrt.com.StereoClip.A.XTrans",  "A X
   Translation", Output_Width/2.0, -10000.0, 10000.0 )
      RegisterParameterDouble( "vizrt.com.StereoClip.A.YTrans",  "A Y
   Translation", 0.0, -10000.0, 10000.0 )
      RegisterParameterDouble( "vizrt.com.StereoClip.A.XAnchor", "A X
   Anchor",      0.0, -10000.0, 10000.0 )
      RegisterParameterDouble( "vizrt.com.StereoClip.A.YAnchor", "A Y
   Anchor",      0.0, -10000.0, 10000.0 )
      RegisterParameterSliderDouble( "vizrt.com.StereoClip.A.Sharp",   "A
   Sharpness",     0.5, 0.0, 1.0, 200 )

      RegisterParameterDouble( "vizrt.com.StereoClip.B.HScale",  "B H
   Scaling",     2.0, 0.0, 10.0 )
      RegisterParameterDouble( "vizrt.com.StereoClip.B.VScale",  "B V
   Scaling",     1.0, 0.0, 10.0 )
      RegisterParameterDouble( "vizrt.com.StereoClip.B.HSoft",   "B H
   Softness",    0.0, 0.0, 1.0 )
      RegisterParameterDouble( "vizrt.com.StereoClip.B.VSoft",   "B V
   Softness",    0.0, 0.0, 1.0 )
      RegisterParameterDouble( "vizrt.com.StereoClip.B.XTrans",  "B X
   Translation", -Output_Width/2.0, -10000.0, 10000.0 )
      RegisterParameterDouble( "vizrt.com.StereoClip.B.YTrans",  "B Y
   Translation", 0.0, -10000.0, 10000.0 )
      RegisterParameterDouble( "vizrt.com.StereoClip.B.XAnchor", "B X
   Anchor",      0.0, -10000.0, 10000.0 )
      RegisterParameterDouble( "vizrt.com.StereoClip.B.YAnchor", "B Y
   Anchor",      0.0, -10000.0, 10000.0 )
      RegisterParameterSliderDouble( "vizrt.com.StereoClip.B.Sharp",   "B
   Sharpness",     0.5, 0.0, 1.0, 200 )

      RegisterParameterBool(   "vizrt.com.StereoClip.changed",   "Update",
   false )

   end sub

   sub OnParameterChanged(parameterName As String)
      if  parameterName == "vizrt.com.StereoClip.changed" then
         this.Scene.Map[ "vizrt.com.StereoClip.changed" ] = CStr(
   GetParameterInt( parameterName ))

      elseif  parameterName == "vizrt.com.StereoClip.A.HScale" then
         this.Scene.Map[ parameterName ] = CStr( GetParameterDouble(
   parameterName ))

      elseif  parameterName == "vizrt.com.StereoClip.A.VScale" then
         this.Scene.Map[ parameterName ] = CStr( GetParameterDouble(
   parameterName ))

      elseif  parameterName == "vizrt.com.StereoClip.A.HSoft" then
         this.Scene.Map[ parameterName ] = CStr( GetParameterDouble(
   parameterName ))

      elseif  parameterName == "vizrt.com.StereoClip.A.VSoft" then
         this.Scene.Map[ parameterName ] = CStr( GetParameterDouble(
   parameterName ))

      elseif  parameterName == "vizrt.com.StereoClip.A.XTrans" then
         this.Scene.Map[ parameterName ] = CStr( GetParameterDouble(
   parameterName ))
```

```
   elseif  parameterName == "vizrt.com.StereoClip.A.YTrans" then
      this.Scene.Map[ parameterName ] = CStr( GetParameterDouble(
parameterName ))

   elseif  parameterName == "vizrt.com.StereoClip.A.XAnchor" then
      this.Scene.Map[ parameterName ] = CStr( GetParameterDouble(
parameterName ))

   elseif  parameterName == "vizrt.com.StereoClip.A.YAnchor" then
      this.Scene.Map[ parameterName ] = CStr( GetParameterDouble(
parameterName ))

   elseif  parameterName == "vizrt.com.StereoClip.A.Sharp" then
      this.Scene.Map[ parameterName ] = CStr( GetParameterDouble(
parameterName ))

   elseif  parameterName == "vizrt.com.StereoClip.B.HScale" then
      this.Scene.Map[ parameterName ] = CStr( GetParameterDouble(
parameterName ))

   elseif  parameterName == "vizrt.com.StereoClip.B.VScale" then
      this.Scene.Map[ parameterName ] = CStr( GetParameterDouble(
parameterName ))

   elseif  parameterName == "vizrt.com.StereoClip.B.HSoft" then
      this.Scene.Map[ parameterName ] = CStr( GetParameterDouble(
parameterName ))

   elseif  parameterName == "vizrt.com.StereoClip.B.VSoft" then
      this.Scene.Map[ parameterName ] = CStr( GetParameterDouble(
parameterName ))

   elseif  parameterName == "vizrt.com.StereoClip.B.XTrans" then
      this.Scene.Map[ parameterName ] = CStr( GetParameterDouble(
parameterName ))

   elseif  parameterName == "vizrt.com.StereoClip.B.YTrans" then
      this.Scene.Map[ parameterName ] = CStr( GetParameterDouble(
parameterName ))

   elseif  parameterName == "vizrt.com.StereoClip.B.XAnchor" then
      this.Scene.Map[ parameterName ] = CStr( GetParameterDouble(
parameterName ))

   elseif  parameterName == "vizrt.com.StereoClip.B.YAnchor" then
      this.Scene.Map[ parameterName ] = CStr( GetParameterDouble(
parameterName ))

   elseif  parameterName == "vizrt.com.StereoClip.B.Sharp" then
      this.Scene.Map[ parameterName ] = CStr( GetParameterDouble(
parameterName ))
   end if

end sub
```

21 Scripting Data Types and Procedures

This section contains information on all supported data types and procedures, how to share data and several script examples.

This section contains informaiton on the following high-level topics:

- Callback Procedures
- Data Sharing
- Data Types and Procedures

21.1 Callback Procedures

Here is a list of callback procedures recognized by the script engine. When defined in a script, these procedures are called by the system in response to events such as keyboard and mouse input. All of them are supported by container scripts, while only five (**OnInitParameters**, **OnInit**, **OnExecPerField**, **OnExecAction**, **OnParameterChanged**) are meaningful in scene scripts.

Sub **OnInitParameters**()

This procedure must contain all Register... calls (such as **RegisterParameterInt**), which are used to define the script's parameters. For locally defined scripts, this callback is invoked once for every script instance, immediately before **OnInit**, while for plugin scripts, it is called at system startup.

Sub **OnInit**()

Called after a script instance has been created (e.g. by pressing *Compile and Run* or by loading a scene with a script). Initialization code should be placed here.

Sub **OnEnter**()

Called when the mouse cursor enters the area occupied by the script's container in the output window. If the container's **ExactPicking** property, which is *true* by default, is set to *false*, it is sufficient that the mouse cursor hits the container's bounding box for **OnEnter** to be called.

Note: Due to the necessary object hit-testing (picking) using this callback can have impacts on the overall render performance.

Sub **OnLeave**()

The reverse of **OnEnter**: Called when the mouse cursor leaves the container's area. This procedure is called only if **OnEnter** has previously been called for the script's container.

Note: Due to the necessary object hit-testing (picking) using this callback can have impacts on the overall render performance.

Sub **OnEnterSubContainer**(*subContainer* as Container)

Called when the mouse cursor enters the area occupied by a sub-container of the script's container. Otherwise identical to **OnEnter**.

Note: Due to the necessary object hit-testing (picking) using this callback can have impacts on the overall render performance.

Sub **OnLeaveSubContainer**(*subContainer* as Container)

Called when the mouse cursor leaves the area occupied by a sub-container of the script's container.

Note: Due to the necessary object hit-testing (picking) using this callback can have impacts on the overall render performance.

Sub **OnExecPerField**()

Called once for every field.

Sub **OnExecAction**(*buttonId* as Integer)

Called when the user clicks on any push button defined by **RegisterPushButton**. If the script defines several push buttons, use *buttonId* to determine which button has been clicked on.

Sub **OnParameterChanged**()

Called whenever the user changes the value of script parameter defined by any of the **RegisterParameter...** functions.

Sub **OnGuiStatus**()

Called if the GUI wants to refresh the state of the UI. This would be the proper place where the plugin can set the UI state to enabled/disabled with **SendGuiStatus** or shown/hidden with **SendGuiParameterShow**.

Sub **OnKeyDown**(keyCode as Integer)

Called whenever the user presses a key. *keycode* identifies the key pressed. Possible values are: KEY_A, KEY_B, ..., KEY_Z, KEY_0, ..., KEY_9, KEY_HOME, KEY_END, KEY_PAGEUP, KEY_PAGEDN, KEY_UP, KEY_DOWN, KEY_LEFT, KEY_RIGHT, KEY_INSERT, KEY_DELETE, KEY_BEGIN, KEY_MULTIPLY, KEY_DIVIDE, KEY_ESCAPE, KEY_RETURN, KEY_ENTER, KEY_SEPARATOR, KEY_SPACE, KEY_BACKSPACE, KEY_TAB, KEY_CONTROL, KEY_MENU, KEY_ALT, KEY_SHIFT, KEY_F1, ... KEY_F12, KEY_SCROLL_LOCK, KEY_PAUSE, NUMPAD_INSERT, NUMPAD_END, NUMPAD_DOWN, NUMPAD_PGDN, NUMPAD_LEFT, NUMPAD_BEGIN, NUMPAD_RIGHT, NUMPAD_HOME, NUMPAD_UP, NUMPAD_PGUP, NUMPAD0, ..., NUMPAD9

Sub **OnKeyUp**(*keyCode* as Integer)

Called whenever the user releases a key.

Sub **OnButtonDown6DOF**(*button* as Integer, *pos* as Vertex, *rot* as Vertex)

Called when the user clicks on a scene grid. *button* defines the ID of the pressed button, *pos* specifies the 3D world position of the cursor and *rot* gives you the actual rotation.

Sub **OnButtonUp6DOF**(*button* as Integer, *pos* as Vertex, *rot* as Vertex)

Called whenever the user releases a mouse button.

Sub **OnMove6DOF**(*button* as Integer, *pos* as Vertex, *rot* as Vertex)

Called when the user moves the cursor on a scene grid. *button* defines the ID of the pressed button, *pos* specifies the 3D world position of the cursor and *rot* gives you the actual rotation.

Sub **OnMoveRelative6DOF**(*button* as Integer, *pos* as Vertex, *rot* as Vertex)

Called when the user moves the cursor on a scene grid. *button* defines the ID of the pressed button, *pos* specifies the offset vector to the last 6DOF position of the cursor and *rot* gives you the actual rotation.

Sub **OnScale6DOF**(*button* as Integer, *scale* as Vertex)

Called when the user performs a scale operation (e.g.: with a multi-touch device) on a scene grid. *button* defines the ID of the pressed button and *scale* specifies the actual scale factor.

Sub **OnMTHit**(*stroke* as Integer, *x* as Integer, *y* as Integer)

Called when the user touches this object (where this scriptplugininstance resides on) at a multi-touch device. *stroke* gives you the stroke ID of the multi-touch operation and *x* and *y* specify the hit position in screen coordinates. This callback is used to instantiate a certain control in the multi-touch server.

Note: Due to the necessary object hit-testing (picking) using this callback can have impacts on the overall render performance.

Sub **OnMTMenu**(*x* as Integer, *y* as Integer)

Called when the user performs a menu gesture on the multi-touch device. *x* and *y* specify the hit position in screen coordinates.

Note: Due to the necessary object hit-testing (picking) using this callback can have impacts on the overall render performance.

Sub **OnMTControlPZR2D**(*x* as Integer, *y* as Integer, *rot* as Vertex, *scale* as Vertex, *pressure* as Double)

Called when a PZR2D control was registered. *x* and *y* specify the hit position in screen coordinates, *rot* gives you the object rotation, *scale* holds the scale factor and *pressure* contains the current mean contact pressure.

Sub **OnMTControlButton**(*strokes* as Integer, *pressure* as Double)

Called when a Button control was registered. *strokes* contains the current hit count on the button object and *pressure* defines the mean contact pressure.

Sub **OnMTControlInactive**()

Called when a registered multi-touch control became inactive (all strokes were removed from the object).

Sub **OnMouseMove**(*x* as Integer, *y* as Integer)

Called whenever the mouse cursor is moved within the output window. *x* and *y* specify the cursor position in screen coordinates.

Sub **OnLButtonDown**()
Sub **OnMButtonDown**()
Sub **OnRButtonDown**()

Called whenever the user presses the left/middle/right mouse button.

Note: Due to the necessary object hit-testing (picking) using these callbacks can have impacts on the overall render performance.

Sub **OnLButtonUp**()
Sub **OnMButtonUp**()
Sub **OnRButtonUp**()

Called whenever the user releases the left/middle/right mouse button.

Note: Due to the necessary object hit-testing (picking) using these callbacks can have impacts on the overall render performance.

Sub **OnMouseWheel**(*distance* as Integer)

Called whenever the mouse wheel is rotated within the output window.

Sub **OnSharedMemoryVariableChanged**(*map* as SharedMemory, *mapKey* as String)

Called when a variable in a *SharedMemory* map is changed. The *SharedMemory* map and the name of the variable are passed to this procedure as parameters. The name of the variable must previously be registered to the *SharedMemory* map by calling its *RegisterChangedCallback* procedure.

Sub **OnSharedMemoryVariableDeleted**(*map* as SharedMemory, *mapKey* as String)

Called when a variable in a *SharedMemory* map is deleted. The *SharedMemory* map and the name of the variable are passed to this procedure as parameters. The name of the variable must previously be registered to the *SharedMemory* map by calling its *RegisterChangedCallback* procedure.

sub **OnGeometryChanged**(*geom* as Geometry)

Called when a geometry changed which was registered with *RegisterChangedCallback* or *RegisterTextChangedCallback* before.

21.2 Data Sharing

This section describes the data sharing mechanism provided by **shared memory maps** (SSM). An SSM is a map holding user-defined variables indexed by a string. It is exposed to the script interface via the SharedMemory data type. The variables are stored as Variant objects and thus can hold objects of any other type. Any script has access to three SSMs:

- Scene.Map: This is the map local to the current scene. Every scene has one map that can be used to exchange data among the scripts in the scene.
- System.Map: The system-wide map. It allows for data sharing among the scenes currently loaded into memory.
- VizCommunication.Map: A distributed map that enables data sharing among the computers connected to one Viz Graphics Hub.

Example I

Below is an example that shows how a scene map can be used to pass data from one script to another (within the same scene).

The first script defines the **OnEnterSubContainer**, which is called whenever the mouse-cursor enters any of its sub-containers. In it, the name of the subcontainer is stored in the scene map under the key *subcontainer_name*.

```
' Script 1:
Sub OnEnterSubContainer(subContainer as Container)
   Scene.Map["subcontainer_name"] = subContainer.Name
End Sub
```

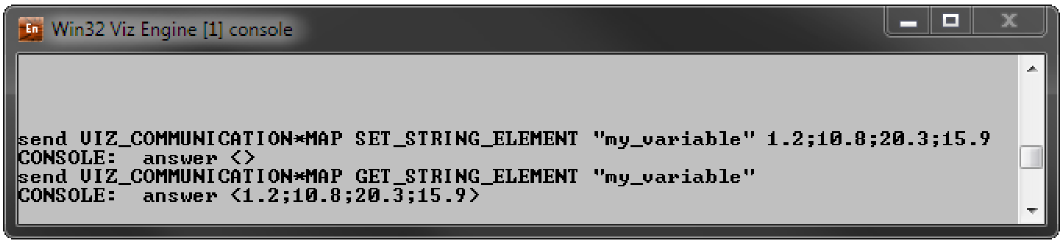
For further details, Sub OnEnterSubContainer(subContainer as Container).

In the second script, the previously stored value is retrieved, cast into a string, and output as text (assuming that the container holding the script has a text geometry).

```
' Script 2:
Sub OnExecPerField()
   Geometry.Text = (String)System.Map["subcontainer_name"]
End Sub
```

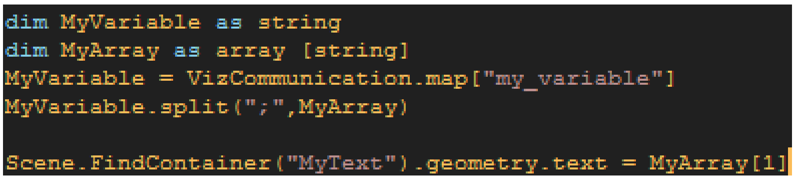
For further details, Sub OnExecPerField().

Example II



```
send VIZ_COMMUNICATION*MAP SET_STRING_ELEMENT "my_variable" 1.2;10.8;20.3;15.9
CONSOLE:   answer <>
send VIZ_COMMUNICATION*MAP GET_STRING_ELEMENT "my_variable"
CONSOLE:   answer <1.2;10.8;20.3;15.9>
```

The above image depicts a command being sent to Viz Engine assigning a value to a variable named `my_variable`. The below image depicts a script using that variable to assign one of its values to a text container named `MyText`.



```
dim MyVariable as string
dim MyArray as array [string]
MyVariable = VizCommunication.map["my_variable"]
MyVariable.split(";",MyArray)

Scene.FindContainer("MyText").geometry.text = MyArray[1]
```

From outside Viz Engine you can access the shared memory by connecting to Viz Engine’s listener port `6100`. For more information on ways to access shared memory, see the Accessing Maps through the Command Interface section.

So, to create and set a value shared by all Viz Engines, that are connected to the same Viz Graphics Hub, you can send the following command using the Viz Engine console:

```
send VIZ_COMMUNICATION*MAP SET_STRING_ELEMENT "my_variable"
1.2;10.8;20.3;15.9
CONSOLE:   answer <>

send VIZ_COMMUNICATION*MAP GET_STRING_ELEMENT "my_variable"
CONSOLE:   answer <1.2;10.8;20.3;15.9>
```

Inside Viz Engine you can also set and retrieve data from the shared memory through scripting.

You address the different areas as follows:

- Scene.map
- System.map
- VizCommunication.map

To set a value

```
send VIZ_COMMUNICATION*MAP SET_STRING_ELEMENT "my_variable"
1.2;10.8;20.3;15.9
```

or

```
VizCommunication.map["my_variable"] = 1.2;10.8;20.3;15.9
```

See AlsoTo retrieve a value

```
Dim MyVariable as string
MyVariable = (string)VizCommunication.map["my_variable"]
```

`string` is here used to format the data from the shared memory to a text value.

To split the values into an array

You can then split the data inside MyVariable into a table called MyArray:

```
Dim MyArray as array [string]
MyVariable.split(";",MyArray)
```

The information in MyVariable has now been split into 4:

```
' MyArray[0] = 1.2
' MyArray[1] = 10.8
' MyArray[2] = 20.3
' MyArray[3] = 15.9
```

To assign a value to a text container

One of these values can then be assigned to a text container:

```
Scene.FindContainer("MyText").geometry.text = MyArray[1]
```

To automatically update graphics

When a shared memory variable is updated you will have to register the value for updates in the scene:

```
Scene.Map.RegisterChangedCallback("my_variable")
```

Changes to `my_variable` will execute a subroutine called `OnSharedMemoryVariableChanged`. This subroutine can do whatever you decide with the data. For further details, see Callback Procedures.

This section also contains information on the following topics:

- Receiving Notification on Map Changes
- VizCommunication.Map
- Accessing Maps via the UDP Interface
- Accessing Maps through the Command Interface
- Accessing Maps from Within Plugins

21.2.1 Receiving Notification on Map Changes

Sometimes you want to be notified when a value in an SSM changes, so you do not have to poll for changes within `OnExecPerField`. This is done by calling a map's `RegisterChangedCallback` procedure, passing it the name of a key that you want to monitor. Now, whenever the variable identified by that key is changed, the `OnSharedMemoryVariableChanged` callback (if present) is invoked. Similarly, the `OnSharedMemoryVariableDeleted` callback is invoked, when the variable is deleted.

Example

```
Sub OnInit()
   Scene.Map.RegisterChangedCallback("subcontainer_name")
End Sub

Sub OnSharedMemoryVariableChanged(map as SharedMemory, mapKey as
String)
   If mapKey = "subcontainer_name" Then
      Geometry.Text = (String)map["subcontainer_name"]
   End If
End Sub
```

Note: *OnSharedMemoryVariableChanged* is passed the map as well as the key whose variable has been changed.

If you pass an empty string ("") to `RegisterChangedCallback`, `OnSharedMemoryVariableChanged` and `OnSharedMemoryVariableDeleted` will be invoked whenever any variable in the map is modified or deleted:

```
   Sub OnInit()
      Scene.Map.RegisterChangedCallback("")
   End Sub

   Sub OnSharedMemoryVariableChanged(map as SharedMemory, mapKey as
   String)
      ...
   End Sub
```

See Also

- Sub OnExecPerField()
- Sub OnSharedMemoryVariableChanged(map as SharedMemory, mapKey as String)
- Sub OnSharedMemoryVariableDeleted(map as SharedMemory, mapKey as String)

21.2.2 VizCommunication.Map

The map exposed by the VizCommunication type is used to share data among the client computers connected to Viz Graphics Hub. Whenever a script on one client adds a new value or changes an existing value in the map, the change is propagated to the other clients via database messaging to update the local copy of each client's map. Note, however, that this does not necessarily mean that all local client maps are identical. When a client starts up after other clients have made changes to the `VizCommunication.Map`, it is not informed of those changes. To work around this limitation, one client can save the entire contents of the map by calling the map's `SaveToDb` procedure, and another client can read it by calling `LoadFromDb`.

21.2.3 Accessing Maps via the UDP Interface

Elements in the distributed memory map (VizCommunication.Map) can be set (but not retrieved) via a UDP interface exposed by Viz Artist. This option is particularly useful when writing external control applications. A C++ library, the VizControlLib, is provided to encapsulate all required communication and serialization logic. The VizControlLib can be found in `<viz3 install directory>/Tools/ControlLib`. Please refer to the header files and the included test project for more information.

21.2.4 Accessing Maps through the Command Interface

There is limited support for getting and setting map elements via the command interface. The maps are accessible through these locations:

- `<scene>*MAP` for the scene specific map
- `GLOBAL*MAP` for system-wide map

- VIZ_COMMUNICATION*MAP for the distributed map

Here is a list of commands supported by each of these maps:

- SET_BOOLEAN_ELEMENT and GET_BOOLEAN_ELEMENT
- SET_INTEGER_ELEMENT and GET_INTEGER_ELEMENT
- SET_DOUBLE_ELEMENT and GET_DOUBLE_ELEMENT
- SET_STRING_ELEMENT and GET_STRING_ELEMENT

Examples:

```
send VIZ_COMMUNICATION*MAP SET_DOUBLE_ELEMENT "my_double" 1.2
send VIZ_COMMUNICATION*MAP GET_DOUBLE_ELEMENT "my_double"
=> Answer = "1.2"
```

21.2.5 Accessing Maps from Within Plugins

Functions for getting and setting map elments are provided by the plugin interface. For more information, refer to the *evPlugin_SHARED_MEMORY.h* file, which is part of the plugin SDK (<viz3 install directory>/Plugin/src).

21.3 Data Types and Procedures

This section contains a complete reference of the Viz Artist’s supported data types and procedures:

Table 36:

Data Types and Procedures			
Alpha	DateTime	Light	Stage
Array[Type]	Director	Material	String
Base	Double	Matrix	StringMap
BezierHandle	Expert	PluginInstance	System
Boolean	Geometry	Position	Texture
Camera	Grid	Rotation	Timecode
Center	Image	Scaling	Uuid
Channel	InfoText	Scene	Variant
ClipChannel	Integer	Script	Vertex
Color	Key	ScriptSettings	VizCommunication
Container	Keyframe	SharedMemory	Global Procedures

21.3.1 Alpha

Base type

Base

Description

An alpha object. This is used to control a container's alpha value.

Properties

Table 37:

Property	Description
Active as Double	
Scene as Scene [read-only]	(*Inherited from* Base) Gets the current scene.
Stage as Stage [read-only]	(*Inherited from* Base) Gets the current stage.
System as System [read-only]	(*Inherited from* Base) Gets system wide data.
Value as Double	The alpha value [0.0, 100.0]
VizCommunication as VizCommunication [read-only]	(*Inherited from* Base) Gets the VizCommunication object.
VizId as Integer	(*Inherited from* Base) Gets or sets the internal id of this object.

Member Procedures

Table 38:

Name	Description
Function **FindChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as Director), it returns null. You can limit the search to a particular director by using the syntax "directorName$channelName" for the channelName argument. Nested directors may be specified like this: "directorName1$directorName2$channelName".

Table 38:

Name	Description
Function **FindKeyframeOfObject**(keyframeName as String) as Keyframe	(*Inherited from* Base) Finds a keyframe acting on the object. If this function is called on a type that cannot be animated (such as Director), it returns null. You can limit the search to a particular director by using the syntax "directorName$keyframeName" for the keyframeName argument. Nested directors may be specified like this: "directorName1$directorName2$keyframeName".
Function **FindOrCreateChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) This function works like FindChannelOfObject except that if no animation channel with the specifed name exists, a new one is created.
Function **GetChannelsOfObject**([out] v as Array[Channel]) as Integer	(*Inherited from* Base) Fills the array v with the animation channels acting on this object, returning the number of channels. If this function is called on a type that cannot be animated (such as Director), v will be empty.
Function **IsAnimated**() as Boolean	(*Inherited from* Base) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as Director), it always returns false.
Sub **SetChanged**()	(*Inherited from* Base) Mark this object as changed.

21.3.2 Array[Type]

Description

A dynamic, one-dimensional array. The element type must be specified in square brackets. Use the index operator `[]` to access individual elements. Negative indices refer to elements at the end of the array.

Properties

Table 39:

Property	Description
LBound as Integer [read-only]	The index of the first elment (always 0)
Size as Integer	The size of the array (this member can be set to change the size)
UBound as Integer [read-only]	The index of the last elment (-1 if the array is empty)

Member Procedures

Table 40:

Name	Description
Sub **Clear**()	Erases all elements
Sub **Erase**(index as Integer)	Erases the element at 'index'.
Sub **Insert**(index as Integer, element as Type)	Inserts a new element at 'index'.
Sub **Pop**()	Removes the last element
Sub **Push**(element as Type)	Inserts a new element at the end of the array.
Function **Sort**(object as Base, comparisonFunction as String) as Boolean	Sorts the array. *comparisonFunction* must be the name of a user-defined comparison function that takes two arguments of type *Type* and returns an integer. The return value must be -1 if the the first arguement is smaller than the second, 1 if the first argument is greater than the second, and 0 if the arguments are equal. *object* is the this pointer passed to the comparison function. It may be *null* if the camparison function does not access any member variables. The comparison function must be defined in the same script as the *Type* (if it is a structure) or in the script where *Sort* is called.

21.3.3 Base

Description

A Viz object. This is the base class of all other Viz types.

Properties

Table 41:

Property	Description
Scene as Scene [read-only]	Gets the current scene.
Stage as Stage [read-only]	Gets the current stage.
System as System [read-only]	Gets system wide data.
VizCommunication as VizCommunication [read-only]	Gets the VizCommunication object.
VizId as Integer	Gets or sets the internal id of this object.
NumChannels as Integer [read-only]	**Deprecated**. Use *IsAnimated* or *GetChannelsOfObject* instead.

Member Procedures

Table 42:

Name	Description
Function **FindChannelOfObject**(channelName as String) as Channel	Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$channelName" for the *channelName* argument. Nested directors may be specified like this: "directorName1$directorName2$channelName".

Table 42:

Name	Description
Function **FindKeyframeOfObject**(keyframeName as String) as Keyframe	Finds a keyframe acting on the object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$keyframeName" for the *keyframeName* argument. Nested directors may be specified like this: "directorName1$directorName2$keyframeName".
Function **FindOrCreateChannelOfObject**(channelName as String) as Channel	This function works like FindChannelOfObject except that if no animation channel with the specifed name exists, a new one is created.
Function **GetChannelsOfObject**([out] v as Array[Channel]) as Integer	Fills the array *v* with the animation channels acting on this object, returning the number of channels. If this function is called on a type that cannot be animated (such as *Director*), *v* will be empty.
Function **IsAnimated**() as Boolean	Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as *Director*), it always returns *false*.
Sub **SetChanged**()	Mark this object as changed.
Function FindChannel(channelName as String) as Channel	**Deprecated**. Use *FindChannelOfObject*
Function FindKeyframe(keyframeName as String) as Keyframe	**Deprecated**. Use *FindKeyframeOfObject*
Function GetChannels([out] v as Array[Channel]) as Integer	**Deprecated**. Use *GetChannelsOfObject*
Function GetNumberOfAnimationChannels() as Integer	**Deprecated**. Use *IsAnimated* instead.

21.3.4 BezierHandle

Base type

Base

Description

A spline handle of a keyframe.

Properties

Table 43:

Property	Description
LeftAngle as Double	
LeftInterpolationMode as Integer	HANDLE_INTERPOLATION_MODE_SMOOTH HANDLE_INTERPOLATION_MODE_LINEAR
LeftWeight as Double	
LockMode as Integer	HANDLE_LOCK_MODE_LOCKED HANDLE_LOCK_MODE_SINGLE HANDLE_LOCK_MODE_MIRROR
RightAngle as Double	
RightInterpolationMode as Integer	HANDLE_INTERPOLATION_MODE_SMOOTH HANDLE_INTERPOLATION_MODE_LINEAR
RightWeight as Double	
Scene as Scene [read-only]	(*Inherited from* Base) Gets the current scene.
Stage as Stage [read-only]	(*Inherited from* Base) Gets the current stage.
System as System [read-only]	(*Inherited from* Base) Gets system wide data.
VizCommunication as VizCommunication [read-only]	(*Inherited from* Base) Gets the VizCommunication object.
VizId as Integer [read-only]	(*Inherited from* Base) Gets or sets the internal id of this object.

Member Procedures

Table 44:

Name	Description
Function **FindChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$channelName" for the *channelName* argument. Nested directors may be specified like this: "directorName1$directorName2$channelName".
Function **FindKeyframeOfObject**(keyframeName as String) as Keyframe	(*Inherited from* Base) Finds a keyframe acting on the object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$keyframeName" for the *keyframeName* argument. Nested directors may be specified like this: "directorName1$directorName2$keyframeName".
Function **FindOrCreateChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) This function works like FindChannelOfObject except that if no animation channel with the specifed name exists, a new one is created.
Function **GetChannelsOfObject**([out] v as Array[Channel]) as Integer	(*Inherited from* Base) Fills the array *v* with the animation channels acting on this object, returning the number of channels. If this function is called on a type that cannot be animated (such as *Director*), v will be empty.
Function **IsAnimated**() as Boolean	(*Inherited from* Base) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as *Director*), it always returns *false*.
Sub **SetChanged**()	(*Inherited from* Base) Mark this object as changed.

21.3.5 Boolean

Description

A flag that is either true or false.

21.3.6 Camera

Base type

Base

Description

A camera object. Camera objects are returned by the Scene's CurrentCamera and GetCamera members.

Properties

Table 45:

Property	Description
ActiveExt as Boolean	Activates/Deactivates the external camera.
Cx as Double	The center shift in x
CxExt as Double [read-only]	The center shift in x of the external camera
Cy as Double	The center shift in y
DecoupleExternal as Boolean	If true, viz receives external camera data without copying them to its internal camera structures for rendering.
Direction as Position	The direction of the camera
FocalPlane as Double	The focal plane
Fovx as Double	The field of view in x
FovxExt as Double [read-only]	The field of view in x of the external camera
Fovy as Double	The field of view in y
FovyExt as Double [read-only]	The field of view in y of the external camera
Matrix as Matrix [read-only]	The model matrix of the camera.
Pan as Double	The pan
PanExt as Double [read-only]	The pan of the external camera

Table 45:

Property	Description
Position as Position	The camera position
PositionExt as Position [read-only]	The position of the external camera
PositionTrackingActive as Boolean	Is position tracking active?
PositionTrackingContainer as Container	The tracking object for position tracking
ProjectionMatrix as Matrix [read-only]	The projection matrix of the camera.
RotationTrackingActive as Boolean	Is rotation tracking active?
RotationTrackingContainer as Container	The tracking object for rotation tracking
Scene as Scene [read-only]	(*Inherited from* Base) Gets the current scene.
Stage as Stage [read-only]	(*Inherited from* Base) Gets the current stage.
System as System [read-only]	(*Inherited from* Base) Gets system wide data.
Tilt as Double	The tilt
TiltExt as Double [read-only]	The tilt of the external camera
Twist as Double	The twist
TwistExt as Double [read-only]	The twist of the external camera
View as Integer	The view type: CAM_VIEW_PERSPECTIVE CAM_VIEW_ORTHOGONAL
ViewMatrix as Matrix [read-only]	The view matrix of the camera.
VizCommunication as VizCommunication [read-only]	(*Inherited from* Base) Gets the VizCommunication object.
VizId as Integer	(*Inherited from* Base) Gets or sets the internal id of this object.

Member procedures

21.3.7 Center

Table 46:

Name	Description
Function **FindChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$channelName" for the *channelName* argument. Nested directors may be specified like this: "directorName1$directorName2$channelName".
Function **FindKeyframeOfObject**(keyframeName as String) as Keyframe	(*Inherited from* Base) Finds a keyframe acting on the object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$keyframeName" for the *keyframeName* argument. Nested directors may be specified like this: "directorName1$directorName2$keyframeName".
Function **FindOrCreateChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) This function works like FindChannelOfObject except that if no animation channel with the specifed name exists, a new one is created.
Function **GetChannelsOfObject**([out] v as Array[Channel]) as Integer	(*Inherited from* Base) Fills the array v with the animation channels acting on this object, returning the number of channels. If this function is called on a type that cannot be animated (such as *Director*), *v* will be empty.
Function **IsAnimated**() as Boolean	(*Inherited from* Base) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as *Director*), it always returns *false*.

Table 46:

Name	Description
Sub **ResetPositionTracking**()	
Sub **ResetRotationTracking**()	
Sub **SetChanged**()	(*Inherited from* Base) Mark this object as changed.

21.3.8 Channel

Base type

Base

Description

A channel of a director.

Properties

Table 47:

Property	Description
Actor as Base [read-only]	
AnimationObject as Base [read-only]	
CommandName as String [read-only]	
Director as Director [read-only]	
FirstKeyframe as Keyframe [read-only]	Gets the leftmost keyframe.
KeyframeCount as Integer [read-only]	Gets the number of keyframes.
LastKeyframe as Keyframe [read-only]	Gets the rightmost keyframe.
Name as String	
PostLoopActive as Boolean	
PostLoopCounter as Integer	
PostSwingActive as Boolean	
PreLoopActive as Boolean	
PreLoopCounter as Integer	
PreSwingActive as Boolean	
Scene as Scene [read-only]	(*Inherited from* Base) Gets the current scene.

Table 47:

Property	Description
Stage as Stage [read-only]	(*Inherited from* Base) Gets the current stage.
System as System [read-only]	(*Inherited from* Base) Gets system wide data.
VizCommunication as VizCommunication [read-only]	(*Inherited from* Base) Gets the VizCommunication object.
VizId as Integer	(*Inherited from* Base) Gets or sets the internal id of this object.
NumKeyframes as Integer *[read-only]*	*Deprecated. Use* **KeyframeCount**.

Member procedures

Table 48:

Name	Description
Function **AddKeyframe**(time as Double) as Keyframe	
Function **CopyToDirector**(destination as Director) as Boolean	
Sub **Delete**()	
Function **DeleteKeyframe**(kf as Keyframe) as Boolean	
Function **DeleteKeyframe**(index as Integer) as Boolean	
Function **FindChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$channelName" for the *channelName* argument. Nested directors may be specified like this: "directorName1$directorName2$channelName".
Function **FindKeyframe**(keyframeName as String) as Keyframe	Searches for a keyframe named keyframeName.

Table 48:

Name	Description
Function **FindKeyframeOfObject**(keyframeName as String) as Keyframe	(*Inherited from* Base) Finds a keyframe acting on the object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$keyframeName" for the *keyframeName* argument. Nested directors may be specified like this: "directorName1$directorName2$keyframeName".
Function **FindOrCreateChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) This function works like FindChannelOfObject except that if no animation channel with the specifed name exists, a new one is created.
Function **GetChannelsOfObject**([out] v as Array[Channel]) as Integer	(*Inherited from* Base) Fills the array v with the animation channels acting on this object, returning the number of channels. If this function is called on a type that cannot be animated (such as *Director*), *v* will be empty.
Function **GetKeyframe**(n as Integer) as Keyframe	Returns the n'th keyframe of the channel.
Function **IsAnimated**() as Boolean	(*Inherited from* Base) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as Director), it always returns false.
Function **MoveToDirector**(destination as Director) as Boolean	
Sub **Rebuild**()	Rebuilds the channel after it has been modified. Normally, there is no need to call this procedure.
Sub **SetChanged**()	(*Inherited from* Base) Mark this object as changed.

21.3.9 ClipChannel

Base type

Base

Description

A ClipChannel object. The *ClipChannel* member of the Base type provides access to the clip channel. Use System.GetClipChannel(number as Integer), whereas *number* >= 0, to get a ClipChannel instance. examples section ClipChannel.

Properties

Table 49:

Property	Description
AutoRun as Boolean	Get/Set the autorun mode. If *true* then the clip start playing automatically upon load.
ClipChannelID as Integer [read-only]	Get the clip channel number.
CurrentFrame as Integer [read-only]	Get the current frame.
CurrentFramePending as Integer [read-only]	Get the current frame of the pending clip.
Duration as Integer [read-only]	Get the number of frames of the current clip.
DurationPending as Integer [read-only]	Get the number of frames of the pending clip.
FrameIn as Integer	Get/Set the *In* frame.
FrameInActive as Integer	Get/Set the *In* frame on the active clip player.
FrameInPending as Integer	Get/Set the *In* frame on the pending clip player.
FrameOut as Integer	Get/Set the *Out* frame.
FrameOutActive as Integer	Get/Set the *Out* frame on the active clip player.
FrameOutPending as Integer	Get/Set the *Out* frame on the pending clip player.
LoopMode as Boolean	Get/Set the loop mode. If *true* (=default), then the clip will be looped.
LoopModeActive as Boolean	Get/Set the loop mode on the active clip player. If *true* (=default), then the clip will be looped.
LoopModePending as Boolean	Get/Set the loop mode on the pending clip player. If *true* (=default), then the clip will be looped.

Table 49:

Property	Description
ModeOnLoadError as Integer	Determines the mode upon an error situation when loading a clip. none|pause|stop|flush
ModeOnLoadError_FLUSH as Integer [read-only]	Constant for *mode on load error* error flush.
ModeOnLoadError_NONE as Integer [read-only]	Constant for *mode on load error* none.
ModeOnLoadError_PAUSE as Integer [read-only]	Constant for *mode on load error* pause.
ModeOnLoadError_STOP as Integer [read-only]	Constant for *mode on load error* stop.
PendingClipSupported as Boolean [read-only]	Determines if support for loading pending clips is available.
PlayBackSpeed as Double	Get/Set the playback speed.
PlayBackSpeedActive as Double	Get/Set the playback speed on the active clip player.
PlayBackSpeedPending as Double	Get/Set the playback speed on the pending clip palyer.
PlayMode as Integer [read-only]	Get the current mode of the clip channel.
PlayMode_ERROR as Integer [read-only]	Constant for *current mode* ERROR.
PlayMode_NONE as Integer [read-only]	Constant for *current mode* NONE.
PlayMode_PLAY as Integer [read-only]	Constant for *current mode* PLAYING.
PlayMode_PLAY_PENDING as Integer [read-only]	Constant for *current mode* PLAYING_PENDING.
PlayMode_SCRUB as Integer [read-only]	Constant for *current mode* SCRUBBING.
PlayMode_SCRUB_EOF as Integer [read-only]	Constant for *current mode* SCRUBBING_EOF.
PlayMode_STOP as Integer [read-only]	Constant for *current mode* STOP.
PlayModePending as Integer [read-only]	Get the current mode of the pending clip player.
RepeatMode as Integer	Get/Set the *repeat mode* for playback and pause. none|field|frame.
RepeatMode_FIELD as Integer [read-only]	Constant for *repeat mode* field.

Table 49:

Property	Description
RepeatMode_FRAME as Integer [read-only]	Constant for *repeat mode* frame.
RepeatMode_NONE as Integer [read-only]	Constant for *repeat mode* none.
RepeatModeActive as Integer	Get/Set the repeat mode for playback and pause on the active clip player. none|field|frame.
RepeatModePending as Integer	Get/Set the repeat mode for playback and pause on the pending clip player. none|field|frame.
ReverseFields as Boolean	If *true* swap fields when playing interlaced material backwards.
ReverseFieldsActive as Boolean	If *true* swap fields when playing interlaced material backwards on the active clip player.
ReverseFieldsPending as Boolean	If *true* swap fields when playing interlaced material backwards on the pending clip player.
Scene as Scene [read-only]	(*Inherited from* Base) Gets the current scene.
Stage as Stage [read-only]	(*Inherited from* Base) Gets the current stage.
System as System [read-only]	(*Inherited from* Base) Gets system wide data.
UpscaleLuma as Boolean	If *true* then upscale the luma value in the separate key file.
UpscaleLumaActive as Boolean	If *true* then upscale the luma value in the separate key file on the active clip player.
UpscaleLumaPending as Boolean	If *true* then upscale the luma value in the separate key file on the pending clip player.
VizCommunication as VizCommunication [read-only]	(*Inherited from* Base) Gets the VizCommunication object.
VizId as Integer	(*Inherited from* Base) Gets or sets the internal id of this object.

21.3.10 Color

Description

A color object consisting of 4 members: red, green blue, alpha.

Properties

Table 50:

Property	Description
Alpha as Double	The alpha component [0.0, 1.0]
Blue as Double	The blue component [0.0, 1.0]
Green as Double	The green component [0.0, 1.0]
Red as Double	The red component [0.0, 1.0]
Rgb as Vertex	The red/green/blue components combined into a Vertex value

21.3.11 Container

Base type

Base

Description

A container object.

Properties

Table 51:

Property	Description
Active as Boolean	The visibility state
Alpha as Alpha	The alpha object
Center as Center	The center object
ChildContainer as Container [read-only]	The first child container
ChildContainerCount as Integer [read-only]	Gets the number of child containers.
ExactPicking as Boolean	Enables/Disables exact picking for OnEnter/OnLeave.

Table 51:

Property	Description
Expert as Expert	The expert object
FirstChildContainer as Container [read-only]	The first child container
Geometry as Geometry	The geometry object
InfoText as InfoText	The infotext object
Key as Key	The key object
LastChildContainer as Container [read-only]	The last child container
Material as Material	The material object
Matrix as Matrix [read-only]	The model matrix of the cotainer.
Merged as Boolean [read-only]	Indicates whether the container is merged.
MergedChildContainer as Container [read-only]	The first child container within a merged object
MergedParentContainer as Container [read-only]	The merged parent container
Name as String	The name of the container
NextContainer as Container [read-only]	The next container in the tree
Open as Boolean	Indicates whether the container is open, i.e. its subcontainers are visible in the tree
ParentContainer as Container [read-only]	The parent container
Position as Position	The position object
PreviousContainer as Container [read-only]	The previous container
Rotation as Rotation	The rotation object
Scaling as Scaling	The scaling object
Scene as Scene [read-only]	(*Inherited from* Base) Gets the current scene.
Script as Script [read-only]	The script object: it can be used to dynamically call member procedures of the container
ScriptPluginInstance as PluginInstance [read-only]	The script's plugin instance

Table 51:

Property	Description
ScriptSettings as ScriptSettings [read-only]	The script's local settings
Stage as Stage [read-only]	(*Inherited from* Base) Gets the current stage.
System as System [read-only]	(*Inherited from* Base) Gets system wide data.
Texture as Texture	The texture object
Uuid as Uuid [read-only]	The UUID of the container
VizCommunication as VizCommunication [read-only]	(*Inherited from* Base) Gets the VizCommunication object.
VizId as Integer	(*Inherited from* Base) Gets or sets the internal id of this object.
NumChildContainers as Integer *[read-only]*	*Deprecated. Use* **ChildContainerCount** *instead.*
Transform as Transform	*Deprecated.*

Member procedures

Table 52:

Name	Description
Function **AddContainer**(where as Integer) as Container	Inserts and returns a new container; where = TL_NEXT TL_PREVIOUS TL_DOWN.
Function **AddScreenOffset**(dx as Integer, dy as Integer) as Vertex	This procedure is identical to *ScreenOffsetToContainerOffset* except that it automatically adds the resulting container offset to the local container position. This procedure is typically called in OnMouseMove to implement dragging of containers.
Function **ContainsMouseCursor**() as Boolean	Returns true if and only if the mouse curser is inside the container. For this method to work correctly, it is required that at least one enter or leave callback be defined on the container or on a super-container.

Table 52:

Name	Description
Function **CopyTo**(destination as Container, where as Integer) as Container	Copies the container; where = TL_NEXT TL_PREVIOUS TL_DOWN.
Function **CreateAlpha**() as Alpha	Creates and returns an alpha object.
Function **CreateExpert**() as Expert	
Function **CreateFunction**(location as String) as PluginInstance	Creates a plugin instance by name, e.g. "BUILT_IN*FUNCTION*Autofollow"
Function **CreateGeometry**(location as String) as Geometry	Creates a geometry by name, e.g. "BUILT_IN*GEOM*Circle" (for built-ins) or "GEOM*ObjectName" (for pool objects)
Function **CreateInfoText**() as InfoText	
Function **CreateKey**() as Key	
Function **CreateMaterial**() as Material	
Function **CreateMaterial**(location as String) as Material	
Function **CreateTexture**(name as String) as Texture	Creates a texture object by name, e.g. "IMAGE*ImageName"
Function **CreateTexture**(image as Image) as Texture	
Sub **Delete**()	Deletes the container.
Sub **DeleteAlpha**()	Deletes the alpha object.
Sub **DeleteChildren**()	Deletes the container's child containers.
Sub **DeleteExpert**()	
Sub **DeleteFunction**(location as String)	Deletes a plugin instance by name, e.g. "Autofollow"
Sub **DeleteGeometry**()	
Sub **DeleteInfoText**()	
Sub **DeleteKey**()	
Sub **DeleteMaterial**()	
Sub **DeleteTexture**()	

Table 52:

Name	Description
Function **FindChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$channelName" for the *channelName* argument. Nested directors may be specified like this: "directorName1$directorName2$channelName".
Function **FindKeyframeOfObject**(keyframeName as String) as Keyframe	(*Inherited from* Base) Finds a keyframe acting on the object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$keyframeName" for the *keyframeName* argument. Nested directors may be specified like this: "directorName1$directorName2$keyframeName".
Function **FindOrCreateChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) This function works like FindChannelOfObject except that if no animation channel with the specifed name exists, a new one is created.
Function **FindSiblingSubContainer**(name as String) as Container	Searches among the container's siblings (as well as the container itself) and their sub-containers for a container with the specified name. Otherwise identical to FindSubContainer.
Function **FindSubContainer**(name as String) as Container	Searches for a sub-container with the specified name. The search mode is breadth-first. It's possible to combine multiple calls to this function into one call by concatenating the individual container names using $ as a separator. For example, FindSubContainer("name1$name2") has the same effect as FindSubContainer("name1").FindSubContainer("name2").

Table 52:

Name	Description
Function **FindSuperContainer**(name as String) as Container	Searches for a super-container with the specified name.
Sub **GetBoundingBox**(v1_out as Vertex, v2_out as Vertex)	Returns in *v1_out* and *v2_out* the left-bottom-back and right-top-front vertices of the container's bounding box.
Function **GetBoundingBoxDimensions**() as Vertex	Returns the dimensions (width, height, depth) of the container's bounding box. (The used bounding box was calculated in the last frame.)
Function **GetChannelsOfObject**([out] v as Array[Channel]) as Integer	(*Inherited from* Base) Fills the array v with the animation channels acting on this object, returning the number of channels. If this function is called on a type that cannot be animated (such as *Director*), *v* will be empty.
Function **GetChildContainerByIndex**(index as Integer) as Container	Returns the child container specified by *index.*
Function **GetChildContainerContainingMouseCursor**() as Container	Returns the first child container containing the mouse cursor.
Sub **GetContainerAndSubContainers**([out] v as Array[Container], traverseMergedGeometries as Boolean)	
Function **GetDirector**() as Director	Finds the first director containing an animation of this container.
Function **GetDirectorOfMergedGeometry**() as Director	If this container holds a merged geometry with an embedded animation, this function returns the corresponding director.
Function **GetFirstActiveChildContainer**() as Container	Returns the first (=topmost) active child container.
Function **GetFunctionPluginInstance**(pluginName as String) as PluginInstance	Provides access to the parameters of a function plugin residing on the container.
Function **GetGeometryPluginInstance**() as PluginInstance	Provides access to the parameters of a geometry plugin residing on the container.

Table 52:

Name	Description
Function **GetIndexOfFirstActiveChildContainer**() as Integer	Returns the 0-based index of the first active child container within the container (or -1 if there is no active child).
Function **GetLocalIndex**() as Integer	Returns the 0-based index of the container within its parent.
Sub **GetScreenBounds**(v1_out as Vertex, v2_out as Vertex)	Returns in *v1_out* and *v2_out* the left-bottom and right-top screen coordinates defining the rectangle occupied by the container. (The used bounding box was calculated in the last frame.)
Function **GetScreenPointToGeomSurface**(x as Integer, y as Integer, pos as Vertex) as Boolean	This procedure calculates the geometry intersection with the projected ray from the specified screen position and returns true on success. The intersection point is returned in world coordinates.
Function **GetScreenPositionOrtho**(x as Integer, y as Integer) as Boolean	This procedure gets the container's screen position in the same way as you can do it via GUI in the Transformation Editor. There can't be any rotation on the object.
Function **GetScreenSizeOrtho**(x as Integer, y as Integer) as Boolean	This procedure gets the container's screen size in the same way as you can do it via GUI in the Transformation Editor. There can't be any rotation on the object.
Function **GetShaderPluginInstance**() as PluginInstance	Provides access to the parameters of a shader plugin residing on the container.
Sub **GetTransformedBoundingBox**(v1_out as Vertex, v2_out as Vertex)	Returns in *v1_out* and *v2_out* the left-bottom-back and right-top-front vertices of the container's bounding box with applied transformation matrix.
Function **GetTransformedBoundingBoxDimensions**() as Vertex	Returns the dimensions (width, height, depth) of the container's bounding box with applied transformation matrix. (The used bounding box was calculated in the last frame.)

Table 52:

Name	Description
Function **GetUVCoordinatesOfScreenPoint**(x as Integer, y as Integer) as Vertex	Transforms (*x*, *y*) into (*u*, *v*) coordinates that are relative to the x/y cross section of the container's bounding box. The (*u*, *v*) coordinates are returned in the x and y components of the returned vertex (the z coordinate is always 0.0). If (*x*, *y*) are within the bounding-box area, u and v will be within the range [0.0, 1.0].
Sub **HideAllChildContainers**()	Makes all child containers invisible.
Function **IsAnimated**() as Boolean	(*Inherited from* Base) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as *Director*), it always returns *false*.
Function **IsOnTop**() as Boolean	Returns true if and only if the container is on top at the mouse cursor position. For this procedure to work correctly, it is required that at least one enter or leave callback be defined on the container or on a super-container.
Function **IsScreenPointInBoundingBox**(x as Integer, y as Integer) as Boolean	Returns true if the screen point (*x*, *y*) is inside the area covered by the container's bounding box. (The used bounding box was calculated in the last frame.)
Function **IsVertexInBoundingBox**(v as Vertex) as Boolean	Returns true if *v* (which must be specified in world coordinates) is inside the container's bounding box. (The used bounding box was calculated in the last frame.)
Sub **LocalPosToScreenPos**(pos as Vertex, [out] dX as Double, [out] dY as Double)	Converts local position to the position on the screen.
Function **LocalPosToWorldPos**(pos as Vertex) as Vertex	Converts the local position to a world coordinate space position.
Sub **MoveAllChannelsToDirector**(director as Director)	Moves the animation channels to the specified director.

Table 52:

Name	Description
Function **MoveTo**(destination as Container, where as Integer) as Container	Moves the container; where = TL_NEXT TL_PREVIOUS TL_DOWN.
Function **MoveToKeepTransformation**(destination as Container, where as Integer) as Container	Moves the container preserving the containers transformation; where = TL_NEXT TL_PREVIOUS TL_DOWN.
Sub **RecomputeMatrix**()	Recalculate transformation matrix of the container. This function is useful if you can't wait one field for the updated matrix. Use it carefully and only where it is really needed (current transformation matrix, transformed bounding box, screen-position or screen-size)
Function **ScreenOffsetToContainerOffset**(dx as Integer, dy as Integer) as Vertex	Transforms a pixel offset into the corresponding offset in local container coordinates. In other words, the vector returned by this procedure is the vector you need to add to the container's local position (i.e. *Position.Xyz*) in order to make the container move (*dx*, *dy*) pixels on the screen.
Function **ScreenPosToLocalGrid**(dX as Double, dY as Double) as Vertex	Converts the screen position to current grid intersection point. The returned coordinates are in local container coordinate space.
Function **ScreenPosToLocalPos**(dX as Double, dY as Double) as Vertex	Converts the screen position to the local container position. This would mean that if you assign the returned vertex to the local container's position then it moves exactly to the passed screen position (orthogonal to camera).
Sub **SetCenterPositionLocked**(center as Vertex)	This procedure sets the container's center point without affecting its position.

Table 52:

Name	Description
Sub **SetCenterScreenGridPositionLocked**(dX as Double, dY as Double)	This procedure sets the container's center point to the passed screen position, using the current grid, without affecting the container position.
Sub **SetCenterScreenPositionLocked**(dX as Double, dY as Double)	This procedure sets the container's center point to the passed screen position without affecting the container position.
Sub **SetChanged**()	(*Inherited from* Base) Mark this object as changed.
Function **SetScreenPositionOrtho**(x as Integer, y as Integer) as Boolean	This procedure sets the container's screen position in the same way as you can do it via GUI in the Transformation Editor. There can't be any rotation on the object.
Function **SetScreenSizeOrtho**(x as Integer, y as Integer) as Boolean	This procedure sets the container's screen size in the same way as you can do it via GUI in the Transformation Editor.
Sub **ShowAllChildContainers**()	Makes all child containers visible.
Sub **ShowOneChildContainer**(index as Integer)	Makes the child container specified by *index* visible and all other child containers invisible.
Sub **Update**()	Redraws the container's entry in the scene tree GUI. Ignored in on-air mode.
Function **WorldPosToLocalPos**(pos as Vertex) as Vertex	Converts the world position to a local coordinate space position.
Function ScreenOffsetToWorldOffset(dx as Integer, *dy as* Integer*) as* Vertex	*Deprecated.*

21.3.12 DateTime

Description

A type storing date/time information.

Properties

Table 53:

Property	Description
DayOfMonth as Integer	Day of month (1–31)
DayOfWeek as Integer	Day of week (0–6; Sunday = 0)
DayOfYear as Integer	Day of year (0–365)
Hour as Integer	Hours after midnight (0–23)
Minute as Integer	Minutes after hour (0–59)
Month as Integer	Month (1–12)
Second as Integer	Seconds after minute (0–59)
TotalSeconds as Integer	
Year as Integer	Year

Member procedures

Table 54:

Name	Description
Sub Normalize()	Computes *DayOfWeek* and *DayOfYear* members and forces the other members to standard ranges.
Function ToString() as String	Returns a date+time string of the form *Wed Jan 02 02:03:55 1980*.

21.3.13 Director

Base type

Base

Description

A director in the stage.

Properties

Table 55:

Property	Description
ActionChannelCount as Integer [read-only]	Gets the number of action channels.
ChildDirector as Director [read-only]	The topmost child director
EventChannel as Channel [read-only]	Gets the event channel.
Field as Integer [read-only]	The director field
Name as String	The name of the director
NextDirector as Director [read-only]	The next director in the stage tree
Offset as Double	The offset of the director (wrt to the parent director)
ParentDirector as Director [read-only]	The parent director
PreviousDirector as Director [read-only]	The previous director in the stage tree
Reverse as Boolean	Enables/Disables reverse playing.
Scene as Scene [read-only]	(*Inherited from* Base) Gets the current scene.
Stage as Stage [read-only]	(*Inherited from* Base) Gets the current stage.
System as System [read-only]	(*Inherited from* Base) Gets system wide data.
Time as Double	The director time
TotalOffset as Double [read-only]	The sum of all offsets (of this director and its parents)
VizCommunication as VizCommunication [read-only]	(*Inherited from* Base) Gets the VizCommunication object.
VizId as Integer	(*Inherited from* Base) Gets or sets the internal id of this object.
NumActionChannels as Integer [read-only]	Deprecated. Use *ActionChannelCount*.
NumAnimationChannels as Integer [read-only]	Deprecated. Call *GetAnimationChannels* to get a list of animation channels.

Member procedures

Table 56:

Name	Description
Function **AddDirector**(where as Integer) as Director	Inserts a new director; where = TL_NEXT TL_PREVIOUS TL_DOWN
Sub **ContinueAnimation**()	Continues the animation.
Function **CopyTo**(destination as Director, where as Integer) as Director	Copies a director; where = TL_NEXT TL_PREVIOUS TL_DOWN
Sub **Delete**()	Removes this director from the stage.
Function **FindActionChannel**(channelName as String) as Channel	Searches for an action channel named *channelName*.
Function **FindChannel**(channelName as String) as Channel	Searches for an animation channel named *channelName*.
Function **FindChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$channelName" for the *channelName* argument. Nested directors may be specified like this: "directorName1$directorName2$channelName".
Function **FindKeyframe**(keyframeName as String) as Keyframe	Searches for a keyframe named keyframeName.

Table 56:

Name	Description
Function **FindKeyframeOfObject**(keyframeName as String) as Keyframe	(*Inherited from* Base) Finds a keyframe acting on the object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$keyframeName" for the *keyframeName* argument. Nested directors may be specified like this: "directorName1$directorName2$keyframeName".
Function **FindOrCreateChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) This function works like FindChannelOfObject except that if no animation channel with the specifed name exists, a new one is created.
Function **FindSubDirector**(name as String) as Director	Searches for a sub-director named *name*.
Function **GetActionChannel**(index as Integer) as Channel	Returns the action channel at *index*.
Sub **GetAnimationChannels**([out] v as Array[Channel])	Fills the specified array with the director's animation channels.
Function **GetChannelsOfObject**([out] v as Array[Channel]) as Integer	(*Inherited from* Base) Fills the array v with the animation channels acting on this object, returning the number of channels. If this function is called on a type that cannot be animated (such as *Director*), *v* will be empty.
Sub **GetKeyframes**([out] v as Array[Keyframe])	Fills the specified array with the director's keyframes.
Function **HasAnimationChannels**() as Boolean	Determines whether this director contains any animation channels (not counting channels in sub-directors).
Function **IsAnimated**() as Boolean	(*Inherited from* Base) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as *Director*), it always returns *false*.
Function **IsAnimationRunning**() as Boolean	Returns true if the animation of this director is active.

Table 56:

Name	Description
Sub **MoveTo**(destination as Director, where as Integer)	Moves a director; where = TL_NEXT TL_PREVIOUS TL_DOWN
Sub **SetChanged**()	(*Inherited from* Base) Mark this object as changed.
Sub **Show**(time as Double)	Sets director time.
Sub **StartAnimation**()	Starts the animation in this director and its sub-directors.
Sub **StopAnimation**()	Stops the animation of this director and its sub-directors.
Sub **StopAnimation**(recursive as Boolean)	Stops the animation of this director and, optionally, its sub-directors.
Function GetAnimationChannel(index as Integer*) as* Channel	*Deprecated. Use GetAnimationChannels instead.*

21.3.14 Double

Description

A double precision floating-point number (64 bits).

21.3.15 Expert

Base type

Base

Description

A container's Expert property. This type provides access to the parameters of the Expert plugin.

Properties

Table 57:

Property	Description
BackFace as Boolean	Get/Set back face rendering.
ExactPicking as Boolean	Get/Set exact picking mode.
LineAntialiasing as Boolean	Get/Set line antialiasing.

Table 57:

Property	Description
Mirror as Integer	Get/Set mirror mode. Possible settings are: EXPERT_MIRROR_NONE EXPERT_MIRROR_X EXPERT_MIRROR_Y EXPERT_MIRROR_Z
Outline as Boolean	Get/Set outline rendering.
RenderMode as Integer	Get/Set render mode. Possible settings are: EXPERT_RENDER_ADD EXPERT_RENDER_BLEND EXPERT_RENDER_REV_SUBTRACT EXPERT_RENDER_SUBTRACT EXPERT_RENDER_MULTIPLY EXPERT_RENDER_REV_MULTIPLY
Scene as Scene [read-only]	(*Inherited from* Base) Gets the current scene.
SeparateSpecular as Boolean	Get/Set separate specular rendering.
ShadeModel as Integer	Get/Set shading model. Possible modes are: EXPERT_SHADE_FLAT EXPERT_SHADE_SMOOTH
Stage as Stage [read-only]	(*Inherited from* Base) Gets the current stage.
System as System [read-only]	(*Inherited from* Base) Gets system wide data.
TransparancyMode as Integer	Get/Set transparancy mode. Possible settings are: EXPERT_TRANSP_BLEND EXPERT_TRANSP_MASK EXPERT_TRANSP_FASTEST EXPERT_TRANSP_HIGH_QUALITY
TwoSidedLighting as Boolean	Get/Set two-sided lighting.
VizCommunication as VizCommunication [read-only]	(*Inherited from* Base) Gets the VizCommunication object.
VizId as Integer	(*Inherited from* Base) Gets or sets the internal id of this object.
Wireframe as Boolean	Get/Set drawing in wireframe mode.

Table 57:

Property	Description
WireframeWidth as Double	Get/Set line width in wireframe mode (typically ranges from 0.0 to 10.0).
ZBuffer as Boolean	Get/Set drawing of Z-Buffer.

Member procedures

Table 58:

Name	Description
Function FindChannelOfObject(channelName as String) as Channel	(*Inherited from* Base) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$channelName" for the *channelName* argument. Nested directors may be specified like this: "directorName1$directorName2$channelName".
Function FindKeyframeOfObject(keyframeName as String) as Keyframe	(*Inherited from* Base) Finds a keyframe acting on the object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$keyframeName" for the *keyframeName* argument. Nested directors may be specified like this: "directorName1$directorName2$keyframeName".
Function FindOrCreateChannelOfObject(channelName as String) as Channel	(*Inherited from* Base) This function works like FindChannelOfObject except that if no animation channel with the specifed name exists, a new one is created.
Function GetChannelsOfObject([out] v as Array[Channel]) as Integer	(*Inherited from* Base) Fills the array *v* with the animation channels acting on this object, returning the number of channels. If this function is called on a type that cannot be animated (such as *Director*), *v* will be empty.

Table 58:

Name	Description
Function IsAnimated() as Boolean	(*Inherited from* Base) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as *Director*), it always returns *false*.
Sub SetChanged()	(*Inherited from* Base) Mark this object as changed.

21.3.16 Geometry

Base type

Base

Description

A container's geometry. Depending on whether the geometry is a plugin or a text, this type provides access to the parameters of the plugin or the text.

Properties

Table 59:

Property	Description
Active as Boolean	
Direction as Integer	The text direction: TEXT_DIRECTION_LEFT_TO_RIGHT TEXT_DIRECTION_RIGHT_TO_LEFT TEXT_DIRECTION_TOP_TO_BOTTOM TEXT_DIRECTION_BOTTOM_TO_TOP
Kerning as Double	The kerning value
LineSpacing as Double	The line spacing
Name as String [read-only]	The name of the geometry
Orientation as Integer	The text orientation: TEXT_ORIENTATION_LEFT TEXT_ORIENTATION_RIGHT TEXT_ORIENTATION_CENTER TEXT_ORIENTATION_CHAR
PluginInstance as PluginInstance [read-only]	Gets the geometry's plugin data (if this is a geometry plugin).

Table 59:

Property	Description
Scene as Scene [read-only]	(*Inherited from* Base) Gets the current scene.
Stage as Stage [read-only]	(*Inherited from* Base) Gets the current stage.
System as System [read-only]	(*Inherited from* Base) Gets system wide data.
Text as String	The text string (if this is a text geometry)
Uuid as Uuid [read-only]	The UUID of the geometry
VerticalOrientation as Integer	The vertical text orientation: TEXT_ORIENTATION_VERTICAL_TOP TEXT_ORIENTATION_VERTICAL_BOTTOM TEXT_ORIENTATION_VERTICAL_CENTER TEXT_ORIENTATION_VERTICAL_FIRST_LINE
VizCommunication as VizCommunication [read-only]	(*Inherited from* Base) Gets the VizCommunication object.
VizId as Integer	(*Inherited from* Base) Gets or sets the internal id of this object.
WordSpacing as Double	The word spacing

Member procedures

Table 60:

Name	Description
Function **FindChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$channelName" for the *channelName* argument. Nested directors may be specified like this: "directorName1$directorName2$channelName".

Table 60:

Name	Description
Function **FindKeyframeOfObject**(keyframeName as String) as Keyframe	(*Inherited from* Base) Finds a keyframe acting on the object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$keyframeName" for the *keyframeName* argument. Nested directors may be specified like this: "directorName1$directorName2$keyframeName".
Function **FindOrCreateChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) This function works like FindChannelOfObject except that if no animation channel with the specifed name exists, a new one is created.
Function **GetChannelsOfObject**([out] v as Array[Channel]) as Integer	(*Inherited from* Base) Fills the array v with the animation channels acting on this object, returning the number of channels. If this function is called on a type that cannot be animated (such as *Director*), *v* will be empty.
Function **GetParameterBool**(parName as String) as Boolean	Gets a geometry parameter of type Boolean.
Function **GetParameterColor**(parName as String) as Color	Gets a geometry parameter of type Color.
Function **GetParameterContainer**(parName as String) as Container	Gets a geometry parameter of type Container.
Function **GetParameterDouble**(parName as String) as Double	Gets a geometry parameter of type Double.
Function **GetParameterInt**(parName as String) as Integer	Gets a geometry parameter of type Integer.
Function **GetParameterString**(parName as String) as String	Gets a geometry parameter of type String.
Function **IsAnimated**() as Boolean	(*Inherited from* Base) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as *Director*), it always returns *false*.

Table 60:

Name	Description
Sub **RegisterChangedCallback**()	Causes *OnGeometryChanged* to be invoked whenever the geometry changes.
Sub **RegisterTextChangedCallback**()	Causes *OnGeometryChanged* to be invoked whenever the geometry's text changes.
Sub **SetChanged**()	(*Inherited from* Base) Mark this object as changed.
Sub **SetParameterBool**(parName as String, v as Boolean)	Sets a geometry parameter of type Boolean.
Sub **SetParameterColor**(parName as String, v as Color)	Sets a geometry parameter of type Color.
Sub **SetParameterContainer**(parName as String, v as Container)	Sets a geometry parameter of type Container.
Sub **SetParameterDouble**(parName as String, v as Double)	Sets a geometry parameter of type Double.
Sub **SetParameterInt**(parName as String, v as Integer)	Sets a geometry parameter of type Integer.
Sub **SetParameterString**(parName as String, v as String)	Sets a geometry parameter of type String.
Sub **UnregisterChangedCallback**()	Removes the binding set up by *RegisterChangedCallback*.

21.3.17 Grid

Base type

Base

Description

A grid object.

Properties

Table 61:

Property	Description
Locked as Boolean	Gets or sets grid lock.
Scene as Scene [read-only]	(*Inherited from* Base) Gets the current scene.

Table 61:

Property	Description
Stage as Stage [read-only]	(*Inherited from* Base) Gets the current stage.
System as System [read-only]	(*Inherited from* Base) Gets system wide data.
Type as Integer	Gets or sets the grid type. Possible types are: GRID_TYPE_XY GRID_TYPE_XZ GRID_TYPE_ZY GRID_TYPE_FREE GRID_TYPE_NONE
VizCommunication as VizCommunication [read-only]	(*Inherited from* Base) Gets the VizCommunication object.
VizId as Integer	(*Inherited from* Base) Gets or sets the internal id of this object.

Member procedures

Table 62:

Name	Description
Function **FindChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$channelName" for the *channelName* argument. Nested directors may be specified like this: "directorName1$directorName2 $channelName".

Table 62:

Name	Description
Function **FindKeyframeOfObject**(keyframeName as String) as Keyframe	(*Inherited from* Base) Finds a keyframe acting on the object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$keyframeName" for the *keyframeName* argument. Nested directors may be specified like this: "directorName1$directorName2$keyframeName".
Function **FindOrCreateChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) This function works like FindChannelOfObject except that if no animation channel with the specifed name exists, a new one is created.
Function **GetChannelsOfObject**([out] v as Array[Channel]) as Integer	(*Inherited from* Base) Fills the array v with the animation channels acting on this object, returning the number of channels. If this function is called on a type that cannot be animated (such as *Director*), *v* will be empty.
Function **IsAnimated**() as Boolean	(*Inherited from* Base) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as *Director*), it always returns *false*.
Sub **MoveToContainer**(container as Container)	Moves grid to the specified container and aligns it.
Sub **SetChanged**()	(*Inherited from* Base) Mark this object as changed.

21.3.18 Image

Base type

Base

Description

An image object. This is used to retrieve an image's properties.

Properties

Table 63:

Property	Description
Name as String [read-only]	The name of the image
Scene as Scene [read-only]	(*Inherited from* Base) Gets the current scene.
Stage as Stage [read-only]	(*Inherited from* Base) Gets the current stage.
System as System [read-only]	(*Inherited from* Base) Gets system wide data.
Uuid as Uuid [read-only]	The UUID of the image
VizCommunication as VizCommunication [read-only]	(*Inherited from* Base) Gets the VizCommunication object.
VizId as Integer	(*Inherited from* Base) Gets or sets the internal id of this object.

Member procedures

Table 64:

Name	Description
Function **FindChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$channelName" for the *channelName* argument. Nested directors may be specified like this: "directorName1$directorName2$channelName".

Table 64:

Name	Description
Function **FindKeyframeOfObject**(keyframeName as String) as Keyframe	(*Inherited from* Base) Finds a keyframe acting on the object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$keyframeName" for the *keyframeName* argument. Nested directors may be specified like this: "directorName1$directorName2$keyframeName".
Function **FindOrCreateChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) This function works like FindChannelOfObject except that if no animation channel with the specifed name exists, a new one is created.
Function **GetChannelsOfObject**([out] v as Array[Channel]) as Integer	(*Inherited from* Base) Fills the array v with the animation channels acting on this object, returning the number of channels. If this function is called on a type that cannot be animated (such as *Director*), v will be empty.
Function **IsAnimated**() as Boolean	(*Inherited from* Base) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as *Director*), it always returns *false*.
Sub **SetChanged**()	(*Inherited from* Base) Mark this object as changed.

21.3.19 InfoText

Base type

Base

Description

A container's InfoText property. This type provides access to the parameters of the InfoText plugin.

Properties

Table 65:

Property	Description
Scene as Scene [read-only]	(*Inherited from* Base) Gets the current scene.
Stage as Stage [read-only]	(*Inherited from* Base) Gets the current stage.
System as System [read-only]	(*Inherited from* Base) Gets system wide data.
Text as String	Get/Set info text.
VizCommunication as VizCommunication [read-only]	(*Inherited from* Base) Gets the VizCommunication object.
VizId as Integer	(*Inherited from* Base) Gets or sets the internal id of this object.

Table 66:

Name	Description
Function **FindChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$channelName" for the *channelName* argument. Nested directors may be specified like this: "directorName1$directorName2$channelName".
Function **FindKeyframeOfObject**(keyframeName as String) as Keyframe	(*Inherited from* Base) Finds a keyframe acting on the object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$keyframeName" for the *keyframeName* argument. Nested directors may be specified like this: "directorName1$directorName2$keyframeName".

Table 66:

Name	Description
Function **FindOrCreateChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) This function works like FindChannelOfObject except that if no animation channel with the specifed name exists, a new one is created.
Function **GetChannelsOfObject**([out] v as Array[Channel]) as Integer	(*Inherited from* Base) Fills the array *v* with the animation channels acting on this object, returning the number of channels. If this function is called on a type that cannot be animated (such as *Director*), *v* will be empty.
Function **IsAnimated**() as Boolean	(*Inherited from* Base) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as *Director*), it always returns *false*.
Sub **SetChanged**()	(*Inherited from* Base) Mark this object as changed.

21.3.20 Integer

Description

A 32-bit signed integer.

21.3.21 Key

Base type

Base

Description

A key object. This is used to control a container's key object.

Properties

Table 67:

Property	Description
Active as Boolean	
AlphaValue as Double	
AutoAlpha as Boolean	

Table 67:

Property	Description
DrawKey as Boolean	
DrawRGB as Boolean	
Scene as Scene [read-only]	(*Inherited from* Base) Gets the current scene.
Stage as Stage [read-only]	(*Inherited from* Base) Gets the current stage.
System as System [read-only]	(*Inherited from* Base) Gets system wide data.
VizCommunication as VizCommunication [read-only]	(*Inherited from* Base) Gets the VizCommunication object.
VizId as Integer	(*Inherited from* Base) Gets or sets the internal id of this object.

Member procedures

Table 68:

Name	Description
Function **FindChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$channelName" for the *channelName* argument. Nested directors may be specified like this: "directorName1$directorName2$channelName".
Function **FindKeyframeOfObject**(keyframeName as String) as Keyframe	(*Inherited from* Base) Finds a keyframe acting on the object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$keyframeName" for the *keyframeName* argument. Nested directors may be specified like this: "directorName1$directorName2$keyframeName".

Table 68:

Name	Description
Function **FindOrCreateChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) This function works like FindChannelOfObject except that if no animation channel with the specifed name exists, a new one is created.
Function **GetChannelsOfObject**([out] v as Array[Channel]) as Integer	(*Inherited from* Base) Fills the array *v* with the animation channels acting on this object, returning the number of channels. If this function is called on a type that cannot be animated (such as *Director*), *v* will be empty.
Function **IsAnimated**() as Boolean	(*Inherited from* Base) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as *Director*), it always returns *false*.
Sub **SetChanged**()	(*Inherited from* Base) Mark this object as changed.

21.3.22 Keyframe

Base type

Base

Description

An animation keyframe.

Properties

Table 69:

Property	Description
ActionString as String	Gets or sets the command string of an action keyframe.
BoolValue as Boolean	Gets or sets the value of a boolean keyframe.
Channel as Channel [read-only]	Gets the channel that contains this keyframe.
ColorValue as Color	Gets or sets the value of a color keyframe.

Table 69:

Property	Description
EventDirection as Integer	Get or sets the direction of an event keyframe. Possible values are: ED_BOTH, ED_NORMAL, ED_REVERSE
EventDuration as Double	Get or sets the duration of an event keyframe (in seconds).
EventValue as Integer	Get or sets the type of an event keyframe. Possible values are: ET_STOP, ET_LOCAL_STOP, ET_TAG.
FloatValue as Double	Gets or sets the value of a float keyframe.
IntValue as Integer	Gets or sets the value of an integer keyframe.
Locked as Boolean	Gets or sets the locked status of a path point keyframe.
MaterialValue as Material [read-only]	Gets or sets the value of a material keyframe.
Name as String	Gets or sets the keyframe's name.
NextKeyframe as Keyframe [read-only]	Gets the next keyframe.
NumberOfAxes as Integer [read-only]	Returns the number of spline axes.
PathHandle as BezierHandle [read-only]	Gets the associated path handle if this is a position keyframe. If not, Null is returned.
PreviousKeyframe as Keyframe [read-only]	Gets the previous keyframe.
Scene as Scene [read-only]	(*Inherited from* Base) Gets the current scene.
Stage as Stage [read-only]	(*Inherited from* Base) Gets the current stage.
System as System [read-only]	(*Inherited from* Base) Gets system wide data.
Time as Double	Gets or sets the keyframe's time (in seconds).
VizCommunication as VizCommunication [read-only]	(*Inherited from* Base) Gets the VizCommunication object.
VizId as Integer	(*Inherited from* Base) Gets or sets the internal id of this object.

Table 69:

Property	Description
XyzValue as Vertex	Gets or sets the value of a a rotation or scaling keyframe or a path point keyframe.
NumAxes as Integer *[read-only]*	*Deprecated. Use NumberOfAxes.*

Member procedures

Table 70:

Name	Description
Function **Delete**() as Boolean	Deletes the keyframe.
Function **FindChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$channelName" for the *channelName* argument. Nested directors may be specified like this: "directorName1$directorName2$channelName".
Function **FindKeyframeOfObject**(keyframeName as String) as Keyframe	(*Inherited from* Base) Finds a keyframe acting on the object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$keyframeName" for the *keyframeName* argument. Nested directors may be specified like this: "directorName1$directorName2$keyframeName".
Function **FindOrCreateChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) This function works like FindChannelOfObject except that if no animation channel with the specifed name exists, a new one is created.

Table 70:

Name	Description
Function **GetBezierHandle**(axis as Integer) as BezierHandle	Returns the bezier handle for the specified axis. The *axis* parameter is only used if the keyframe has multiple splines, which is the case for rotation, scaling, and material keyframes. When called on a position keyframe, this procedure returns the handle for the distance spline. To get the path handle, use the *PathHandle* property instead.
Function **GetChannelsOfObject**([out] v as Array[Channel]) as Integer	(*Inherited from* Base) Fills the array *v* with the animation channels acting on this object, returning the number of channels. If this function is called on a type that cannot be animated (such as *Director*), *v* will be empty.
Function **IsAnimated**() as Boolean	(*Inherited from* Base) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as *Director*), it always returns *false*.
Sub **SetChanged**()	(*Inherited from* Base) Mark this object as changed.
Sub **UpdateAnimation**()	Update animation to reflect changes to keyframe. (This update isn't performed in OnAir mode where the update is done as soon as the animation is started.)

21.3.23 Light

Base type

Base

Description

A light object. Light objects are returned by the Scene's *GetLight* member.

Properties

Table 71:

Property	Description
Attenuation as Double	

Table 71:

Property	Description
Color as Color	
Direction as Rotation	
Position as Position	
PositionTrackingActive as Boolean	
PositionTrackingContainer as Container	
RotationTrackingActive as Boolean	
RotationTrackingContainer as Container	
Scene as Scene [read-only]	(*Inherited from* Base) Gets the current scene.
Stage as Stage [read-only]	(*Inherited from* Base) Gets the current stage.
System as System [read-only]	(*Inherited from* Base) Gets system wide data.
Type as Integer	The light type: LIGHT_TYPE_LOCAL LIGHT_TYPE_SPOT LIGHT_TYPE_INFINITE
VizCommunication as VizCommunication [read-only]	(*Inherited from* Base) Gets the VizCommunication object.
VizId as Integer	(*Inherited from* Base) Gets or sets the internal id of this object.

Member procedures

Table 72:

Name	Description
Function **FindChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$channelName" for the *channelName* argument. Nested directors may be specified like this: "directorName1$directorName2$channelName".
Function **FindKeyframeOfObject**(keyframeName as String) as Keyframe	(*Inherited from* Base) Finds a keyframe acting on the object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$keyframeName" for the *keyframeName* argument. Nested directors may be specified like this: "directorName1$directorName2$keyframeName".
Function **FindOrCreateChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) This function works like FindChannelOfObject except that if no animation channel with the specifed name exists, a new one is created.
Function **GetChannelsOfObject**([out] v as Array[Channel]) as Integer	(*Inherited from* Base) Fills the array *v* with the animation channels acting on this object, returning the number of channels. If this function is called on a type that cannot be animated (such as *Director*), *v* will be empty.
Function **IsAnimated**() as Boolean	(*Inherited from* Base) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as *Director*), it always returns *false*.
Sub **ResetPositionTracking**()	
Sub **ResetRotationTracking**()	

Table 72:

Name	Description
Sub **SetChanged**()	(*Inherited from* Base) Mark this object as changed.

21.3.24 Material

Base type

Base

Description

A material object. This is used to control a container's material.

Properties

Table 73:

Property	Description
Alpha as Double	The alpha value [0.0, 1.0]
Ambient as Color	The ambient color
Color as Color	The color
Diffuse as Color	The diffuse color
Emission as Color	The emission color
Lit as Boolean	Controls whether lighting is enabled.
Scene as Scene [read-only]	(*Inherited from* Base) Gets the current scene.
Shininess as Double	The shininess [0.0, 1.0]
Specular as Color	The specular color
Stage as Stage [read-only]	(*Inherited from* Base) Gets the current stage.
System as System [read-only]	(*Inherited from* Base) Gets system wide data.
VizCommunication as VizCommunication [read-only]	(*Inherited from* Base) Gets the VizCommunication object.
VizId as Integer	(*Inherited from* Base) Gets or sets the internal id of this object.

Member procedures

Table 74:

Name	Description
Function **FindChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$channelName" for the *channelName* argument. Nested directors may be specified like this: "directorName1$directorName2$channelName".
Function **FindKeyframeOfObject**(keyframeName as String) as Keyframe	(*Inherited from* Base) Finds a keyframe acting on the object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$keyframeName" for the *keyframeName* argument. Nested directors may be specified like this: "directorName1$directorName2$keyframeName".
Function **FindOrCreateChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) This function works like FindChannelOfObject except that if no animation channel with the specifed name exists, a new one is created.
Function **GetChannelsOfObject**([out] v as Array[Channel]) as Integer	(*Inherited from* Base) Fills the array *v* with the animation channels acting on this object, returning the number of channels. If this function is called on a type that cannot be animated (such as *Director*), *v* will be empty.
Function **IsAnimated**() as Boolean	(*Inherited from* Base) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as *Director*), it always returns *false*.
Sub **SetChanged**()	(*Inherited from* Base) Mark this object as changed.

21.3.25 Matrix

Description

A 4x4 matrix. The index operator [] can be used to access individual entries, with indices 0, 1, 2, 3 referring to the entries of the first column, 4, 5, 6, 7 to the second, etc.

Member procedures

Table 75:

Name	Description
Sub **Frustum**(left as Double, right as Double, bottom as Double, top as Double, near as Double, far as Double)	
Function **GetDeterminant**() as Double	
Function **GetRotation**() as Vertex	
Function **GetScaling**() as Vertex	
Function **GetScaling2**() as Vertex	Returns the squared scaling factors.
Function **GetTranslation**() as Vertex	
Function **Invert**() as Boolean	
Sub **LoadIdentity**()	
Sub **LoadMatrix**()	Performs a glLoadMatrix().
Sub **MultMatrix**()	Performs a glMultMatrix().
Sub **Ortho**(left as Double, right as Double, bottom as Double, top as Double, near as Double, far as Double)	
Sub **Perspective**(fov as Double, aspect as Double, near as Double, far as Double)	
Sub **Rotate**(angle as Double, v as Vertex)	
Sub **Rotate**(v as Vertex)	
Sub **RotateX**(angle as Double)	
Sub **RotateY**(angle as Double)	
Sub **RotateZ**(angle as Double)	
Sub **Scale**(v as Vertex)	
Sub **Translate**(v as Vertex)	

Table 75:

Name	Description
Sub Transpose()	

21.3.26 PluginInstance

Base type

Base

Description

This type provides access to the parameters of geometry or function plugin. Objects of this type are returned by members of Geometry and Container.

Properties

Table 76:

Property	Description
Active as Boolean	Gets or sets the plugin instance's active state.
MTAlignGrid as Boolean	Gets or sets grid alignment on or off (default).
MTDOFEventDistribution as Boolean	Distribute DOF events to other plugin instances (default) or fire them only on the local instance.
MTGrid as Grid	Gets or sets the grid for the current plugin instance.
MTRegistered as Boolean [read-only]	Is true if a control type was already registered for the current plugin instance.
MTUseCenter as Boolean	Gets or sets automatic center shift on (default) or off.
PluginName as String [read-only]	Gets the plugin name.
Scene as Scene [read-only]	(*Inherited from* Base) Gets the current scene.
Script as Script [read-only]	Provides dynamic access to the script functions of the plugin instance. (This only makes sense for scripted plugins.)
Stage as Stage [read-only]	(*Inherited from* Base) Gets the current stage.

Table 76:

Property	Description
System as System [read-only]	(*Inherited from* Base) Gets system wide data.
VizCommunication as VizCommunication [read-only]	(*Inherited from* Base) Gets the VizCommunication object.
VizId as Integer	(*Inherited from* Base) Gets or sets the internal id of this object.

Member procedures

Table 77:

Name	Description
Function **FindChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$channelName" for the *channelName* argument. Nested directors may be specified like this: "directorName1$directorName2$channelName".
Function **FindKeyframeOfObject**(keyframeName as String) as Keyframe	(*Inherited from* Base) Finds a keyframe acting on the object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$keyframeName" for the *keyframeName* argument. Nested directors may be specified like this: "directorName1$directorName2$keyframeName".
Function **FindOrCreateChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) This function works like FindChannelOfObject except that if no animation channel with the specifed name exists, a new one is created.

Table 77:

Name	Description
Function **GetChannelsOfObject**([out] v as Array[Channel]) as Integer	(*Inherited from* Base) Fills the array *v* with the animation channels acting on this object, returning the number of channels. If this function is called on a type that cannot be animated (such as *Director*), *v* will be empty.
Function **GetParameterBool**(parName as String) as Boolean	Gets a parameter value of type Boolean.
Function **GetParameterColor**(parName as String) as Color	Gets a parameter value of type Color.
Function **GetParameterContainer**(parName as String) as Container	Gets a parameter value of type Container.
Function **GetParameterDouble**(parName as String) as Double	Gets a parameter value of type Double.
Function **GetParameterInt**(parName as String) as Integer	Gets a parameter value of type Integer.
Function **GetParameterString**(parName as String) as String	Gets a parameter value of type string.
Function **IsAnimated**() as Boolean	(*Inherited from* Base) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as *Director*), it always returns *false*.
Sub **MTRegister**(stroke as Integer, controlType as Integer)	Registers a control for a certain stroke in the multitouch core. Control types: MT_CONTROL_BUTTON MT_CONTROL_PZR2D MT_CONTROL_PZR2D_TILT
Sub **MTRegisterEx**(stroke as Integer, controlType as Integer, params as Array[Double])	Same as *MTRegister* but with the additional possibility to set parameters for the specified control.
Sub **MTSetPosition**(x as Integer, y as Integer)	Sets the position of the container
Sub **MTUnregister**()	Unregisters a control in the multitouch core.
Sub **PushButton**(parName as String)	Calls the OnExecAction handler of the push button identified by *parName*.
Sub **SetChanged**()	(*Inherited from* Base) Mark this object as changed.

Table 77:

Name	Description
Sub **SetParameterBool**(parName as String, v as Boolean)	Sets a parameter value of type Boolean.
Sub **SetParameterColor**(parName as String, v as Color)	Sets a parameter value of type Color.
Sub **SetParameterContainer**(parName as String, v as Container)	Sets a parameter value of type Container
Sub **SetParameterDouble**(parName as String, v as Double)	Sets a parameter value of type Double.
Sub **SetParameterInt**(parName as String, v as Integer)	Sets a parameter value of type Integer.
Sub **SetParameterString**(parName as String, v as String)	Sets a parameter value of type Dtring.

21.3.27 Position

Base type

Base

Description

A position object. This is used to control the position of a container, camera, or light.

Properties

Table 78:

Property	Description
Scene as Scene [read-only]	(*Inherited from* Base) Gets the current scene.
Stage as Stage [read-only]	(*Inherited from* Base) Gets the current stage.
System as System [read-only]	(*Inherited from* Base) Gets system wide data.
VizCommunication as VizCommunication [read-only]	(*Inherited from* Base) Gets the VizCommunication object.
VizId as Integer	(*Inherited from* Base) Gets or sets the internal id of this object.
x as Double	Gets or sets the x component.

Table 78:

Property	Description
xyz as Vertex	Gets or sets the x, y, z components combined into a vertex.
y as Double	Gets or sets the y component.
z as Double	Gets or sets the z component.

Member procedures

Table 79:

Name	Description
Function **FindChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$channelName" for the *channelName* argument. Nested directors may be specified like this: "directorName1$directorName2$channelName".
Function **FindKeyframeOfObject**(keyframeName as String) as Keyframe	(*Inherited from* Base) Finds a keyframe acting on the object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$keyframeName" for the *keyframeName* argument. Nested directors may be specified like this: "directorName1$directorName2$keyframeName".
Function **FindOrCreateChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) This function works like FindChannelOfObject except that if no animation channel with the specifed name exists, a new one is created.

Table 79:

Name	Description
Function **GetChannelsOfObject**([out] v as Array[Channel]) as Integer	(*Inherited from* Base) Fills the array *v* with the animation channels acting on this object, returning the number of channels. If this function is called on a type that cannot be animated (such as *Director*), *v* will be empty.
Function **IsAnimated**() as Boolean	(*Inherited from* Base) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as *Director*), it always returns *false*.
Sub **SetChanged**()	(*Inherited from* Base) Mark this object as changed.

21.3.28 Rotation

Base type

Base

Description

A rotation object. This is used to control the position of a container or light.

Properties

Table 80:

Property	Description
Scene as Scene [read-only]	(*Inherited from* Base) Gets the current scene.
Stage as Stage [read-only]	(*Inherited from* Base) Gets the current stage.
System as System [read-only]	(*Inherited from* Base) Gets system wide data.
VizCommunication as VizCommunication [read-only]	(*Inherited from* Base) Gets the VizCommunication object.
VizId as Integer	(*Inherited from* Base) Gets or sets the internal id of this object.
x as Double	Gets or sets the x component.
xyz as Vertex	Gets or sets the x, y, z components combined into a vertex.

Table 80:

Property	Description
y as Double	Gets or sets the y component.
z as Double	Gets or sets the z component.

Member procedures

Table 81:

Name	Description
Function **FindChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$channelName" for the *channelName* argument. Nested directors may be specified like this: "directorName1$directorName2$channelName".
Function **FindKeyframeOfObject**(keyframeName as String) as Keyframe	(*Inherited from* Base) Finds a keyframe acting on the object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$keyframeName" for the *keyframeName* argument. Nested directors may be specified like this: "directorName1$directorName2$keyframeName".
Function **FindOrCreateChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) This function works like FindChannelOfObject except that if no animation channel with the specifed name exists, a new one is created.
Function **GetChannelsOfObject**([out] v as Array[Channel]) as Integer	(*Inherited from* Base) Fills the array *v* with the animation channels acting on this object, returning the number of channels. If this function is called on a type that cannot be animated (such as *Director*), *v* will be empty.

Table 81:

Name	Description
Function **IsAnimated**() as Boolean	(*Inherited from* Base) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as *Director*), it always returns *false*.
Sub **SetChanged**()	(*Inherited from* Base) Mark this object as changed.

21.3.29 Scaling

Base type

Base

Description

A scaling object. This is used to control the scaling of a container.

Properties

Table 82:

Property	Description
Scene as Scene [read-only]	(*Inherited from* Base) Gets the current scene.
Stage as Stage [read-only]	(*Inherited from* Base) Gets the current stage.
System as System [read-only]	(*Inherited from* Base) Gets system wide data.
VizCommunication as VizCommunication [read-only]	(*Inherited from* Base) Gets the VizCommunication object.
VizId as Integer	(*Inherited from* Base) Gets or sets the internal id of this object.
x as Double	Gets or sets the x component.
xyz as Vertex	Gets or sets the x, y, z components combined into a vertex.
y as Double	Gets or sets the y component.
z as Double	Gets or sets the z component.

Member procedures

Table 83:

Name	Description
Function **FindChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$channelName" for the *channelName* argument. Nested directors may be specified like this: "directorName1$directorName2$channelName".
Function **FindKeyframeOfObject**(keyframeName as String) as Keyframe	(*Inherited from* Base) Finds a keyframe acting on the object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$keyframeName" for the *keyframeName* argument. Nested directors may be specified like this: "directorName1$directorName2$keyframeName".
Function **FindOrCreateChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) This function works like FindChannelOfObject except that if no animation channel with the specifed name exists, a new one is created.
Function **GetChannelsOfObject**([out] v as Array[Channel]) as Integer	(*Inherited from* Base) Fills the array *v* with the animation channels acting on this object, returning the number of channels. If this function is called on a type that cannot be animated (such as *Director*), *v* will be empty.
Function **IsAnimated**() as Boolean	(*Inherited from* Base) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as *Director*), it always returns *false*.
Sub **SetChanged**()	(*Inherited from* Base) Mark this object as changed.

21.3.30 Scene

Base type

Base

Description

A scene object. The Scene member of the Base type provides access to the current scene.

Properties

Table 84:

Property	Description
AlphaClearValue as Integer	
AutoKeyActive as Boolean	
AutoKeyPosition as Vertex	
BackgroundColor as Color	
CurrentCamera as Camera [read-only]	The current camera
CurrentCameraIndex as Integer	The index of the current camera
Location as String [read-only]	The location of the scene
LocationPath as String [read-only]	The full db location of the scene
Map as SharedMemory [read-only]	The scene's memory map, used for data sharing within a scene.
Name as String [read-only]	The name of the scene
RootContainer as Container [read-only]	The topmost container in the scene tree
Scene as Scene [read-only]	(*Inherited from* Base) Gets the current scene.
Script as Script [read-only]	The script object: it can be used used to dynamically call member procedures of the scene
ScriptPluginInstance as PluginInstance [read-only]	The script's plugin instance
Stage as Stage [read-only]	(*Inherited from* Base) Gets the current stage.
System as System [read-only]	(*Inherited from* Base) Gets system wide data.
Uuid as Uuid [read-only]	The UUID of the scene

Table 84:

Property	Description
VizCommunication as VizCommunication [read-only]	(*Inherited from* Base) Gets the VizCommunication object.
VizId as Integer	(*Inherited from* Base) Gets or sets the internal id of this object.

Member procedures

Table 85:

Name	Description
Sub **AddLogMessage**(message as String)	Adds a log message to the scene log.
Function **CreateRootContainer**() as Container	
Function **FindChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$channelName" for the *channelName* argument. Nested directors may be specified like this: "directorName1$directorName2$channelName".
Function **FindContainer**(name as String) as Container	Searches the scene tree for a container named name. The search mode is breadth-first. It's possible to combine a call to this function and multiple calls to Container.FindSubContainer into a single call by concatenating the individual container names using $ as a separator. For example, scene.FindContainer("name1$name2") has the same effect as scene.FindContainer("name1").FindSubContainer("name2").

Table 85:

Name	Description
Function **FindKeyframeOfObject**(keyframeName as String) as Keyframe	(*Inherited from* Base) Finds a keyframe acting on the object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$keyframeName" for the *keyframeName* argument. Nested directors may be specified like this: "directorName1$directorName2$keyframeName".
Function **FindOrCreateChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) This function works like FindChannelOfObject except that if no animation channel with the specifed name exists, a new one is created.
Function **GetCamera**(cameraIndex as Integer) as Camera	
Function **GetChannelsOfObject**([out] v as Array[Channel]) as Integer	(*Inherited from* Base) Fills the array *v* with the animation channels acting on this object, returning the number of channels. If this function is called on a type that cannot be animated (such as *Director*), *v* will be empty.
Function **GetGrid**(gridNumber as Integer) as Grid	Gets a grid from the scene.
Function **GetLight**(lightIndex as Integer) as Light	
Function **GetScenePluginInstance**(pluginName as String) as PluginInstance	Provides access to the parameters of a scene plugin instance.
Function **GetUnlockedGrid**() as Grid	Gets the next unused grid from the scene.
Function **IsAnimated**() as Boolean	(*Inherited from* Base) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as *Director*), it always returns *false*.
Function **IsBacklayer**() as Boolean	Returns true, if the script is executed within Backlayer
Function **IsFrontlayer**() as Boolean	Returns true, if the script is executed within Frontlayer

Table 85:

Name	Description
Function **IsMainlayer**() as Boolean	Returns true, if the script is executed within Mainlayer
Function **ScreenPosToWorldGrid**(dX as Double, dY as Double) as Vertex	Converts the screen position to current grid intersection point.
Function **ScreenPosToWorldPos**(dX as Double, dY as Double) as Vertex	Converts the screen position to the world position. This would mean that if you assign the returned vertex to a container (on root level - no parent containers) position then it moves exactly to the passed screen position (orthogonal to camera).
Sub **SetChanged**()	(*Inherited from* Base) Mark this object as changed.
Sub **UpdateSceneTree**()	In authoring mode, this method causes the scene tree GUI to be rebuilt. Ignored in on-air mode.
Sub **WorldPosToScreenPos**(pos as Vertex, [out] dX as Double, [out] dY as Double)	Converts the world position to the position on the screen.

21.3.31 Script

Description

The container's script object. This is used to dynamically call member procedures of another container.

21.3.32 ScriptSettings

Description

A script's local settings.

Properties

Table 86:

Property	Description
CallButtonDownForTopObjectsOnly as Boolean	If set to true, the OnXButtonDown procedures are called only if the container contains the mouse cursor and is on top at the mouse cursor position. Default = *true*.
CallEnterLeaveForTopObjectsOnly as Boolean	If set to true, OnEnter/OnLeave/ OnEnterSubContainer/ OnLeaveSubContainer is called only if the container is on top at the mouse cursor position. Default = *true*.
CallMouseMoveForTopObjectsOnly as Boolean	If set to true, OnMouseMove is called only if the container contains the mouse cursor and is on top at the mouse cursor position. Default = *false*.
CallEnterLeaveForBackgroundObjects as Boolean	*Deprecated. Use CallEnterLeaveForTopObjectsOnly instead.*
CallMouseMoveForBackgroundObjects as Boolean	*Deprecated. Use CallMouseMoveForTopObjectsOnly instead.*

21.3.33 SharedMemory

Base type

Base

Description

A storage object that holds a map of Variant objects indexed by a string. It can be used for system-wide or scene-wide data sharing. To access an individual value, use the index operator. Example:

```
' Script 1:
dim a = 123 ' define an integer variable
System.Map["my_global_variable"] = a
'
' Script 2:
dim a = (Integer)System.Map["my_global_variable"]
```

Properties

Table 87:

Property	Description
Scene as Scene [read-only]	(*Inherited from* Base) Gets the current scene.
Stage as Stage [read-only]	(*Inherited from* Base) Gets the current stage.
System as System [read-only]	(*Inherited from* Base) Gets system wide data.
VizCommunication as VizCommunication [read-only]	(*Inherited from* Base) Gets the VizCommunication object.
VizId as Integer	(*Inherited from* Base) Gets or sets the internal id of this object.

Member procedures

Table 88:

Name	Description
Function **ContainsKey**(key as String) as Boolean	Tests whether an element with the specified key is in the map.
Function **CreateKey**(key as String) as Boolean	Creates a new empty variable.
Function **DeleteKey**(key as String) as Boolean	Deletes a variable.
Function **FindChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$channelName" for the *channelName* argument. Nested directors may be specified like this: "directorName1$directorName2$channelName".

Table 88:

Name	Description
Function **FindKeyframeOfObject**(keyframeName as String) as Keyframe	(*Inherited from* Base) Finds a keyframe acting on the object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$keyframeName" for the *keyframeName* argument. Nested directors may be specified like this: "directorName1$directorName2$keyframeName".
Function FindOrCreateChannelOfObject(channelName as String) as Channel	(*Inherited from* Base) This function works like FindChannelOfObject except that if no animation channel with the specifed name exists, a new one is created.
Function **GetChannelsOfObject**([out] v as Array[Channel]) as Integer	(*Inherited from* Base) Fills the array *v* with the animation channels acting on this object, returning the number of channels. If this function is called on a type that cannot be animated (such as *Director*), *v* will be empty.
Sub **GetKeys**(keys as Array[String])	Fills the passed array with all defined keys.
Function **IsAnimated**() as Boolean	(*Inherited from* Base) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as *Director*), it always returns *false*.
Function **LoadFromDb**(path as String) as Boolean	Loads the contents of the map from a database file. The file is specified by *path, which must be an absolute path, such as "/User/mem_map"*.
Sub **RegisterChangedCallback**(key as String)	Causes *OnSharedMemoryVariableChanged* to be invoked whenever the variable identified by *key* is changed and *OnSharedMemoryVariableDeleted* in response to the key being deleted. If you pass an empty string for *key*, the callbacks will be invoked whenever *any* variable in the map is changed or deleted.

Table 88:

Name	Description
Function **SaveToDb**(path as String) as Boolean	Saves the contents of the map into a database file. The file is specified by path, which must be an absolute *path, such as "/User/mem_map".*
Sub **SetChanged**()	(*Inherited from* Base) Mark this object as changed.
Sub **UnregisterChangedCallback**(key as String)	Removes the binding set up by *RegisterChangedCallback*.

21.3.34 Stage

Base type

Base

Description

A stage object. The Stage member of the Base type provides access to the current scene's stage.

Properties

Table 89:

Property	Description
RootDirector as Director [read-only]	Gets the topmost director in the stage.
Scene as Scene [read-only]	(*Inherited from* Base) Gets the current scene.
Stage as Stage [read-only]	(*Inherited from* Base) Gets the current stage.
System as System [read-only]	(*Inherited from* Base) Gets system wide data.
VizCommunication as VizCommunication [read-only]	(*Inherited from* Base) Gets the VizCommunication object.
VizId as Integer	(*Inherited from* Base) Gets or sets the internal id of this object.

Member procedures

Table 90:

Name	Description
Sub **ContinueAnimation**()	Continues the animation.
Function **CreateRootDirector**() as Director	If there is no root director this method creates and returns a new one.
Function **FindChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$channelName" for the *channelName* argument. Nested directors may be specified like this: "directorName1$directorName2$channelName".
Function **FindDirector**(directorName as String) as Director	Searches for a director named directorName.
Function **FindKeyframeOfObject**(keyframeName as String) as Keyframe	(*Inherited from* Base) Finds a keyframe acting on the object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$keyframeName" for the *keyframeName* argument. Nested directors may be specified like this: "directorName1$directorName2$keyframeName".
Function **FindOrCreateChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) This function works like FindChannelOfObject except that if no animation channel with the specifed name exists, a new one is created.
Function **GetChannelsOfObject**([out] v as Array[Channel]) as Integer	(*Inherited from* Base) Fills the array *v* with the animation channels acting on this object, returning the number of channels. If this function is called on a type that cannot be animated (such as *Director*), *v* will be empty.

Table 90:

Name	Description
Function **IsAnimated**() as Boolean	(*Inherited from* Base) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as *Director*), it always returns *false*.
Function **IsAnimationRunning**() as Boolean	Returns true if the animation active.
Sub **SetChanged**()	(*Inherited from* Base) Mark this object as changed.
Sub **StartAnimation**()	Starts the animation in the directors selected in the diretor tree.
Sub **StopAnimation**()	Stops the animation in all directors.

21.3.35 String

Description

A character string. Use the '&' operator to concatenate strings.

Properties

Table 91:

Property	Description
Length as Integer [read-only]	Gets the length of the string. Checking the length of a doublebyte string: You have to make sure to use the correct function for calculating double byte strings. Normal Doublebyte Fonts (Asian/Cyrillic) are calculated wrong if you use the string.length() function. Solution: To get the prober length, you have to convert it using Utf8ToAnsi()

Member procedures

Table 92:

Name	Description
Sub **AnsiToUtf8**()	Converts an ANSI encoded string (using the system code-page) to UTF-8.
Sub **Append**(str as String)	Appends 'str' to the string
Function **EndsWith**(str as String) as Boolean	Returns true if the string ends with 'str'
Sub **Erase**(pos as Integer, len as Integer)	Erases a range of 'len' characters starting at 'pos'
Sub **EraseChar**(pos as Integer)	Erases the character at 'pos'
Function **Find**(substring as String) as Integer	Searches for a substring
Function **FindFirstNotOf**(chars as String) as Integer	Searches for the first character that is not equal to any character within 'chars'
Function **FindFirstOf**(chars as String) as Integer	Searches for the first character that is equal to any character within 'chars'
Function **FindLastNotOf**(chars as String) as Integer	Searches for the last character that is not equal to any character within 'chars'
Function **FindLastOf**(chars as String) as Integer	Searches for the last character that is equal to any character within 'chars'
Function **GetChar**(pos as Integer) as String	Returns the character at 'pos'
Function **GetSubstring**(pos as Integer, len as Integer) as String	Returns a substring
Sub **Insert**(pos as Integer, str as String)	Inserts 'str' before position 'pos'
Sub **Join**(strings as Array[String], joiner as String)	Concatenates individual strings
Function **Left**(len as Integer) as String	Extracts 'len' characters from the left end of the string
Sub **MakeLower**()	Converts the string to an lower-case string
Sub **MakeUpper**()	Converts the string to an upper-case string
Function **Match**(regex as String) as Boolean	Returns true if the regular expression 'regex' is found int the string

Table 92:

Name	Description
Sub **PadLeft**(totalWidth as Integer, paddingChar as String)	Right-aligns the characters of the string
Sub **PadRight**(totalWidth as Integer, paddingChar as String)	Left-aligns the characters of the string
Sub **Prepend**(str as String)	Inserts 'str' at the beginning of the string
Sub **Replace**(pos as Integer, len as Integer, str as String)	Replaces the substring specified by 'pos' and 'len' with 'str'
Sub **ReplaceChar**(pos as Integer, str as String)	Replaces the character at 'pos' with 'str'
Function **RFind**(substring as String) as Integer	Searches backward for a substring
Function **Right**(len as Integer) as String	Extracts 'len' characters from the right end of the string
Sub **Split**(separators as String, [out] result as Array[String])	Splits a string into substrings; if 'separators' is empty, whitespace sequences act as separators. Example: dim a as string = "Monday Thuesday Wednesday" dim b as Array[String] a.split(" ",b) b will be ["Monday","Thuesday","Wednesday"]
Function **StartsWith**(str as String) as Boolean	Returns true if the string starts with 'str'
Function **Substitute**(regex as String, subst as String, global as Boolean) as Integer	Replaces 'regex' with 'subst'. If 'global' is true, all occurences of 'regex' are replaced. Otherwise, only the first occurrece is replaced.
Sub **TclSplit**([out] result as Array[String])	Splits a string TCL-style-wise, using whitespace as separators (TCL-style means: Braced and double-quoted strings are never split even if they contain whitespace).
Sub **Trim**()	Removes whitspace characters from both ends of the string
Sub **Utf8ToAnsi**()	Converts a UTF-8 encoded string to ANSI (using the system code-page).

21.3.36 StringMap

Description

An associative array that maps strings to Variant objects. Use the index operator `[]` to access individual elements.

Member procedures

Table 93:

Property	Description
Sub **Clear**()	
Function **ContainsKey**(key as String) as Boolean	
Sub **DeleteKey**(key as String)	
Sub **GetKeys**(keys as Array[String])	Fills the passed array with all defined keys.

21.3.37 System

Base type

Base

Description

This type provides access to system-wide parameters and procedures.

Properties

Table 94:

Property	Description
BackScene as Scene [read-only]	Returns the scene that is in the back layer of the scene/render editor.
CurrentDirectoryPath as String	Gets or sets the current directory path.
CurrentRefreshRate as Double [read-only]	Gets the current refresh rate in seconds.
FrontScene as Scene [read-only]	Returns the scene that is in the front layer of the scene/render editor.
HostName as String [read-only]	Gets the host name of the local computer.

Table 94:

Property	Description
IPAddressString as String [read-only]	Gets the IP address of the local computer. The format is a string that looks like "10.211.54.199".
IsPostRecording as Boolean [read-only]	Determines if in post mode.
LButtonState as Boolean [read-only]	True if the left mouse button is pressed.
MainScene as Scene [read-only]	Returns the scene that is in the main layer of the scene/render editor.
Map as SharedMemory [read-only]	The global memory map, used for system-wide data sharing.
MButtonState as Boolean [read-only]	True if the middle mouse button is pressed.
MouseX as Integer [read-only]	Gets the x position of the mouse cursor (in screen coordinates).
MouseY as Integer [read-only]	Gets the y position of the mouse cursor (in screen coordinates).
OutputAspect as Double [read-only]	Gets the output aspect ratio.
OutputFieldDominance as Integer [read-only]	Gets the field dominance of current the output field. (0=first field or progressive, 1=second field;)
OutputInterlaced as Boolean [read-only]	Determines if the output format is interlaced.
OutputRefreshRate as Double [read-only]	Gets the output refresh rate in seconds.
OutputRetraceCounter as Integer [read-only]	Gets the retrace counter.
OutputSystem as Integer [read-only]	Gets the output system. (0=INACTIVE, 3=480I_5994_SMPTE259, 4=576I_5000_SMPTE259, 5=720P_5994_SMPTE296, 6=720P_6000_SMPTE296, 10=1080I_5000_SMPTE274, 11=1080I_5994_SMPTE274, 12=1080I_6000_SMPTE274, 21=720P_5000_SMPTE296, 27=FULL_SCREEN, 28=USER_DEFINED).
PostCurrentTime as Double [read-only]	Get the current time in post mode.

Table 94:

Property	Description
PostFieldDominance as Integer [read-only]	Gets the field dominance in post mode. (0=first field or progressive, 1=second field;)
PostRetraceCounter as Integer [read-only]	Gets the retrace counter in post mode.
RButtonState as Boolean [read-only]	True if the right mouse button is pressed.
RenderEditorHeight as Integer [read-only]	Gets the height in pixels of the render editor.
RenderEditorWidth as Integer [read-only]	Gets the width in pixels of the render editor.
RenderWindowHeight as Integer [read-only]	Gets the height in pixels of the render window.
RenderWindowWidth as Integer [read-only]	Gets the width in pixels of the render window.
Scene as Scene [read-only]	(Inherited from Base) Gets the current scene.
ScreenHeight as Integer [read-only]	Gets the height in pixels of the screen.
ScreenWidth as Integer [read-only]	Gets the width in pixels of the screen.
ShowMouseCursor as Boolean	Gets or sets the visibility state of the mouse cursor.
Stage as Stage [read-only]	(*Inherited from* Base) Gets the current stage.
System as System [read-only]	(*Inherited from* Base) Gets system wide data.
VideoHeight as Integer [read-only]	Gets the height in pixels of the output video.
VideoWidth as Integer [read-only]	Gets the width in pixels of the output video.
VizCommunication as VizCommunication [read-only]	(*Inherited from* Base) Gets the VizCommunication object.
VizId as Integer	(*Inherited from* Base) Gets or sets the internal id of this object.
RefreshRate as Double *[read-only]*	*Deprecated.*

Member procedures

Table 95:

Name	Description
Function **DeleteFile**(filename as String) as Boolean	Deletes the file *filename*, returns true when successfully
Function **DirectoryExists**(directory as String) as Boolean	Returns true if the directory exists
Function **ExploreDocument**(filename as String, activate as Boolean) as Boolean	Opens a directory in the Windows Explorer. If *activate* is false, the Explorer window is minimized.
Function **FileExists**(filename as String) as Boolean	Returns true if the file exists
Function **FindChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$channelName" for the *channelName* argument. Nested directors may be specified like this: "directorName1$directorName2$channelName".
Function **FindKeyframeOfObject**(keyframeName as String) as Keyframe	(*Inherited from* Base) Finds a keyframe acting on the object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$keyframeName" for the *keyframeName* argument. Nested directors may be specified like this: "directorName1$directorName2$keyframeName".
Function **FindOrCreateChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) This function works like FindChannelOfObject except that if no animation channel with the specifed name exists, a new one is created.

Table 95:

Name	Description
Function **GetChannelsOfObject**([out] v as Array[Channel]) as Integer	(*Inherited from* Base) Fills the array *v* with the animation channels acting on this object, returning the number of channels. If this function is called on a type that cannot be animated (such as *Director*), *v* will be empty.
Function **GetClipChannel**(param_1 as Integer) as ClipChannel	Gets the ClipChannel A (parameter = 0) or B (parameter = 1)
Function **GetDirectoryContent**(dirname as String, pattern as String) as Array[String]	Returns an array with all files in the directory *dirname* matching the *pattern*, e.g.: *.jpg
Function **GetElapsedTime**() as Double	Gets the time (in seconds) that has passed since Viz was started.
Function **GetNumberOfClipChannel**() as Integer	Gets the number of available Clip Channels
Function **GetNumberOfTimecodeSources**() as Integer	Returns the number of registered timecodes.
Function **GetTimecode**(Device as String, Type as String) as Timecode	Returns a timecode object from *device* with *type*.
Function **GetTimecodeByIndex**(Index as Integer) as Timecode	Returns a timecode object by its *index*.
Function **IsAnimated**() as Boolean	(*Inherited from* Base) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as *Director*), it always returns *false*.
Function **IsKeyDown**(keyCode as Integer) as Boolean	Returns true if the specified key is pressed.
Function **LoadBackScene**(scenepath as String) as Scene	Loads a scene file into the back layer of the scene/render editor.
Function **LoadFrontScene**(scenepath as String) as Scene	Loads a scene file into the front layer of the scene/render editor.
Function **LoadMainScene**(scenepath as String) as Scene	Loads a scene file into the main layer of the scene/render editor.
Function **LoadTextFile**(filepath as String, [out] result as String) as Boolean	Loads the text file identified by *filepath* into *result*. Returns false if the file does not exist or is not accessible.

Table 95:

Name	Description
Function **MouseOverIPDefineServer**(serverAddress as String, serverPort as Integer, localPort as Integer) as Integer	Define another VizMouseServer
Function **MouseOverIPQuitServer**() as Boolean	Kill the all defined VizMouseServer
Function **MouseOverIPQuitServer**(serverID as Integer) as Boolean	Kill this VizMouseServer
Function **MouseOverIPRegister**(serverID as Integer) as Boolean	Register the Viz Engine as client to receive mouse events from this VizMouseServer
Function **MouseOverIPRegister**() as Boolean	Register the Viz Engine as client to receive mouse events from all defined VizMouseServer
Function **MouseOverIPUndefineAllServers**() as Boolean	Undefine all known VizMouseServers.
Function **MouseOverIPUndefineServer**(serverID as Integer) as Boolean	Undefine the VizMouseServer with the ID serverID.
Function **MouseOverIPUndefineServer**(serverAddress as String, serverPort as Integer, localPort as Integer) as Boolean	Unefine the VizMouseServer with the address *serverAddress*, the server port *serverPort* and the local listening port *localPort*
Function **MouseOverIPUnregister**() as Boolean	Unregister the Viz Engine from all defined VizMouseServer
Function **MouseOverIPUnregister**(serverID as Integer) as Boolean	Unregister the Viz Engine from this VizMouseServer
Function **MouseOverIPUnregisterAll**() as Boolean	Unregister all Viz Engines from all defined VizMouseServer
Function **MouseOverIPUnregisterAll**(serverID as Integer) as Boolean	Unregister all Viz Engines from this VizMouseServer
Function **OpenDocument**(filename as String, activate as Boolean) as Boolean	Opens a document file. If *activate* is false, the application loading the file is minimized.
Sub **PlaySound**(filename as String)	Plays a sound file.

Table 95:

Name	Description
Function **SaveTextFile**(filepath as String, text as String) as Boolean	Saves the contents of *text* into the file identified by *filepath*. Returns false if the file cannot be written to.
Function **SendCommand**(command as String) as String	Sends a command to the local Viz Engine.
Function **SendRemoteCommand**(hostname as String, port as Integer, commandString as String, timeoutInMs as Integer) as String	Sends a command string to a Viz Engine running on another PC. If *timeoutInMs* > 0, the procedure waits at most *timeoutInMs* milliseconds for an answer, and susequently returns that answer. If *timeoutInMs* = 0, the procedure returns immediatly, ignoring any answer.
Sub **SetChanged**()	(*Inherited from* Base) Mark this object as changed.
Function **StartApplication**(filename as String, parameters as String, activate as Boolean) as Boolean	Launches an application. If *activate* is false, the application window is minimized.
Sub **StopSound**()	Stops sound playback. See *PlaySound*.
Function **TcpSend**(hostname as String, port as Integer, commandString as String, timeoutInMs as Integer) as String	Sends a command string via TCP/IP to a remote socket. Unlike *SendRemoteCommand*, this procedure sends the command string as is, that is without a viz command identifier attached. The *timeoutInMs* has the same meaning as for *SendRemoteCommand*.
Sub **TcpSendAsync**(sharedMemoryKey as String, hostname as String, port as Integer, commandString as String, timeoutInMs as Integer)	Sends a command string via TCP/IP to a remote socket. Unlike *TcpSend*, this procedure works asynchron and the return result is stored in a shared memory map, at your defined *sharedMemoryKey*. The defined *sharedMemoryKey* or its RegisterChangedCallback will be searched in the maps in the following order: *Scene.Map*, *System.Map* and *VizCommunication.Map* If *timeoutInMs* is set to 0 then *TcpSendAsync* will wait until a answer is received or as long as no error occurs (e.g. host doesn't exist).

Table 95:

Name	Description
Sub **TcpSendAsyncStatus**(sharedMemoryKey as String, hostname as String, port as Integer, commandString as String, timeoutInMs as Integer, sharedMemoryKeyStatus as String)	Same as *TcpSendAsync* but with additional status Shared Memory Key which informs you about the transmission process. The following Messages (Strings) will be returned in *sharedMemoryKeyStatus*: SENDING RECEIVED ERROR TIMEOUT The *timeoutInMs* has the same meaning as for *TcpSendAsync*.

21.3.38 Texture

Base type

Base

Description

A container's texture object.

Properties

Table 96:

Property	Description
Active as Boolean	
EffectType as Integer	The effect type: TX_EFFECT_BLUR TX_EFFECT_MOSAIC
EffectValue as Integer	The effect value as an integer ranging 0-100
EffectValueDouble as Double	The effect value as a double ranging 0.0-100.0
EnvironmentType as Integer	The environment type: TX_ENV_BLEND TX_ENV_DECAL TX_ENV_MODULATE
Image as Image	The texture image

Table 96:

Property	Description
MapPosition as Vertex	The mapping position
MapRotation as Vertex	The mapping rotation
MapScaling as Vertex	The mapping scaling
MapType as Integer	The mapping type: TX_MAP_VERTEX TX_MAP_LINEAR TX_MAP_REFLECT TX_MAP_SPHERICAL TX_MAP_CYLINDRICAL TX_MAP_CUBE
QualityType as Integer	The quality type: TX_QUAL_PIXEL TX_QUAL_LINEAR TX_QUAL_MIPMAP TX_QUAL_SHARPEN
Scene as Scene [read-only]	(*Inherited from* Base) Gets the current scene.
SharpenValue as Integer	The sharpen value as an integer ranging 0-100
SharpenValueDouble as Double	The sharpen value as a double ranging 0.0-100.0
Stage as Stage [read-only]	(*Inherited from* Base) Gets the current stage.
System as System [read-only]	(*Inherited from* Base) Gets system wide data.
VizCommunication as VizCommunication [read-only]	(*Inherited from* Base) Gets the VizCommunication object.
VizId as Integer	(*Inherited from* Base) Gets or sets the internal id of this object.
WrapType as Integer	The wrap type: TX_WRAP_REPEAT TX_WRAP_CLAMP

Member procedures

Table 97:

Name	Description
Function **FindChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$channelName" for the *channelName* argument. Nested directors may be specified like this: "directorName1$directorName2$channelName".
Function **FindKeyframeOfObject**(keyframeName as String) as Keyframe	(*Inherited from* Base) Finds a keyframe acting on the object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$keyframeName" for the *keyframeName* argument. Nested directors may be specified like this: "directorName1$directorName2$keyframeName".
Function **FindOrCreateChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) This function works like FindChannelOfObject except that if no animation channel with the specifed name exists, a new one is created.
Function **GetChannelsOfObject**([out] v as Array[Channel]) as Integer	(*Inherited from* Base) Fills the array *v* with the animation channels acting on this object, returning the number of channels. If this function is called on a type that cannot be animated (such as *Director*), *v* will be empty.
Function **IsAnimated**() as Boolean	(*Inherited from* Base) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as *Director*), it always returns *false*.
Sub **SetChanged**()	(*Inherited from* Base) Mark this object as changed.

21.3.39 Timecode

Base type

Base

Properties

Table 98:

Property	Description
Scene as Scene [read-only]	(*Inherited from* Base) Gets the current scene.
Stage as Stage [read-only]	(*Inherited from* Base) Gets the current stage.
System as System [read-only]	(*Inherited from* Base) Gets system wide data.
VizCommunication as VizCommunication [read-only]	(*Inherited from* Base) Gets the VizCommunication object.
VizId as Integer	(*Inherited from* Base) Gets or sets the internal id of this object.

Member procedures

Table 99:

Name	Description
Function **FindChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$channelName" for the *channelName* argument. Nested directors may be specified like this: "directorName1$directorName2$channelName".

Table 99:

Name	Description
Function **FindKeyframeOfObject**(keyframeName as String) as Keyframe	(*Inherited from* Base) Finds a keyframe acting on the object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$keyframeName" for the *keyframeName* argument. Nested directors may be specified like this: "directorName1$directorName2$keyframeName".
Function **FindOrCreateChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) This function works like FindChannelOfObject except that if no animation channel with the specifed name exists, a new one is created.
Sub **Get**([out] Hours as Integer, [out] Minutes as Integer, [out] Seconds as Integer, [out] Frames as Integer)	Writes the current value of the timecode into *Hours*, *Minutes*, *Seconds* and *Frames*.
Function **GetChannelsOfObject**([out] v as Array[Channel]) as Integer	(*Inherited from* Base) Fills the array *v* with the animation channels acting on this object, returning the number of channels. If this function is called on a type that cannot be animated (such as *Director*), *v* will be empty.
Function **GetDevice**() as String	Returns the device name of the timecode.
Function **GetString**() as String	Returns the current value of the timecode as string.
Function **GetType**() as String	Returns the type of the timecode.
Function **IsAnimated**() as Boolean	(*Inherited from* Base) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as *Director*), it always returns *false*.
Sub **SetChanged**()	(*Inherited from* Base) Mark this object as changed.

21.3.40 Uuid

Description

A Universally Unique Identifier.

Properties

Table 100:

Property	Description
FullString as String [read-only]	Returns the uuid as a string, enclosed in angle brackets, e.g. "7410EF6B-4271-4EFA-8E8E30212D9AD2A5".
String as String [read-only]	Returns the uuid as a string, e.g. "7410EF6B-4271-4EFA-8E8E30212D9AD2A5".

Member Procedures

Table 101:

Name	Description
Sub **GenerateNewUuid**()	Generates a new UUID.

21.3.41 Variant

Description

A variant type. Objects of this type can hold objects of any other type.

21.3.42 Vertex

Description

A vertex/vector consisting of 3 double components. The components can be accessed either via the x, y and z members, or using the index operator []. For example, v.y and v[1] both refer to the y component.

Properties

Table 102:

Property	Description
X as Double	The x component
Y as Double	The y component
Z as Double	The z component

Member procedures

Table 103:

Name	Description
Function **DistanceToLine**(linePoint1 as Vertex, linePoint2 as Vertex) as Double	
Function **DistanceToPlane**(planeNormal as Vertex, planePoint as Vertex) as Double	
Function **DistanceToPlaneSigned**(planeNormal as Vertex, planePoint as Vertex) as Double	
Function **MirrorAroundLine**(linePoint1 as Vertex, linePoint2 as Vertex) as Vertex	
Function **MirrorAroundVector**(v as Vertex) as Vertex	
Function **Norm**() as Double	Returns the length of the vector.
Function **Normalize**() as Double	Normalizes the vertex, i.e. scales it to length 1.
Function **Project**(viewMat as Matrix, projMat as Matrix, x as Integer, y as Integer, width as Integer, height as Integer) as Boolean	
Function **Project**(mat as Matrix, halfwidth as Integer, halfweight as Integer) as Boolean	
Function **ProjectOnLine**(linePoint1 as Vertex, linePoint2 as Vertex) as Vertex	
Function **ProjectOnVector**(v as Vertex) as Vertex	
Function **SquaredNorm**() as Double	Returns the squared length of the vector.
Sub **Unproject**(viewMat as Matrix, projMat as Matrix, x as Integer, y as Integer, width as Integer, height as Integer)	

Table 103:

Name	Description
Function **Unproject**(viewMat as Matrix, projMat as Matrix, x as Integer, y as Integer, width as Integer, height as Integer, plane1 as Vertex, plane2 as Vertex, plane3 as Vertex) as Boolean	
Sub GlNormal()	*Deprecated.*
Sub GlVertex()	*Deprecated.*

21.3.43 VizCommunication

Base type

Base

Description

This class maintains a group of viz hosts that you can simultaneously send commands to. In addition, there is a SharedMemory member (Map) that shares its data among all engines connected to the database.

Properties

Table 104:

Property	Description
Map as SharedMemory [read-only]	Distributed shared memory map. Note that other Viz Engine machines are notified of changes made to this map only if they are running at the time the change is made.
Scene as Scene [read-only]	(*Inherited from* Base) Gets the current scene.
Stage as Stage [read-only]	(*Inherited from* Base) Gets the current stage.
System as System [read-only]	(*Inherited from* Base) Gets system wide data.
VizCommunication as VizCommunication [read-only]	(*Inherited from* Base) Gets the VizCommunication object.
VizId as Integer	(*Inherited from* Base) Gets or sets the internal id of this object.

Member procedures

Table 105:

Name	Description
Sub **AddHostToGroup**(hostname as String, port as Integer)	Adds a hostname to the group of hostnames.
Sub **ClearHostGroup**()	Clears the group of hostnames.
Function **FindChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$channelName" for the *channelName* argument. Nested directors may be specified like this: "directorName1$directorName2$channelName".
Function **FindKeyframeOfObject**(keyframeName as String) as Keyframe	(*Inherited from* Base) Finds a keyframe acting on the object. If this function is called on a type that cannot be animated (such as *Director*), it returns *null*. You can limit the search to a particular director by using the syntax "directorName$keyframeName" for the *keyframeName* argument. Nested directors may be specified like this: "directorName1$directorName2$keyframeName".
Function **FindOrCreateChannelOfObject**(channelName as String) as Channel	(*Inherited from* Base) This function works like FindChannelOfObject except that if no animation channel with the specifed name exists, a new one is created.
Function **GetChannelsOfObject**([out] v as Array[Channel]) as Integer	(*Inherited from* Base) Fills the array *v* with the animation channels acting on this object, returning the number of channels. If this function is called on a type that cannot be animated (such as *Director*), *v* will be empty.

Table 105:

Name	Description
Function **IsAnimated**() as Boolean	(*Inherited from* Base) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as *Director*), it always returns *false*.
Sub **SendVizCommandToHostGroup**(command as String)	Sends a viz command to the group of hostnames (which was previously filled by calls to *AddHostToGroup*).
Sub **SetChanged**()	(*Inherited from* Base) Mark this object as changed.

21.3.44 Global Procedures

Table 106:

Name	Description
Function **Abs**(x as Double) as Double	
Function **Acos**(x as Double) as Double	
Function **AngleBetweenVectors**(v1 as Vertex, v2 as Vertex) as Double	
Function **Asc**(s as String) as Integer	Returns the ASCII/UTF8 code (0-255) for the specified character.
Function **Asin**(x as Double) as Double	
Function **Atan**(x as Double) as Double	
Function **Atan2**(x as Double, y as Double) as Double	
Function **CBool**(v as Integer) as Boolean	Converts an integer to a boolean.
Function **CBool**(v as Double) as Boolean	Converts a double to a boolean.
Function **CBool**(v as Uuid) as Boolean	Converts a UUID to a boolean.
Function **CColor**(r as Double, g as Double, b as Double) as Color	Constructs a color object.
Function **CColor**(r as Double, g as Double, b as Double, a as Double) as Color	Constructs a color object.
Function **CDbl**(v as Boolean) as Double	Converts a boolean to a double.
Function **CDbl**(v as Integer) as Double	Converts an integer to a double.

Table 106:

Name	Description
Function **CDbl**(v as String) as Double	Converts a string to a double.
Function **Ceil**(x as Double) as Double	Returns the smallest integer that is larger than or equal to *x*.
Function **Chr**(charCode as Integer) as String	Returns the character associated with the specified ASCII/UTF8 code (0–255).
Function **CInt**(v as Boolean) as Integer	Converts a boolean to an integer.
Function **CInt**(v as Double) as Integer	Converts a double to an integer.
Function **CInt**(v as String) as Integer	Converts a string to an integer.
Function **Cos**(x as Double) as Double	
Function **Cosh**(x as Double) as Double	
Function **CStr**(v as Boolean) as String	Converts a boolean to a string.
Function **CStr**(v as Integer) as String	Converts a integer to a string.
Function **CStr**(v as Double) as String	Converts a double to a string.
Function **CStr**(v as Uuid) as String	Converts a UUID to a string.
Function **CUuid**(v as String) as Uuid	Constructs a uuid from a string value.
Function **CVertex**(v as Boolean) as Vertex	Constructs a vertex from a bool value, assigning the value to x, y, and z.
Function **CVertex**(v as Integer) as Vertex	Constructs a vertex from an integer value, assigning the value to x, y, and z.
Function **CVertex**(v as Double) as Vertex	Constructs a vertex from an double value, assigning the value to x, y, and z.
Function **CVertex**(x as Double, y as Double, z as Double) as Vertex	Constructs a vertex from 3 double values.
Function **Determinant**(a as Vertex, b as Vertex, c as Vertex) as Double	
Function **Distance**(v1 as Vertex, v2 as Vertex) as Double	Returns the distance between two vertices.
Function **Distance2**(v1 as Vertex, v2 as Vertex) as Double	Returns the squared distance between two vertices.
Function **DoubleToString**(value as Double, precision as Integer) as String	Converts a double value to a string.
Function **DoubleToString**(value as Double, precision as Integer, width as Integer) as String	Converts a double value to a string.

Table 106:

Name	Description
Function **Exp**(x as Double) as Double	
Function **Fix**(x as Double) as Double	Returns *Floor(x)* if x larger than or equal to 0.0, otherwise *Ceil(x)*.
Function **Floor**(x as Double) as Double	Returns the largest integer that is smaller than or equal to x.
Function **GetCurrentTime**() as DateTime	Gets the current time.
Function **GetParameterBool**(parameterName as String) as Boolean	Returns the value of the Boolean parameter *parameterName*.
Function **GetParameterColor**(parameterName as String) as Color	Returns the value of the Color parameter *parameterName*.
Function **GetParameterContainer**(parameterName as String) as Container	Returns the value of the Container parameter *parameterName*.
Function **GetParameterDouble**(parameterName as String) as Double	Returns the value of the Double parameter *parameterName*.
Function **GetParameterInt**(parameterName as String) as Integer	Returns the value of the Integer parameter *parameterName*.
Function **GetParameterString**(parameterName as String) as String	Returns the value of the String parameter *parameterName*.
Function **GetTickCount**() as Integer	Returns the processor time in milliseconds. Note that the timing resolution provided by this procedure is fairly low. For more accurate timing, use *System.GetElapsedTime()* instead.
Function **Int**(x as Double) as Double	Identical to Floor.
Function **IntToString**(value as Integer, width as Integer) as String	Converts an integer value to a right-aligned string, filling empty space with blanks.
Function **IntToString**(value as Integer, width as Integer, addLeadingZeros as Boolean) as String	Converts an integer value to a right-aligned string, filling empty space with blanks or (if *addLeadingZeros* is true) with zeros.
Function **Len**(s as String) as Integer	Returns the length of the specified string.

Table 106:

Name	Description
Function **LineLineIntersection**(line1Point as Vertex, line1Direction as Vertex, line2Point as Vertex, line2Direction as Vertex) as Vertex	
Function Log(x as Double) as Double	
Function **Max**(a as Integer, b as Integer) as Integer	Returns the larger of the two values.
Function **Max**(a as Double, b as Double) as Double	Returns the larger of the two values.
Function **Min**(a as Double, b as Double) as Double	Returns the smaller of the two values.
Function **Min**(a as Integer, b as Integer) as Integer	Returns the smaller of the two values.
Function **PlaneLineIntersection**(planePoint as Vertex, planeNormal as Vertex, linePoint as Vertex, lineDirection as Vertex) as Vertex	
Sub **Println**(v as Vertex)	Prints a vertex.
Sub **Println**()	Prints a newline.
Sub **Println**(v as Boolean)	Prints a boolean.
Sub **Println**(v as Integer)	Prints an integer.
Sub **Println**(v as Double)	Prints a double.
Sub **Println**(v as String)	Prints a string.
Sub **Println**(v as Variant)	Prints an object of type *Variant*.
Sub **Println**(v as Uuid)	Prints a uuid.
Function **Random**() as Double	Returns a random number of type double in the range [0, 1).
Function **Random**(i as Integer) as Integer	Returns a random integer in the range 0 through i–1.
Sub **RegisterDirSelector**(parameterName as String, label as String, default as String)	Registers a directory parameter.

Table 106:

Name	Description
Sub **RegisterFileSelector**(parameterName as String, label as String, defaultPath as String, defaultFile as String, filter as String)	Registers a file parameter.
Sub **RegisterInfoText**(infoText as String)	Call this function to provide documentation for the script (accessible via the 'i'-button).
Sub **RegisterParameterBool**(parameterName as String, label as String, defaultName as Boolean)	Registers a parameter of type Boolean.
Sub **RegisterParameterColor**(parameterName as String, label as String, defaultValue as Color)	Registers a parameter of type Color.
Sub **RegisterParameterContainer**(parameterName as String, label as String)	Registers a parameter of type Container.
Sub **RegisterParameterDouble**(parameterName as String, label as String, defaultValue as Double, minValue as Double, maxValue as Double)	Registers a parameter of type Double.
Sub **RegisterParameterDropDown**(parameterName as String, label as String, default as Integer, entries as Array[String], width as Integer, height as Integer)	Registers a dropdown box. *width* defines the parameter's width in number of characters and *height* sets the dropdown box's height in number of lines.
Sub **RegisterParameterHList**(parameterName as String, label as String, default as Integer, entries as Array[String], separatorChar as String, width as Integer, height as Integer)	Registers a list box. *separatorChar* defines the character which is used to separate the values. *width* and *height* are defined in pixels.
Sub **RegisterParameterImage**(parameterName as String, label as String)	Registers a parameter of type Image.
Sub **RegisterParameterInt**(parameterName as String, label as String, defaultValue as Integer, minValue as Integer, maxValue as Integer)	Registers a parameter of type Integer.

Table 106:

Name	Description
Sub **RegisterParameterList**(parameterName as String, label as String, default as Integer, entries as Array[String], width as Integer, height as Integer)	Registers a list box. *width* and *height* are defined in pixels.
Sub **RegisterParameterSliderDouble**(parameterName as String, label as String, defaultValue as Double, minValue as Double, maxValue as Double, width as Integer)	Registers a parameter of type Double with slider appearance. *width* specifies the parameter's width in pixels.
Sub **RegisterParameterSliderInt**(parameterName as String, label as String, defaultValue as Integer, minValue as Integer, maxValue as Integer, width as Integer)	Registers a parameter of type Integer with slider appearance. *width* specifies the parameter's width in pixels.
Sub **RegisterParameterString**(parameterName as String, label as String, defaultString as String, widgetSize as Integer, maxLength as Integer, charSet as String)	Registers a parameter of type String. If *charSet* is empty, all characters are allowed.
Sub **RegisterParameterText**(parameterName as String, defaultText as String, width as Integer, height as Integer)	Registers a parameter of type String that is entered in a text box. *width* and *height* specify the size of the text box in pixels.
Sub **RegisterPluginVersion**(version as Integer, release as Integer, patchlevel as Integer)	You can use this to set information about your script-plugin's version. It's recommended to set version numbers especially for VSL plugins.
Sub **RegisterPushButton**(parameterName as String, label as String, buttonId as Integer)	Registers a push button.
Sub **RegisterRadioButton**(parameterName as String, label as String, defaultIndex as Integer, buttonNames as Array[String])	Register a block of radio buttons.
Function **Round**(x as Double) as Double	Rounds *x* to the nearest integer

Table 106:

Name	Description
Sub **SendGuiParameterShow**(parameterName as String, state as Integer)	Shows or hides a parameter in the GUI. state can be set to *SHOW* or *HIDE*.
Sub **SendGuiParameterUpdate**(parameterName as String, state as Integer)	Enables or disables GUI parameter updates. state can be set to *ENABLED* or *DISABLED*.
Sub **SendGuiRefresh**()	Sends a refresh command to the GUI
Sub **SendGuiStatus**(parameterName as String, state as Integer)	Enables or disables a parameter in the GUI. state can be set to *ENABLED* or *DISABLED*.
Sub **SetAnimateParameter**(animate as Boolean)	Disable or enable animation on the last registered parameter.
Sub **SetHideParameter**()	Hides the last registered parameter in the GUI.
Function **Sign**(x as Double) as Integer	Returns -1 if x less than 0.0, 1 if x larger than 0.0, 0 if x equals 0.0.
Function **Sin**(x as Double) as Double	
Function **Sinh**(x as Double) as Double	
Function **Sqr**(x as Double) as Double	Returns *x*x*.
Function **Sqrt**(x as Double) as Double	Computes the square root of *x*.
Function **Tan**(x as Double) as Double	
Function **TriangleArea**(a as Vertex, b as Vertex, c as Vertex) as Double	
Function **TriangleCenter**(a as Vertex, b as Vertex, c as Vertex) as Vertex	
Sub **UpdateGuiParameterDoubleDefMinMax**(parameterName as String, defaultValue as Double, minValue as Double, maxValue as Double)	Sets new default, min and max value for parameters like Double, SliderDouble etc.
Sub **UpdateGuiParameterEntries**(parameterName as String, entries as Array[String])	Sets new entries for parameters like List or DropDown
Sub **UpdateGuiParameterIntDefMinMax**(parameterName as String, defaultValue as Integer, minValue as Integer, maxValue as Integer)	Sets new default, min and max value for parameters like Int, SliderInt etc.

Table 106:

Name	Description
Function IsKeyDown(keyCode as Integer*) as* Boolean	*Deprecated. Use System.IsKeyDown.*
Function IsLButtonDown() as Boolean	*Deprecated. System.LButtonState.*
Function IsMButtonDown() as Boolean	*Deprecated. System.MButtonState.*
Function IsRButtonDown() as Boolean	*Deprecated. System.RButtonState.*
Function SendCommand(command as String*) as* String	*Deprecated. Use System.SendCommand.*
Function SendRemoteCommand(hostname as String, *port as* Integer, *commandString as* String, *timeoutInMs as* Integer*) as* String	*Deprecated. Use System.SendRemoteCommand.*
Function TcpSend(hostname as String, port as Integer, *commandString as* String, *timeoutInMs as* Integer*) as* String	*Deprecated. Use System.TcpSend.*

22 Glossary

This section describes words and acronyms that are used throughout the documentation.

Table 107: Glossary

Name (acronym)	Description
5.1	Delivers five audio channels and 1 Low-frequency Effect (LFE) channel from a 6 channel source.
6DoF	Six degrees of freedom (6DoF) is used in interactive scenes to move and rotate objects in 3D on a defined grid.
7.1	Delivers seven audio channels and one Low-frequency Effect (LFE) channel from an 8 channel source.
AES	Audio Engineering Society (AES). Also known for the digital audio standard frequently called AES/EBU, officially known as AES3, is used for carrying digital audio signals between various devices; hence, the term AES/EBU connectors.
AGP	Accelerated Graphics Port (AGP) is a high-speed point-to-point channel for attaching a graphics card to a computer's motherboard, primarily to assist in the acceleration of 3D computer graphics.
Anamorphic widescreen	Anamorphic widescreen is a videographic technique utilizing rectangular (wide) pixels to store a widescreen image to standard 4:3 aspect ratio.

Table 107: Glossary

Name (acronym)	Description
ANC data	Ancillary data refers to a means which by non-video information (such as audio, other forms of essence, and metadata) may be embedded within the serial digital interface (SDI). Ancillary data packets are commonly divided into two types, depending on where they are located. Ancillary packets located in the *horizontal blanking region* (after EAV but before SAV), regardless of line, are known as horizontal ancillary data, or HANC. HANC is commonly used for higher-bandwidth data, and/or for things that need to be synchronized to a particular line; the most common type of HANC is *embedded audio*. Ancillary packets located in the *vertical blanking region*, and after SAV but before EAV, are known as vertical ancillary data, or VANC. VANC is commonly used for low-bandwidth data, or for things that only need be updated on a per-field or per-frame rate. Closed *caption data* and *VPID* are generally stored as VANC. Note: ANC packets which lie in the dataspace which is in both the horizontal and vertical intervals, is considered to be HANC and not VANC.
Anti aliasing	Anti-aliasing is the technique of minimizing the distortion artifacts known as aliasing when representing a high-resolution signal at a lower resolution.
Audio codec	An audio codec is a hardware device or a computer program that compresses and decompresses digital audio data according to a given audio file format or streaming audio format.
AVI	Audio Video Interleave (AVI) is a multimedia container format introduced by Microsoft in November 1992 as part of its Video for Windows technology.
Bi-level sync	In standard-definition applications, a bi-level sync signal, often with a colorburst signal in facilities that have analog equipment. Typically, this is either in NTSC or PAL format. As the resulting signal is usually indistinguishable from an all-black television signal of the same format, this sort of reference is commonly known as black or black burst. See SPG and Tri-level sync.
BIOS	Basic Input/Output System (BIOS) is a system that defines the firmware interface. The initial function of the BIOS is to identify, test, and initialize system devices, and prepare the machine into a known state.
Black burst	See SPG and Bi-level sync.

Table 107: Glossary

Name (acronym)	Description
BNC	The BNC (Bayonet Neill Concelman) connector is a very common type of RF connector used for terminating coaxial cable.
BSOD	The Blue Screen of Death (also known as a stop error or blue screen) is an error screen displayed by some operating systems, most notably Microsoft Windows, after encountering a critical system error which can cause the system to shut down, to prevent damage.
Configuration file	Holds all configurable settings used by Viz Engine and various integrated applications. The configuration file can be found under Viz Engine's installation directory; C:\Program Files\vizrt\viz3\.
CVBS	Composite video (aka CVBS or Composite Video Burst Signal) is the format of an analog television (picture only) signal before it is combined with a sound signal and modulated onto an RF carrier. It is a composite of the three source signals Y, U and V (YUV) with sync pulses.
D10	Television: Type D-10 Stream Specifications: MPEG-2 4:2:2P @ ML for 525/60 and 625/50. SMPTE D10 (356M) is a professional video compression format composed of MPEG-2 Video 4:2:2 Intra-frame (I-frame) only and 8 channel AES3 audio streams.
D12	Television: 6.35-mm Type D-12 Component Format: Digital Recording at 100 Mb/s 1080/60i, 1080/50i, 720/60p. Also known as Digital Betacam.
DIF	DV-DIF (Digital Video Digital Interface Format) is an interface format for DV, the Digital Video family of related video encodings (DV, DVCAM, and DVCPRO). DV Audio/Video data can be stored as raw DV data stream file (data is written to a file as the data is received over FireWire, file extensions are .dv and .dif) or the DV data can be packed into AVI container files.
Digital Voodoo	Digital Voodoo is a broadcast capture and playback card used for uncompressed Standard Definition 10-bit SDI solution on Macintosh computers.

Table 107: Glossary

Name (acronym)	Description
DirectSound	DirectSound is a software component of the Microsoft DirectX library, that resides on a computer with the Windows operating system. It provides a direct interface between applications and the sound card drivers on Windows. It provides audio data to the sound card, recording and mixing sound; adding effects to sound for example reverb, echo, flange; using hardware controlled buffers for extra speed; positioning sounds in 3D space (3D audio spatialization), capturing sounds from a microphone or other input and controlling capture effects during audio capture.
DV	Digital Video.
DV25	DV25 applies a 5:1 compression and a 4:1:1 color space and achieves a fixed data rate of 25Mbps.
DV50	DV50 applies a 3.3:1 compression and a 4:2:2 color space and achieves a fixed data rate of 50Mbps.
DVCAM	Sony's DVCAM is a professional variant of the DV standard that uses the same cassettes as DV and MiniDV, but transports the tape 33% faster.
DVCPRO	Panasonic's DVCPRO family is specifically created for electronic news gathering use, with good linear editing capabilities and robustness.
DVI	Digital Visual Interface (DVI) is a video interface standard designed to maximize the visual quality of digital display devices such as flat panel LCD computer displays and digital projectors.
EAS	The Emergency Alert System (EAS) is a national warning system in the United States of America (USA). The official EAS enables the President of the USA to speak to the citizens of the USA within 10 minutes.
EAV	Horizontal scan lines of video pixel data are delimited in the stream by 4-byte long SAV (Start of Active Video) and EAV (End of Active Video) code sequences. SAV codes also contain status bits indicating line position in a video field or frame. Line position in a full frame can be determined by tracking SAV status bits, allowing receivers to “synchronize” with an incoming stream. ANC data.
EBU	European Broadcasting Union. AES.

Table 107: Glossary

Name (acronym)	Description
GenLock	Genlock (Generator Lock) is a common technique where the video output of one source, or a specific reference signal, is used to synchronize other television picture sources together. GenLock type is set in Viz Config's Video Output section.
Glyph	A glyph is an element of writing. Glyphs may also be Ligatures, that is, compound characters, or diacritics.
HANC	See ANC data.
HD	High-definition television or HDTV is a digital television broadcasting system with higher resolution than traditional television systems (standard-definition TV, or SDTV).
I/O	I/O is short for Input/Output, and is a common term used for hardware connectors.
IEEE	The Institute of Electrical and Electronics Engineers, Inc.
IRE	Institute of Radio Engineers. Is now part of the IEEE.
IRE unit	An IRE unit is used in the measurement of composite video signals. The range of a video signal is defined to be 1.0 volts peak to peak (p/p) which again is divided into the picture and the synchronization portion of the signal. The picture portion is defined as 0 IRE to 100 IRE. The synch portion is defined as -40 IRE to 0 IRE; hence, the total range of IRE values is 140 units making each IRE unit equivalent to 1/140 volts. Picture sources are typically set with *black* somewhere between 7.5 to 10 IRE. The maximum signal brightness is usually set to just below 100 IRE. A third signal, the burst signal, is present from +20IRE to -20IRE. It is a short sample of the color subcarrier frequency, and is used to control how colors are displayed on screen.
LED	Light-emitting-diode.
Ligature	In writing and typography, a ligature occurs where two or more graphemes (fundamental unit in a written language) are joined as a single Glyph.
Lip-sync	Lip-sync (short for lip synchronization) is a technical term for matching lip movements with voice. The term refers to a technique in movie and broadcast production for synchronization of visual and audio signals.
LTC	Linear TimeCode. VITC.

Table 107: Glossary

Name (acronym)	Description
MCX	MCX connectors are coaxial RF connectors. They use a snap-on interface and usually have a 50 Ohm impedance. They offer broadband capability from direct current (DC) to 6 GHz.
MDIF	Matrox Dongle Information File (.MDIF) can be generated and sent to Vizrt in order to upgrade the Matrox X.mio video card to a higher class model.
MDUF	Matrox Dongle Upgrade File (.MDUF) is the file, provided by Vizrt, that will upgrade the Matrox X.mio video card to a higher class model.
MMCX	MMCX (micro-miniature coaxial) connectors are coaxial RF connectors similar to MCX but smaller.
Mono	Monaural (often shortened to mono) sound reproduction is single-channel.
MPEG	Moving Picture Experts Group, is the name of a family of standards used for coding audio-visual information (e.g., movies, video, music) in a digital compressed format.
MXF	Material Exchange Format (MXF) is an object-based file format that wraps video, audio, and other bitstreams, optimized for content interchange or archiving by creators and/or distributors, and intended for implementation in devices ranging from cameras and video recorders to computer systems.
NTSC	National Television System Committee (NTSC) is the analog television system used in the north, mid and to some extend in south America and also in some Asian countries.
OB	Outside Broadcasting is the production of television or radio programmes (typically to cover news and sports events) from a mobile television studio.
OP-Atom	MXF Operational Pattern Atom. OP-Atom was designed to address the needs of non-linear video and audio editing. Reference: SMPTE 390M-2004 Television - Material Exchange Form at (MXF) - Specialized Operational Pattern “Atom” (Simplified Representation of a Single Item).
PAL	Phase Alternating Line, is a color-encoding system used in broadcast television systems in large parts of the world.

Table 107: Glossary

Name (acronym)	Description
PCI	PCI Local Bus (usually shortened to PCI), or Conventional PCI, specifies a computer bus for attaching peripheral devices to a computer motherboard.
PCIe	PCI Express was designed to replace the general-purpose PCI (Peripheral Component Interconnect) expansion bus, the high-end PCI-X bus and the AGP graphics card interface.
PCI-X	PCI-X (Peripheral Component Interconnect eXtended) is a computer bus and expansion card standard that enhanced the PCI Local Bus for higher bandwidth demanded by servers.
PCM	Pulse-code modulation (PCM) is a digital representation of an analog signal.
Quad	Quadraphonic (or quadrophonic) sound – also called 4.0 stereo – uses four channels in which speakers are positioned at the four corners of the listening space, reproducing signals that are (wholly or in part) independent of one another.
Rack unit	A rack unit or U (less commonly, RU) is a unit of measure used to describe the height of equipment intended for mounting in a 19-inch rack or a 23-inch rack (the dimension referring to the width of the rack).
RF	A radio frequency (RF) connector is an electrical connector designed to work at radio frequencies in the multi-megahertz range.
RGB	RGB stands for Red Green Blue, and its color model is an additive color model in which red, green, and blue light are added together in various ways to reproduce a broad array of colors.
RGBA	RGBA stands for Red Green Blue Alpha. It uses the RGB color model with extra information, and is normally used as an transparency/opacity channel.
Ring buffer	A circular buffer or ring buffer is a data structure that uses a single, fixed-size buffer as if it were connected end-to-end. This structure lends itself easily to buffering data streams.

Table 107: Glossary

Name (acronym)	Description
Safe Area (aka Action Safe Area)	Depending on how a television set is adjusted, viewers can see a larger area than the Title Area (aka Title Safe Area). The safe area is in Viz Artist shown as a green rectangle in the preview window. This area can be considered a “safe margin” to create a buffer and avoid that elements are butted up against the edge of the screen. However, the area outside the safe area is often used to place graphics (for example tickers that run horizontally across the screen) that butt up against the edges of the screen.
SATA	The Serial Advanced Technology Attachment (SATA) computer bus has the primary function of transferring data between the motherboard and mass storage devices (such as hard disk drives and optical drives) inside a computer.
SAV	See EAV and ANC data.
SD	Standard-definition television or SDTV refers to television systems that have a resolution that meets standards but not considered either enhanced definition or high definition.
SDI	Serial digital interface (SDI) refers to a family of video interfaces standardized by the SMPTE.
SECAM	Séquentiel couleur à mémoire, French for “Sequential Color with Memory”), is an analog color television system first used in France.
Shader	A shader in the field of computer graphics is a set of software instructions, which is used by the graphics resources primarily to perform rendering effects. A shader is a part of the renderer, which is responsible for calculating the color of an object; hence, the Viz Engine renderer has a shader that handles the video input (normally provided as interlaced YUV images). To be able to use the YUV images in a CG scene they need to be RGB, and in addition they need to be de-interlaced. All this is done by Viz Engine's shader. The shader not only takes two, but four input images when it does the color conversion from YUV to RGB and the de-interlacing. As this is a very heavy process it can only be done in SD.
SMPTE	Society of Motion Picture and Television Engineers.

Table 107: Glossary

Name (acronym)	Description
SPG	A sync pulse generator (SPG) is a special type of generator which produces synchronization signals, with a high level of stability and accuracy. These devices are used to provide a master timing source for a video facility. See Bi-level sync and Tri-level sync.
Squeezeback	Squeezeback (or squeeze back) is a screen size adjustment process that reduces the size of a display area to allow other items (such as logos, text crawls or other graphics) to be seen in the new exposed display area. Squeezebacks are often seen during the closing credits of a show to introduce the viewer to upcoming content along with other promotions.
Stereo	Stereophonic sound (often shortened to Stereo) is the reproduction of sound, using two or more independent audio channels.
Sub-D	Sub-D, D-subminiature or D-sub is a common type of electrical connector used particularly in computers.
Surround sound	See Quad, 5.1 and 7.1.
Title Area (aka Title Safe Area)	The title area is in Viz Artist shown as a purple rectangular area which is far enough in from the four edges, such that text or graphics show neatly: with a margin and without distortion.
Tri-level sync	In some high-definition applications, a tri-level sync signal is used instead. This signal is virtually identical to the synchronization signal used in component analogue video (CAV); and is similar to the synchronization signals used in VGA (the main difference being, in VGA the horizontal and vertical syncs are carried on different wires; whereas TLS signals include both H and V syncs). See SPG and Bi-level sync.
Universal Naming Convention (UNC)	Specifies a common syntax to describe the location of a network resource, such as a shared file, directory, or printer. \\ComputerName\SharedFolder\Resource
VANC	See ANC data.
VBI	Vertical blanking interval (VBI) is the time difference between the last line of one frame or field of a raster display, and the beginning of the next.
Video codec	A video codec is a device or software that enables video compression and/or decompression for digital video.
VITC	Vertical Interval TimeCode. LTC and VBI.

Table 107: Glossary

Name (acronym)	Description
Writing system	A writing system is a type of symbolic system used to represent elements or statements expressible in language.
XLR	The XLR connector is an electrical connector, and a standard for broadcast audio equipment. The 3-pin XLR3 is used as a balanced audio connector for both analog and digital audio. Note that Viz Engine's Matrox board only supports digital audio.
YUV	YUV is a color space typically used as part of a color image pipeline. Y' stands for the luma component (the brightness) and U and V are the chrominance (color) components. The Y'UV color model is used in the NTSC, PAL, and SECAM composite color video standards.

Index