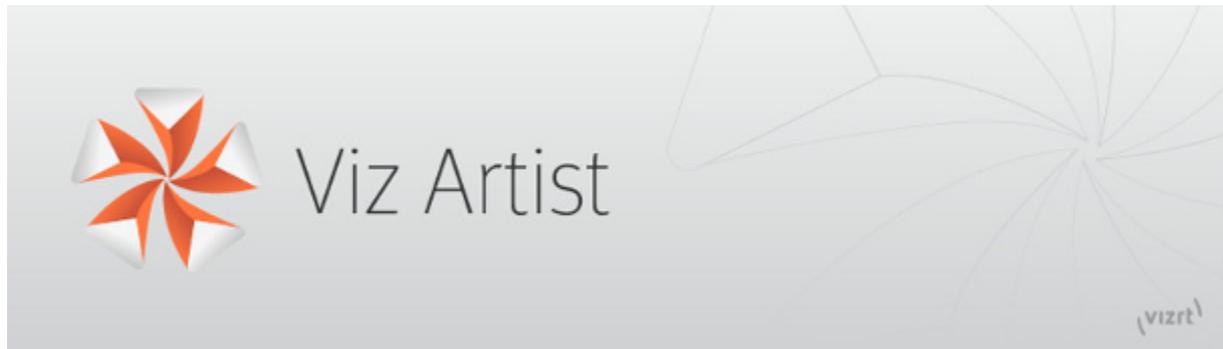


Viz Artist
User's Guide
3.7



Copyright © 2015 Vizrt. All rights reserved.

No part of this software, documentation or publication may be reproduced, transcribed, stored in a retrieval system, translated into any language, computer language, or transmitted in any form or by any means, electronically, mechanically, magnetically, optically, chemically, photocopied, manually, or otherwise, without prior written permission from Vizrt.

Vizrt specifically retains title to all Vizrt software. This software is supplied under a license agreement and may only be installed, used or copied in accordance to that agreement.

Disclaimer

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

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

11.08.2015

Table of Contents

1 Introduction	27
1.1 Document Structure	27
1.2 Related Documents	28
1.3 Customer Feedback and Suggestions	29
1.4 Customer Support Request	30
1.4.1 Submitting a Support Request	30
1.4.2 Attach Log Files to a Customer Request	32
2 Getting Started	33
2.1 Viz Artist/Engine Folders	33
2.2 Viz Artist Startup and Close	34
2.2.1 Viz Artist 64-Bit or 32-Bit Version	38
2.2.2 Viz Configuration	38
2.3 Viz Command Line Options	39
3 Workbench Description	41
3.1 Main Window	41
3.1.1 Dual Monitor Configuration	42
3.2 Main Menu	43
3.2.1 Main Menu Left	43
3.2.2 Main Menu Right	46
3.3 Scene Editor	48
3.3.1 Scene Editor Buttons	49
3.3.2 Layer Manager	50
3.3.3 Snapshot	52
3.3.4 Performance Bar	53
3.3.5 Grid Toolbar	54
3.4 Working with the Scene Editor	56
3.4.1 Scene Editor Functions	56
3.4.2 Scene Editor Context Menu	59
3.5 Scene Editor Views	62
4 Scene Tree Knowledge and Skills	65
4.1 Open a Scene	65
4.2 Scene Tree	66
4.3 Scene Tree Menu	67
4.3.1 Tree Button	68

4.3.2 Sort Button.....	69
4.3.3 Search Button.....	70
4.3.4 Media Assets [Manager].....	73
4.4 Favorites Bar.....	75
4.5 Containers.....	76
4.5.1 Container Icons.....	77
4.5.2 Container Context Menu.....	79
4.5.3 Container Editor.....	81
4.5.4 Add Containers to a Scene.....	82
4.5.5 Select Multiple Containers.....	83
4.5.6 Copy Containers.....	84
4.5.7 Move Containers.....	84
4.5.8 Rename Containers.....	85
4.5.9 Edit Container Properties.....	86
4.5.10 Delete Containers.....	86
4.5.11 Merge Containers.....	87
4.5.12 Group Containers.....	89
4.6 Modify Container Properties.....	90
4.6.1 Manipulate Container Properties.....	90
4.6.2 Multiple Selection Container Properties.....	92
4.6.3 Editing Properties in the Scene Editor.....	94
4.6.4 Transformation Editor.....	94
4.6.5 Copy Properties.....	97
4.6.6 Geometry Editor.....	98
4.6.7 Text Editor.....	100
4.6.8 Texture Editor.....	107
4.6.9 Plugin Editor.....	113
4.6.10 Material Editor.....	114
5 Manage Work and Graphic Hub.....	117
5.1 Graphic Hub.....	117
5.2 Server Area.....	118
5.2.1 Server Tree Bar.....	119
5.2.2 Server Tree Menu.....	119
5.3 Folder Tree.....	120
5.3.1 Folders.....	120
5.4 What are Elements.....	124
5.4.1 Scenes.....	125

5.4.2 Geometries.....	125
5.4.3 Materials.....	126
5.4.4 Materials Advanced.....	127
5.4.5 Images.....	127
5.4.6 Fonts.....	128
5.4.7 Audio Clips.....	128
5.5 Element Area.....	129
5.5.1 Element List.....	129
5.5.2 Element Context Menu.....	132
5.5.3 Content Switcher.....	134
5.5.4 Server Menu.....	136
5.5.5 Properties Panel.....	137
5.6 Working with Elements.....	140
5.6.1 Common Element Procedures.....	140
5.6.2 Active Elements.....	143
5.6.3 Linked Elements.....	145
5.6.4 Referenced Elements.....	146
5.6.5 Working with Scene Elements.....	146
5.6.6 Working with Geometry Elements.....	149
5.6.7 Working with Material Elements.....	149
5.6.8 Working with Material Advanced Elements.....	150
5.6.9 Working with Image Elements.....	152
5.6.10 Working with Fontstyle Elements.....	154
5.6.11 Working with Audio (Clips) Elements.....	155
5.7 Add Properties to Containers and Scenes.....	158
5.8 Assign Keywords to Elements.....	163
5.9 Image Editor.....	169
5.9.1 General Image Information.....	170
5.9.2 Apply and Preview Effects.....	173
5.10 Fontstyle Editor.....	173
5.10.1 Available Modifiers.....	175
5.10.2 Working with Modifiers.....	176
5.10.3 Preview Menu.....	180
5.10.4 Fontstyle Viewer.....	181
5.11 Material Editor.....	182
5.11.1 Create a Material.....	182
5.11.2 Material Color Properties and Schemes.....	183

5.12 Element Search.....	186
5.12.1 Search Panel.....	187
5.12.2 Search Parameters Area.....	188
5.12.3 Search Menu.....	188
5.12.4 Property Search.....	189
5.12.5 Reference Search.....	190
5.12.6 Keyword Search.....	195
5.12.7 Directory Search.....	196
5.12.8 Combination Search.....	197
5.12.9 Search Result Area.....	198
5.13 SmartView Search.....	200
5.13.1 Search String Syntax.....	201
5.13.2 SmartView Searches.....	202
5.13.3 File Sets.....	203
5.14 Background Loading.....	205
5.14.1 Automatic Background Loading.....	205
5.14.2 Manual Background Loading.....	205
5.14.3 Background Loading Limitations.....	206
5.15 Built Ins.....	207
5.15.1 Content Tabs.....	208
5.15.2 Plugin Context Menu.....	208
5.15.3 View and Search for Plugins.....	210
5.15.4 Add Plugins to Containers.....	212
6 Global Tasks.....	215
6.1 Scene Settings.....	215
6.1.1 Global Settings.....	216
6.1.2 Media Asset Properties.....	219
6.1.3 Rendering.....	224
6.1.4 Plugin.....	228
6.1.5 Script.....	229
6.1.6 Clipper.....	230
6.1.7 HDR (High Dynamic Range).....	231
6.1.8 Virtual Studio.....	232
6.2 Video Clip Playback Considerations.....	232
6.2.1 3 and 4-Point Loops in Clip Player.....	233
6.2.2 General, Mixed and Special Modes.....	235
6.2.3 Clip Handling.....	235

6.2.4 Settings.....	236
6.2.5 Scripting and Plugin interface.....	237
6.2.6 IO Matrix.....	237
6.3 Video Playback.....	239
6.3.1 Video Application.....	240
6.3.2 Video Clip Storage.....	240
6.3.3 Video Clip Playback Use Cases.....	241
6.3.4 Video Clip Playback.....	244
6.3.5 Live Video Feeds.....	251
6.3.6 Live Feed from a Video Stream.....	253
6.3.7 Keying Mode for Video.....	256
6.3.8 Keying Mode Parameters.....	259
6.3.9 Seamless Input Channel Switcher.....	263
6.4 Supported Codecs.....	264
6.4.1 Matrox Codecs.....	264
6.4.2 PAL Codecs.....	265
6.4.3 NTSC Codecs.....	270
6.4.4 720p50 and 720p59.94 (60M) Codecs.....	275
6.4.5 1080i25 and 1080i29.97 (30M) Codecs.....	281
6.4.6 1080p50 and 1080p59.94 (60M) Codecs.....	287
6.4.7 QuickTime Writer Plugin Codecs.....	288
6.4.8 DVS and Bluefish Codecs.....	289
6.5 Import Files and Archives.....	289
6.5.1 Import Window.....	290
6.5.2 Supported Files for Import.....	291
6.5.3 Import Files.....	294
6.5.4 Import Archives.....	296
6.6 Archiving.....	300
6.6.1 Create an Archive.....	301
6.7 Deploy Elements.....	302
6.7.1 Deploy Panel Overview.....	303
6.7.2 Deploy Panel Procedures.....	303
6.8 Post Rendering.....	305
6.8.1 Post Render Screen.....	306
6.8.2 Post Render a Scene.....	306
6.8.3 Post Render Templates.....	309
6.8.4 Post Rendering Plugins.....	311

6.8.5 Advanced Issues with Video Codecs.....	313
6.9 Integration with 3rd Party Software.....	314
6.9.1 Adobe After Effects Integration.....	315
6.9.2 CINEMA 4D Integration.....	315
6.9.3 TriCaster Integration.....	317
7 Lights.....	319
7.1 Types Of Light.....	319
7.2 Light Editor.....	320
7.2.1 Light Editor Left Panel.....	320
7.2.2 Light Editor Right Panel.....	321
7.2.3 Light Editor Procedures.....	323
7.3 Light Visualization.....	327
7.4 Light Source Animation.....	329
8 Cameras.....	331
8.1 Camera Editor.....	331
8.1.1 Left Panel Parameters.....	332
8.1.2 Right Panel Parameters.....	333
8.1.3 Camera Views.....	334
8.1.4 Parameters for Perspective View.....	336
8.1.5 Parameters for Orthogonal View.....	340
8.1.6 Parameters for Window View.....	340
8.1.7 Configure a Virtual Window.....	341
8.2 Stereo Settings.....	342
8.3 Stereoscopy Best Practices.....	343
8.4 Stereoscopic Output Using Shutter Glasses.....	344
8.5 Change Camera Parameters in Orthogonal Views.....	345
8.6 Track Objects with a Camera.....	346
8.7 Receive Tracking Data from a Real Camera.....	348
8.8 Copy Properties from one Camera to Another.....	348
8.9 Camera Selection.....	349
8.9.1 Camera Selection in the Scene Editor.....	351
8.9.2 Camera Selection in the Scene Editor Views.....	352
8.10 Camera Animation.....	352
9 Animation Stage.....	355
9.1 Stage Tree Area.....	356
9.1.1 Stage Tree.....	356
9.1.2 Stage Tree Bar.....	357

9.1.3 Stage Tree Menu.....	357
9.2 Stage Editor.....	360
9.2.1 Stage Editor Menu.....	361
9.2.2 Stage Editor Bar.....	361
9.3 Timeline Editor.....	363
9.4 Timeline Marker.....	364
9.5 Director Control Panel.....	365
9.6 Director Editor.....	366
9.7 Using the Master Clip Feature.....	368
9.8 Actor Editor.....	368
9.9 Channel Editor.....	369
9.9.1 Common Channel Editor Properties.....	369
9.9.2 Position Channel Editor Properties.....	370
9.9.3 Action Channel Editor Properties.....	371
9.10 Dopesheet Editor.....	372
9.11 Spline Editor.....	374
9.11.1 Spline Editor Procedures.....	374
9.11.2 Editing Handles.....	375
9.11.3 Position in the Spline Editor.....	376
9.11.4 Rotation in the Spline Editor.....	376
9.11.5 Scale in the Spline Editor.....	376
9.11.6 Material in the Spline Editor.....	377
9.11.7 Additional Best Practices for the Spline Editor.....	377
9.12 Stage Object Editor.....	378
9.13 Key Frame Editor.....	378
9.13.1 Common Key Frame Editor Properties.....	379
9.13.2 Locked vs Unlocked Key Frames.....	379
9.13.3 Action Key Frame Editor Properties.....	380
9.13.4 Position Key Frame Editor Properties.....	381
9.13.5 Rotation Key Frame Editor Properties.....	383
9.13.6 Scale Key Frame Editor Properties.....	384
9.13.7 Material Key Frame Editor Properties.....	385
9.13.8 Clip Key Frame Editor Properties.....	386
9.14 Event Editor.....	387
10 Create Animations.....	389
10.1 Working with Directors in the Stage Tree.....	389
10.2 Working with Directors in the Dopesheet Editor.....	392

10.3 Working with Events in a Director	393
10.4 Working with Actors in the Stage Tree	396
10.5 Working with Actors in the Dopesheet Editor	398
10.6 Working with Channels in the Stage Tree	399
10.7 Working with Channels in the Dopesheet Editor	400
10.8 Working with Action Channels in the Stage Tree	401
10.9 Working with Action Channels in the Dopesheet Editor	403
10.10 Working with Key Frames	403
10.10.1 Reverse the Order of Selected Key Frames	406
10.11 Basic Animation Functions	407
10.12 Creating a Basic Animation	410
10.13 Create an Advanced Animation	413
10.14 Further Advanced Animation Topics	416
10.14.1 Animating Channels	416
10.14.2 Path Control	418
10.14.3 Spline Control	419
10.14.4 Animate Audio	421
10.14.5 Animate Cameras	422
10.14.6 Animate Light	423
10.14.7 Merging Animations	423
10.14.8 Multi Select Animation	423
10.15 Create an Over the Shoulder Scene	424
10.16 Create a Stand-alone Scene	435
10.17 Create Transition Effects	442
11 Geometry Plugins	447
11.1 Default	447
11.1.1 2D Line	448
11.1.2 2D Patch	449
11.1.3 2D Ribbon	452
11.1.4 Alpha Map	458
11.1.5 Arrow	459
11.1.6 Circle	460
11.1.7 Cog Wheel	461
11.1.8 Cone	463
11.1.9 Connector	464
11.1.10 Cube	465
11.1.11 Cycloid	467

11.1.12 Cylinder	468
11.1.13 Cylinder3	470
11.1.14 Dexter	471
11.1.15 DisplacementMap	484
11.1.16 Eclipse	485
11.1.17 Fade Rectangle	487
11.1.18 Filecard	488
11.1.19 Graph	489
11.1.20 Graph2D	490
11.1.21 Icosahedron	492
11.1.22 Image FX	493
11.1.23 Noggi	499
11.1.24 Pointer	501
11.1.25 Polygon	502
11.1.26 Rectangle	509
11.1.27 Ring	511
11.1.28 Roll	512
11.1.29 SoftClip Draw Pixels	513
11.1.30 Sphere	515
11.1.31 Spline Path	515
11.1.32 Spline Strip	517
11.1.33 Spring	521
11.1.34 Star	522
11.1.35 Torus	523
11.1.36 Trio Scroll	524
11.1.37 Wall	526
11.1.38 Wave	528
11.2 PixelFX Plugins	529
11.2.1 pxLensEnergyBolt	531
11.2.2 Lens Flare Geometries	531
11.2.3 pxLensRays	532
11.3 Primitives	533
11.3.1 N Quad	533
11.4 RealFX Plugins	534
11.4.1 Cloth	535
11.4.2 RFXSmoke	537
11.5 Ticker	542

11.5.1 Scroller	542
11.6 Visual Data Tools	545
11.6.1 Overview	546
11.6.2 Area Chart	546
11.6.3 Bar Chart	550
11.6.4 Line Chart	553
11.6.5 Pie Chart	556
11.6.6 Scatter Chart	559
11.6.7 Stock Chart	561
12 Container Plugins	565
12.1 Arrange	565
12.1.1 Circle Arrange	565
12.1.2 Grid Arrange	567
12.1.3 Time Displacement	569
12.2 Container	572
12.2.1 BoundingBox	572
12.2.2 Cobra	573
12.2.3 Coco	574
12.2.4 Colin	575
12.2.5 Cora	577
12.2.6 Corena	578
12.2.7 Toggle	579
12.3 Container FX	583
12.3.1 Common Container FX Properties	584
12.3.2 CFX 2D Follow	585
12.3.3 CFX Alpha	586
12.3.4 CFX Arrange	587
12.3.5 CFX Color	588
12.3.6 CFX Explode	588
12.3.7 CFX Jitter Alpha	590
12.3.8 CFX Jitter Color	590
12.3.9 CFX Jitter Position	591
12.3.10 CFX Jitter Scale	592
12.3.11 CFX Plus Plus	593
12.3.12 CFX Rotate	594
12.3.13 CFX Scale	595
12.4 Control	596

12.4.1 Common Control Plugin Properties	597
12.4.2 Apply Shared Memory	597
12.4.3 Control Action	600
12.4.4 Control Action Table	601
12.4.5 Control Audio	602
12.4.6 Control Bars	603
12.4.7 Control Chart	603
12.4.8 Control Clip	604
12.4.9 Control Clock	605
12.4.10 Control Condition	606
12.4.11 Control Container	607
12.4.12 Control Data Action	608
12.4.13 Control Datapool	608
12.4.14 Control DP Object	609
12.4.15 Control FeedView	610
12.4.16 Control Geom	611
12.4.17 Control Hide in Range	611
12.4.18 Control Hide on Empty	612
12.4.19 Control Image	613
12.4.20 Control Key Frame	614
12.4.21 Control List	615
12.4.22 Control Map	617
12.4.23 Control Material	618
12.4.24 Control Num	618
12.4.25 Control Object	619
12.4.26 Control Omo	624
12.4.27 Control Parameter	624
12.4.28 Control Payload	626
12.4.29 Control Pie	626
12.4.30 Control Scaling	628
12.4.31 Control Sign Container	629
12.4.32 Control SoftClip	629
12.4.33 Control Stoppoint	630
12.4.34 Control Targa Alpha	631
12.4.35 Control Targa Clip	631
12.4.36 Control Text	632
12.4.37 Control Video	633

12.4.38 Control World	634
12.4.39 Placeholder	635
12.5 Default	636
12.5.1 Trio Scroll Element	636
12.6 Feed	637
12.6.1 Feed Activate	638
12.6.2 Feed View	638
12.6.3 Hide in Range	639
12.7 Global	640
12.7.1 Alpha	640
12.7.2 Audio	641
12.7.3 Clipper	644
12.7.4 Expert	646
12.7.5 Extrude	648
12.7.6 Glow	650
12.7.7 HDR	651
12.7.8 Key	651
12.7.9 Look-At	653
12.7.10 Mask Source and Mask Target	654
12.7.11 Lighting	656
12.7.12 Z-Sort	658
12.7.13 Projector Source and Projector Target	661
12.7.14 Script	662
12.7.15 Shadow Caster and Shadow Receiver	664
12.7.16 Synchronized Properties	667
12.7.17 Video Clip	668
12.7.18 Window Mask	669
12.8 Multi-Touch Plugins	670
12.8.1 Plugin Event and Notification System	671
12.8.2 MtTelestrator Plugin	671
12.8.3 MtButton Plugin	674
12.8.4 Mt2D Control Plugin	678
12.8.5 Mt3D Control Plugin	682
12.8.6 MtNavigator Plugin	686
12.9 PixelFX	689
12.9.1 pxLensMulti	689
12.10 Presenter	690

12.10.1 Bar	690
12.10.2 Bar Value	691
12.10.3 Bar Values	693
12.10.4 Pie Slice	694
12.10.5 Pie Values	695
12.11 pxColorWorks	697
12.11.1 PixelFX plugins	697
12.11.2 pxAddSubtract	698
12.11.3 pxBlackAndWhite	698
12.11.4 pxBrightContrast	699
12.11.5 pxColorMatch	699
12.11.6 pxGamma	699
12.11.7 pxHueRotate	699
12.11.8 pxMask	700
12.11.9 pxSaturation	700
12.11.10 pxStack	700
12.11.11 pxTint	701
12.12 RealFX	702
12.12.1 RFXColliderSrc	703
12.12.2 RFXColliderTgt	704
12.12.3 RFXLatLong	704
12.12.4 RFXMagnet	705
12.12.5 RFXTurb	705
12.12.6 RFXVortex	706
12.13 Script Plugins	706
12.14 SplineFX	706
12.14.1 2D Follow	707
12.15 TextFX	708
12.15.1 Common Text FX Properties	708
12.15.2 Convert Case	709
12.15.3 Mark Text	710
12.15.4 Text FX Alpha	711
12.15.5 Text FX Arrange	712
12.15.6 Text FX Color	713
12.15.7 Text FX Color Per Vertex	714
12.15.8 Text FX Emoticons	715
12.15.9 Text FX Explode	716

12.15.10 Text FX Jitter Alpha	718
12.15.11 Text FX Jitter Position	718
12.15.12 Text FX Jitter Scale	719
12.15.13 Text FX Plus Plus	720
12.15.14 Text FX Rotate	722
12.15.15 Text FX Scale	723
12.15.16 Text FX Size	723
12.15.17 Text FX Slide	724
12.15.18 Text FX Vertex Explode	725
12.15.19 Text FX Write	726
12.16 Texture	727
12.16.1 BrowserCEF	728
12.16.2 Grabbit	734
12.16.3 GraffitiTex	741
12.16.4 Image Clip	743
12.16.5 Light Blur	746
12.16.6 MoViz	747
12.16.7 Noise	750
12.16.8 SoftClip	751
12.16.9 Tex Component	754
12.17 Ticker	755
12.17.1 Scroller Action	755
12.18 Time	757
12.18.1 Analog Watch	757
12.19 Tools	759
12.19.1 Advanced Counter	759
12.19.2 Autofollow	760
12.19.3 Autorotate	762
12.19.4 Bounding Actions	763
12.19.5 Clock Rotation	764
12.19.6 Cloner	765
12.19.7 Colorize	766
12.19.8 Counter	768
12.19.9 DVE Follow	769
12.19.10 Heartbeat	769
12.19.11 Image Link	771
12.19.12 Jack	772

12.19.13 Level Of Detail (LOD)	773
12.19.14 Magnify	775
12.19.15 Match It	776
12.19.16 Max Size	777
12.19.17 Max Size Lines	778
12.19.18 Object Zoom	779
12.19.19 Omo	780
12.19.20 Pablo	781
12.19.21 Parliament	782
12.19.22 Rotations Order	783
12.19.23 Slide Show	783
12.19.24 System Time	784
12.19.25 Temo	785
12.19.26 TextBG	786
12.19.27 Text Link	788
12.19.28 Text Parameters	789
12.19.29 VCF Parameter	790
12.20 Transformation	792
12.20.1 Justifier	792
12.21 Visual Data Tools	793
12.21.1 Data Fit	793
12.21.2 Data Import	795
12.21.3 Data Label	797
12.21.4 Data Storage	800
13 Shader Plugins	803
13.1 Default	803
13.2 Effects	803
13.2.1 Chroma Keyer	804
13.2.2 Fluid	806
13.2.3 Frame Mask	807
13.2.4 Image Mask	808
13.2.5 Soft Mask	809
13.2.6 Water Shader	810
13.3 Filter	811
13.3.1 Blend Image	812
13.3.2 Blur	813
13.3.3 Color Balance	814

13.3.4 Radial Blur	815
13.3.5 Sepia	815
13.3.6 Sharpen	816
13.4 Material	817
13.4.1 Anisotropic Light	817
13.4.2 Bump Map	818
13.4.3 Cartoon	819
13.4.4 Gooch	820
13.4.5 Normal Map	821
13.4.6 Simple Bump Map	821
13.5 PixelFX	823
13.5.1 Common Properties	823
13.5.2 pxBcubic	824
13.5.3 pxCCBase	824
13.5.4 pxEqualize	824
13.5.5 pxGradient	824
13.5.6 pxInvert	825
13.5.7 pxLensDistort	826
13.5.8 pxMotionBlur	826
13.5.9 pxNoise	827
13.5.10 pxPixelate	827
13.5.11 pxPosterize	828
13.5.12 pxRecolor	828
13.5.13 pxRipple	828
13.5.14 pxSparkle	828
13.5.15 pxTurbDissolve and pxTurbWipe	829
13.5.16 pxTurbulence	830
13.5.17 pxTwirl	830
13.5.18 pxWaves	830
13.6 PixelFXLensFlare	831
13.7 RealFX	831
13.7.1 RFx2DSphere	833
13.7.2 RFxClouds	833
13.7.3 RFxFlame	833
13.8 RTT Advanced Materials	834
13.8.1 Anisotropic Light Shader	837
13.8.2 Brushed Metal Shader	838

13.8.3 Bump Shader	840
13.8.4 Bump Optimized Shader	842
13.8.5 Fabric Shader	844
13.8.6 Glass Shader	845
13.8.7 Gooch Shader	847
13.8.8 Lacquered Surfaces Shader	849
13.8.9 Metal Reflection Shader	850
13.8.10 Microstructure Shader	851
13.8.11 Monitor Shader	852
13.8.12 Velvet Shader	854
13.9 Texture	855
13.9.1 Drop Shadow	855
13.9.2 Emboss	856
13.9.3 MultiTexture	857
14 Scene Plugins	859
14.1 Default	859
14.1.1 VCF	859
14.2 Image	861
14.2.1 Background Clip	861
14.3 Libero	863
14.3.1 EVSControl Plugin	863
14.3.2 Properties	863
14.3.3 Example Scenes	865
14.4 Lineup	865
14.4.1 Tree Status	865
14.5 Multi-Touch Plugins	866
14.5.1 MtSensor Plugin	866
14.6 Script Plugins	868
14.7 Texture	868
14.7.1 Graffiti	868
14.8 Tools	870
14.8.1 Level Of Detail (LOD) Manager	871
15 Media Assets	873
15.1 Media Asset Workflow	874
15.1.1 Media Assets as DVE	874
15.1.2 Media Assets as Texture	877
15.1.3 Media Assets as a Background or Foreground	879

15.2 Channel Folder Media Assets	880
15.3 Container Folder Media Assets	883
15.3.1 Dynamic Scene Media Asset	883
15.3.2 Dynamic Texture Media Asset	885
15.3.3 Viewport Tile Media Asset	885
16 On Air Mode	887
16.1 Control Buttons	888
16.2 Director Control Panel	889
16.3 Viz Artist Performance	889
16.3.1 Performance Analyzer	890
16.3.2 Performance Editor	891
16.4 On Air Information	892
16.4.1 Basic Tab	893
16.4.2 Clients Tab	894
16.5 License Information	894
17 Transition Logic	895
17.1 Standalone versus Transition Logic Scene Design	896
17.2 Toggle Layer	897
17.3 State Transition Animation	897
17.3.1 Defining States	898
17.3.2 Identifying States	899
17.4 Cross Animation	901
17.5 Geometry Animation	901
17.6 Background Scene	902
17.7 Foreground Scene	902
17.8 Tutorial	903
17.8.1 Create a Project	904
17.8.2 Create a Background Scene	904
17.8.3 Create a Foreground Scene	907
18 Scripting	911
18.1 How Programming Works	911
18.2 Program Structure	912
18.3 Variables	912
18.3.1 Basic Data Types	913
18.4 Procedures	915
18.4.1 Sub-routines	915
18.4.2 Functions	916

18.5 Operators	916
18.6 Conditions	918
18.7 Loops	919
18.7.1 Condition-controlled Loops	919
18.7.2 Count-controlled Loops	919
18.8 Script Editor	920
18.9 Create and Run Scripts	921
18.10 Create Script-based Plugins	923
18.11 Control 3D Stereoscopic Clip Playback	925
18.12 Program Examples	926
18.12.1 Initializing Scripts	926
18.12.2 Calculating Values	927
18.12.3 Registering Parameters	930
18.12.4 Registering Parameters and Moving Objects	931
18.12.5 Mouse Events I (Set)	932
18.12.6 Mouse Events II (Start)	932
18.12.7 Mouse Events III (Start, Stop and Reverse)	933
18.12.8 Mouse Events IV (Grow and Shrink)	933
18.12.9 Mouse Events V (Advanced)	934
18.12.10 Planets	938
18.12.11 Tetris	940
18.12.12 Clip Channel	945
18.12.13 Clip Channel: Stereo	959
18.12.14 General Script Examples	961
19 Scripting Data Types and Procedures	967
19.1 Callback Procedures	967
19.2 Data Sharing	970
19.2.1 Script Examples	971
19.2.2 Receiving Notification on Map Changes	973
19.2.3 VizCommunication.Map	974
19.2.4 Map Access Through a TCP or UDP Interface	974
19.2.5 Map Access through the Command Interface	974
19.2.6 Map Access from Within Plugins	975
19.3 Data Types and Procedures	975
19.3.1 Alpha	976
19.3.2 Array[Type]	977
19.3.3 Base	978

19.3.4 BezierHandle	980
19.3.5 Boolean	981
19.3.6 Camera	981
19.3.7 Center	983
19.3.8 Channel	984
19.3.9 ClipChannel	987
19.3.10 Color	990
19.3.11 Container	990
19.3.12 DateTime	998
19.3.13 Director	999
19.3.14 Double	1002
19.3.15 Expert	1002
19.3.16 Geometry	1005
19.3.17 Grid	1008
19.3.18 Image	1009
19.3.19 InfoText	1011
19.3.20 Integer	1012
19.3.21 Key	1012
19.3.22 Key Frame	1014
19.3.23 Light	1017
19.3.24 Material	1019
19.3.25 Matrix	1020
19.3.26 PluginInstance	1021
19.3.27 Position	1024
19.3.28 Rotation	1026
19.3.29 Scaling	1028
19.3.30 Scene	1029
19.3.31 Script	1032
19.3.32 ScriptSettings	1032
19.3.33 Shared Memory	1033
19.3.34 Stage	1035
19.3.35 String	1037
19.3.36 StringMap	1039
19.3.37 System	1040
19.3.38 Texture	1046
19.3.39 Timecode	1048
19.3.40 Uuid	1050

19.3.41 Variant	1050
19.3.42 Vertex	1051
19.3.43 VizCommunication	1052
19.3.44 Global Procedures	1054
20 Shared Memory (SHM)	1061
20.1 External Data Input	1061
20.1.1 TCP Protocol	1062
20.1.2 UDP Protocol	1063
20.1.3 Plugin API	1064
20.1.4 Command Interface	1065
20.2 Internal Data (Interactive Scene)	1065
20.3 Synchronization	1066
20.3.1 TCP and UDP Synchronization	1066
20.3.2 External Control Synchronization	1067
20.3.3 Command Synchronization	1068
20.4 Snapshot	1068
21 Integration with 3rd Party Applications and Files	1071
21.1 Working with Adobe After Effects Files	1071
21.1.1 Import of Adobe After Effect Files	1071
21.1.2 Import Limitations	1072
21.2 Working with CINEMA 4D Files	1073
21.2.1 Using the CINEMA 4D LiveLink	1073
21.2.2 Current Limitations	1074
21.3 Working with FBX Files	1076
21.3.1 Scene Graph Traversal	1076
21.3.2 Transformations	1077
21.3.3 Export FBX Files from 3ds Max and Maya	1079
21.4 Working with TriCaster	1082
22 Keyboard and Mouse Shortcuts	1085
22.1 Application Controls and Shortcuts	1085
22.2 Integer/Float Controls	1086
22.3 Server Area Shortcuts	1086
22.4 Scene Tree Shortcuts	1089
22.5 Scene Editor Shortcuts	1091
22.6 Stage Shortcuts	1094
22.7 Import Shortcuts	1097
22.8 On Air Shortcuts	1098

22.9 Polygon Plugin Editor Shortcuts	1098
22.10 Script Editor Shortcuts	1100

1 Introduction

This User Guide gives the information required to understand the Viz Artist User interface, and how to do certain tasks.

The term Viz is used for the programs installed and run on the computer. This is used as a general reference for all modes of the program:

- Viz Artist
- Viz Engine (see the [Viz Engine Administrator Guide](#))
- Viz Configuration (see [Viz Configuration](#))

What mode of program that can run is determined from the hardware dongle that is attached to the actual machine (the different modes and the hardware dongle are detailed in this User Guide).

This section contains information on the following topics:

- [Document Structure](#)
- [Related Documents](#)
- [Customer Feedback and Suggestions](#)
- [Customer Support Request](#)

1.1 Document Structure

This section describes the structure and content of this User Guide.

Title	Content
Introduction	This section gives details on related documentation and how to contact Vizrt for Customer support. Details are also given for Customers to provide feedback on their Vizrt product.
Getting Started	This section describes the location of the installation files, how to log in to a Graphic hub and start Viz Artist, and how to close Viz Artist.
Workbench Description	This section introduces Viz Artist and its basic concepts. It 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 show a Scene from different camera angles.
Scene Tree Knowledge and Skills	This section describes the Viz Scene Tree, and shows the knowledge and Skills needed to be familiar with the Scene Tree and how it operates.
Manage Work and Graphic Hub	This section describes briefly the Graphic Hub, and how to manage Elements and the workflow of Elements.
Global Tasks	This section describes how to do Global tasks, such as how to work with the Scene Settings , Video Playback ,

Title	Content
	Import Files and Archives , Archiving and Deploy Elements of graphics to video clips and/or still images.
Lights	This section describes how to work with Lights .
Cameras	This section describes how to work with Cameras in scenes used in 2D and 3D (stereoscopy) broadcast production.
Animation Stage Create Animations	These sections describe how to work with the Animation Stage and Create Animations .
Geometry Plugins Container Plugins Shader Plugins Scene Plugins Media Assets	These sections describe all the built-ins. Most of the plugin sections also have small procedures with illustrations that help to get started with a Scene design.
On Air Mode	This section describes the on air mode, that is used to test and analyze the performance of Viz scenes.
Transition Logic	The section describes the core aspects of transition logic Scene design with a small tutorial at the end, to give guidance with transition logic designs.
Scripting Scripting Data Types and Procedures	These sections describe the Viz scripting language, how to use it and the available data types and procedures.
Shared Memory (SHM)	This section describes how to work with Shared Memory. Some use cases are described to give a better understanding of Shared Memory and its functionality.
Integration with 3rd Party Applications and Files	This section describes the 3rd party applications and files, which can be integrated into and used with Viz Artist.
Keyboard and Mouse Shortcuts	This section give details on the keyboard and mouse shortcuts for each part of the Viz Artist GUI.
Glossary	The section gives an extensive description of keywords and acronyms referenced throughout the document, with simple and to the point explanations.

See Also

- [Related Documents](#)

1.2 Related Documents

- [Viz Engine Administrator Guide](#) : Contains information on how to install the Viz Engine software, supported hardware and how to configure Viz Artist/Engine.

- [Viz Configuration](#): See the [Viz Engine Administrator Guide](#) for detailed information on how to configure Viz Artist/Engine, with the Viz Artist GUI and the system Config file
- **Viz Engine Plugin SDK reference**: Contains information on how to create a customized Viz plugin.
- **Viz Artist Script reference**: Contains information on how to create scripts for a scene.

See Also

- [Document Structure](#)

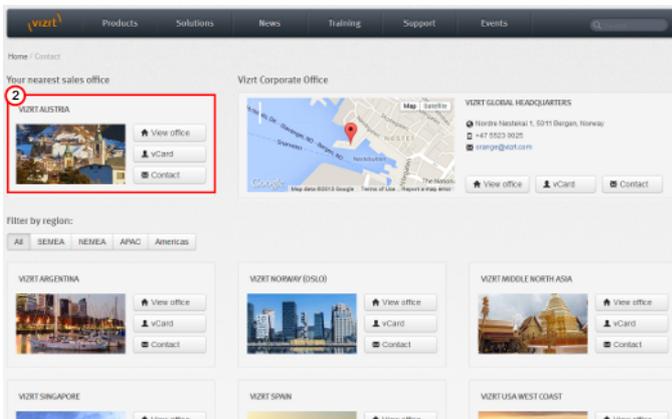
1.3 Customer Feedback and Suggestions

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

1. Go to www.vizrt.com.
2. Click on **Contact** (1).



3. The Vizrt office which is nearest to your location will be shown (2), or select from the list of Vizrt offices.
4. Click on 'Contact'.



5. Complete the required details in the window that opens.

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

6. Click on **CONTACT ME**.

A Vizrt representative will contact you as soon as possible.

See Also

- [Customer Support Request](#)

1.4 Customer Support Request

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

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

This section contains information on the following topics:

- [Submitting a Support Request](#)
- [Attach Log Files to a Customer Request](#)

1.4.1 Submitting a Support Request

Before a Support Request is submitted make sure that you:

Read:

- The relevant User Guide or Guides
- The release notes

and Check:

- That the system is configured correctly
- That you have the specified hardware, tested and recommended versions

Always refer to your Vizrt Service Level Agreement document.

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

This section contains information on the following topics and procedures:

- [Content of a Support Request](#)
- [To Submit a Support Request](#)
- [Attach Log Files to a Customer Request](#)

Content of a Support Request

The report should contain information about these topics:

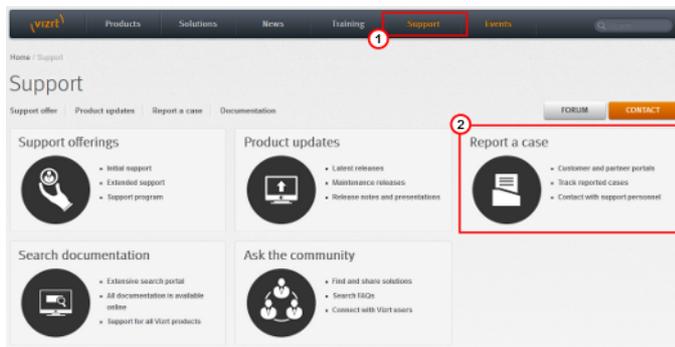
- **Problem Description:** Include a good description of what the problem is and how to reproduce it. Remember to use simple English.
- **Screen Shots and Illustrations:** Use to simplify the message.
- **System Log Files:** Send the system log files (see [Attach Log Files to a Customer Request](#)).
- **Dump Files:** Send any dump files generated from a crash (<viz data folder>).
- **System Viz Config File:** Send the system config file (<viz data folder>).
- **Software Configuration:** Add exact versions of software (-build) used.

- **Hardware Configuration:** Add exact versions of hardware used.
- **System Setup:** Describe differences in the installation, if any, from the recommended setup.
- **System Network:** Add a description of how the network, bandwidth, routers, and switches are configured.

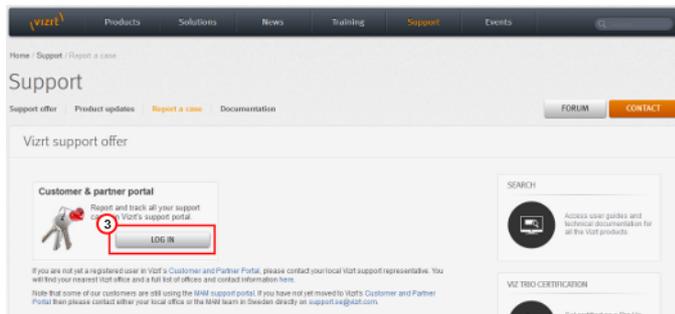
Always refer to your Vizrt Service Level Agreement document.

To Submit a Support Request

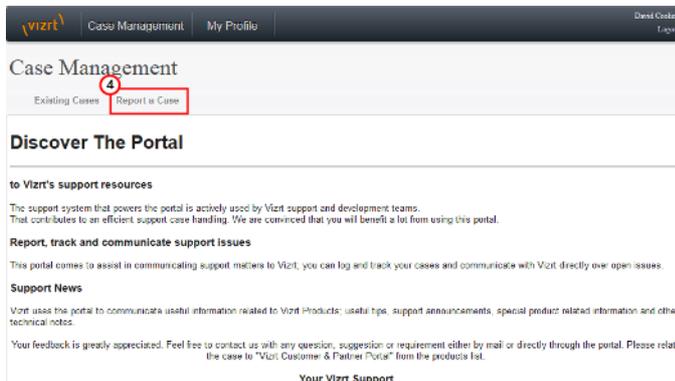
1. Go to www.vizrt.com.
2. Click on Support (1).
3. Click on Report a case (2).



4. Click on **LOG IN** (3).
5. Log-in to the Customer and Partner portal.



6. Click on **Report a Case** (4).



7. In the online form complete the required minimum information (shown by a red asterisk):
 - **Contact:** Your name
 - **Account:** Your account
 - **Product:** The product the support request refers to.
 - **Case Type:** The type of support request required
8. Click **SAVE**.
9. In the saved Support Case that opens, complete the various text boxes and upload any required documents, files, etc. (see [Content of a Support Request](#)).

Note: The entered text or uploaded documents / files are automatically added.

To track the status of open support tickets, log in to the Customer and Partner portal. Add information or communicate about the cases directly with the support team.

1.4.2 Attach Log Files to a Customer Request

Log files provide important information when dealing with an error situation or when diagnosing performance issues. Viz Artist/Engine can write various log files that can be valuable when solving issues. For more information about the log files available see the **Viz Artist/Engine Log Files** in the [Viz Configuration](#) (see the [Viz Engine Administration Guide](#)).

Consult your customer representative when deciding which logs files to attach. As a general rule: the more log files the better. Always try to include log files for:

- The actual software/hardware configuration in use, as detailed as possible.
- The relevant Viz Artist/Engine log files.

2 Getting Started

This section describes the location of the installation files, how to log in to a Graphic hub and start Viz Artist, and how to close Viz Artist.

This section contains information on the following topics:

- [Viz Artist/Engine Folders](#)
- [Viz Artist Startup and Close](#)
- [Viz Command Line Options](#)

2.1 Viz Artist/Engine Folders

This section details the location of the Viz Artist/Engine default installation and data folders.

This section contains details on the following topics:

- [Installation Folders](#)
- [Data Folders](#)

See Also

- [User Account Control \(UAC\)](#) in the [Viz Engine Administrator Guide](#)

Installation Folders

The default installations folders are:

Windows System	Viz Artist/Engine Platform	Installation Folder
Windows 7 32-bit	32 bit	C:\Program Files\vizrt\Viz3\
	64 bit	N/A
Windows 7 64-bit	32 bit	C:\Program Files (x86)\vizrt\Viz3\
	64 bit	C:\Program Files\vizrt\Viz3\ C:\Program Files (x86)\vizrt\Viz3Gui

In this User Guide any reference to the Viz Artist/Engine installation folder, e.g. C:\Program Files\vizrt\Viz3\, is replaced with the text <viz install folder>.

IMPORTANT! In a Viz Artist/Engine 64-bit installation, the GUI is installed in a 32-bit folder (because the GUI is a 32-bit application).

Note: Viz Config files that already exist from a previous installation are automatically copied on install time if the installation folder did not change. If Viz Engine/Artist is installed in a different installation folder then Viz Config files must be manually copied from the old installation folder to <viz data folder>.

Data Folders

Files which are created or modified by Viz Artist/Engine are located at `%ProgramData%\vizrt\viz3`, which usually resolves to `C:\ProgramData\vizrt\viz3`. This folder is referenced as `<viz data folder>` throughout this User Guide, and contains, for example:

- Viz Config files
- Script plugins
- Crash dump files

Temporary files are located at: `%TMP%\vizrt\viz3`, which usually resolves to `C:\Users\<user name>\AppData\Local\Temp\vizrt\viz3`. This folder is referenced as `<viz temp folder>` throughout this User Guide.

Script Plugins are located in the `<viz data folder>\ScriptPlugins` subfolder.

Note: Script Plugin files that already exist from a previous installation are automatically copied on install time if the installation folder did not change. If Viz Engine/Artist is installed in a different installation folder then Script Plugin files must be manually copied from the old installation folder to `<viz data folder>\ScriptPlugins`

2.2 Viz Artist Startup and Close

Viz Artist is started from various desktop options. To start and run Viz Artist a Graphic Hub must also be available.

Viz Artist can also be started with various options included (see [Viz Command Line Options](#)). To start Viz Artist with an option or options, the command for that option must be added before Viz Artist startup.

This section contains information on the following topics procedures:

- [To Start Viz Artist](#)
- [To Add a Viz Artist Startup Option](#)
- [To Log in a New User after Viz Artist has Started](#)
- [Log in History](#)
- [To View the Log in History](#)
- [To Close \(Quit\) or Restart Viz Artist](#)
- [Viz Artist 64-Bit or 32-Bit Version](#)
- [Viz Configuration](#)

See Also

- [Database](#) (in [Viz Configuration](#) ([Viz Engine Administrator Guide](#)))
- [Graphic Hub User Guide](#)

To Start Viz Artist

1. Start Viz Artist:
 - <double-click> the desktop icon  , or

- <right-click> the desktop icon and select **Open** from the context menu, or
 - Use the Windows **Start** menu (example: *Start menu -> All Programs -> Vizrt -> Viz 3.x 64bit-> Viz Artist 3.x 64bit*)
2. In the Graphic Hub login window select the required details:
-  : Select a Host from the drop-down list

Note: In some circumstances (e.g., server located in different sub-net) the host will not show in the list. The host name must then be typed in.

-  : Select a Graphic Hub from the drop-down list
-  : Select a User from the drop-down list
-  : Type the correct password for the selected User



- **Log me in automatically:** If auto log in is available, click to activate auto log in

Note: When auto log in is active Viz Artist will automatically log in into a Graphic Hub and startup.

3. Click on:
- **Log in:** Log in to the selected Graphic Hub and startup Viz Artist
 - **Cancel:** Exit login.

After a correct log in to a Graphic Hub, Viz Artist starts and opens the main screen. See [Workbench Description](#) for an overview of the Viz Artist main screen.

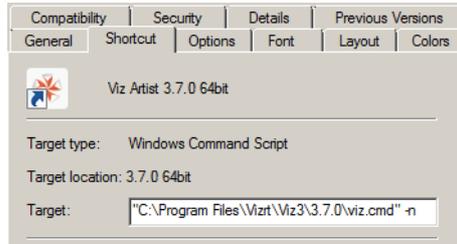
The log in credentials are saved in the [Database](#) section of the [Viz Configuration](#) ([Viz Engine Administrator Guide](#)) where it can be modified as required.

To Add a Viz Artist Startup Option

1. Open the Viz Artist Properties panel:
- <Right-click> the desktop icon and select **Properties** in the context menu  , or

- Use the Windows **Start** menu (example: *Start menu -> All Programs -> Vizrt -> Viz 3.x 64bit-> Viz Artist 3.x 64bit*), <right-click> the Viz Artist link and select **Properties** in the context menu
2. In the Properties window, apply a command in the **Target** field (see [Viz Command Line Options](#)).

Example: "<viz install folder> viz.cmd" -n

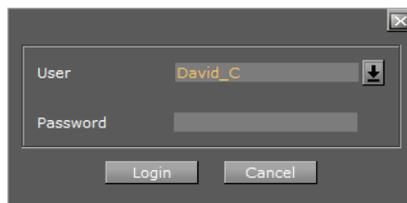


3. Click **OK**.
4. Start Viz Artist (see [To Start Viz Artist](#)).

To Log in a New User after Viz Artist has Started



1. Click on **Server**.
2. Click on the Server area drop-down menu (1).
3. Click on **Change User...** (2).



4. In the Change User window select a User from the drop down menu.
5. Enter a valid password for the selected User.
6. Click on:

- **Login** to change User, or
- **Cancel** to abort the procedure.

Log in History

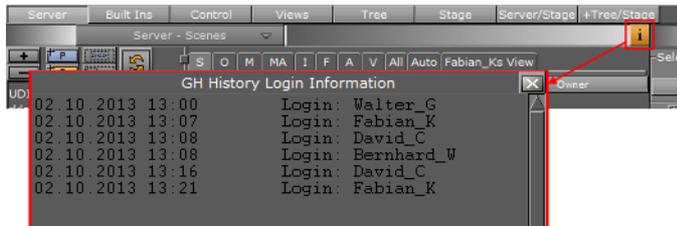
Viz Artist creates a Graphic Hub log in history (view only) for each User who logs in to the current Viz Artist session.

When Viz Artist starts up the **startup** User is logged. If a new User logs in, this is recorded and the Graphic Hub History icon will change from grey to orange. This shows that another User, other than the **startup** User, has logged in to the current Viz Artist session.

When the **startup** User logs in again the Graphic Hub log in history icon will show grey again.

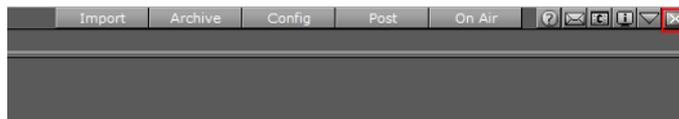
To View the Log in History

- Click on the Graphic Hub History icon, at any time, to open the log in history window.



To Close (Quit) or Restart Viz Artist

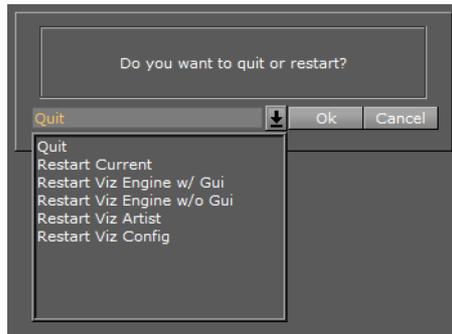
1. Start the Quit or Restart, either:
 - Click on the **Quit** button (top right [Button Bar](#))



or

- Press double <Esc>, or
 - Press <Alt+Q>, or
 - Press <Alt+F4>
2. In the **Do you want to quit or restart?** panel:
 - Click **OK** to Quit (Quit (close) is the default option), or
 - Click on the drop down arrow to open the Quit or Restart menu
 3. In the Quit or Restart menu, select from the available options.

- Click **OK** or **Cancel**.

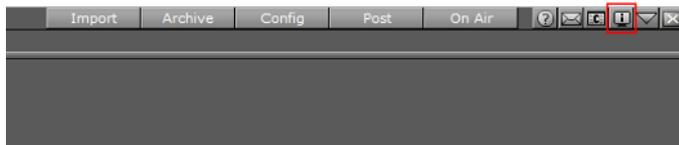


See Also

- [Viz Configuration](#)
- [Viz Command Line Options](#)
- Installing the Viz Engine ([Viz Engine Administrator Guide](#))
- [Viz Configuration](#) ([Viz Engine Administrator Guide](#))

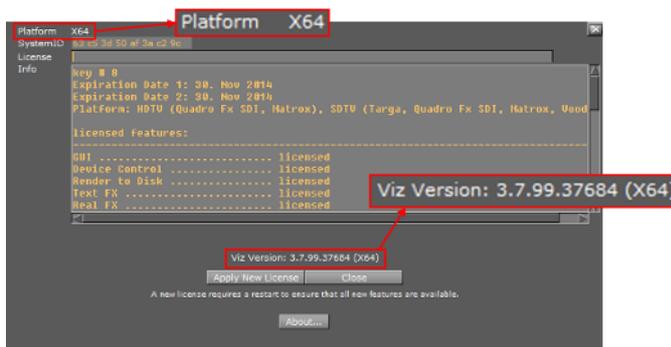
2.2.1 Viz Artist 64-Bit or 32-Bit Version

To check if Viz Artist is a 64-Bit or 32-Bit installation, click on the **Show Licence Information** button:



In the Licence Information window the Platform and Viz Version number show as:

- Platform Win32 or X64
- Viz Version 3.7.XX.XXX (Win32) or (X64)



2.2.2 Viz Configuration

Viz is mainly configured by the [Viz Configuration](#) panel (see the [Viz Engine Administrator Guide](#)).

All configuration settings are stored in a Viz Config file, found in the *<viz data folder>*. The Viz Config file uses the machine hostname to uniquely identify which machine Viz Artist/Engine is installed on, for example, *Viz-<hostname>-0-0.cfg*.

Any changes to the hostname will affect the Viz Artist/Engine configuration. If a hostname is changed, a new Viz Config file is created with a default setup. The old Viz Config file is not deleted, but left unused. If required, use the command **-g <config file>** to reassign the old Viz Config file.

IMPORTANT! Folder, File and Path Naming Convention: Do not use the Hash tag (#) in any folder, file or path name. If used in Folder, File and Path Names, text after a Hash tag will be ignored.

See Also

- [Viz Artist Startup and Close](#)
- [Viz Command Line Options](#)
- [Viz Configuration](#) (see the [Viz Engine Administrator Guide](#))

2.3 Viz Command Line Options

Viz Artist and Viz Engine can be started with various options included (see [To Add a Viz Artist Startup Option](#)). The table below details the available option commands.

Command	Description
-3D	To playback 3D Stereo Clips. This sets Viz Artist 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: This is for clips only
-B <path>	Specify the path where Viz Engine stores its temporary data (see Viz Artist/Engine Folders)
-c	Start in Viz Configuration mode (see also -u1, -u2, -u3)
-C	Start without a console
-g <config file>	Start with a predefined Viz Config file. This allows a User to have more than one startup option
-i	Enable pre-initialization of textures. Textures are generated on the graphics card immediately after loading an image
-l	Specify a console title to distinguish Viz Engines in a Dual Channel setup (example: -l <title>)
-n	Start in Viz Engine mode (see also -u1, -u2, -u3)
-o <scene>	Start in Viz Engine mode with an optional auto load scene in the Main Layer (example: -o SCENE*<path>)
-o <layer> <scene>	Start in Viz Engine mode with an optional auto load Scene and loads it into a specific layer: 0 ... Back Layer

Command	Description
	1 ... Main Layer 2 ... Front Layer (example: <code>-o 2 SCENE*<path></code>)
<code>-T</code>	Keep the Viz Engine Console always on top
<code>-u1, -u2, -u3</code> (up to a maximum of 24)	For systems with two or more graphic cards (i.e. Trio One Channel / Dual Channel): Specify GPU1 (-u1), GPU2 (-u2), etc., to run the program on (see <code>-c</code> and <code>-n</code>), for example: <code>"<viz install folder> viz.cmd" -c -u2</code> <code>"<viz install folder> viz.cmd" -n -u1</code> When either Trio One Channel or Dual Channel Viz Artist versions are installed, GPU's are selected by default. Note: Viz Artist will run on one GPU only. Only the control GPU should have a display connected and will be used for Artist/Preview.
<code>-v</code>	Enable verbose mode. This enables all possible information to be shown in the Viz Engine Console
<code>-V</code>	Show the last rendered frame on video output (only Digital Voodoo cards)
<code>-Y <path></code>	Specify the path where Viz Engine stores its program data (see Viz Artist/Engine Folders)

See Also

- [To Add a Viz Artist Startup Option](#)
- [Viz Artist Startup and Close](#)
- [Viz Configuration](#)

3 Workbench Description

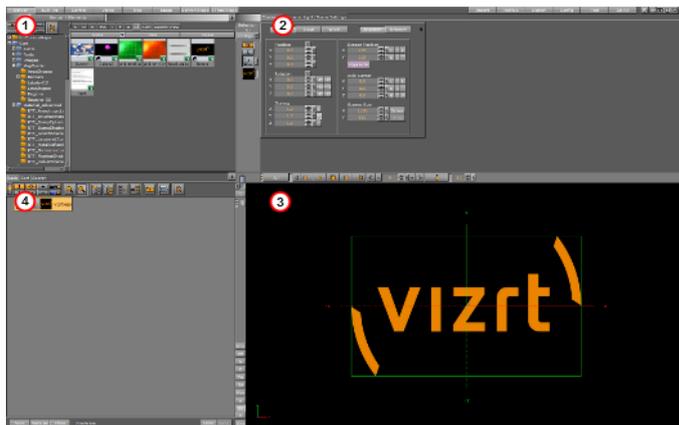
This Section gives a Workbench Description (overview) of Viz Artist and its main panels.

This section contains information on the following topics:

- [Main Window](#)
- [Main Menu](#)
- [Scene Editor](#)
- [Working with the Scene Editor](#)
- [Scene Editor Views](#)

3.1 Main Window

After a successful start up and log in (see [Viz Artist Startup and Close](#)), the Main window shows:



At the top of the Main window is the [Main Menu](#) bar. Below the Main menu bar is the display area. The content of the display area changes according to selections made in the Main menu.

The Viz Artist GUI can also be configured to work on two monitors (see [Dual Monitor Configuration](#))

At system startup the area is divided into four panels:

Fig Item	Panel	Description
1	Server Area	From the Server area it is possible to control Elements, such as Materials , Images , and Fonts which are located in the Graphic Hub.
2	Properties	Properties for Containers (Container contents), Cameras , Lights and Scene Settings are controlled from the Property panel, where size, position and other settings are modified, if required.

Fig Item	Panel	Description
3	Scene Editor	A visualization of the Scene is shown in the Scene Editor panel.
4	Scene Tree	Elements, in the Graphic Hub database, can be added to the Scene Tree , which is the logical visualization of a Scene. The Scene Tree is built up of Containers, which can hold items such as Materials or Plugins.

Note: The panel areas can be resized.

3.1.1 Dual Monitor Configuration

Viz Artist can be configured to work on two monitors (see **User Interface** in [Viz Configuration](#) in the [Viz Engine Administrator Guide](#)). The Viz Artist GUI screen options are:

- [Mosaic Horizontal or Mosaic Vertical](#)
- [Dual Screen](#)

Mosaic Horizontal or Mosaic Vertical

Mosaic Horizontal (1): The Server, Properties and Scene Tree panels show on the left monitor and the Scene Editor shows on the right monitor.

Mosaic Vertical (2): The Server, Properties and Scene Tree panels show on the top monitor and the Scene Editor panel shows on the bottom monitor.

In both Horizontal and Vertical view the monitor which shows the Scene Editor panel can be set to show either, (O) Reset size to screen size or (R) Set size to output aspect (3).



Dual Screen

The Scene Editor panel opens in a new window, and the elements cannot be dragged to the Scene Editor panel.



3.2 Main Menu

The buttons in the Main menu make it possible to change the view of the display area.



This section contains information on the following topics:

- [Main Menu Left](#) (1)
- [Main Menu Right](#) (2)

3.2.1 Main Menu Left



This section contains information on the following topics:

- [Server Button](#)
- [Built Ins Button](#)
- [Control Button](#)
- [Views Button](#)
- [Tree Button](#)
- [Stage Button](#)
- [Server/Stage Button](#)
- [+Tree/Stage Button](#)

Server Button

From the Server area it is possible to control the database elements, 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 shows at system startup.

Shows the following panels:

- Server (top left)
- Scene Tree (bottom left)
- Property (top right)
- Scene Editor (bottom right).

See Also

- [Manage Work and Graphic Hub](#)
- [Server Menu](#)
- [Server Tree Menu](#)

Built Ins Button

From the Plugins area it is possible to control the built in plugins such as Geometries, Container, Shader, Texture and Scene plugins. The features vary according to which plugin packages that are licensed and installed on each machine.

Shows the following panels:

- Plugins (top left)
- Scene Tree (bottom left)
- Property (top right)
- Scene Editor (bottom right).

See Also

- [Built Ins](#)
- [Scene Editor](#)
- [Image Editor](#)
- [Plugin](#)

Control Button

In the Control area, parameters, which are exposed in a Scene (for modification in an external control), can be viewed, modified or created. The Control Objects panel shows the exposed parameters when a Control plugin is used, while in the Control Channels panel, parameters from the Properties panel can be dragged and saved as exposed parameters.

Shows the following panels:

- Controls (top left)
- Scene Tree (bottom left)
- Property (top right)
- Scene Editor (bottom right)

Views Button

The Views area shows three configurable camera views, in addition to the one in the Scene Editor area. The additional camera views make it possible to monitor the scene from up to four various angles simultaneously.

Shows the following panels:

- Views (left)
- Property (top right)
- Scene Editor (bottom right)

See Also

- [Scene Editor Views](#)

Tree Button

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.

Shows the following panels:

- Scene Tree (left)
- Property (top right)
- Scene Editor (bottom right)

See Also

- [Scene Tree Knowledge and Skills](#)

Stage Button

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).

Shows the following panels:

- Stage (top)
- Scene Tree (bottom left)
- Scene Editor (bottom right)

See Also

- [Animation Stage](#)
- [Create Animations](#)

Server/Stage Button

The Server/Stage view is similar to the regular Stage view, but here the scene tree has been replaced with the database content.

Shows the following panels:

- Stage (top)
- Server (bottom left)
- Scene Editor (bottom right)

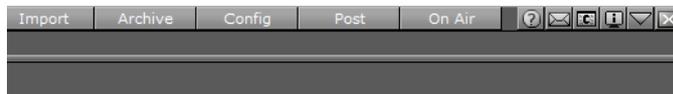
+Tree/Stage Button

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.

Shows the following panels:

- Stage (top)
- Server (bottom left)
- Scene Tree (bottom center)
- Scene Editor (bottom right)

3.2.2 Main Menu Right



This section contains information on the following topics:

- [Import Button](#)
- [Archive Button](#)
- [Deploy Button](#)
- [Config Button](#)
- [Post Button](#)
- [On Air Button](#)
- [Button Bar](#)

Import Button

Click to open the Import panels. Use these panels to transfer archives and various files into the database. For more information see the [Import Files and Archives](#) section.

Archive Button

Click to open the Archive panels. Use these panels to create an archive of multiple database elements into a single file, which is then located on the hard drive (see [Archiving](#)).

Deploy Button

Click to open the Deploy panels (available when Deploy is enabled in **Database** (see [Viz Configuration](#) in the [Viz Engine Administrator Guide](#)). Use these panels to copy Elements, with all their necessary data, from one server to another (see [Deploy Elements](#)).

Config Button

Click to open the [Viz Configuration](#) panel. Use this panel to change system options and functionality (see [Viz Configuration](#) in the [Viz Engine Administrator Guide](#)).

The content of the Viz Configuration panel changes with different platform types. An installation on a machine that contains a video board will run the software in Video mode, a machine with no video board will run the software in Standard PC mode. To find the platform type select the License Information button (see [Main Menu Right](#)).

Viz Configuration can also be opened, as a separate application, from *Start -> All Programs -> Vizrt -> Viz 3 -> Config*.

Note: After any modifications in the Viz Configuration settings, the system must be restarted for the changes to take effect.

Post Button

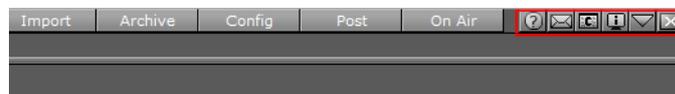
Opens the Post Rendering panel. From this panel it is possible to export image chains or Scene clips (see [Post Rendering](#)).

On Air Button

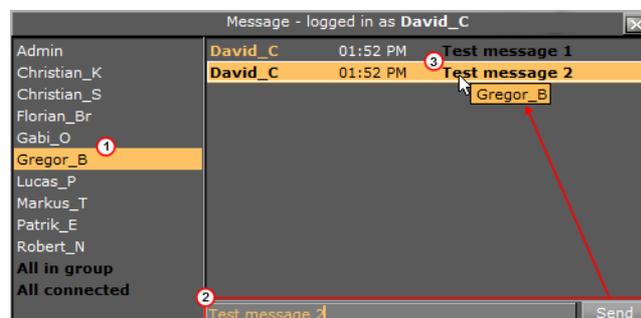
Opens the On Air panel. This panel switches the application from the modelling tool to a plain rendering engine. The application waits for control commands through its communication port, or it can be controlled manually from the control buttons when Viz is set in [On Air Mode](#).

By default, the port number is set to 6100. The port number and other communication settings can be modified in the **Communication** section in the [Viz Configuration](#).

Button Bar



- **Documentation:** Opens the Viz Artist Users Guide.
- **Message Box:** Opens the Message window. This window shows server warnings, and chat messages can be sent/received to/from other users connected to the database (3). To send a message, click on a recipient (1), type a message and click on Send (2) (or press <Enter>).



- **Show Commands:** Opens the Viz Engine Console window. This window shows 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.

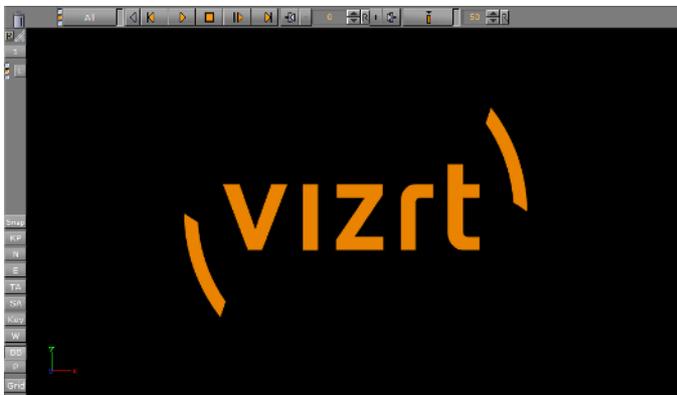
Tip: Use keyboard shortcut <Alt+C> to open the Viz Engine Console window.

-  License Information: Opens the License Information and About window. This window shows the status of the current license such as license expiration date, platform type (Standard PC or Video), licensed plugin packages, Viz version number, etc. This is also where the license can be renewed. For details on how to renew the license see the Viz Engine Administrator Guide.

Click the **About...** button to view the About statement and see all legal notices.

-  Minimize: Minimizes the Viz Artist window.
-  Close: Opens the Quit or Restart menu. In the Quit or Restart menu click on the drop down arrow to open the shut down menu (see [To Close \(Quit\) or Restart Viz Artist](#)).

3.3 Scene Editor



The Scene Editor has a set of useful functions that lets the designer see how the graphics will look when keyed, seen from a different angle in Viz Virtual Studio (virtual sets), positioned on-screen, performance of the scene and more.

On the left side of the Scene Editor are the [Scene Editor Buttons](#). These buttons make it possible to show additional information in the Scene Editor, as well as providing more options while working.

This section contains information on the following topics:

- [Scene Editor Buttons](#)
- [Layer Manager](#)
- [Snapshot](#)
- [Performance Bar](#)
- [Grid Toolbar](#)

See Also

- [Working with the Scene Editor](#)
- [Scene Editor Shortcuts](#)
- [Camera Controls and Shortcuts](#)
- [Orthogonal View Controls and Shortcuts](#)

- [Animation Controls and Shortcuts](#)

3.3.1 Scene Editor Buttons

Click the Scene Editor buttons once to enable the features, click them again to disable.



- **Reset** (R): Set the Viz Artist layout to default size
- **Camera Selection** (1): See [Camera Selection](#)
- **Layer Manager** (L): Show or hide the Layer Manager (see [Layer Manager](#))
- **Snap** (Snap): Takes a snapshot of the current content in the Scene Editor (see [Snapshot](#))
- **Key Preview** (KP): Activates the preview of the key signal
- **Normal Vertices** (N): Shows the normal vertices. For example, it is practical to use this feature when the lighting is not reflected in the required manner. Enabling this feature shows the vertices as purple handles. These handles makes it easier to predict how the light will be reflected
- **Events** (E): Enables the handling of interactive script/plugin events. Interactive scripts and plugins are those related to mouse or keyboard actions
- **Title Area** (TA): Shows the title area as the outline of a light blue rectangle in the Scene Editor

Tip: The size of the title area can be defined under the [Rendering](#) scene settings.

- **Safe Area** (SA): Shows the safe area as the outline of a green rectangle in the Scene Editor

Tip: The size of the safe area can be defined under the [Rendering](#) scene settings.

- **Key** (Key): Shows what the key output will look like
- **Wire Frame Mode** (W): Renders the scene in wire frame mode
- **Bounding Box** (BB): Enables the bounding box visualization for the selected Container

- **P** **Performance Bar:** Shows the [Performance Bar](#)
- **Grid** **Grid:** Opens the [Grid Toolbar](#)

This section also contains information on the following topics:

- [Snapshot](#)
- [Performance Bar](#)
- [Grid Toolbar](#)

3.3.2 Layer Manager

All scenes are, by default, designed to run in the Main layer (2), but can be played out in the Front (1) and Back (3) layer as well. The Layer Manager can be used to test combinations of Scenes in the three layers.

A Scene set in the Main Layer can be edited in the Scene Tree and the Scene Editor panel. Scenes set in the Front or Back Layer can be edited by a double-click on the relevant Scene.



Picking in the [Scene Editor](#) is done on the loaded layer. The [Grid Toolbar](#) also works for the current layer.

This section contains the following topics and procedure:

- [Self Layers](#)
- [Layer Manager Context Menu](#)
- [To Add Scenes to the Front or Back Layer](#)
- [To Edit a Scene Used as a Front or Back Layer](#)

Self Layers

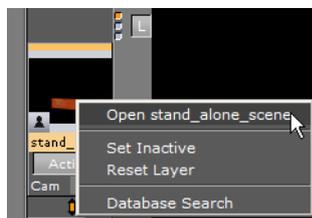
The same Scene can be set to the Front and Back layer, which creates what is known as self layer(s).

This allows the Scenes to be linked, and the adjustment of the camera settings for the Front layer. When saving a Scene, any self layer settings are preserved.

The three small indicators (4) show which layer is active in the Scene Tree, and are referred to as traffic lights.



Layer Manager Context Menu



- **Open <scene>**: Open the Scene for editing.
- **Set Inactive/Active**: Show or hide the layer
- **Reset layer**: Remove the Scene from the layer (the same as drag and drop into the bin).
- **Database Search**: Open the location of the Scene in the Server View.

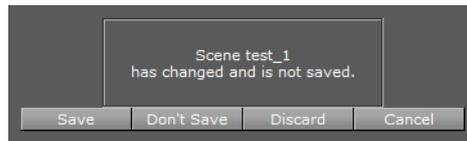
To Add Scenes to the Front or Back Layer

- Drag and drop Scenes onto the layer placeholder.

To Edit a Scene Used as a Front or Back Layer

1. Double-click the Scene to be edited.
2. If another Scene in one of the three layers was modified, select an option from the message which opens:
 - **Save**: Save the changes made in the last selected Scene
 - **Don't Save**: Any changes made in the previously selected Scene will stay, but not saved
 - **Discard**: Remove any changes made in the previously selected Scene.

- **Cancel:** Cancel the operation



See Also

- [Scene Editor Buttons](#)
- [Working with the Scene Editor](#)
- [Timeline Editor](#)

3.3.3 Snapshot

A snapshot can be taken of the current content in the Scene Editor.

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



This section also contains information on the following procedure:

- [To Take a Snapshot](#)

To Take a Snapshot

1. Snapshot RGB or RGBA:
 - **RGB:** Click the **Snap** button
 - **RGBA:** Press **Shift** and click the **Snap** button.
2. In the Snapshot dialog box that shows, enter the **name**, **size**, and **format** for the image.
3. 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.

See Also

- [Scene Editor Buttons](#)
- [Performance Bar](#)

- [Grid Toolbar](#)

3.3.4 Performance Bar

A Performance Bar can be shown at the bottom of the Scene Editor. The Performance Bar gives an indication of the performance (frames per second) the Scene is rendering in.



Note: A Performance Bar can also be shown in [On Air Mode](#) (see [Viz Artist Performance](#)).

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

- [To View the Performance Bar](#)
- [To View all Performance Bar Indicators](#)
- [Performance Bar Properties](#)

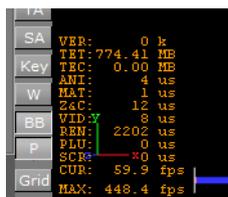
To View the Performance Bar

- Click **P** in the Scene Editor.

Note: This shows the Current (CUR) and Maximum (MAX) parameters (see [Performance Bar Properties](#)).

To View all Performance Bar Indicators

- Middle-click **P**



Performance Bar Properties

- **VER (Vertices):** Shows the number of vertices in the scene.
- **TET (AllocTexSize):** Shows the total allocated size of Texture memory.
- **TEC (TexSize):** Shows the size of the current in use Texture memory.

Note: The actual drawing of Textures has an impact on the performance, more visible textures cause more render time. If texture memory is low it is possible that Textures are not visible in the Scene. The Texture memory usage is an indication for this.

- **ANI (Animation):** Shows how many microseconds all active directors and animation channels take. This indicator is linked to the yellow bar.
- **MAT (Matrix):** Transforms each container in the scene into world coordinate space. This indicator is linked to the cyan bar.
- **Z&C (Z-Sort):** 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.
- **VID (Video):** Shows how many microseconds video input (live video Media Asset) 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.
- **REN (Rendering):** 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.
- **PLU (Plugin):** Indicates how much time in microseconds all active plugins spend in each render cycle. This indicator is linked to the orange bar.
- **SCR (Script):** Shows the consumed time in microseconds from all active scripts. This indicator is linked to the dark green bar.
- **CUR (Current):** 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.
- **MAX (Maximum):** 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.
- **Blue bar:** Shows the number of frames used
- **Green bar:** Shows the number of frames available

The Performance Bar refresh rate can be modified in **User Interface** (see [Viz Configuration](#) in the [Viz Engine Administrator Guide](#))

See Also

- [Scene Editor Buttons](#)
- [Snapshot](#)
- [Grid Toolbar](#)
- [Viz Artist Performance](#)

3.3.5 Grid Toolbar

The Grid toolbar can be shown in the lower part of the Scene Editor. Grids are used to align containers.

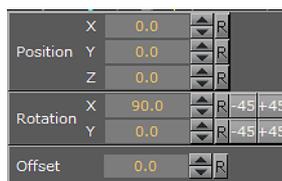


- **Plane Type:** Shows which plane the grid should lie in:
 - **Align X/Y:** Aligns the grid to the X/Y-plane.
 - **Align X/Z:** Aligns the grid to the X/Z-plane.

- **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 an object).
- **Off:** Shows that the grid is set to OFF.
-  **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 Scene Editor.
-  **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 Scene Editor, the position of the object will snap 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 Scene Editor.
-  **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 three buttons, detailed below, 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 as illustrated in the image of the Free Plane Editor options below:



Adjusting the **Position** or **Rotation** in the Free Plane Editor will change the grid's own Position (x,y,z) or Rotation (x,y). Changing the Offset will adjust the grid's location along the normal vector of the grid XY plane (Z position).

To Enable the Grid Toolbar

- Click the Scene Editor's **Grid** button.

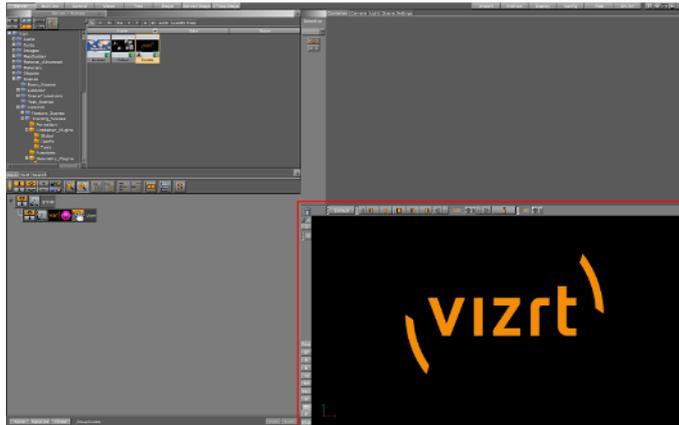
See Also

- [Scene Editor Buttons](#)
- [Snapshot](#)
- [Performance Bar](#)

- [Layer Manager](#)
- [Working with the Scene Editor](#)
- [Timeline Editor](#)

3.4 Working with the Scene Editor

Some transformation parameters of a Container or Containers in a Scene can be edited in the Scene Editor. Furthermore you can switch some properties of a container ON and OFF.



This section contains information of the following topics:

- [Scene Editor Functions](#)
- [Scene Editor Context Menu](#)

See Also

- [Scene Editor Shortcuts](#)
- [Camera Controls and Shortcuts](#)
- [Orthogonal View Controls and Shortcuts](#)
- [Animation Controls and Shortcuts](#)

3.4.1 Scene Editor Functions

This section contains information of the following topics:

- [To Select/Deselect an Object](#)
- [To Copy an Object](#)
- [To Move an Object](#)
- [To Move an Object on its Z Axis](#)
- [To Select Multiple Objects](#)
- [To Deselect Multiple Objects](#)
- [To Add Properties to Multi Selected Objects](#)
- [To Move Multi Selected Objects](#)
- [To Move Multi Selected Objects on Their Z Axis](#)

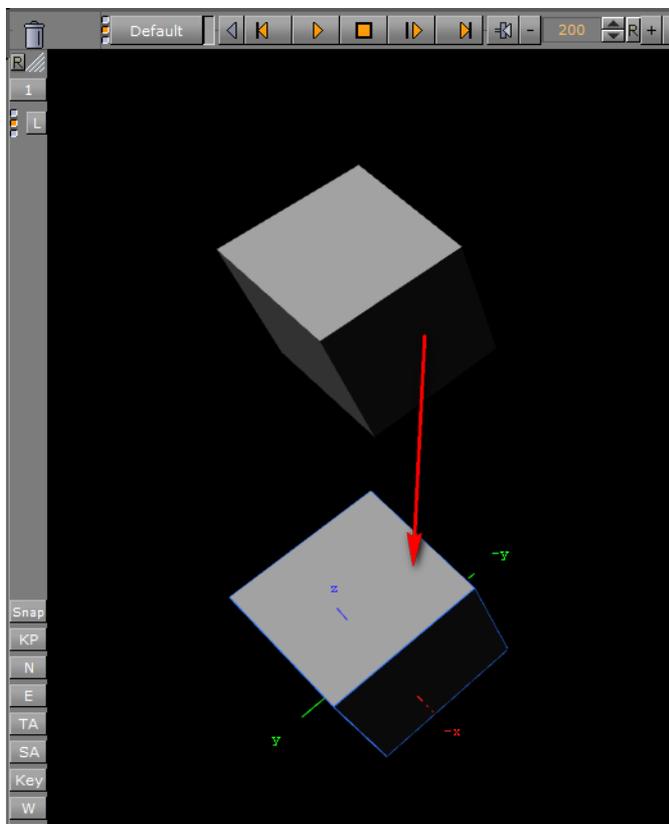
To Select/Deselect an Object

1. Left-click on an object, to select that object.

.....
Note: This is done on a per pixel basis.
.....

- A bounding box shows which object has been selected (if the Bounding Box option is enabled, see [Scene Editor Buttons](#)).
 - The selected object's Container is highlighted in the Scene Tree
 - Click on the Transformation editor in the Properties panel to show to the transformation properties.
2. Press **Space** or click in an empty space to deselect an object.

To Copy an Object



1. Hold <Ctrl> down and click and hold the **left mouse button**
2. Drag and drop the object to the new location.

To Move an Object

1. Left-click on an object, to select that object.
2. Keep the left button pressed and move the object within the Scene Editor.
3. Release the left button

To Move an Object on its Z Axis

1. Left-click on an object, to select that object.

2. Keep the left button pressed and press the right button. Hold both buttons down.
3. Move the mouse to the left or right to move selected object in its Z axis.

Multiple Objects Selected

When multiple objects are selected in the Scene Editor, the Transformation editor, in the Properties Panel, will show the editable properties for all selected objects.

All selected objects will be included in one bounding box (if the Bounding Box option is enabled, see [Scene Editor Buttons](#)). The selected object's Containers are also highlighted in the Scene Tree.

Note: When more than one object is selected the Group coordinate system will show and a bounding box will cover all selected objects. When Local and World is selected a bounding box for each selected object will show.

To Select Multiple Objects

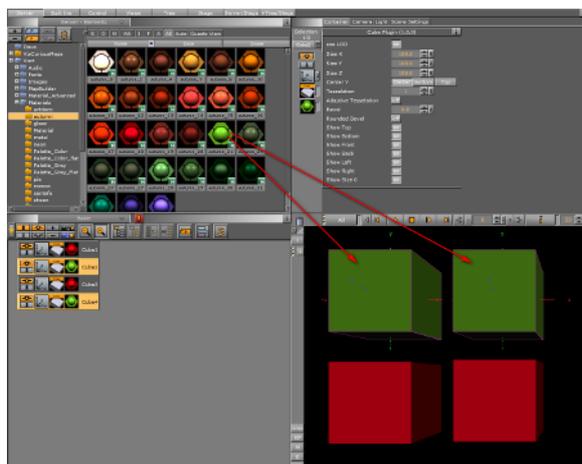
1. Hold the <Ctrl> key and click on each object to select, as required, or
2. Click in an empty space in the Scene Editor and drag around the required objects to select.

To Deselect Multiple Objects

1. Press the **SPACE** bar, or
2. Click in an empty space on the Scene Editor.

Note: Toggle the SPACE bar to select or deselect multiple objects

To Add Properties to Multi Selected Objects



1. Multi select the required objects.
2. From the Server Area select the required Element.
3. Drag the Element to one of the selected objects.
4. All selected objects will take on the property.

To Move Multi Selected Objects

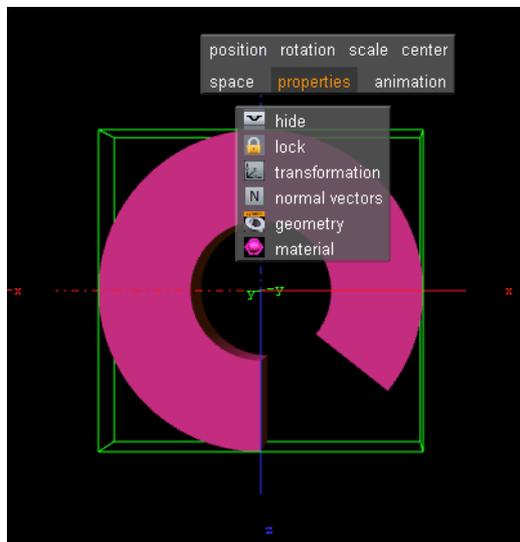
1. Select the required objects (see how [To Select Multiple Objects](#))
2. Left-click one of the objects to be moved and move the multi selected objects within the Scene Editor.
3. Release the left button.

To Move Multi Selected Objects on Their Z Axis

1. Select the required objects (see how [To Select Multiple Objects](#))
2. Left-click on one of the multi selected objects.
3. Keep the left button pressed and press the right button. Hold both buttons down.
4. Move the mouse to the left or right to move selected objects in its Z axis.

3.4.2 Scene Editor Context Menu

In the Scene Editor right-click and hold to open a context menu:



Note: The context menu selection is available on a single Container or multiple Containers.

- **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 coordinate systems:
 - **Parent/Group:** Refers to changes that can be made to the Container or Containers in its Parent or Group coordinate system.
 - **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.

Note: Multi selected objects: When more than one object is selected Group will show and a bounding box will cover all selected objects. When Local and World is selected a bounding box for each selected object will show.

- **Properties** (only applicable if a Container is selected):
 - **Hide/Show:** Hide or show the Container in the Scene Editor.
 - **Lock/Unlock:** Lock and unlock the container. When locked the Container cannot be modified in the Scene Editor.
 - **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**.
 - **Material:** Select enlightened or colored.
 - Any other properties of the container can set to on and off (except materials).
- **Animation:** Shows the animation path of when [Position Key Frame Editor Properties](#) (only applicable if a container is selected and it has an animation defined).

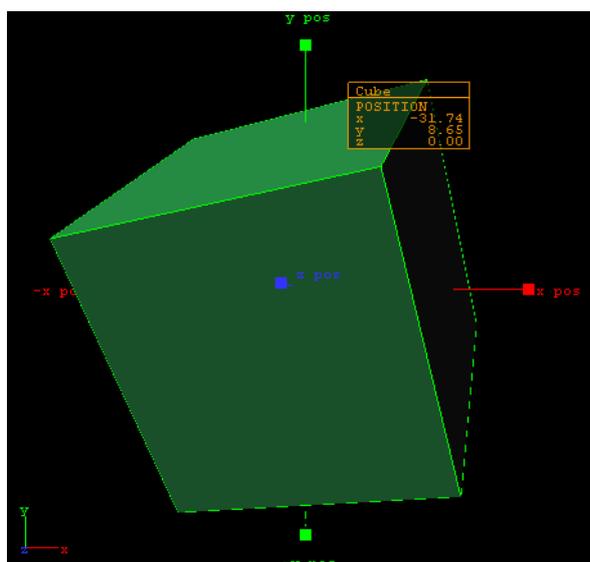
This section contains information of the following topics:

- [To Select an Option from the Context Menu](#)
- [To Change an Object's Position](#)
- [To Rotate an Object](#)
- [To Scale an Object](#)
- [To Move the Axis Center of an Object](#)

To Select an Option from the Context Menu

1. Right-click and hold.
2. Move the cursor to the required option.
3. Release to select.

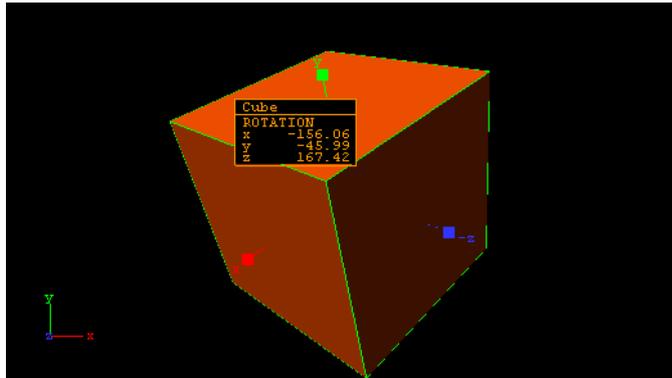
To Change an Object's Position



1. Select an object.
2. Right-click the object and select **position** from the context menu.
3. Grab and drag a handle.

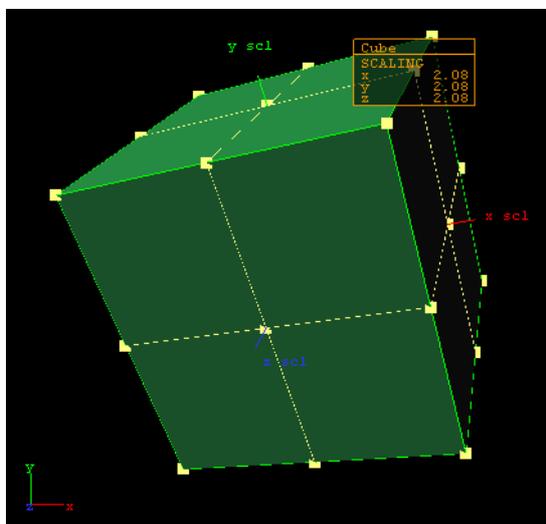
The container will follow the movement along the axes shown in the selected camera view. During this operation a status window is shown, giving you information about the position.

To Rotate an Object



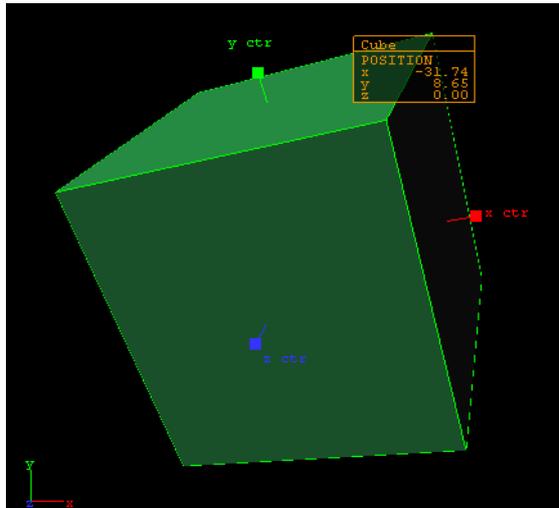
1. Right-click to open the context menu.
2. Select **rotation**.
3. Drag the handles side ways to rotate on all the axes.

To Scale an Object



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

To Move the Axis Center of an Object



1. Right-click to open the context menu
2. Select **center**.
This will show the axis of the selected container with handles on their ends.
3. Grab one of the handles and change the center position by dragging them around the Scene Editor.

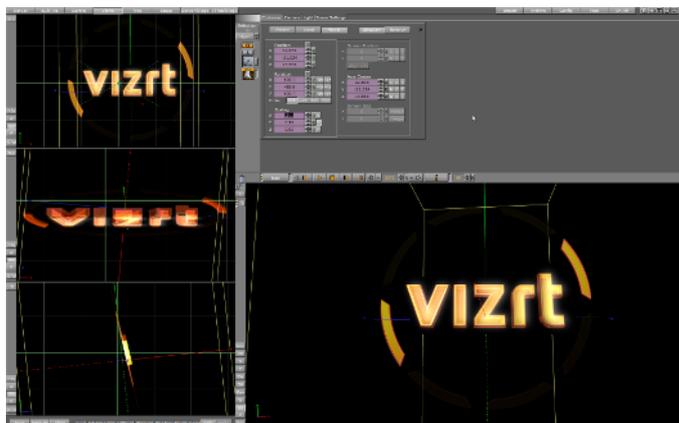
See Also

- [Layer Manager](#)
- [Timeline Editor](#)

3.5 Scene Editor Views

To get a better overview of an object, or objects, in a Scene, and to visualize their position in relation to lights and cameras, open the Scene Editor Views function.

To open the views panes click on the [Views Button](#) in the [Main Menu](#).



Each of the three views, in principle, has the same functionality as the [Scene Editor](#) and let you do the same operations on objects as is possible in the [Scene Editor](#).

As well as the [Scene Editor](#), the three side views can be set to view through one of the 16 editable cameras or through one of the six predefined orthogonal views (see [Camera Selection](#)). When set to one of the predefined orthogonal views this allows the position and direction of lights and cameras, to be modified.

If the [Scene Editor](#) is set to show one of the predefined orthogonal side views, there are some special shortcuts that may be used to pan and zoom:

- **Pan:** Left-click, hold and drag in the background (not the objects).
- **Zoom:** Hold down **Z** (zoom in) or **X** (zoom out) and draw a rectangle with the left button pressed, or position the cursor over a panel and press the **Z** or **X** keys multiple times 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 show as symbols.

Click on a camera in the side views and that camera's focus is set in the [Scene Editor](#), and handles to edit the position and direction will show (drag the handles to change position and direction).

The same functionality is available for lights (see [Lights](#)). The light currently selected and those lights which have their **Vis** status set to **On**, will be visible, as symbols. Their position and direction can be edited with the cursor.

See Also

- [Light Editor](#)
- [Cameras](#)

4 Scene Tree Knowledge and Skills

Scenes are root Elements that hold other Elements. Created Scenes are stored in the Graphic Hub database and are available through the Server area. For general information about Scenes and procedures on how to create, import, rename, and delete Scenes, see [Working with Scene Elements](#).

This section contains information on the following topics:

- [Open a Scene](#)
- [Scene Tree](#)
- [Scene Tree Menu](#)
- [Favorites Bar](#)
- [Containers](#)
- [Modify Container Properties](#)

4.1 Open a Scene

To Open a Scene in the Database

1. Select **Server** from the [Main Menu Left](#).
2. In the Server Tree, select the folder or project where the Scene is.
3. Open the Scene:
 - Double-click the Scene to open it, or
 - Drag the Scene to the Scene Tree area, or
 - Drag the Scene to the renderer window.

The content of the Scene will be shown as Containers in the Scene Tree, and visualized in the renderer window.

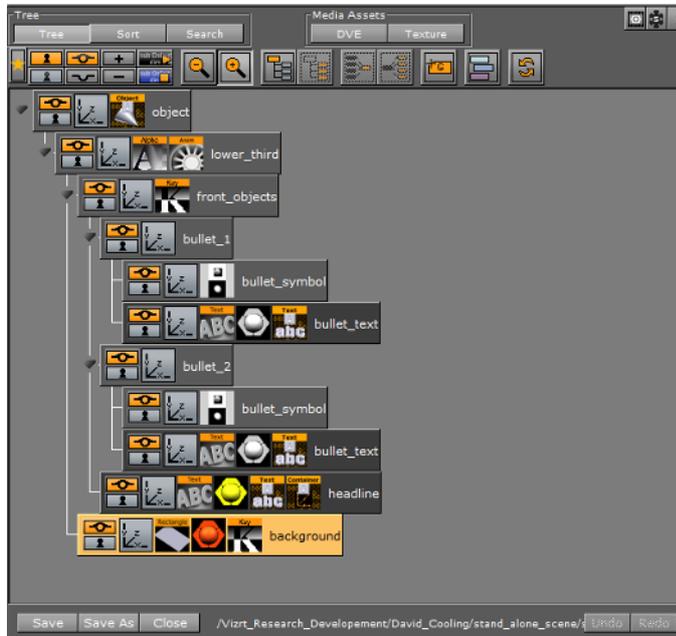
Note: [Image Editor](#) or Scenes are marked with a Keyhole icon  .

Tip: If the folder or project's path is unknown, search for the Scene in the [SmartView Search](#) function under the <Username>s View tab in the Server area (see [Content Switcher](#)).

Tip: The Status bar in the lower left part of the [Main Window](#) (between the Close and Undo buttons) shows the name and path of the current Scene. Click on the path to select the Scene in the Server area.

4.2 Scene Tree

The Scene Tree is the logical visualization of all the properties in a Scene. The Scene Tree consists of Containers that hold properties. The properties can be Geometries, Images, Materials, Texts, Transparency functions, Key functions, and many more.



To build a Scene add properties, like Image Elements or Geometries (Built-Ins), to the Scene Tree (see [Add Properties to Containers and Scenes](#)).

The Containers in the Scene Tree are organized in a hierarchy and grouped in logical divisions. Containers and Groups are explained in more detail in the following paragraphs.

This section contains information on the following topics and procedures:

- [To View the Scene Tree](#)

To View the Scene Tree

The Scene Tree is available when these buttons in the [Main Menu](#) are clicked on:

- [Server Button](#)
- [Built Ins Button](#)
- [Control Button](#)
- [Tree Button](#)
- [Stage Button](#)
- [+Tree/Stage Button](#)

See Also

- [Scene Tree Menu](#)
- [Favorites Bar](#)
- [Containers](#)

- [Group Containers](#)
- [Scene Tree Shortcuts](#)
- [Basic View Controls and Shortcuts](#)
- [Sort View Shortcuts](#)

4.3 Scene Tree Menu

At the top of the Scene Tree area is the control menu:



- **Tree**
 - **Tree:** Shows an overview of all the Containers in the Scene (see [Tree Button](#)).
 - **Sort:** Sorts the Elements in the Scene Tree based upon criteria that are set in the Display area (see [Sort Button](#)).
 - **Search:** Search Elements in the Scene Tree (see [Search Button](#)).
- **Media Assets [Manager]:** In the [Media Assets \[Manager\]](#), add Media Assets to a Scene as:
 - [DVE](#)
 - [Texture](#)
- **! Scene Log:** Shows if an error occurs. Opens the Log window, which gives information about the loading process of the Scene, such as errors, for example if a plugin is not found. Click on it to open the Scene Log.
 - If the log window is closed, but not cleared, the exclamation mark icon will show in gray. It will show red again if another fault occurs.
- **Media Asset Properties Editor:** Click on the icon at any time to open the [Media Asset Properties](#) Editor panel of the Scene Settings.
- **Scene Plugin:** Click on the icon at any time to open the [Plugin](#) panel of the Scene Settings. In the Plugin panel Scene plugins can be added to the current Scene (see [To Add a Scene Plugin](#)). This icon shows orange when a Scene contains a Scene Plugin. Click on the icon to view the Scene plugin properties panel.
- **i Scene Info:** Opens the Scene Information window, which gives information about the Scene entered by the designer. Shows orange if information is available.

This section contains information on the following topics:

- [Tree Button](#)
- [Sort Button](#)
- [Search Button](#)
- [Media Assets \[Manager\]](#)

4.3.1 Tree Button

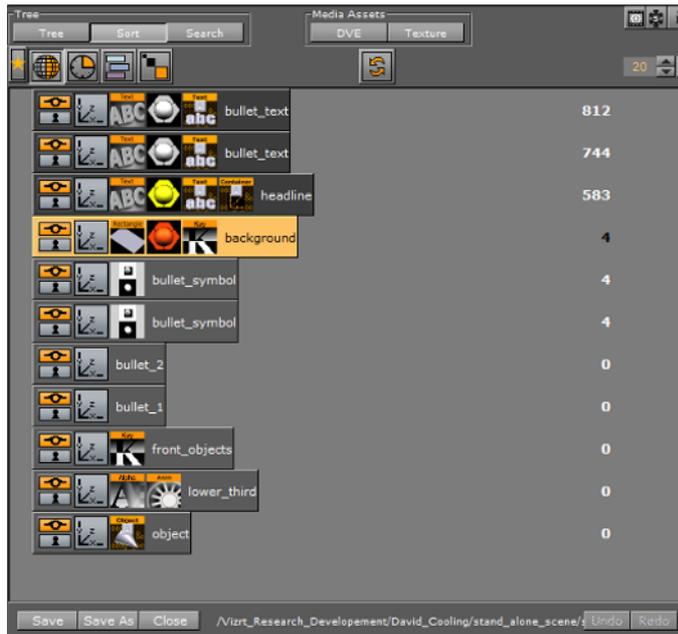
If the Tree button is clicked, all Containers are shown in the tree hierarchy (see [Scene Tree View](#)).



- **Show/Hide Favorites:** Show or hide the [Favorites Bar](#)
- **Lock/Unlock:** Lock or unlock the ability to modify the transformation of selected Containers in the [Scene Editor](#). Middle mouse click to lock or unlock all Containers
- **Show/Hide:** Show or hide selected Containers in the Scene Editor, and eventually in the final rendered output. Middle mouse click to show or hide all Containers
- **Expand/Collapse:** Expand or collapse the Scene Tree
- **Start/Stop:** Start (compiles) or stop all Container scripts
- **Zoom In/Zoom Out:** Zoom in (larger Containers) or Zoom out (smaller Containers) on the Container tree. Zoom out applies smaller icons to the Containers. This gives a better overview, especially in complex Scenes
- **Group/Ungroup:** Create a new group (see [Group Containers](#)), and places the selected Containers as Sub-Containers. The Ungroup button moves the Sub-Containers up one level
- **Merge/Split:** Merge the selected Containers to one compound object, or split an existing compound object. For more information, see [Merge Containers](#)
- **Create Group:** Create a new empty group at the top of the Scene Tree. Drag the icon to place a new group at a specific place in the Scene Tree (see [To Add Root or Sub-Containers](#))
- **Color:** Apply a color to 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](#)). Text for global colors can be modified in **Colors** (see **User Interface** in [Viz Configuration](#) (see the [Viz Engine Administrator Guide](#)))
- **Refresh:** Refresh the Scene Tree. Use this button if scripts have been deployed to change the Scene Tree

-  **Unique:** Show unique group objects with the same Texture only once. This option is only available if the Containers are sorted by Texture size (click the Texture Size button).
-  **Refresh:** Refresh the Scene Tree.
-  The input box, on the right, defines the number of search results shown. Click <R> to set to default number of search results (20).

Scene Tree Sort View



4.3.3 Search Button

If the Search button is clicked, the Elements in the Scene Tree can be searched, based on various search criteria.



The different buttons are used to search for specific properties of the Scene (see [Search Options](#)) or highlight the search results (see [Show Search Results](#)). Search results are highlighted in the Scene Tree, and the number of search results are shown (see [Show Search Results](#)).

Example: If a plugin (for example *Extrude*) is dragged to the *Dragsite for Properties*, a click on the *Show all* button will highlight all occurrences of that plugin in the Scene Tree.

Search Options



- **Name:** Search for Containers with a specified name. Type the search string in the Name field (1). Wildcards (for example *, ? and []) may be used in the search, see [Wildcard Text Description](#) for details

Note: By default, text wildcard expansion "*" are enabled at the *start* and at the *end* of the search string. This means that searching for "player" will find any node named "player" in the scene tree and also nodes named for instance *playername* or *secondplayername*.

Click on the drop arrow (2) to show extended search parameters. Options are to search for Parameter, Values and Plugin names within the Scene (see [To Do an Extended Search](#))

Note: When the extended parameters option is active, all other search options, apart from Name, are inactive.

-  **Dragsite for Properties:** Drag an Element from the Server area, from a Container in the Scene Tree or from the [Favorites Bar](#) to search for its occurrence within the Scene
-  **Hidden:** Search for all hidden Containers
-  **Locked:** Search for all locked Containers
-  **Inactive Properties:** Search for Containers that contain properties that are set to inactive
-  **Color:** Searches for all Containers with the assigned color tile (see [Container Context Menu](#))

Show Search Results

-  **Show Next:** Show the next result
-  **Show Previous:** Show the previous result
-  **Show First:** Show the first result
-  **Show Last:** Show the last result
-  **Show All:** Show all results
-  **Search Results:** Shows how many results were found

Tip: To show all results is especially practical to do a scene cleanup.

Tip: If a specific plugin is no longer to be used in a Scene, make a search for the plugin and select **Show All** to highlight all the results. Drag one of the highlighted plugins to the trash can (all the highlighted plugins will be deleted from the Scene).

Tip: If **Show all** search results is selected, the transformation properties can be changed for all highlighted Containers.

Tip: If **Show all** search results is selected, a plugin or an element (except a Scene) can be added to all the highlighted Containers (drag a plugin or an element to one of the highlighted Containers).

Tip: Click the show next or previous buttons to change individual Container properties (see [Container Editor](#)).

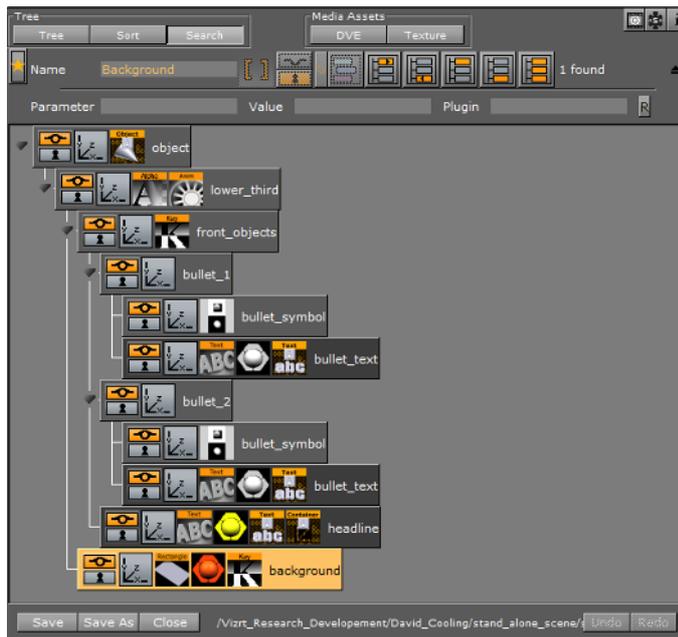
To Do an Extended Search



1. Go to the Scene Tree and click **Search**.
2. Click drop arrow to show the extended search options (1).
3. Enter a search criteria. The search criteria can be a combination of Parameter, Value and Plugin name. Below are some examples:
 - **Parameter:** tessellation, width, *show*
 - **Value:** 20, *10.0*
 - **Plugin:** *rect*
4. Press <Enter>.
5. Use the [Show Search Results](#) buttons to navigate the Scene Tree.

Tip: Use the label names in the plugin editors as a search criteria. If there is no hit, use the asterisk (*) symbol.

Scene Tree Search View



4.3.4 Media Assets [Manager]

In the Media Assets Manager panel [Media Assets](#) can be added to a Scene as [DVE](#) or [Texture](#).



This section contains the following topics:

- [Media Assets Added as DVE](#)
- [Media Assets Added as Texture](#)

Media Assets Added as DVE

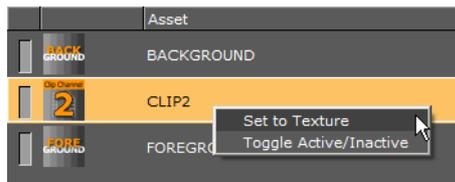
For detailed information see [Media Assets as DVE](#) in [Media Asset Workflow](#).



Right-click the Media Asset to open its context menu:

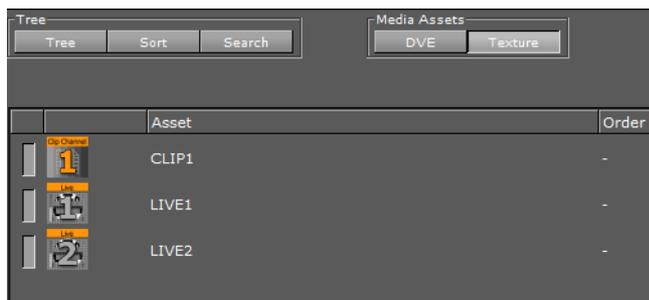
Tip: Media Assets can also be multi selected and set to Texture or made Active / Inactive.

- **Set to Texture:** Set the Media Asset to Texture
- **Toggle Active/Inactive:** Set the **Media Asset** to Active or Inactive



Media Assets Added as Texture

For detailed information see [Media Assets as Texture](#) in [Media Asset Workflow](#).

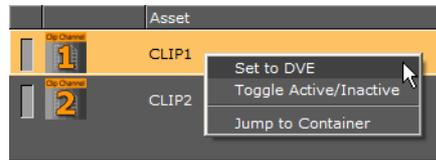


Right-click the Media Asset to open its context menu:

Tip: Media Assets can also be multi selected and set to DVE or made Active / Inactive.

- **Set to DVE:** Set the Media Asset to DVE
- **Toggle Active/Inactive:** Set the **Media Asset** to Active or Inactive

- **Jump to Container:** Jump (go) to the Container that holds the Media Asset in the Scene Tree (if a Container was created for the Media Asset (see [To Add a Media Asset as Texture](#))).



See Also

- [Media Assets](#)
- [Video Playback](#)
- [To Add a Plugin to Multiple Containers](#)

4.4 Favorites Bar

For quicker access to Elements that are used often, add Elements to the Favorites bar (see [To add an Element as a Favorite](#)). To open or close the Favorites bar click on the Star icon (1) at the top of the Scene Tree area.



All Elements, except Scenes and Audio files, can be added as favorites. Geometry, Container and Shader plug-ins can also be added to the favorites. Favorites are saved as part of the user settings to the database.

Tip: To view more favorites, use the scroll bar or resize the favorites panel, as required.

Click on the buttons below the Favorites bar (2) to show or not show favorite Elements from the categories detailed here:

- **G:** Geometries
- **M:** Materials
- **MA:** Materials Advanced
- **I:** Images
- **F:** Fonts
- **GP:** Geometry plugins
- **CP:** Container plug-ins
- **SH:** Shader plug-ins

Elements can also be fast selected. Click on an Element button with:

- **<Ctrl+Click>:** Selects all
- **<Shift+Click>:** Deselects all

- **<Alt+Click>**: Only show the clicked type

This section also contains information on the following procedures:

- [To add an Element as a Favorite](#)
- [To Remove an Element as a Favorite](#)
- [To Remove an Element as a Favorite](#)

To add an Element as a Favorite

1. Right-click the Element in the Server or Built Ins area.
2. Select **Add as Favorite...**,

or

1. Click on the Star icon to open the Favorites bar (see [Favorites Bar](#)).
2. Drag an Element from the Server or Built Ins area to the Favorites bar.

Note: Elements can also be added as favorites from the search results window, after a Graphic Hub database search (see [Import Files and Archives](#)).

To Remove an Element as a Favorite

1. Right-click the Element in the Server or Built Ins area.
2. Select **Delete from Favorites...**,

or

1. Click on the Star icon to open the Favorites bar (see [Favorites Bar](#)).
2. Drag an Element from the Favorites bar to the trash can.

To Add a Favorite Element to a Container

1. Select the target Container(s).
2. Right-click the required Element in the Favorites bar.
3. Select Add to Container(s).

or

1. Select the target Container(s).
2. Drag the required Element from the Favorites bar to the target Container

Note: If more than one Container is selected the Element will populate all selected Containers.

4.5 Containers

A Container combines a set of properties to one block used in the Scene (a Scene can contain many Containers). Containers can also be multi selected to modify or move many Containers at once.

This section contains information on these topics:

- [Container Icons](#)
- [Container Context Menu](#)
- [Container Editor](#)
- [Add Containers to a Scene](#)
- [Select Multiple Containers](#)
- [Copy Containers](#)
- [Move Containers](#)
- [Rename Containers](#)
- [Edit Container Properties](#)
- [Delete Containers](#)
- [Merge Containers](#)
- [Group Containers](#)

See Also

- [Modify Container Properties](#)
- [Multiple Selection Container Properties](#)
- [Multiple Select Transformation](#)
- [Add Properties to Containers and Scenes](#)

4.5.1 Container Icons



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

-  **Show/Hide:** Shows if the Container and all its Sub-Containers are visible/invisible in the [Scene Editor](#). Click the icon to change the state (see also [Tree Button](#))
-  **Locked/Unlocked:** Shows if the Container is locked so that properties cannot be manipulated in the [Scene Editor](#), or unlocked so that properties can be manipulated. Click the icon to change the state (see also [Tree Button](#))

Example: An unlocked element can be resized and repositioned in the Scene Editor.

-  **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 shown in the Scene Tree. If the arrow points downwards, the Container is expanded and the Sub-Containers are shown 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 geometries 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.



- **Geometry Elements:** Shows the geometry held by the Container (e.g. a merged or imported geometry). The thumbnail icon reflects the geometry.



- **Material Elements:** Shows 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 Material Elements](#) and [Material Editor](#)).

Note: A Sub-Container can inherit color from a parent Container.



- **Image Elements:** Shows 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. Click the icon to open the [Image Editor](#).



- **Font Elements:** Shows 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 Built Ins:** Shows the geometry held by the Container. The icon reflects the built-in [Geometry Plugins](#), in this example a cone.



- **Container Built Ins:** Shows that the Container holds a Container function. The icon reflects the [Container Plugins](#), in this example an alpha.



- **Media Asset Built Ins:** Shows that the Container holds a Media Asset. The icon reflects the [Media Assets](#), in this example a Clip Channel.



- **Animation:** Shows that the Container contains animated elements (see [Create Animations](#)).

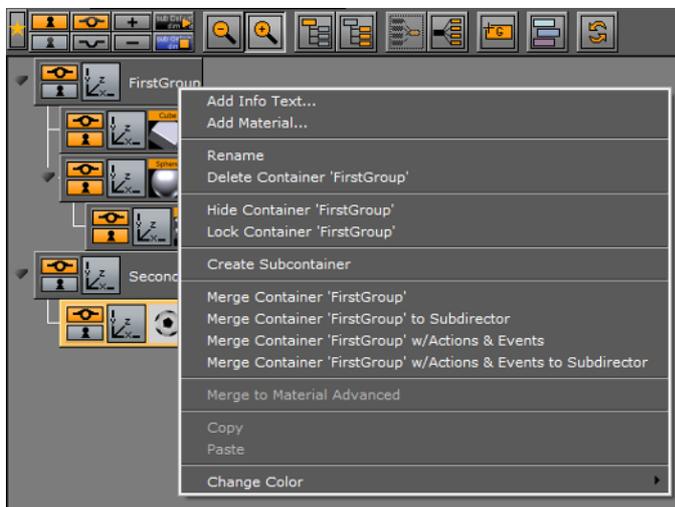
4.5.2 Container Context Menu

A right-click on a Container in the Scene Tree opens a Container context menu.

This section contains information on these topics:

- [Right-click on a Container Name](#)
- [Right-click on a Container Property](#)
- [Right-click on a Container Media Asset Property](#)

Right-click on a Container Name



Do a right-click on the Container name, to show these context menu options (dependent on the Container contents):

- **Add Info Text:** Makes it possible to add text-based information on a Container. Also available as a plugin by clicking **Built-ins** and then double-clicking the **Container Plugins** tab.
- **Add Material:** Adds a Material to the selected Container. See also [Working with Material Elements](#).
- **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 Scene Editor.
- **Lock/Unlock Container [name]:** Disables or enables the ability to modify the transformation of the selected Container in the Scene Editor.
- **Create Sub Container:** Creates a new empty Sub-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 (see [Merge 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 Material Advanced Elements](#)).
- **Change Color:** 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.

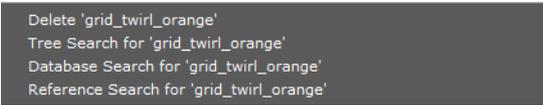
Tip: To modify the color of multiple containers, multi select the required Containers and click on  in the [Tree Button](#).

See Also

- How to configure **Colors** (see **User Interface** in [Viz Configuration](#) (see the [Viz Engine Administrator Guide](#)))

Right-click on a Container Property

Do a right-click on a Container property to show these additional menu options:



```

Delete 'grid_twirl_orange'
Tree Search for 'grid_twirl_orange'
Database Search for 'grid_twirl_orange'
Reference Search for 'grid_twirl_orange'

```

- **Delete [property]:** Deletes the property from the Container.
- **Tree Search for [property]:** Selects the [Search Button](#), and searches the Scene Tree for instances of this property.
- **Database Search for [property]:** Searches for the property in the database, and shows the result in the Server area.

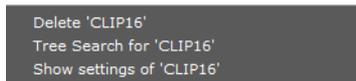
Note: This menu option is only available for images.

- **Reference Search for [property]:** Searches for the property in the Graphic Hub database, and shows the result in the Search window (see [Reference Search](#)).

Note: This menu option is only available for images.

Right-click on a Container Media Asset Property

Do a right-click on a Container Media Asset property, to show these additional menu options:



- **Delete <Media Asset>:** Deletes the Media Asset from the Container.
- **Tree Search for <Media Asset>:** Selects the [Search Button](#), and searches the Scene Tree for instances of this property.
- **Show settings of <Media Asset>:** Opens the Texture View (Media Asset Manager) and the Media Assets tab of the Scene Settings in the Property panel (see [Media Asset Workflow](#)).

4.5.3 Container Editor

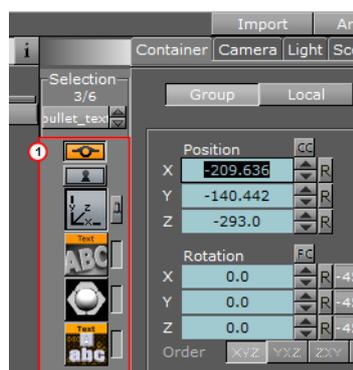
When a Container is highlighted in the Scene Tree a list of all properties and plugins added to that Container, shows in the Container Editor (Container Menu (1)). Click on each icon to view its editor menu.

If more than one Container is selected, select which Container to view from the **Selection** panel.

This section contains the following topics:

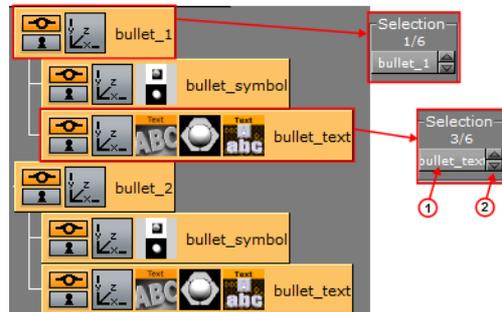
- [Container Editor Properties](#)

Container Editor Properties



- **Selection:** When multiple Containers are selected choose which Container properties to view (see [Multiple Select Transformation](#) and [Multiple Selection Container Properties](#)):

Either use click the Container name (1) to show a context menu or use the up and down arrows (2), to select a Container.



- **Container Menu (1):** Shows a list of all properties and plugins added to the Container selected in the Scene Tree, or the Container selected in **Selection**. Click on an icon to open an editor (see [Modify Container Properties](#))
 -  **Show/Hide:** Show or hide a Container in the [Scene Editor](#), and eventually in the final rendered output. Click to show or hide the Container.
 -  **Lock/Unlock:** Enable or disable the ability to modify the transformation of a Container in the [Scene Editor](#). Click to lock or unlock the Container.

Some editors like the [Transformation Editor](#) enable properties for one or multiple selected Containers to be set. Other editors have the option to enable or disable certain properties (see [Manipulate Container Properties](#)).

4.5.4 Add Containers to a Scene

Containers can be added to the Scene by dragging an element onto the Scene Tree, Scene Editor or Property Editor.

By dragging an element to the Scene Editor, the Container will be added at the top of the Scene Tree.

Tip: 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.

This section contains the following topics:

- [To Add Root or Sub-Containers](#)
- [To Add Empty Containers](#)
- [To Add an Empty Container](#)
- [To Add a Container at a Specific Location in the Scene Tree](#)

To Add 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.

To Add 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.

4.5.5 Select Multiple Containers

More than one Container can be selected in the Scene Tree.

When multiple Containers are selected in the Scene tree, properties can be applied to all the selected Elements (see [Multiple Selection Container Properties](#)). This multiple selection feature is global.

This section contains the following topics and procedures:

- [To Select Multiple Containers](#)
- [To Select all Containers](#)
- [To Select a Range of Containers](#)

See Also

- [Multiple Selection Container Properties](#)
- [Multiple Select Transformation](#)
- [Containers](#)

To Select Multiple Containers

- Press <Ctrl> and click on each Container

To Select all Containers

- With the cursor over the Scene Tree, press <Ctrl+A>

To Select a Range of Containers

1. Click on a Container.
2. Press <Shift> and click the last required Container.

All Containers in between are selected.

4.5.6 Copy Containers

To Copy a Container Within a Scene

- Hold <Ctrl> and drag 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 and drag the Containers to their new position, and then click the right mouse button to release.

4.5.7 Move Containers

To Move a Container Within a Scene

- Drag it to the new position, or
- Use <Ctrl+X> and <Ctrl+V> to cut and paste the Container(s) to the new position.

To Retain Transformation Properties While Moving a Container Within a Scene

Pressing the <Alt> button while moving a Container (or group of Containers) within the Scene will retain the transformation properties. This lets a Container, which is part of a Group, to be moved to another group while the absolute transformation of the Container will not change, although the parent Containers (Groups) have different transformations.

For example:



1. Group 1: The Group position is X 100.0 and Y 100.0. The Container position is X 0.0 and Y 0.0.
2. Group 2: The Group position is X 0.0 and Y 0.0.

When a Container is moved from Group 1 to Group 2, while pressing the <Alt> button, the moved Container will inherit its former Group's position values (i.e. X 100.0 and Y 100.0), so its absolute transformation stays the same.

Moving it back, while holding the <Alt> button, its transformation properties will be readjusted. Meaning, if you adjust the moved Container's position X to 200.0, before

moving it back to its original Group, its position X will be 100.0 and position Y 0.0 after the move.

1. Add two group Containers to the Scene Tree
2. Rename the groups to Group 1 and Group 2
3. Open Group 1's transformation editor and set Position X and Y to 100.0
4. Add a geometry (e.g. Cylinder) as a Sub-Container of Group 1
5. Add material to the Cylinder
6. Press and hold the <Alt> button and place the Cylinder as a Sub-Container of Group 2
 - When the Container is moved a letter R will show in the drag and drop window.
7. Release the <Alt> button
8. Open the Cylinder Container's transformation editor and set Position X to 200.0
9. Press and hold the <Alt> button and place the Cylinder as a Sub-Container of Group 1 and check that the transformation properties are:
 - Group 1: Position X 100.0 and Y 100.0
 - Cylinder: Position X 100.0 and Y 0.0

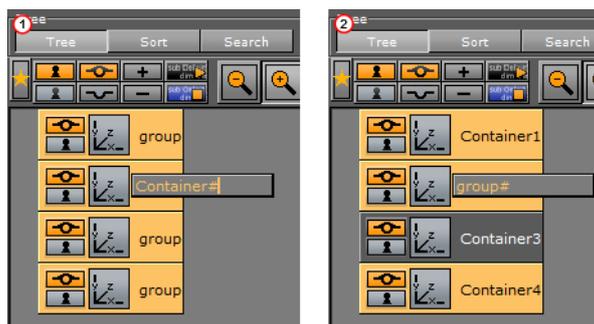
4.5.8 Rename Containers

To Rename a Container

- Double-click the Container name, or
- Right-click a Container. Click on Rename in the Container context menu.

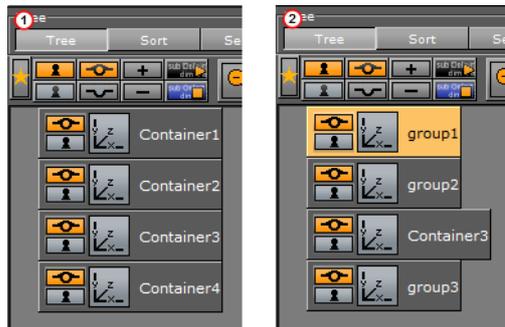
To Rename Multiple Containers

1. Multi select the required containers (see [Select Multiple Containers](#)).
2. Rename one of the selected containers and add # to the end of the text:
 - Example 1: Container#
 - Example 2: group#



3. Press <Enter>.
4. All selected Containers will be renamed.
 - Example 1: Container1, Container2, Container3, etc.

- Example 2: group1, group2, group3, etc.



See Also

- The [Corena](#) plugin

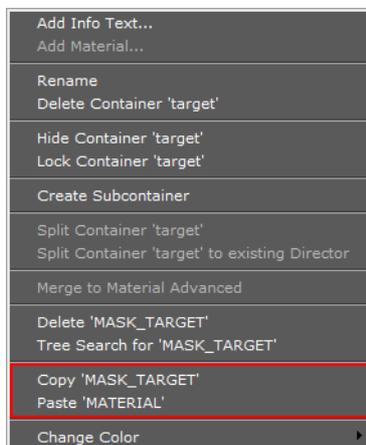
4.5.9 Edit Container Properties

To Edit the Properties of a Container

- Select the Container in the Scene Tree, or in the Scene Editor. Its editor will then show in the Properties Panel (see [Modify Container Properties](#)).

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.



4.5.10 Delete Containers

To Delete a Container from a Scene

1. Drag a Container to 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.

4.5.11 Merge Containers

Containers can be merged into compound objects and saved as reusable components to the Graphic 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 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 "*Old merge style, convert to new merge style*" shows. This is a well-known [Autofollow](#) plugin 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/Engine versions to the official 3.2.2 release and later by opening the Scene and converting it to the new merge style.

This section contains the following procedures:

- [To Merge Containers into a Compound Object](#)
- [To Split a Merged Compound Object](#)
- [To Save a Compound Object to a Graphic Hub](#)

To Merge Containers into a Compound Object



1. Select a Container which contains Sub-Containers.
2. Merge the Container with all its Sub-Containers:
 - Click the **Merge** button, or
 - Press <Ctrl+M>, or
 - or right-click the Container and select **Merge Container [Name]** from the context menu.



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

To Split a Merged Compound Object



1. Select a merged Container.
2. Split the Container:
 - Click the Split button, or
 - Press <Ctrl+Shift+M>, or
 - Right-click the merged Container and select **Split Container [Name]** from the context menu,

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

To Save a Compound Object to a Graphic Hub

1. Right-click a Container (with or without Sub-Containers),
2. In the context menu select **Merge Container [name]**.
3. Drag the compound object's icon from the property editor or Scene Tree and drop it onto a folder/project in the **Server Area**.
4. Type a descriptive name for the object in the dialog box
5. 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

- [Group Containers](#)
- **User Interface** section (see [Viz Configuration](#) in the [Viz Engine Administrator Guide](#)) for setting default merge and split behaviors

4.5.12 Group Containers

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 shown in the Scene Tree. If the arrow points downwards (▼), the Container is expanded and the next level of Sub-Containers are shown.

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



1. Select the Containers to be grouped.
2. Click the **Group** (<Ctrl+G>) button from the Display area.
This adds 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



1. Select the parent Container
2. 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

- [To Add Empty Containers](#)

4.6 Modify Container Properties

When a Container is selected in the Scene Tree, the selected Container property, for example the transformation properties, can be modified in the Properties panel.

Elements stored in the Graphic Hub database can be added to Scenes:

- Geometry or Fontstyle Elements 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 a geometry of a Container, or applied on an existing Container as a Texture
- Geometry plugins can also be added to the Scene

See [Add Properties to Containers and Scenes](#) for details on how to add properties to Containers.

This section contains information on the following topics:

- [Manipulate Container Properties](#)
- [Multiple Selection Container Properties](#)
- [Transformation Editor](#)
- [Copy Properties](#)
- [Geometry Editor](#)
- [Text Editor](#)
- [Texture Editor](#)
- [Plugin Editor](#)
- [Material Editor](#)

4.6.1 Manipulate Container Properties

There are a few operations that apply to all Container properties.

Note: Container properties are Elements and Plugins which have been applied to a Container (see [What are Elements](#) and [Built Ins](#))

This section contains information on the following procedures:

- [To Replace a Property Applied to a Container](#)
- [To Remove a Property from a Container](#)
- [To Save Properties in the Database](#)
- [To Enable/Disable Properties on a Container](#)

See Also

- [Multiple Select Transformation](#)
- [Multiple Selection Container Properties](#)

To Replace a Property Applied to a Container

- Drag the new Element or Plugin to the Container.

To Remove a Property from a Container

- Drag the property from the Container to the trash can, or
- Drag the property from the [Container Editor](#) to the trash can, or
- Right-click the property on the Container in the Scene Tree, and select **Delete <Name of the Property>**.

Note: Every Container must contain transformation properties, so the transformation property cannot be removed.

To Save Properties in the Database

If a scene specific property has been created, and it is likely that the property should be reused in another scene, it is possible to save the property in the database.

- Drag the property from the Container to the required folder/project in the Server area.

To Enable/Disable Properties on a Container



Note: Element properties are automatically enabled when added to a Container.

This is a practical tool when multiple items are applied to a Container.

- Click the switch next to the icon which represents the item in the [Container Editor](#).

See Also

- [Containers](#)

4.6.2 Multiple Selection Container Properties

When multiple Containers are selected in the Scene tree, properties can be applied to all the selected Elements. This multiple selection feature is global. Select the required Containers then apply the appropriate property.

This section contains the following topics and procedures:

- [To Add Properties to Multiple Containers](#)
- [To Change Specific Properties on Multiple Containers](#)
- [To Select a Container in a Selection](#)

See Also

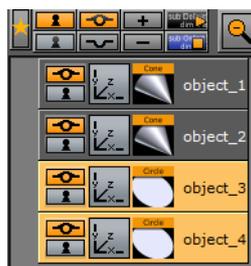
- [Select Multiple Containers](#)
- [Multiple Select Transformation](#)

To Add Properties to Multiple Containers

1. Multi select the required Containers (see [Select Multiple Containers](#)).
 2. Add an Element or Plugin:
 1. Right-click on an Element or Plugin in the Server View.
 2. Select **Add to Container(s)** (if available).
- or
- Drag an Element or Plugin to one of the selected Containers

Example:

1. Create four Containers, named object_1 to object_4.
2. Multi select two Containers.
3. Right-click on the Cone Geometry, and select **Add to Container(s)**.
4. Multi select two Containers.
5. Right-click on the Circle Geometry, and select **Add to Container(s)**.



6. Multi select all four containers.

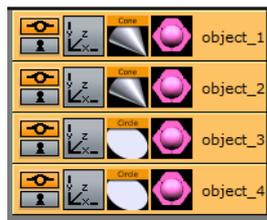
7. Right-click on a Material, and select **Add to Container(s)**.



To Change Specific Properties on Multiple Containers

1. Multi select the required Containers (see [Select Multiple Containers](#)).

Note: If several, or all, of the Containers in a Scene are selected and contain the property to be modified, then all selected Container properties of this type will be changed.



2. Change a parameter of the Circle Geometry.

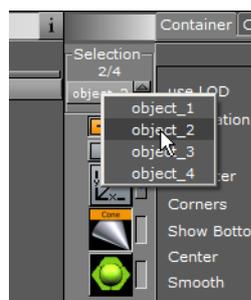
The modified parameter will apply to both selected Containers, which contain the Circle Geometry (object_3 and object_4)

If, in the example above, the Material property is modified, the change will show on all selected Containers in the Scene.

To Select a Container in a Selection

Select a different Container in the selection without losing the multi selection.

- In the **Selection** panel select another Container. Either:
 - Use the up and down arrows, or
 - Click on the Container, as shown in the image below



Tip: Visit www.vizrt.com to see a short video illustrating the Viz Artist 3.7 Multi-selection feature.

4.6.3 Editing Properties in the Scene Editor

Some transformation parameters can be changed directly in the Scene Editor through the manipulation of handles (see [Scene Editor Context Menu](#)).

4.6.4 Transformation Editor

Every Container in a Scene contains transformation properties. The transformation editor holds information about the most basic properties of a Container, such as position and rotation.

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).

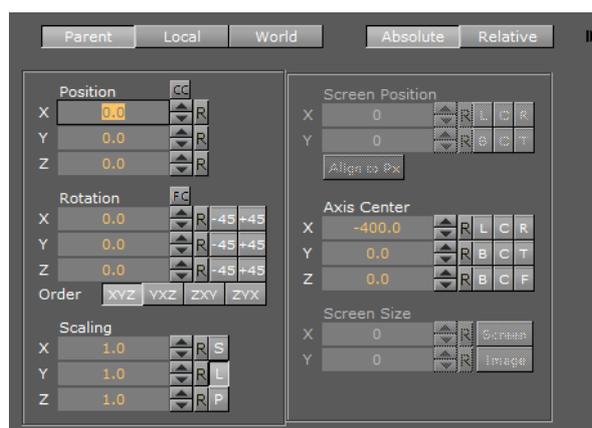
This section contains information on the following topics and procedures:

- [Transformation Editor Properties](#)
- [To Open the Transformation Editor](#)
- [To Stick Transformation Properties to the Container View](#)
- [Transformation Principles](#)
- [Multiple Select Transformation](#)

See Also

- [Container Editor](#)

Transformation Editor Properties



- **Parent / Local / World:** See [Transformation Principles](#)
- **Absolute / Relative:** See [Transformation Principles](#)
- **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
 - **Order:** Set the order of rotation, in which the three angles (X,Y,Z) are applied to the Container (not available when multiple Containers are selected)
 - **FC (Face Camera):** Changes the rotation in such a way that the object faces the camera
- **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 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.
 - **Align to Px (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.
- **Axis Center:** Sets the axis origin of the object. The rotation axis of the object will be visible in the Scene Editor 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.
- **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**

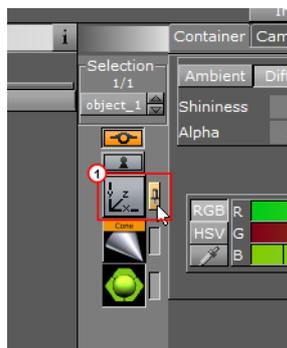
- Click the Expand button to show this information to the right of the transformation editor. This information includes parent Containers plus all *visible* child Containers.
- **Width, Height, and Depth:** Shows the real size of the object in units (not pixels).
- **Containers:** Shows the number of 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.

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 panel.

To Stick Transformation Properties to the Container View

- Click the button (1) next to the transformation editor button in the Container view.



Transformation Principles

Containers can be manipulated in several contexts:

- **Parent/Group:** Refers to changes that can be made to the Container or Containers in its Parent or Group coordinate system.

Note: Multi selected objects: When more than one object is selected Group will show and a bounding box will cover all selected objects (see [Multiple Select Transformation](#)).

- **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.

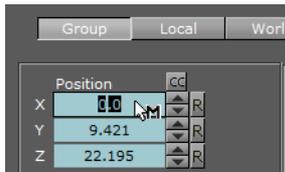
- **Relative:** Refers to changes that can be made to Containers relative to current values.

Multiple Select Transformation

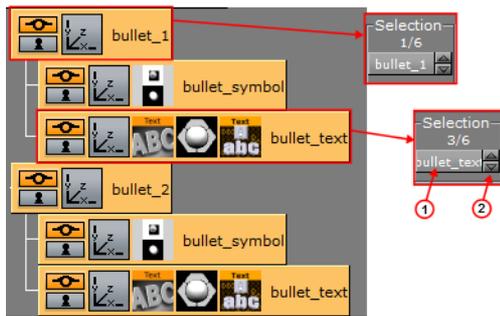
With multiple select transformation, a group of Containers can be moved to a new position, which is determined by parameters of one selected Container in the selection. All Containers keep their relative position to the selected Container.

All changes made in the [Transformation Editor](#) will be on each selected Container:

- **Group** is selected and **M** (multi) will show with the mouse pointer when inside the Transformation editor.



The selected Container and its position in the group is shown in the Selection panel (i.e. 1/6, 3/6).



In the [Container Editor](#) use the up and down arrows (2) to scroll through the selected Containers, or click on the Container name (1) to show a context menu. Change the parameters, as required, for the selected Container.

See Also

- [Multiple Selection Container Properties](#)

4.6.5 Copy 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

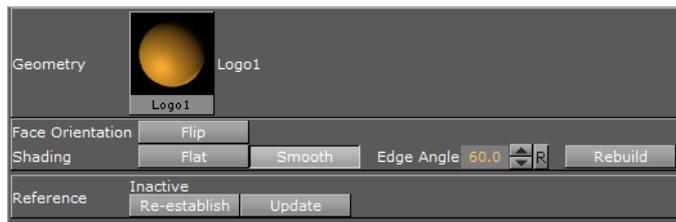
1. Open the [Transformation Editor](#) of the source Container.
2. Select the Container in the Scene Tree that should be given new properties.

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 is done correctly, an icon with the property name will be shown while dragging it.

4.6.6 Geometry Editor

When adding an Geometry to a Scene, a new Container is created. Geometries that have been created with external 3D programs and imported into Viz Artist can be modified in the Geometry editor.

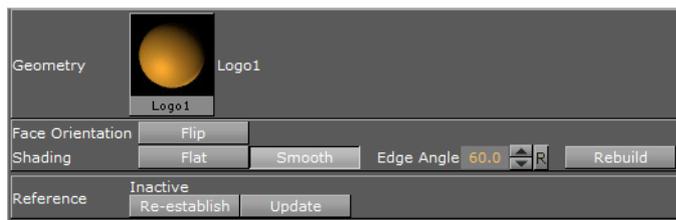


Note: Face Orientation and Shading are only applicable if the reference is deactivated.

This section contains information on the following topic and procedure:

- [Geometry Editor Properties](#)
- [To Open the Geometry Editor](#)

Geometry Editor Properties



- **Geometry:** Shows a thumbnail of the Geometry, and also its name. Click the thumbnail to select the Geometry in the database.
-
- Note:** This can only be done when the reference to the database is active.
-
- **Face Orientation:** After importing Geometries 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 Geometries 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 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:** 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 geometry either on the Container or in the Properties editor.

See Also

- [Add Containers to a Scene](#)
- [Working with Geometry Elements](#)

4.6.7 Text Editor

In the Text editor, text can be added to a Scene and the visualization of the Scene specific text can be modified.



At the top of the Text editor is an input box. The characters in this input box are shown in the Scene Editor. By default, the text equals the font name.

The Text editor contains five tabs:

- [Basic Tab](#)
- [Shadow Tab](#)
- [Clock Tab](#)
- [Characters Tab](#)
- [Special Tab](#)

More advanced modifications can be done 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.

This section contains information on the following procedures and topics:

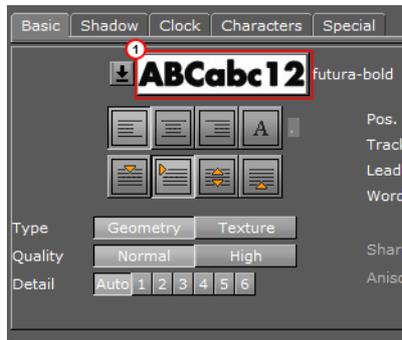
- [To Create a Linked Fontstyle](#)
- [To Modify Text in the Text Editor](#)
- [Basic Tab](#)
- [Shadow Tab](#)
- [Clock Tab](#)
- [Characters Tab](#)
- [Special Tab](#)
- [Special Fonts](#)

To Create a Linked Fontstyle

See [Linked Elements](#).

1. Click on the Basic tab.

- Click on and drag the Fontstyle (1) to the target folder in the Server Area.



To Modify Text in the Text Editor

- Click the text icon either on the Container or in the Container Editor .

Note: When a font is dragged to the Scene, the Text Editor opens automatically, and a new Container will be created with the Font as a Geometry.

- In the Text Editor type text in the text input box. The text is shown in the Scene Editor panel.
- Use the five tabbed panels to modify the text as required:
 - [Basic Tab](#)
 - [Shadow Tab](#)
 - [Clock Tab](#)
 - [Characters Tab](#)
 - [Special Tab](#)

Basic Tab

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



-  Click the arrow to open a list of 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.

-  **Character Orientation:** 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.

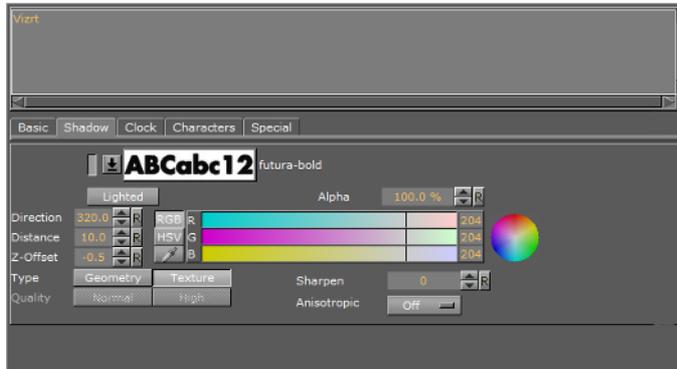
Note: To change the kerning between two individual characters only, place the cursor between those characters, and then use ARROW UP/DOWN while holding <Alt> to increase/decrease the kerning. Press <Alt> and <Shift> to increase the size of the steps.

- **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 Tab

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



From the Shadow panel, shadow can be added to the text. Click the small switch left of the font visualization to enable text shadowing.

Note: All the modifiers that are applied to the text in the Fontstyle Editor are applied to the shadow too.

- Click the arrow to open a list of all available fonts created from the same base font as selected. To use another font, select it from the list. To use a totally different font in the scene, drag a new font from the Server area onto the visualization of the text.
- **Lighted:** Lights the font with the current light settings (see [Light Editor](#)).
- **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.

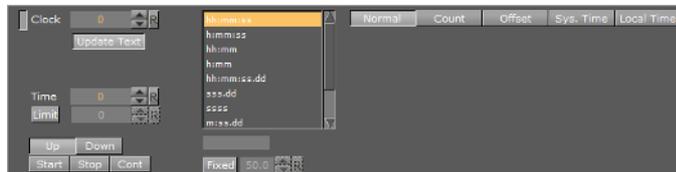
- **Sharpen:** Makes it possible to sharpen the shadow.

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 Tab](#)).

Clock Tab

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, show 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.

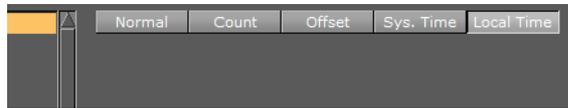
Note: If counting down to 0, the Limit must be set to 0 seconds, or the clock will continue to count below 0. If counting upwards starting at 0, make sure that the Limit is not 0, as the clock will not count at all.

- **Up and Down:** Specifies if the clock should count upwards or countdown.
- **Start, Stop and Continue:** Makes the clock start, stop and continue to count.
- **Format:** Sets the format of the clock display.

Note: More formats (20) can be defined in **Clock Formats** (see [Viz Configuration](#) in the [Viz Engine Administrator Guide](#)).

- **H:** Hours
- **M:** Minutes
- **S:** Seconds
- **D:** 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:** Set 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: 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 required 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
- **Local Time:** Uses the local time

Characters Tab

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

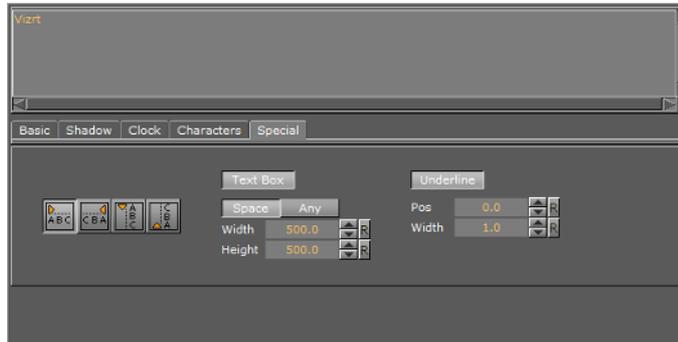


The Characters panel shows the character table in the fonts library. As an alternative to using the keyboard, the symbols from the table can be dragged onto the input box

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 Tab

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



-     **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 shows as an orange outline in the Scene Editor if enabling the Bounding Box option in the [Scene Editor](#).

Note: The Text Box option is only available if the Text Orientation is set from left to right. Without the text box, the text is shown 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: There is no hyphenation function. Only the text that fits within the text box definition is shown in the Scene Editor.

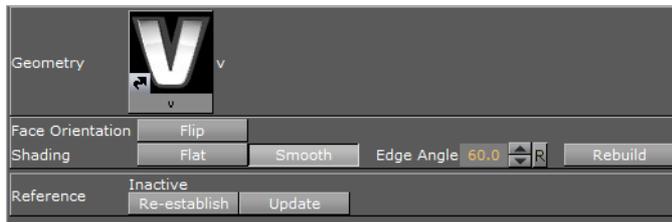
- **Width:** Defines the width of the text box.
- **Height:** Defines the height of the text box.
- **Underline:** Click the Underline button to enable the Underline option, which makes it possible to underline the text.
 - **Pos:** Defines the position of the underline, from -20 to 20.
 - **Width:**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

Special Fonts

Special options apply for multibyte and geometry fonts:

- **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.



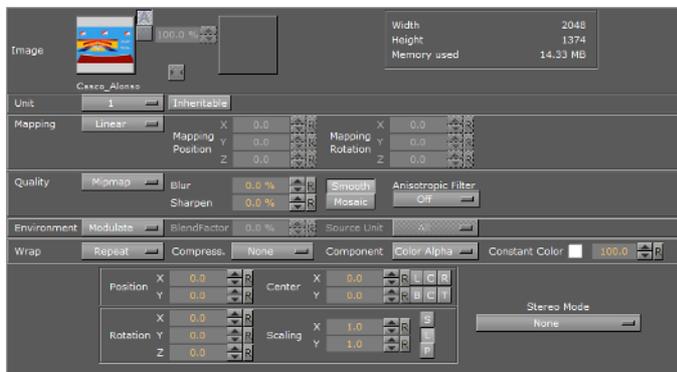
Note: Be aware that it is a one way operation. It is not possible to merge characters into text again, so by performing the splitting operation, the text cannot be edited anymore.

See Also

- [Add Containers to a Scene](#)
- [Working with Fontstyle Elements](#)
- [Fontstyle Editor](#)

4.6.8 Texture Editor

To apply a texture to a Container drag a texture (e.g. an image) from the Server area to a Container. In the Container an icon (a thumbnail of the texture) will show. If a texture is dragged to the Scene Tree area it will create its own Container, Geometries can be added later on, or the texture can be left as it is in the Scene.



Tip: Modifications to the texture in the Texture editor only apply to the texture in the selected Container.



This section contains information on the following topic and procedure:

- [Texture Editor Properties](#)
- [To Open the Texture Editor](#)

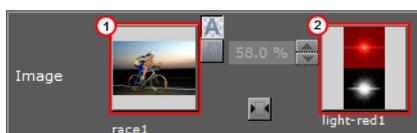
Texture Editor Properties

This section contains the following topics:

Image Properties	Wrap Properties
Unit Properties	Compress. Properties
Mapping Properties	Component and Constant Color Properties
Quality Properties	Position Properties
Environment Properties	Stereo Mode Properties

Image Properties

Shows the added texture:



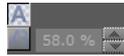
Note: Two image textures can be used in one Container

Note: Only one Media Asset texture can be used.

Tip: Click on a Media Asset image to open the editor for the Media Asset.

- **Default Texture (1):** Shows a thumbnail of the primary texture (the same thumbnail that is shown on the Container). To change the texture drag another texture to the drop zone. If only one texture is used the other functions in the Image panel are inactive.

- **Alpha Texture (2):** Shows the thumbnail of a secondary (alpha) texture, if any. This texture is used as an alpha channel for the Default texture. When two textures are used the Image panel Alpha and Combine functions are active.
- **Alpha Buttons:** When a second image is added the Alpha buttons are active. The top button is the default button, the two images are shown as an alpha image. Click the lower button to view the two images blended, use the slider to blend as required.



- **Combine:** When a second image is added, click this button to create a new single image from the two images. The new image will be created in the currently open folder in the Server area.

Tip: To save the new image in another folder, move to a different folder in the Server area before the Combine button is clicked.

- **Information:** Shows the original width, height, and file size of the primary texture.

Tip: To merge the two textures, 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 Properties

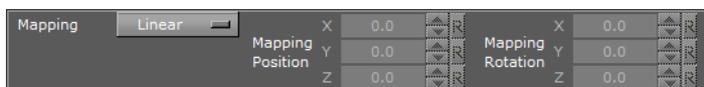
Texture units take a texture and map them to a piece of geometry:



- **1-4:** If the graphics card supports it, 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 Properties

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 plugins 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 shown 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 were 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.
- **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.

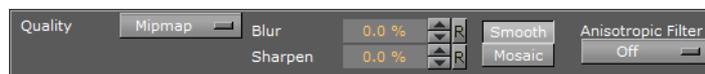
Note: To use the Cube mapping method for texture it is not possible to use the same texture on another object with a different mapping method than Cube.

- **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.

Quality Properties

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 shows 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 Properties

The environment parameters specify the way the texture RGBA information is combined with the object RGBA information:



- **Environment:** Select from:
 - **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.
 - **Interpol.:** 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 Properties

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

Compress. Properties

Texture compression can be used to save texture memory on the graphics card:

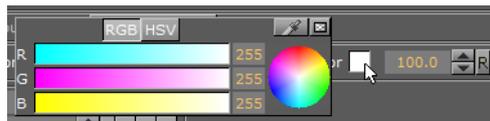


- **None:** No compression
- **DXT1:** Compresses to an RGB image. This format has a 6:1 compression ratio (DXT1 does not store alpha)
- **DXT3:** Compresses to an RGBA image with a 4:1 compression ratio
- **DXT5:** Compresses to an alternative RGBA image that treats the alpha channel different than DXT3 (alpha looks smoother). DXT5 has the same compression ratio as DXT3

Component and Constant Color Properties



- **Component:** Select from:
 - **Color:** Only the RGB color of the texture will be used. Set the alpha to a constant value in item 1
 - **Alpha:** Only the alpha of the texture will be used. Set the RGB color of the texture to a constant color specified in item 1
 - **Color Alpha:** Both the alpha and the RGB color of the texture will be used (default setting)
- **Constant Color:** Set the constant RGB color of the texture when **Alpha** is selected



Position Properties



- **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.

Stereo Mode Properties

Select the correct stereo mode for the inserted image:



- **None:** No Stereo mode.
- **Config:** Images are drawn as set in the [Viz Configuration](#)
- **Over Under Left Top** or **Over Under Right Top:** Two images are stored, one over (top) and the other under (bottom), **left or right eye (top) first**. The image height will be halved, providing half resolution.
- **Side by Side Left** or **Side by Side Right:** Two images are stored, side by side, left or right eye first. The image width will be halved, providing half resolution.

To Open the Texture Editor

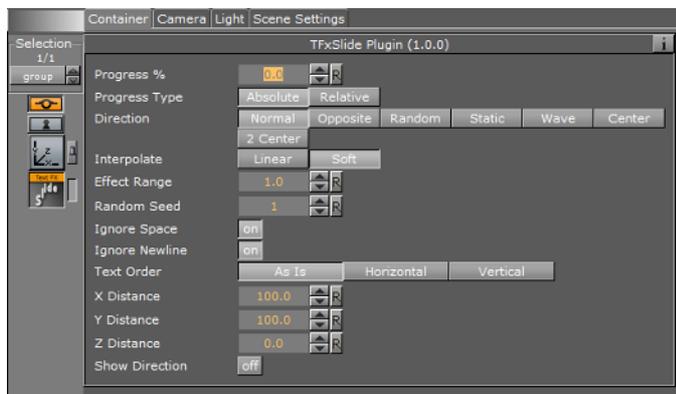
- Click the icon representing the texture either on the Container or in the Properties editor.

See Also

- [Add Containers to a Scene](#)
- [Working with Image Elements](#)
- [Image Editor](#)

4.6.9 Plugin Editor

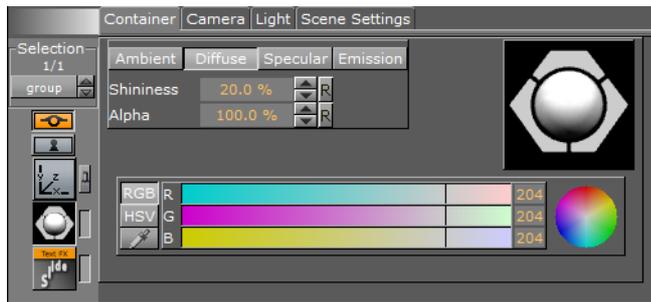
When a Geometry, Container or Shader Plugin is added to a Container its properties will be available in the Properties Panel. Each Plugin may or may not have properties that can be modified.



Click on the information icon () to show details about the selected plugin, if available. Additional information may also be available in the plugin section of this User Guide, see these Plugin sections for more information:

- [Geometry Plugins](#)
- [Container Plugins](#)
- [Shader Plugins](#)

4.6.10 Material Editor



When a Material has been added to a Container, the Container-specific Material can be modified in the Material Editor that opens in the Properties panel. Only the Container-specific Material will be modified, the original Material in the Graphic Hub database will not be affected (see [Working with Material Elements](#)).

This section contains information on the following procedures:

- [To Modify a Scene Specific Material](#)
- [To Save the Container Material](#)

To Modify a Scene Specific Material

1. Click the Material icon in the Container (1) or Properties panel (2). The Properties panel Material Editor (3) is shown.

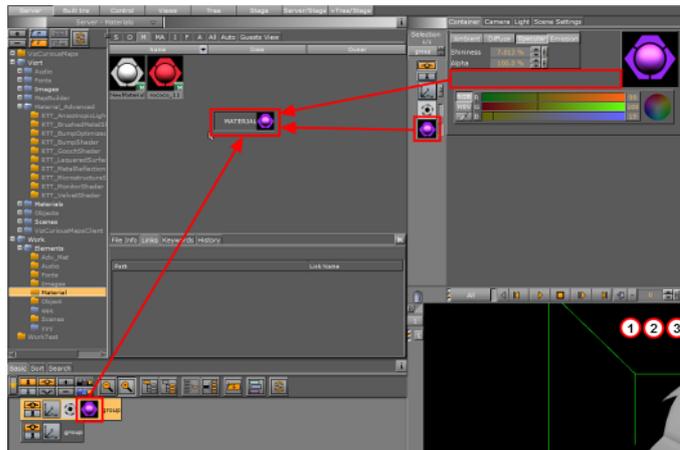


2. Adjust the [Material Color Properties and Schemes](#) as necessary.

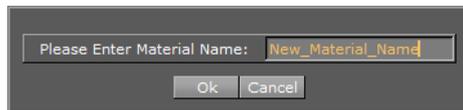
To Save the Container Material

1. Move the modified Material to the Server Area:
 - Drag the material icon from the Container to the Server area, or
 - Drag the material icon from the Properties area to the Server area, or

- Click in the Material properties panel and drag the material icon to the Server area



2. Enter a name for the material.



3. Click **OK**.

See Also

- [Material Editor](#)
- [Light Editor](#)
- [Common Element Procedures](#)

5 Manage Work and Graphic Hub

This section details how to work with Elements, add, edit and search for, and how to add Built ins to a Scene.

This section contains information on the following topics:

- [Graphic Hub](#)
- [Server Area](#)
- [Folder Tree](#)
- [What are Elements](#)
- [Element Area](#)
- [Working with Elements](#)
- [Add Properties to Containers and Scenes](#)
- [Assign Keywords to Elements](#)
- [Image Editor](#)
- [Fontstyle Editor](#)
- [Material Editor](#)
- [Element Search](#)
- [SmartView Search](#)
- [Background Loading](#)
- [Built Ins](#)

5.1 Graphic Hub

The Graphic Hub is the database solution where all Viz Artist Elements are stored. Elements can be either:

- [Scenes](#)
- [Geometries](#)
- [Images](#)
- [Materials](#)
- [Materials Advanced](#)
- [Fonts](#)
- [Audio Clips](#)

To start Viz Artist successfully, the user must log in to a running Graphic 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 Graphic Hub data directory as file objects. The database itself manages the items in terms of properties and Universally Unique Identifiers (UUIDs).

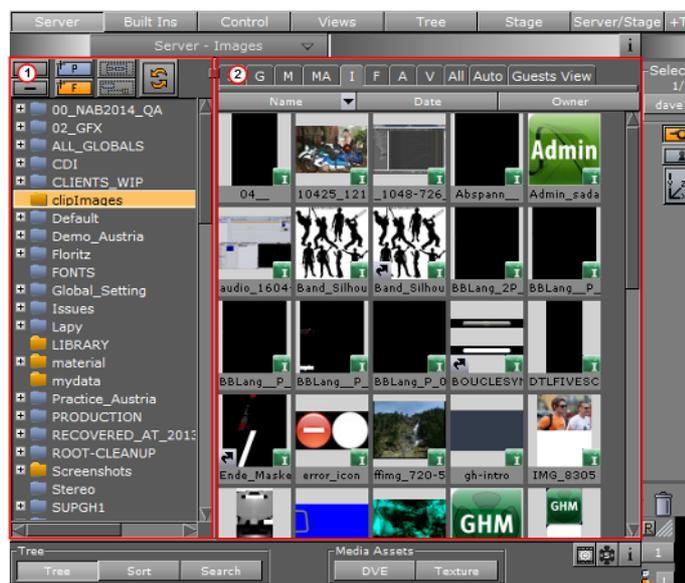
Clients can work with and organize items through the Viz Artist GUI in a logical folder structure provided by the database. Although each item can be listed and linked in more than one folder, it will be in the database only once.

See Also

- [What are Elements](#)
- [Graphic Hub User Guide](#)

5.2 Server Area

The Server area shows an overview of all available Elements in the Graphic Hub (GH) database.



The Server area is divided in two:

- [Folder Tree](#) (1): Shows the logical tree of folders in the GH database.
- [Element Area](#) (2): Shows the Elements in the selected folder.

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

To show the Server area

- Click **Server** in the **Main menu**.
This will show the Server Area together with the scene tree and Scene Editor.

or

- Click **Server/Stage** in the **Main menu**.
This will show the Server Area together with the Stage and Scene Editor.

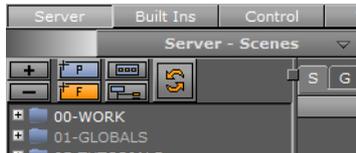
or

- Click **+Tree/Stage** in the **Main menu**.

This will show the Server Area together with the stage, scene tree, and Scene Editor.

5.2.1 Server Tree Bar

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



The Tree bar contains the following options:

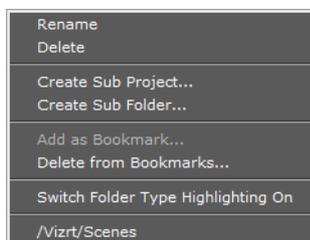
-  **Expand:** Expands the currently selected branch and all its sub-branches (see [To expand or collapse a folder branch](#)).
-  **Collapse:** Collapses the currently selected branch and all its sub-branches (see [To Collapse a Folder Branch Including all its Sub-Branches](#)).
-  **Add New Project (folder):** Makes it possible to add a new project folder to the Server Tree (see [To Create a New Folder](#)).
-  **Add New Folder (regular):** Makes it possible to add a new regular folder to the Server Tree (see [To Create a New Folder](#)).
-  **Restrict to Current Project Folder:** Restricts to working project folder by narrowing down the server view so that only the selected project folder with its sub-folders are shown.

Note: This option is only available if a project is selected in the Server Tree.

-  **Show All Project Folders:** If the tree is narrowed down to only one project folder, the tree is expanded to show all folders.
-  **Refresh:** Refreshes the Server Tree and adds updates.

5.2.2 Server Tree Menu

Right-click a folder in the Server Tree to open the Server Tree context menu.



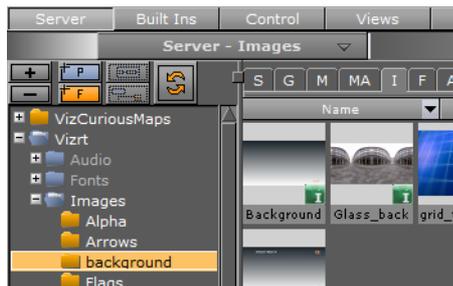
- **Rename:** Change the name of the selected folder (see [To Rename a Folder](#)).
- **Delete:** Deletes the selected folder (see [To Delete a Folder](#)).
- **Create Sub Project...:** Create a sub-project folder under the currently selected project folder (see [To Create a New Folder](#)).

Note: It is not possible to add a sub-project folder to a regular folder (see [Folders](#)).

- **Create Sub Folder...:** Create a sub-folder under the currently selected folder (see [To Create a New Folder](#))
- **Add as Bookmark...:** Adds the selected folder to the list of bookmarks (see [To Bookmark a Folder](#)). Bookmarks are added to the Server Tree menu at the bottom. Click a bookmark in the list to open the corresponding folder
- **Delete from Bookmarks...:** Deletes the selected folder from the list of bookmarks (see [To Delete a Folder Bookmark](#))
- **Switch Folder Type Highlighting <On/Off>:** Set highlighting of folders to On or Off. For example, if a folder (and sub-folders) contain a Scene it will be highlighted when the Scene tab is selected. The default setting of this feature can be set in **User Interface** of the [Viz Configuration](#) (see the [Viz Engine Administrator Guide](#))).
- **<Bookmark(s)>:** Show a list of all current bookmarked folders. Click on a folder bookmark to go to that folder.

5.3 Folder Tree

The Server area folder tree shows a structured storage of Elements, and is located at the left side of the Server area.



Viz Artist can be configured to highlight only the folders in the tree that contain a selected Element type (see **Folder Type Highlighting** in [User Interface](#) (see the [Viz Engine Administrator Guide](#))). The labels of folders, and their sub-folders, that contain none of the selected Element type are grayed out.

Right-click on any folder in the Server Tree to open the [Server Tree Menu](#) .

This section contains information on the following topics:

- [Folders](#)
- [What are Elements](#)
- [Server Tree Menu](#)

5.3.1 Folders

The Server Tree can contain two types of folders:

- Regular folders
- Project folders

The difference between a regular folder and a project folder is that by defining a working project, the tree can be collapsed so that only the currently selected project and its sub-folders and sub-projects folders are shown.

Both types of folders can be dragged around after being added to the Server Tree.

This section contains information on the following topics and procedures:

- [Folder Position in the Server Tree](#)
- [To Create a New Folder](#)
- [To Rename a Folder](#)
- [To Move a Folder](#)
- [To Delete a Folder](#)
- [To expand or collapse a folder branch](#)
- [To Expand a Folder Branch Including all its Sub-Branches](#)
- [To Collapse a Folder Branch Including all its Sub-Branches](#)
- [To Bookmark a Folder](#)
- [To Delete a Folder Bookmark](#)

Folder Position in the Server Tree

The position in the Server Tree where a moved or new folder is placed depends on where in the Server Tree they are dragged and released.



- Release the **Add New <Folder/Project>** button (1) on the left side will create a folder at the same level as the folder highlighted (2).
- Release the **Add New <Folder/Project>** button on the right side will create a sub-folder, of the folder highlighted (3).



To Create a New Folder

Use to create new **Regular** folders or new **Project** folders (see [Folders](#)).

Note: A new project folder cannot be created in a folder.

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

1. In the [Server Tree Bar](#), drag the  (Add New <Folder/Project>) button, to the Server Tree.

2. Create a new folder in the required position in the Server Tree (see [Folder Position in the Server Tree](#)):
 - **Folder level:** Drop the **Button** over a current folder, on the left side of the Server Tree.
 - **Folder Sub level:** Drop the **Button** on a current folder, on the right side of the Server Tree.
3. In the dialog box that opens, type a descriptive name for the folder.
4. Click the **Ok** button.

Or:

1. Click a folder in the Server Tree.
2. In the [Server Tree Bar](#), click the **Add New <Folder/Project>** button.
3. In the dialog box that opens, type a descriptive name for the new folder type.
4. Click the **Ok** button.

Note: If the selected folder is a regular folder and a new project folder is to be created, the new project folder will be created at the next available place up in the Server Tree.

Or:

1. Right-click a folder in the Server Tree.
2. In the [Server Tree Menu](#) that opens, select:
 - **Create Sub Folder...**, or
 - **Create Sub Project...**

Note: Create Sub Project will be only available if a project folder can be created in that position.

3. In the dialog box that opens, type a descriptive name for the new Sub-folder type.
4. Click the **Ok** button.

To Rename a Folder

1. Select the folder to be renamed.
2. Open the rename text box:
 - Right-click a folder in the Server Tree. In the [Server Tree Menu](#) that opens, click on **Rename**, or
 - Press <F2>.
3. In the text box that opens, type a new descriptive name
4. Press <Enter>.

To Move a Folder

1. Drag a folder to its new location.
2. Drop the folder, at its new location, to the left or the right of the Sever area Tree (see [Folder Position in the Server Tree](#)).

Note: A project folder cannot be moved to a folder or a sub-folder.

Note: If a project or folder cannot be dragged and dropped, check the **Allow Folder Move** settings (see **Global Properties** in **Database** of [Viz Configuration](#) in the [Viz Engine Administrator Guide](#))

To Delete a Folder

1. Right-click a folder in the Server Tree.
2. In the [Server Tree Menu](#) that opens, select **Delete**.
3. Click **Yes** to confirm.

Note: If the folder is added as a bookmark, the bookmark will also be deleted (see [To Bookmark a Folder](#)).

Or:

1. Drag a folder to the Trash Can  .
2. Click **Yes** to confirm.

To expand or collapse a folder branch

- Click the   (Expand or Collapse) icon to the left of the folder name, or
- Double click a folder.

To Expand a Folder Branch Including all its Sub-Branches

1. Select a folder in the Server Tree.
2. Click the  (Expand) button in the [Server Tree Bar](#).

To Collapse a Folder Branch Including all its Sub-Branches

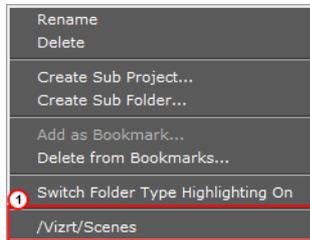
1. Select a folder in the Server Tree.
2. Click the  (Collapse) button in the [Server Tree Bar](#).

To Collapse all Folder Branches

- Double-click the  (Collapse) button in the [Server Tree Bar](#).

To Bookmark a Folder

For User convenience folders can be bookmarked to make it easier to find often used folders. Bookmarked folders are added to the Server Tree menu (1). Click a bookmark in the list to open the corresponding folder.



1. Right-click a folder in the Server Tree.
2. In the [Server Tree Menu](#) that opens, select **Add as Bookmark...**

To Delete a Folder Bookmark

1. Right-click a folder in the Server Tree.
2. In the [Server Tree Menu](#) that opens, select **Delete from Bookmarks...**

5.4 What are Elements

Elements are properties that are used to build up a Scene. Elements can also be complete Scenes.

The [Server Area](#) shows the Elements that are available, for the folder selected, in the Server [Folder Tree](#).

This section contains information on the following topics:

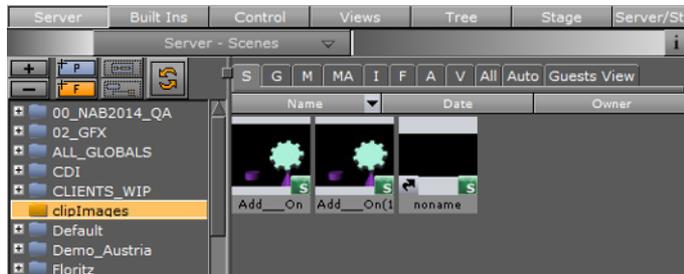
- [Scenes](#)
- [Geometries](#)
- [Materials](#)
- [Materials Advanced](#)
- [Images](#)
- [Fonts](#)
- [Audio Clips](#)

See Also

- [Add Properties to Containers and Scenes](#)

5.4.1 Scenes

Scenes are root Elements that hold other Elements. Scenes are stored in the Graphic Hub database and are available through the Server area.



If a Scene is opened, the [Scene Tree](#) area shows the logical visualization of all the objects within the Scene. The Scene Tree consists of Containers that hold Elements and properties, organized in a hierarchy and grouped in logical divisions. The Elements and properties can be Geometries, Images, Materials, Fonts, transparency functions, key functions, and many more (see [What are Elements](#) and [Built Ins](#)).

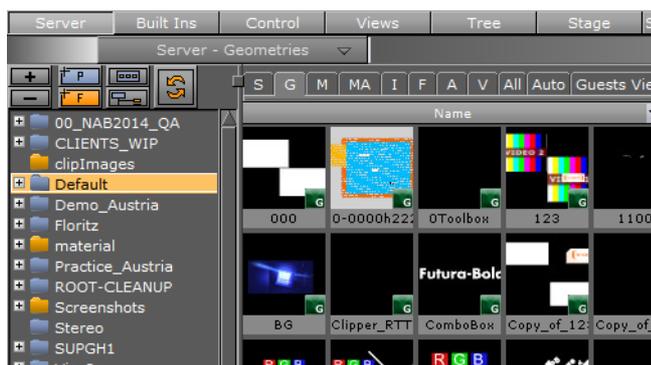
Note: Viz supports the import of scenes from numerous third party file formats (see [Import Files and Archives](#))

See Also

- [Working with Scene Elements](#)
- [Common Element Procedures](#)
- [Built Ins](#)
- [Add Properties to Containers and Scenes](#)

5.4.2 Geometries

Geometries are stored in the Graphic Hub database and are available through the [Server Area](#).



Geometries can be created in Viz Artist or imported from an external modeling program (see [Import Files and Archives](#)).

Viz Artist also has a set of Geometries that are built-in [Geometry Plugins](#). These geometries are common plugins that can be used in Scenes.

An Geometry can be added to a Scene as a standalone Container or as a Geometry property of a Container. Drag the object from the [Element List](#) (or [Favorites Bar](#)) to the Scene Tree, Scene Editor or add to a Container in the Scene Tree (see [Add Containers to a Scene](#)).

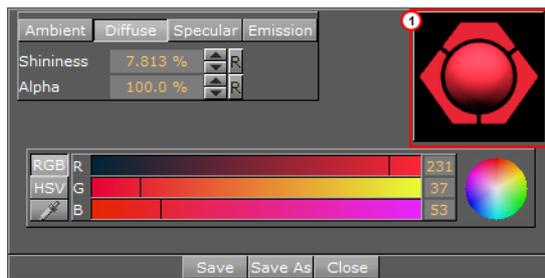
See Also

- [Working with Geometry Elements](#)
- Properties panel [Geometry Editor](#)
- [Common Element Procedures](#)
- [Built Ins](#)
- [Add Properties to Containers and Scenes](#)

5.4.3 Materials

Materials hold colors, customized colors and light effects which can be applied to objects in a Scene.

The Materials icon (1) in the Element area 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.



When a Material has been added to a Scene, the Scene-specific Material can be modified in the Material editor that opens in the Properties panel. Only the Scene specific Material will be modified, the original Material in the Graphic Hub database will not be affected (see [Working with Material Elements](#)).



See Also

- [Working with Material Elements](#)
- Properties panel [Material Editor](#)
- [Material Editor](#)
- [Common Element Procedures](#)
- [Lights](#)

- [Add Properties to Containers and Scenes](#)

5.4.4 Materials Advanced

An Material Advanced is an Element that can combine a Shader plugin, an Image and a Material Element. A combination minimum of two items are required to create the Materials Advanced.

The Materials Advanced Element is stored in the Graphic Hub database and are available in the Server area [Element List](#).



To add a Material Advanced as a Container item, drag the Material Advanced from the [Element List](#) (or [Favorites Bar](#)) to a Container in the Scene Tree.

See Also

- [Working with Material Advanced Elements](#)
- [Common Element Procedures](#)
- [Built Ins](#)
- [Add Properties to Containers and Scenes](#)

5.4.5 Images

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



An Image can be added to a Scene as a standalone Container, or as a Container item. Drag the Image from the [Element List](#) (or [Favorites Bar](#)) to the Scene Tree, Scene Editor or add to a Container in the Scene Tree (see [Add Containers to a Scene](#)).

See Also

- [Working with Image Elements](#)
- [Common Element Procedures](#)
- Properties panel [Texture Editor](#)

- [Image Editor](#)
- [Add Properties to Containers and Scenes](#)

5.4.6 Fonts

Fonts are complete character sets within a unique style. The Fonts are stored in the Graphic Hub database and are available through the Server area.



A Fontstyle can be added to a Scene as a standalone Container or as a Container item. Drag a Fontstyle from the [Element List](#) (or [Favorites Bar](#)) to the Scene Tree, Scene Editor or add to a Container in the Scene Tree (see [Add Containers to a Scene](#)).

See Also

- [Working with Fontstyle Elements](#)
- [Common Element Procedures](#)
- Properties panel [Text Editor](#)
- [Fontstyle Editor](#)
- [Add Properties to Containers and Scenes](#)

5.4.7 Audio Clips

Viz Artist supports Audio clips in formats WAVE, MP3, and Vorbis (ogg), which are stored in the GH database and are available through the Server area.



The Audio clips can be added to Scenes and, or to the animation of a Scene. See:

- [To Add an Audio Clip to a Scene \(Context Menu\)](#)
- [Animate Audio](#)

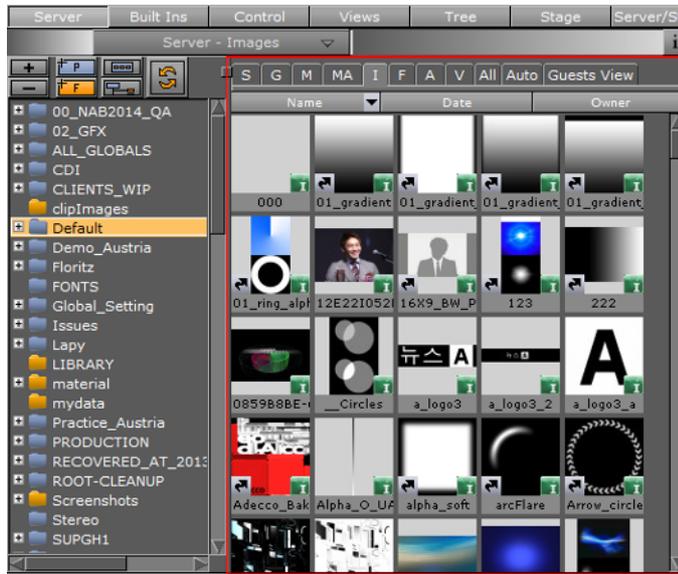
See Also

- [Working with Audio \(Clips\) Elements](#)
- [Common Element Procedures](#)

- [Add Properties to Containers and Scenes](#)

5.5 Element Area

The Element area shows the Elements that are available for the folder selected in the Server Tree.



This section contains information on the following topics:

- [Element List](#)
- [Element Context Menu](#)
- [Content Switcher](#)
- [Server Menu](#)
- [Properties Panel](#)

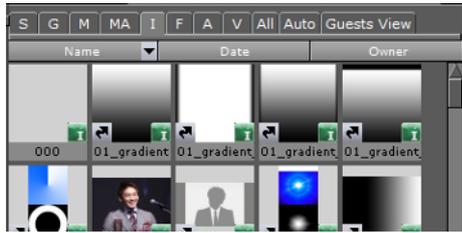
5.5.1 Element List

The Element list shows all Elements in the selected folder.

This section contains information on the following procedures:

- [Default View](#)
- [Details View](#)
- [To Show Elements with Details](#)
- [To Show Elements Details With or Without Icons](#)
- [To Show the Properties Panel](#)
- [To Sort Elements](#)

Default View

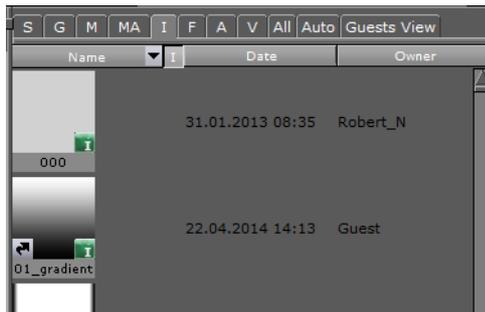


Elements are, by default, shown with a thumbnail and their name, and sorted alphabetically. To show additional information about the Elements see [To Show Elements with Details](#) and [To Show the Properties Panel](#) . Right-click an item in the [Element List](#) to open the [Element Context Menu](#) .

Tip: Holding the pointer over a thumbnail will show the whole Element name as a tool tip.

Details View

The Elements in the list can be shown with information about their owner and creation date.



To Show Elements with Details

- Click on **Detail** from the [Server Menu](#), or
- Hold the pointer over the Server area and press the keyboard shortcut <Ctrl+Y>.

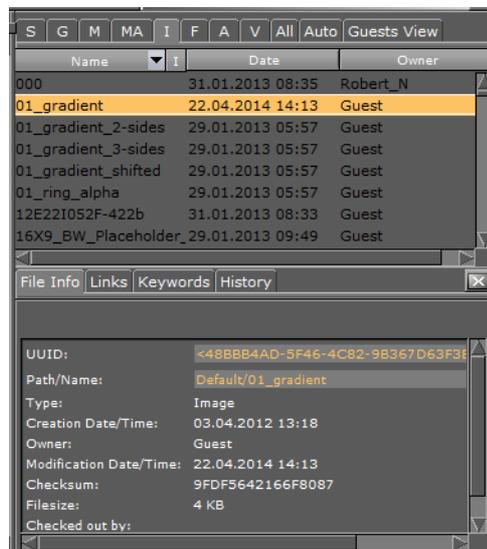
To Show Elements Details With or Without Icons



- When in Detail view toggle click the **I** (1) in the Elements list header to view Elements icons or not. When no icon is selected the Element name shows.

Element Properties

The [Properties Panel](#) will show detailed information about file, links, keywords, and history for the selected Element in the list.



To Show the Properties Panel

- Select **Show Properties** from the [Server Menu](#), or
- Hold the pointer over the Server area and press the keyboard shortcut <Ctrl+P>, or
- Right-click an Element in the [Element List](#). In the menu that opens, select **Show Properties...**

See Also

- [Properties Panel](#)

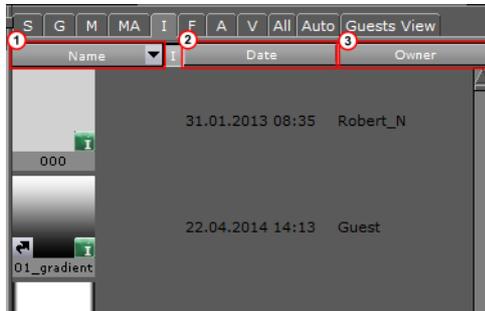
To Sort Elements

Elements can be sorted in [Default View](#) or [Details View](#). Click on a column heading to:

- **Name (1):** Sort Elements by name.
- **Date (2):** Sort Elements by the last date and time when the Element was modified.
- **Owner (3):** Sort the Elements by name of the Element owner.

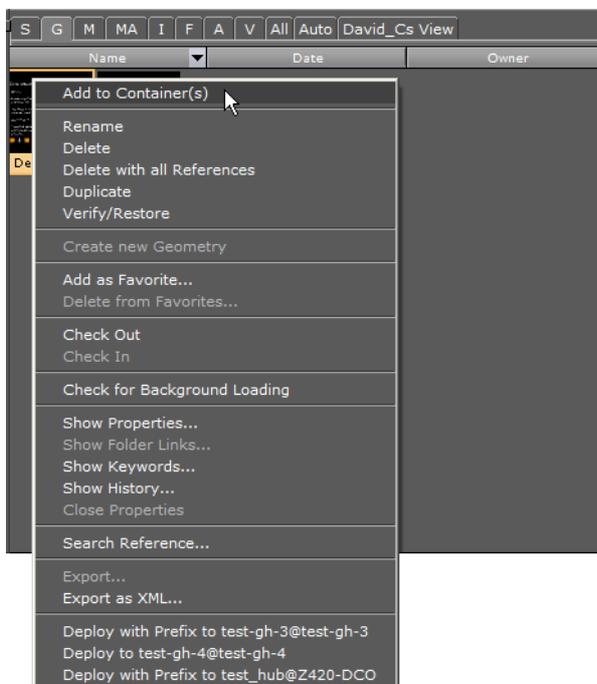
To sort the Element list, click on either the Name, Date or Owner columns. Click on the arrow (shows when a column is clicked on) to sort ascending or descending

Note: By default Elements are sorted alphabetically by name.



5.5.2 Element Context Menu

Right-click an Element in the [Element List](#) to open its context menu, which contains the following options:



- **Add to Container(s):** Add the selected Element to a highlighted Container(s).

Note: Not available for Scene or Audio Elements.

- **Open <Scene name>:** Opens the selected Scene.

Note: Available for Scene Elements only.

- **Rename:** Change the name of the selected Element (see [To Rename an Element](#)).
- **Delete:** Delete the selected Element (see [To Delete an Element](#)).

- **Delete with all references:** Delete the selected Element and all its referenced Elements. Referenced Elements are not deleted if they are referenced by other Elements.
- **Duplicate:** Creates an independent copy of the selected Element. The copy will be placed in the same folder as the original Element (see [To Create an Element Copy in a New Folder](#)).
- **Verify/Restore:** Locates, verifies and attempts to correct a corrupted Element.
- **Switch to Small/Large Icons:** Toggles between small or large Element icons in the [Element List](#).
- **Create New [Element]:** Create a new Element:
 - In the dialog box that opens, type a descriptive name.
 - Click the **Ok** button.

Note: The 'Create new' option is only available for Scenes and Materials.

- **Add as Favorite...:** Adds the selected Element to the [Favorites Bar](#) in the Scene Tree area.

Note: It is not possible to add Scenes or Audio Elements as favorites.

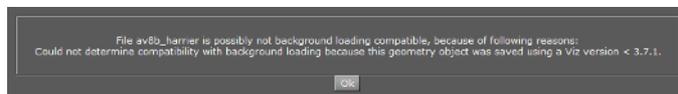
- **Delete from Favorites:** Deletes the selected Element from the [Favorites Bar](#).
- **Check Out:** Checks out the selected Element, so that only the user that checks out the file can modify it (see [To Check Out Elements](#)).

Tip: Other users can see the name of the user that has checked out the file from the [Properties Panel](#).

- **Check In:** Checks in the selected Element, so that all users can modify it (see [To Check In Elements](#)).

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

- **Check for Background Loading:** Click to check if the selected Element is suitable for [Background Loading](#). A message box opens, and shows if the Element is suitable or not. If not suitable, the message gives reasons why.



- **Show Properties...:** Shows the [Properties Panel](#) at the bottom of the Element area, with the File Info tab selected.
- **Show Folder Links...:** Shows the [Properties Panel](#) 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 (see [To Create a Linked Element](#)).

- **Show Keywords...:** Shows the [Properties Panel](#) at the bottom of the Element area, with the Keywords tab selected.
- **Show History...:** Shows the [Properties Panel](#) at the bottom of the Element area, with the History tab selected.
- **Close Properties:** Closes the [Properties Panel](#) .
- **Search Reference...:** Opens the Search window and searches the database for other Elements which are referenced to the selected Element (see [Reference Search](#)).

Note: It is not possible to perform a reference search for materials, as materials are not referenced. Material information is held inside the Scene.

- **Export...:** Selected Image and Audio clip Elements can be exported (see [To Export an Image or Images](#) and [To Export an Audio clip](#)).

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

- **Export as XML...:** Export an XML representation of a Scene and other objects into an XML file. XML files can be used to compare two versions of a Scene using a file comparison utility (aka diff tool).
- **Deploy...:** Deploy an Element, with all its necessary data, from one server to another. For example, if a Scene is deployed, all items referenced by the Scene (Images, Geometries, etc.), will also be deployed (see [To Deploy Selected Elements](#) and [To Deploy Single Elements](#)).

Note: Available when Deploy is enabled and a valid Deploy server is entered in **Deploy Properties** (see **Database** in [Viz Configuration](#) section of the [Viz Engine Administrator Guide](#)).

5.5.3 Content Switcher

Folders can contain various types of Elements.

To view only a selected type of Element, contained in a folder, click an Element type tab from the Content Switcher (1). If the **All** tab is selected all Elements in the selected folder, regardless of type, will be shown.



- **S:** Shows only Scene Elements.
- **G:** Shows only Geometry Elements
- **M:** Shows only Material Elements
- **MA:** Shows only Material Advanced Elements
- **I:** Shows only Image Elements
- **F:** Shows only Fontstyle Elements
- **A:** Shows only Audio Elements

- **All:** Shows all Element types
- **Auto:** Shows the Element type with the most instances in the selected folder. For example, if there are five materials in a folder, but only one instance of each of the other Element types, the materials will be shown in the [Element List](#).
- **<Username>s View:** Personalized view to enter and save free text searches for later use (see [SmartView Search](#)).

This section contains information on the following topics:

- [Content Switcher Menu](#)
- [Element Type History](#)
- [To Show the Element Type History List](#)
- [To Show the Element type Last Used Folder](#)

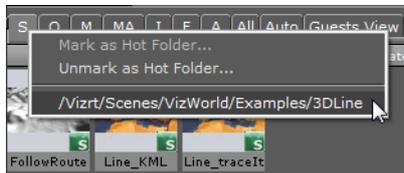
See Also

- [Working with Elements](#)

Content Switcher Menu

A right-click on one of the tabs in the [Content Switcher](#) opens the Content Switcher menu. At the bottom of the menu, below the command separator, is a list of hot folders for the selected Element type. Hot folders are Element specific favorite folders that the User often uses. Selecting a hot folder in the list will open the corresponding folder and switch to the selected Element type.

The Content switcher menu contains the following options:



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

Note: Each Element type has a separate list of Hot Folders.

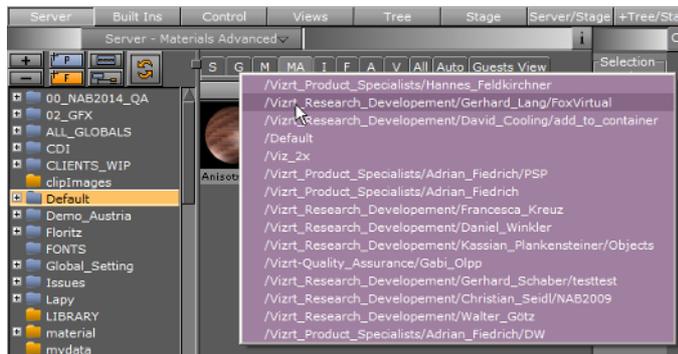
- **Unmark as Hot Folder...:** Removes the currently selected folder from the list of favorites.

Element Type History

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

Note: A 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 Element Type History List



1. Hold the pointer over one of the tabs in the [Content Switcher](#).
2. Press and hold <Ctrl>, and right-click.
3. Move through the list and click to show the selected folder.

Note: The Element list is shown in last used order.

To Show the Element type Last Used Folder

1. Hold the pointer over one of the tabs in the [Content Switcher](#).
2. Press and hold <Ctrl>, and left-click.
3. The content of the last used folder for this Element type will be shown.

5.5.4 Server Menu

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



- **Search:** Opens the Search window (see [Element Search](#)).
- **Cut:** Cuts the currently selected Element, so that it can be pasted in another folder (see [To Move an Element to a New Folder](#)).
- **Copy:** Copies the currently selected Element, so that it can be pasted in another folder (see [To Create an Element Copy in a New Folder](#)).

- **Paste-Link:** Pastes a 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 (see [Linked Elements](#)).
- **Paste-Dup:** Pastes a copied Element as an independent file (see [To Create an Element Copy in a New Folder](#)).
- **Show Properties:** Shows the [Properties Panel](#) at the bottom of the Element area, with the File Info tab selected (see [To Show the Properties Panel](#)).
- **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 shown (see [To Show Elements with Details](#)).
- **Create:** Makes it possible to add a new Element (see [Working with Elements](#)).

Note: The create option is only available for Scenes and Materials.

- **Export:** Makes it possible to export an Element (see [To Export an Image or Images](#) and [To Export an Audio clip](#)).

Note: The export option is only available for Images and Audio elements.

- **Export as XML:** 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.
- **Check Out Files:** Checks out the selected Elements, so that only the User who checked out the files can modify them (see [To Check Out Elements](#)).

Tip: Other users can see the name of the user that has checked out the files from the [Properties Panel](#) .

- **Check In Files:** Checks in the selected Elements, so that all Users can modify them (see [To Check In Elements](#)).

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

- **Reconnect:** Re-establishes the connection to [Graphic Hub](#).
- **Change User...:** Change the current User.

See Also

- [Lights](#)
- [Server Area Shortcuts](#)

5.5.5 Properties Panel

The Properties panel shows detailed information about the selected Element in the [Element List](#). It is shown below the Element list.

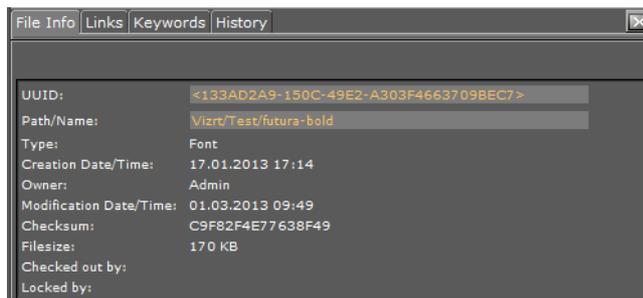
This section contains the following topics and procedures:

- [File Info Tab](#)

- [Links Tab](#)
- [Keywords Tab](#)
- [History Tab](#)
- [Keywords Tab](#)
- [To Show the Properties Panel](#)
- [To Show the Properties Panel with a Selected Tab Open](#)
- [To Close the Properties Panel](#)

File Info Tab

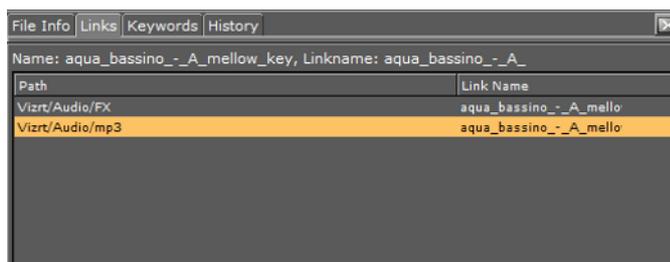
The File Info tab shows various information about the selected Element:



- **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 Fontstyle, 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:** Shows 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 who did the checkout operation (see [Active Elements](#)).
- **Locked By:** If the Element has been locked, it shows the name of the user that has performed the lock operation (see [Active Elements](#)).

Links Tab

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 folder where the linked file is placed, and to show the Element in the [Element List](#).

Keywords Tab

The Keywords tab shows the keywords that are assigned to the selected Element. Keywords can be assigned, for example, when an Element is imported to Viz Artist (see [Assign Keywords to Elements](#)).



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.

History Tab

The History tab shows the history of the selected Element:

Date	User	Changes	SysID	Module	Source
18.12.2012 12:45	Admin	ExternalCopy	--	Viz GH Manager	VizDbServer.10.211.111.19
28.08.2012 09:49	Guest	Created	c2 b4 fa 4a b5 4b 05 3d		

- **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.
- **Module:** Shows the name of the application that changed the data.
- **Source:** Shows the source of the data. For example, the archive name for an imported object or the name of the Graphic Hub from where it was copied.

To Show the Properties Panel

- Select **Show Properties** from the [Server Menu](#), or
- Hold the pointer over the Server area and press the keyboard shortcut <Ctrl+P>, or

- Right-click an Element in the [Element List](#). In the menu that opens, select **Show Properties...**

To Show the Properties Panel with a Selected Tab Open

- Right-click an Element in the [Element List](#). In the menu that opens, select either:
 - **Show Properties...**
 - **Show Folder Links...**
 - **Show Keywords...**
 - **Show History...**

Note: The **Show Folder Links...** option is only available for Elements that are linked (see [Linked Elements](#)).

To Close the Properties Panel

- Right-click in the [Element List](#). In the menu that opens, click on **Close [Properties]**, or
- Click the **Close** icon in the upper right corner of the [Properties Panel](#), or
- Un-tick **Show Properties** in the Server Menu.

5.6 Working with Elements

This section contains information on the following topics:

- [Common Element Procedures](#)
- [Active Elements](#)
- [Active Elements](#)
- [Linked Elements](#)
- [Referenced Elements](#)
- [Working with Scene Elements](#)
- [Working with Geometry Elements](#)
- [Working with Material Elements](#)
- [Working with Material Advanced Elements](#)
- [Working with Image Elements](#)
- [Working with Fontstyle Elements](#)
- [Working with Audio \(Clips\) Elements](#)

5.6.1 Common Element Procedures

These are procedures which are common to all Element types.

This section contains information on the following topics:

- [To View Elements in a Folder](#)
- [To Rename an Element](#)
- [To Delete an Element](#)
- [To Delete an Element with all References](#)

- [To Create an Element Copy in a New Folder](#)
- [To Create an Element Copy in the Same Folder](#)
- [To Move an Element to a New Folder](#)
- [To Create a Linked Element](#)

To View Elements in a Folder

1. Open the Element panel from the [Content Switcher](#):

Element	Tab	Keystroke (hold the pointer over the Server area)
Scenes	S	<Ctrl+1>
Geometries	O	<Ctrl+2>
Materials	M	<Ctrl+3>
Materials Advance	MA	<Ctrl+4>
Images	I	<Ctrl+5>
Fonts	F	<Ctrl+6>
Audio Clips	A	<Ctrl+7>

2. Select a folder in the Server Tree.

Note: Select **All** in the [Content Switcher](#) to view all Element types in the folder selected.

To Rename an Element

1. Select the Element to be renamed.
2. Open the rename text field:
 - Right-click the Element and select **Rename** from the context menu, or
 - Press <F2>.
3. In the text field that shows, type a new descriptive name.
4. Press <Enter>.

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

Note: If renaming a linked Element, only the name of the selected Element will change, not the linked one (see [Linked Elements](#)).

To Delete an Element

1. Select the Element to be deleted.
2. Delete the Element:
 - Drag the Element to the trash can, or
 - Right-click the Element and select **Delete** from the context menu, or

- Press <Delete>.
3. Click on **Yes** to confirm.

Tip: To delete multiple Elements select the Elements to be deleted and drag to the Trash Can.

Note: If deleting a linked Element, only the link to the folder will be removed. The Element itself will remain in the database (see [Linked Elements](#)).

Note: [Referenced Elements](#) cannot be deleted. Referenced Elements are used by other Elements.

To Delete an Element with all References

Deletes the selected Element and all its referenced Elements. [Referenced Elements](#) are not deleted if they are referenced by other Elements.

Note: Referenced elements are deleted if they are referenced to ONLY the selected Element. If they are also referenced to others, then they will not be deleted.

1. Locate the Element to be deleted.
2. Right-click the Element and select **Delete with all references** from the context menu.
3. Click on **Yes** to confirm.

To Create an Element Copy in a New Folder

Use this procedure to create an independent copy of an Element(s) in a new folder:

Note: To select more than one Element press and hold <Ctrl> and click on each required Element or press and hold <Shift> to select a range of Elements.

1. Select the Element(s) to be copied.
2. Press <Ctrl> and drag the Element(s) to the target folder.

or

1. Select the Element(s) to be copied.
2. Select **Copy** from the [Server Menu](#) or <Ctrl+C>.
3. Open the folder where the target Element(s) are to go.
4. Select **Paste-Dup** from the [Server Menu](#) or <Ctrl+Shift+V>

To Create an Element Copy in the Same Folder

Use this procedure to create an independent copy of an Element in the same folder:

1. Right-click the source Element.
2. In the context menu that shows, select **Duplicate**.

3. In the dialog box that opens, type a descriptive name for the new Element.
4. Click the **Ok** button.

or

1. Select the Element(s) to be copied.
2. Select **Copy** from the [Server Menu](#) or <Ctrl+C>.
3. Select **Paste-Dup.** from the [Server Menu](#) or <Ctrl+Shift+V>
4. In the dialog box that opens, type a descriptive name for the new Element.
5. Click the **Ok** button.

To Move an Element to a New Folder

Note: To select more than one Element press and hold <Ctrl> and click on each required Element.

- Drag the Element or Elements from the [Element List](#) onto the target folder in the Server Tree.

or

1. Select the source Element(s).
2. Select **Cut** from the [Server Menu](#), or <Ctrl+X>
3. Open the folder where the target Element(s) are to go.
4. Select **Paste-Dup.** from the [Server Menu](#), or <Ctrl+Shift+V>.

To Create a Linked Element

1. Select the source Element.
2. Press **<Alt>** and drag the Element to the target folder.

or

1. Select the source Element.
2. Select **Copy** from the [Server Menu](#), or <Ctrl+C>
3. Select the folder where the target Element is to go.
4. Select **Paste-Link** from the [Server Menu](#), or <Ctrl+V>.

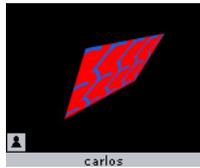
See Also

- [Working with Elements](#)
- [Active Elements](#)
- [Linked Elements](#)
- [Referenced Elements](#)
- [Import Files and Archives](#)

5.6.2 Active Elements

Elements that are in currently work are Active Elements. These Active Elements are locked or checked out.

Element with locked icon



An Active Element has a small icon in the lower left corner of the thumbnail in the [Element List](#) which changes to show different states:

-  **Locked:** If a user connected to the database opens an Element, the Element will be locked for all other users and 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 (see [To Lock an Element](#)).
-  **Checked Out:** If a user, connected to the database, checks out an Element, the Element will be locked for all other users and 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 (see [To Check Out Elements](#) and [To Check In Elements](#)).

Note: The Element will stay checked out (locked) 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 (see [To Lock and Check Out Elements](#)).

This section contains information on the following procedures:

- [To Lock an Element](#)
- [To Check Out Elements](#)
- [To Check In Elements](#)
- [To Lock and Check Out Elements](#)

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 Elements

1. Select the Element or Elements.
2. Check out the Elements:
 - Right-click an Element and click on **Check Out** in the context menu (one Element only), or
 - Hold the pointer over the Server area and use the keyboard shortcut <Ctrl +O>, or
 - Select **Check Out Files...** from the [Server Menu](#).

To Check In Elements

1. Select the Element or Elements.
2. Check in the Elements:
 - Right-click the Element and click **Check In** from the context menu (one Element only), or
 - Press the keyboard shortcut <Ctrl+I> while holding the pointer over the Server area, or
 - Select **Check In Files...** from the [Server Menu](#).

To Lock and Check Out Elements

- An Element is automatically locked when a user opens the Element. It is not possible to do this operation manually (see [To Check Out Elements](#)).

See Also

- [Working with Elements](#)
- [Common Element Procedures](#)
- [Linked Elements](#)
- [Referenced Elements](#)
- [Import Files and Archives](#)

5.6.3 Linked Elements

Every Element in the database is shown in one or more folders.



A linked Element has a small link icon attached to the lower left corner of the thumbnail in the [Element List](#). A linked Element will show, logically, in multiple folders, although only one Element is physically saved in the Graphic Hub database.

To view the UUID of the linked Element, open the [Properties Panel](#) and select the File Info tab. The UUID of the one saved Element is the same for all linked Elements, while information such as path, and name may vary.

If modifying the Element in one of the folders, for example cropping an image, all the linked items will reflect these changes.

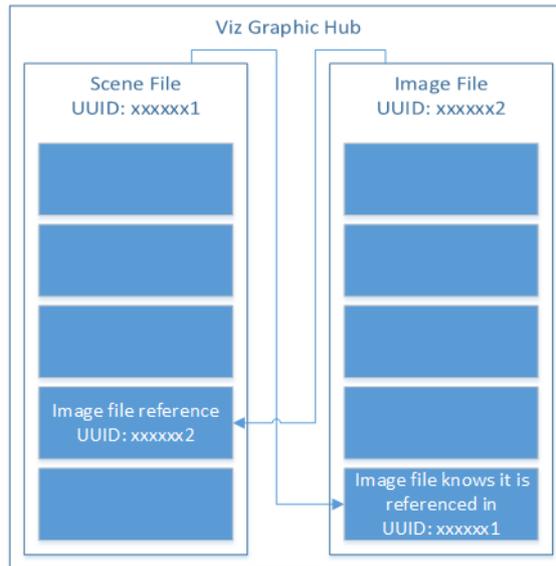
To create a linked Element see [To Create a Linked Element](#).

See Also

- [Working with Elements](#)
- [Active Elements](#)
- [Referenced Elements](#)
- [Import Files and Archives](#)

5.6.4 Referenced Elements

A referenced Element is an Element which is connected to and used by another Element, by a reference only.



Some Elements are, or can be, made up of other Elements, i.e. Geometries. An Geometry Element can be made up of other Elements. These other Elements are not physically part of the Geometry Element, but are used by reference.

Because the Geometry Element uses another Element as part of its output, any changes to the referenced Element will result in changes the Geometry Element itself.

The Element which is referenced cannot be modified without a warning message. If the referenced Element is modified then the Element which is referencing that Element will be modified too - it will look and act differently.

A referenced Element cannot be deleted. The Element which holds the reference has to be deleted before the referenced Element can be deleted from the Graphic Hub.

An Element can reference many other Elements.

See Also

- [Working with Elements](#)
- [Common Element Procedures](#)
- [Active Elements](#)
- [Linked Elements](#)
- [Reference Search](#)
- [Import Files and Archives](#)

5.6.5 Working with Scene Elements

This section contains information on the following topics:

- [To Create a New Scene](#)
- [To Import a Scene](#)

- [To Save a Scene](#)
- [To Save a Scene with a New Name](#)
- [Scene Save as... panel Context Menu](#)
- [To Undo/redo Operations in a Scene](#)

To Create a New Scene

Note: This will create an empty new Scene. Add containers as required, in the [Scene Tree](#)

1. Open the Scene tab (see [To View Elements in a Folder](#)).
2. Select the folder where the new Scene should go.
3. Open the Create New Scene panel:
 - Right-click in the [Element List](#). and select **Create new Scene**, or
 - Open the [Server Menu](#) and select **Create...**, or
 - Hold the pointer over the Server area and press <Ctrl+A>.
4. In the Create New Scene panel enter a text string for the new Scene.
5. Click on **OK**.



To Import a Scene

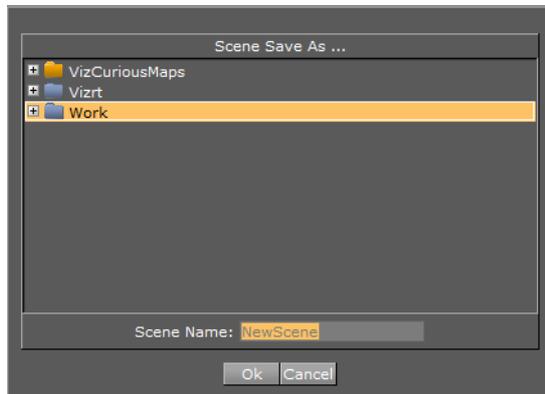
- For information about how to import a Scene (or an Archive) from a hard disk or network share to the database see [Import Files and Archives](#) .

To Save a Scene

- Click the **Save** button in the lower left corner of the window, or
- Hold the pointer over the Scene Tree area and press the keyboard shortcut <Ctrl>+S>.

To Save a Scene with a New Name

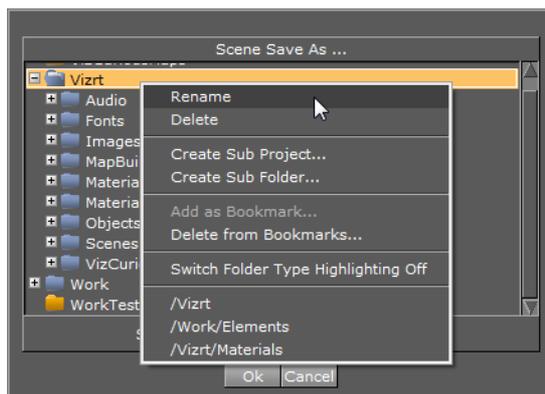
1. Click the **Save As** button in the lower left corner of the window.



IMPORTANT! In the **Scene Save As...** dialog box that opens, type a descriptive name in the **Scene Name** text box.

2. Click the **Ok** button to confirm.

Scene Save as... panel Context Menu



Right-click in the **Scene Save as** panel to open its context menu. Select from:

- **Rename:** Rename the selected folder
- **Delete:** Delete the selected folder
- **Create Sub Project:** Create a new sub Project folder in the selected Project folder
- **Create Sub Folder:** Create a new sub folder in the selected folder
- **Add to Bookmark:** Add selected folder to Bookmarks
- **Delete from Bookmark:** Delete selected folder from Bookmarks
- **Switch Folder Highlighting OFF/ON:** Set highlighting of folders to ON or OFF. For example, if a folder (and sub-folders) contain a Scene it will be highlighted when the Scene tab is selected. This feature can also be set in **User Interface** of the [Viz Configuration](#).
- **<Bookmark(s)>:** Click on a bookmarked folder to select that folder

To Undo/redo Operations in a Scene

- Click the **Undo/Redo** button in the lower right corner of the Scene Tree panel, or

- Hold the pointer over the Scene Tree panel and press the keyboard shortcut <Ctrl +Z/Y>.

Tip: Holding the pointer over the **Undo/Redo** buttons will show 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.

See Also

- [Common Element Procedures](#)

5.6.6 Working with Geometry Elements

A Geometry can be added to a Scene as a standalone Container or as a Geometry property of a Container.

See Also

- What are [Geometries](#)
- [Common Element Procedures](#)
- [Active Elements](#)
- Properties panel [Geometry Editor](#)
- [Geometry Plugins](#)
- [Import Files and Archives](#) Geometries

5.6.7 Working with Material Elements

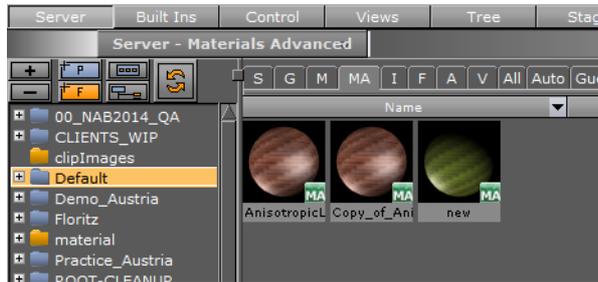
When a Material has been added to a Container, the Container specific Material can be modified in a [Material Editor](#) that opens in the Properties Panel. Only the Container specific Material will be modified, the original Material in the Graphic Hub database will not be affected.

To create a new Material in the Graphic Hub database, see [Material Editor](#).

See Also

- [Active Elements](#)
- [Material Editor](#)
- [Light Editor](#)
- [Common Element Procedures](#)

5.6.8 Working with Material Advanced Elements



This section contains information on the following procedures:

- [To Create a Material Advanced](#)
- [To Edit a Material Advanced](#)

See Also

- [Common Element Procedures](#)
- [Active Elements](#)
- [Materials Advanced](#)
- [Materials](#)
- [Built Ins](#)

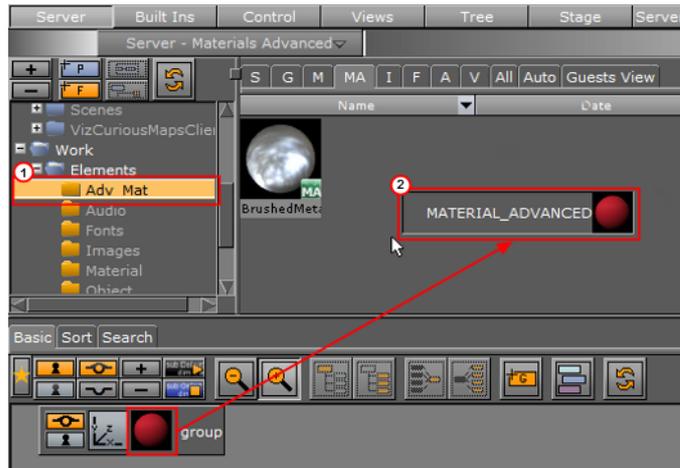
To Create a Material Advanced

1. From the [Element List](#) drag a Material and /or an Image to a Container in the Scene Tree.
2. From the [Built Ins](#) drag a Shader to the same Container in the Scene Tree (if required).

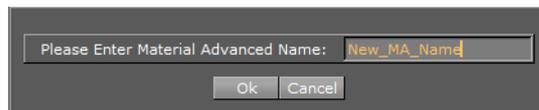
Note: A combination of at least two items (Material, Shader or Image) must be applied to create a Material Advanced.

3. Right-click the Container in the Scene Tree.
4. Click **Merge to Material Advanced**.
The Material, Shader, and Image icons in the Container will then be merged into one combined Material Advanced icon.

5. Select the project or folder where the Material Advanced should be placed (1).



6. Drag the Material Advanced icon from the Container in the Scene Tree to the [Element List](#) (2).



7. In the dialog box that opens, type a descriptive name for the Material Advanced.

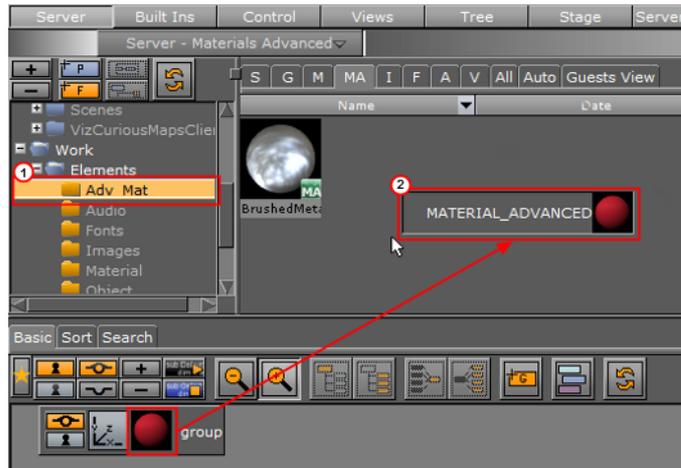
Note: Special characters, such as space, are not allowed in Element names.

8. Click the **Ok** button.

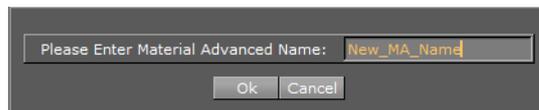
To Edit a Material Advanced

1. Select the project or folder where the existing Material Advanced is placed.
2. Drag the Material Advanced to a Container in the Scene Tree.
3. Right-click the Material Advanced icon on the Container.
4. In the context menu click **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 Material Advanced consists of.
5. Apply new items or modify the Material, Shader, or Image settings.
6. Right-click the Container in the Scene Tree.
7. In the context menu click **Merge to Material Advanced**. The Material, Shader, and Image icons in the Container will merged into one combined Material Advanced icon.

8. Select the project or folder where the Material Advanced should be placed (1).



9. Drag the Material Advanced icon from the Container in the Scene Tree to the [Element List](#) (2).



10. In the dialog box that opens, type a descriptive name for the Material Advanced.

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

1. Click the **Ok** button.

5.6.9 Working with Image Elements

This section contains information on the following procedures:

- [To Add an Image to the Graphic Hub Database](#)
- [To Edit an Image](#)
- [To Export an Image or Images](#)

See Also

- [Common Element Procedures](#)
- [Active Elements](#)
- Properties panel [Texture Editor](#)
- [Image Editor](#)

To Add an Image to the Graphic Hub Database

- **Import Images:** Images can be imported from a hard disk or a network share to the GH database (see [Import Files and Archives](#)), or

Tip: Multiple images can be imported at once.

- **Create a Snapshot:** Images can be created through taking a snapshot of the Scene Editor (see [Snapshot](#)), or
- **Image Editor:** In the Server Tree open an Image in the [Image Editor](#) (double-click the Image), modify (if required) and select **Save as...**

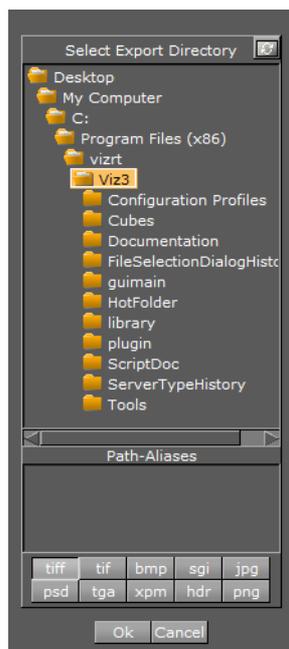
To Edit an Image

1. Double click the image to open the [Image Editor](#).
2. In the [Image Editor](#) that opens, define the new settings.
3. Click the **Save** button.

To Export an Image or Images

An image or multiple images can be exported.

1. Select the image(s) to be exported.
2. Open the **Select Export Directory** panel:
 - Select **Export...** from the [Server Menu](#), or (when only one image is selected)
 - Right-click the image to be exported. In the menu that opens, select **Export...**



3. Select an image format for export.

IMPORTANT! In the **Select Export Directory** panel, select the folder where the image should be exported to.

IMPORTANT! Click the **Ok** button.

Tip: When an RGB image with alpha is exported, an RGBA image will be created.

Note: The name of the exported image will be equal to the name of the original image.

5.6.10 Working with Fontstyle Elements

This section contains information on the following procedures:

- [To Import a Font](#)
- [To Edit a Fontstyle](#)
- [To Create a Fontstyle](#)
- [To Show/Hide Blur or Outline Fontstyles](#)

See Also

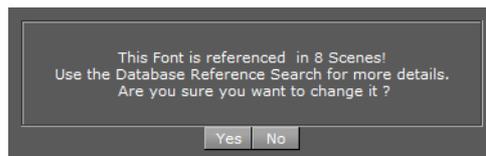
- [Common Element Procedures](#)
- [Active Elements](#)
- Properties panel [Text Editor](#)
- [Fontstyle Editor](#)
- [Import Files and Archives](#) Fontstyles

To Import a Font

- For information about how to import a font from a hard disk or network share to the database see [Import Files and Archives](#).

To Edit a Fontstyle

1. Double-click the Fontstyle to open the [Fontstyle Editor](#).
2. If the Fontstyle is referenced in more than one Scene, a Conformation Panel will open. Click **Yes** to confirm. Click **No** to cancel (see [Reference Search](#) and [Referenced Elements](#)).



3. In the [Fontstyle Editor](#) define the new settings.
4. Click the **Save** button.

See Also

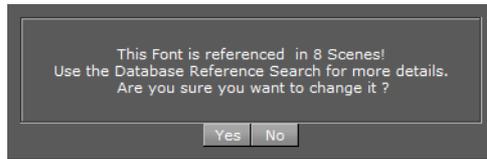
- [Referenced Elements](#)

To Create a Fontstyle

Note: A new Fontstyle will be based on a font which already exists.

1. Double-click the Fontstyle to open the [Fontstyle Editor](#).

2. If the Fontstyle is referenced in more than one Scene, a Conformation Panel will open. Click **Yes** to confirm. Click **No** to cancel (see [Reference Search](#)).



3. In the [Fontstyle Editor](#), define the new settings.
4. Click the **Save As** button.
5. In the **Save As...** panel, type a descriptive name for the new Fontstyle in the text box.
6. Click the **Ok** button.

To Show/Hide Blur or Outline Fontstyles

- Right-click a Fontstyle and select either:
 - **Show/Hide “-Blur-” Fonts**, or
 - **Show/Hide “-Outline-” Fonts**

5.6.11 Working with Audio (Clips) Elements

This section contains information on the following procedures:

- [To Add an Audio Clip to a Scene \(Context Menu\)](#)
- [To Add an Audio Clip to a Scene](#) (manually)
- [To Import an Audio Clip](#)
- [To Export an Audio clip](#)

See Also

- [Common Element Procedures](#)
- [Active Elements](#)

To Add an Audio Clip to a Scene (Context Menu)

1. Highlight the target Container(s).
2. Right-click the [Audio](#) plugin (*Container Plugins -> Global*).
3. Select **Add Audio Clip**.

The Audio plugin is added to the highlighted Container (the Container will show that an animation was created).



4. Click on the [Stage Button](#).
5. Highlight the **Audio Clip** in the Stage tree.

- Click on the **Add key Frame** icon (1).

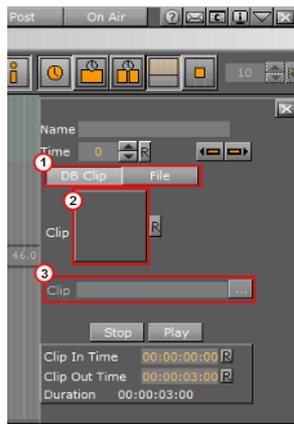


- In the Key Frame panel add an Audio Clip. There are two ways to add an Audio Clip:

- Click **DB Clip** (1).
- Click on **Server/Stage** in the [Main Menu Left](#).
- Select and drag an Audio Clip from the Server Tree panel to the **Clip drop box** (2).

or

- Click **File** (1).
- In the **Clip** line (3), click the  icon to select an Audio clip to use.



To Add an Audio Clip to a Scene

- Add the [Audio](#) plugin (*Container Plugins -> Global*) to a Container (see [Built Ins](#)).
- In the [Main Menu Left](#) click on **Stage**.
- Drag the Container to the [Stage Tree Area](#).

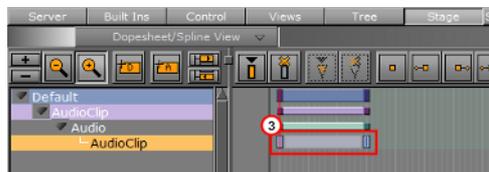


- In the Stage Tree right-click the Container Actor (1)

- Click **Audio > Clip (2)** (the Container will show that an animation was created).



- Click on the Audio Clip Channel (3) in the [Stage Editor](#), to open the Key Frame panel (if not open).

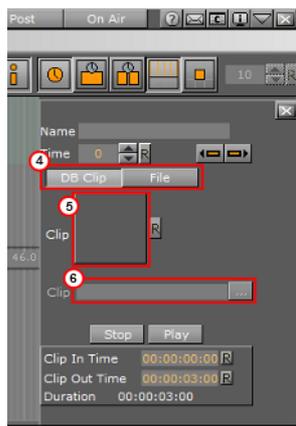


- In the Key Frame panel add an Audio Clip. There are two ways to add an Audio Clip:

- Click **DB Clip (4)**.
- Click on **Server/Stage** in the [Main Menu Left](#).
- Select and drag an Audio Clip from the Server Tree panel to the **Clip drop box (5)**.

or

- Click **File (4)**.
- In the **Clip line (6)**, click the **...** icon to select an Audio clip to use.



See Also

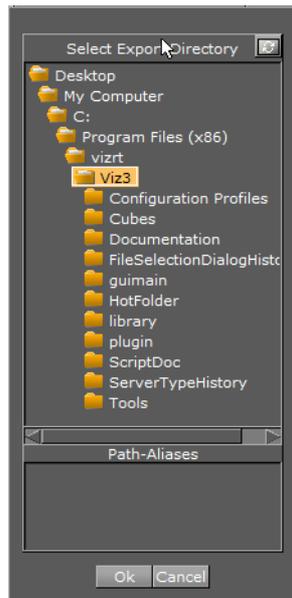
- [Animate Audio](#)

To Import an Audio Clip

- For information about how to import an audio clip (or an archive) from a hard disk or a network share to the database see [Import Files and Archives](#).

To Export an Audio clip

1. Open the Select Export Directory panel:
 - Click on an Audio clip or Audio clips and select **Export...** from the Server Menu, or
 - Right-click the Audio clip to be exported. In the menu that opens, select **Export...**



IMPORTANT! In the Select Export Directory panel, select the folder where the Audio clips should be exported to.

2. Click the **Ok** button.

Note: The name of the exported Audio clips will be equal to the name of the original Audio clips.

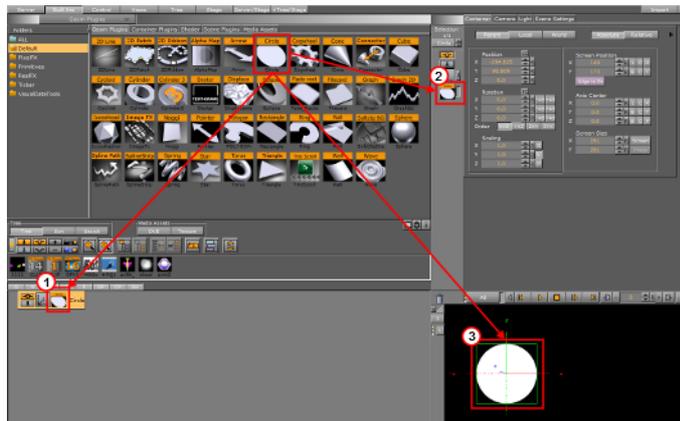
5.7 Add Properties to Containers and Scenes

This section details how to add **Elements** and **Built Ins** to a Container, multiple Containers and Scenes.

Most properties can be dragged to a target Container, or Containers, in the [Scene Tree](#) (1), the Properties Panel (2) or the [Scene Editor](#) (3). Each property also has a context menu where a property can also be added to a Container, multiple Containers and Scenes.

Details how to add **Audio** Elements to a Scene see [Working with Audio \(Clips\) Elements](#).

Details how to add and create Scene Elements see [Working with Scene Elements](#).



This section contains the following procedures:

- [To Add Properties to an Empty Scene Tree](#)
- [To Add Properties to a Container](#)
- [To Multi Drop Properties as Sub Containers](#)
- [To Multi Drop Properties on Containers](#)
- [To Add Multiple Properties to Containers](#)
- [To Add Properties to Multiple Containers](#)
- [To Add a Scene Plugin to a Scene](#)

See Also

- [What are Elements](#)
- [Built Ins](#)
- [Working with Scene Elements](#)
- [Working with Audio \(Clips\) Elements](#)
- [Media Asset Workflow](#)

To Add Properties to an Empty Scene Tree

These properties can be added to an empty Scene Tree:

Elements	Built In
Geometries Images Fontstyles	Geometries Media Assets

1. Drag a property to either:
 - The [Scene Tree](#) (creates a new Container)
 - The [Scene Editor](#) (creates a new Container in the Scene Tree)

Tip: If multiple Geometry or Image Elements are dragged from the **Element List** to the **Scene Tree**, a Container for each selected Element is created.

To Add Properties to a Container

These properties can be added to a Container:

Elements	Built In
Geometries	Geometry Plugins
Materials	Container Plugins
Material Advanced	Shaders
Images	Media Assets
Fontstyles	

Note: Only one Geometry plugin, one Shader or one Media Asset can be applied to a single Container. If a new Geometry plugin, Shader or Media Asset is added the first one will be removed from the Container.

Tip: If a **Shader**, **Media Asset** or a few **Container Plugins** are dragged to an object in the Scene Editor, they will be added to the Container which contains the Object. The **Media Asset** will be added as a Texture. If a Container Plugin can be added this way, the object becomes active when the Container Plugin is moved over the object.

1. Select a Container.
2. Right-click a property and select **Add to Container(s)**. Media Assets give the option to **Add <Media Asset> as DVE** or as **Texture**.

Note: If no Container is selected the property is added to the top container in the Scene Tree.

or

- Drag the property to a Container

Note: Media Assets are added as Texture to the Container.

Material Elements only:

1. Right-click a Container
2. Click on **Add Material...**

To Multi Drop Properties as Sub Containers

These properties can be multi dropped and create Sub Containers:

Elements	Built In
Geometries	Geometry Plugins
Images	Media Assets
Fontstyles	

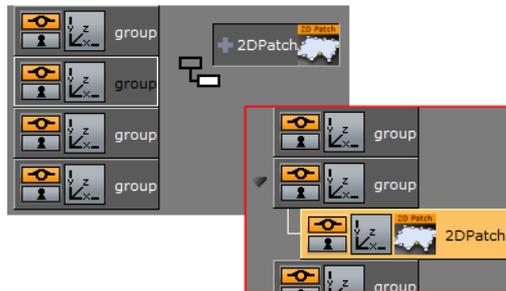
1. Drag a selected property to the Scene Tree (aligned with a Container).

2. Hold the left-click and right-click to add the property to a new Sub-Container of the highlighted Container.
3. Keep the left-click and repeat as required.

Tip: Also works if multiple Geometries are selected.

Note: When the last property to be added has been dropped, release the right-click while the cursor is still aligned the target Container. If the right-click is released when over or aligned with another Container, another (not required) Container or Sub-Container, and Property will be added.

Note: A **Media Asset** is added as a Texture.



To Multi Drop Properties on Containers

These properties can be multi dropped on Containers:

Elements	Built In
Materials	Geometries
Material Advanced	Container Plugins
Image	Shaders
Fontstyles	Media Assets

1. Drag a property over a Container.
2. Hold the left-click, and right-click to add the property to the highlighted Container.
3. Keep the left-click and repeat as required.

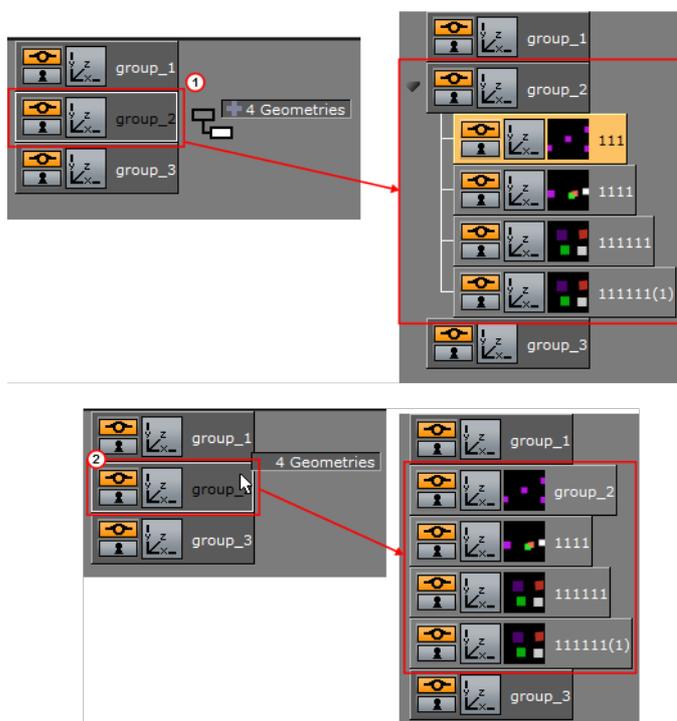
Note: When the last property to be added has been dropped, release the right-click while the cursor is still over the target Container. If the right-click is released when over or aligned with another Container, another (not required) Container or Sub-Container, and Property will be added.

To Add Multiple Properties to Containers

These properties can be Multi Added to a Container:

Elements	Built In
Geometries Images	None

- Multi select the required properties.
- Drag the selected properties to the Scene Tree or a Container:
 - In the Scene Tree** (1(aligned with a Container)): All selected properties are placed in the Scene Tree as sub-containers of the highlighted Container, or
 - Over a Container** (2). The first selected property is added to the highlighted Container and the other selected properties create new Containers, at the same level



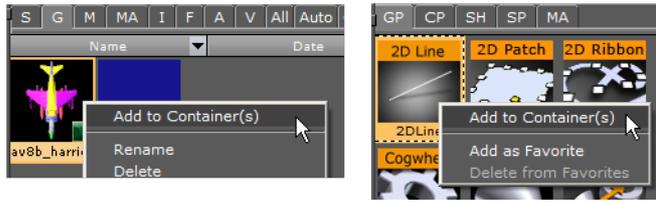
To Add Properties to Multiple Containers

These properties can be added to Multiple Containers:

Elements	Built In
Geometries Materials Material Advanced Image Fontstyles	Geometries Container Plugins Shaders

- Multi select the required Containers (see [Select Multiple Containers](#)).
- Add a property:

- Right-click a property and select **Add to Container(s)**,

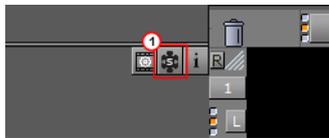


or

- Drag a property to one of the multi selected Containers.

To Add a Scene Plugin to a Scene

1. Open the Plugin tab in [Scene Settings](#):
 - Click on *Scene Settings* -> *Plugin*, or
 - Click on the Scene plugin icon (1)



2. Drag a Scene Plugin icon to the Scene drop area.

or

1. Right-click the Scene Plugin icon to open the [Plugin Context Menu](#).
2. Click on **Add to Scene...**

If a Scene Plugin has been applied to a Scene, the Scene Plugin icon will show orange



5.8 Assign Keywords to Elements

Keywords are used to tag Elements in the Graphic Hub database. New keywords can be added in the [Import Window](#), and when files are imported to the Graphic Hub database. Keywords can be assigned to an Element, up to 20 keywords may be assigned to one Element.

There are four ways to assign keywords. Each method starts from a different workflow point.

This section contains the following procedures:

- [To Create Keywords for Elements](#)
- [To Assign Keywords from the Search Panel](#)
- [To Assign Keywords from the Keyword Panel](#)
- [To Assign a Keyword While Importing](#)
- [To Assign Keywords from the Property Editor](#)
- [To Remove a Keyword from an Element](#)
- [To Delete a Keyword](#)

To Create Keywords for Elements

1. Select **Import** from the [Main Menu Right](#).

Note: The Keywords panel shows when either Fonts, Images, Geometries, Scenes or Audio is selected to import in the [Import Menu](#).



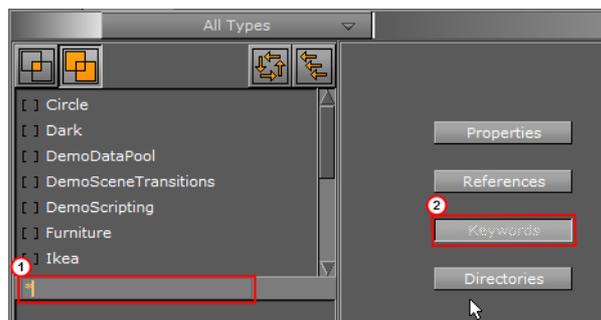
IMPORTANT! In the Keywords panel, type a descriptive keyword in the text box (1).

2. Click the **Add** button (2).

Note: The new keyword is saved and will show in the Keyword list.

To Assign Keywords from the Search Panel

1. Select **Server** from the [Main Menu Left](#).
2. Open the Search panel:
 - In the Server menu, click **Search**, or
 - Hold the pointer over the Server area and press the keyboard shortcut <Ctrl +F>.
3. In the Search panel that opens, click Keywords (2).
4. Search for a Keyword or Keywords (1):
 - To search for all Keywords type * (an asterisk) in the text box, and press **<Enter>**, or
 - To search for a specific Keyword type the name of the Keyword in the text box, and press **<Enter>**.



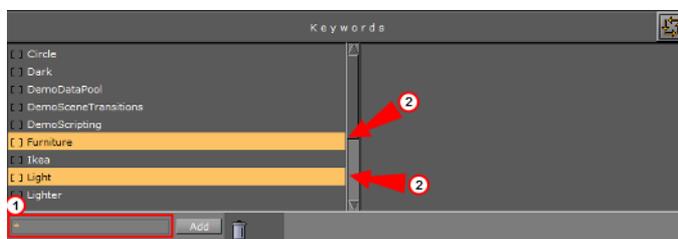
IMPORTANT! Drag one or more Keywords from the Search panel to an Element or selected Elements in the [Element List](#).

- Click **Yes** to confirm.

Note: To view assigned Keywords on an Element, right-click an Element and click on **Show Keywords...** from the context menu.

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. In the dialog box that opens, confirm the assign operation.

To Assign Keywords from the Keyword Panel

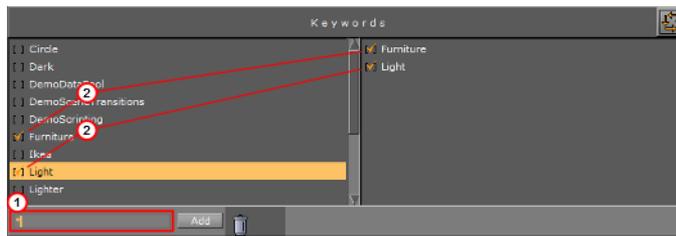


Note: This procedure assumes that there is at least one Element in the Server area.

- Select **Import** from the [Main Menu Right](#).
- In the Keywords panel search for a Keyword or Keywords (1):
 - To search for all Keywords, type * (asterisk) in the text box, and press **<Enter>**, or
 - To search for a specific Keyword, type the name of the Keyword in the text box, and press **<Enter>**.
- Select a Keyword(s) (2).
- Drag the selected keywords to the Element or selected Elements in the Server area.
- Click **Yes** to confirm.

Tip: If the list of Keywords, in the left panel, is long, scroll and tick the required Keywords so that they show in the right panel. From the generated Keyword list in the right panel, highlight them all and drag to the Element.

To Assign a Keyword While Importing



1. Select **Import** from the [Main Menu Right](#).
2. In the Import panel, select the file or files to be imported.
3. Search for a Keyword or Keywords (1):
 - To search for all Keywords, type * (asterisk) in the text box, and press **<Enter>**, or
 - To search for a specific Keyword, type the name of the Keyword in the text box, and press **<Enter>**.
4. Select a Keyword or Keywords (2):
 - Click in the [] (to the left of each Keyword) to select a Keyword (the selected Keywords show in the right side panel), or
 - Click on one or more keywords and drag to the right side panel.
5. Drag the selected Elements to their location in the Server area.

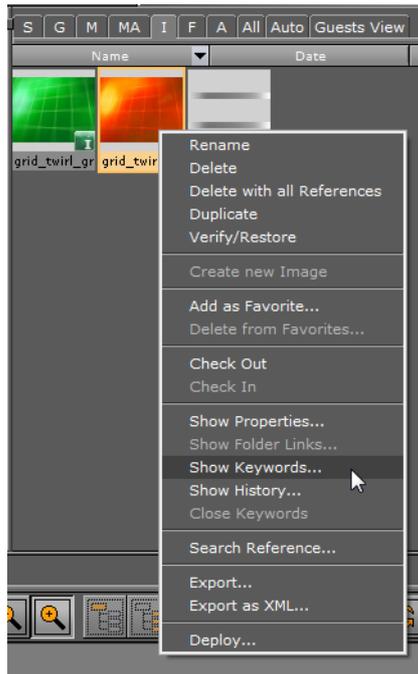
Note: To view assigned Keywords on an Element, right-click an Element, in the Server area, and click on **Show Keywords...** from the context menu.

To Assign Keywords from the Property Editor

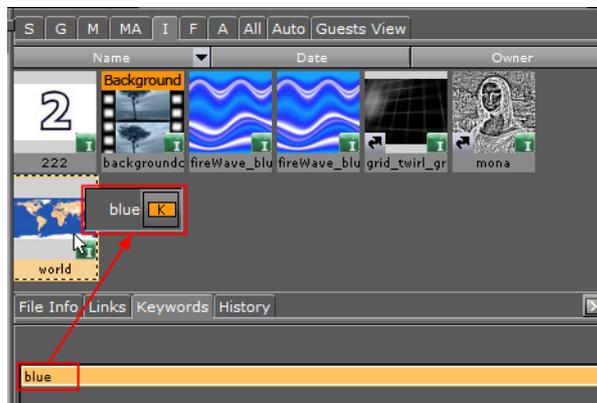
Note: The source and destination Element must be in the same folder.

1. In the Server area, right-click the source Element.

2. Select **Show Keywords...**

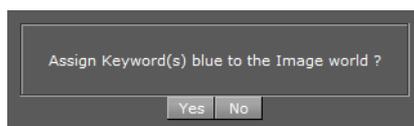


IMPORTANT! In the Keywords tab click the required Keywords, and drag to the required Element in the Server Area.



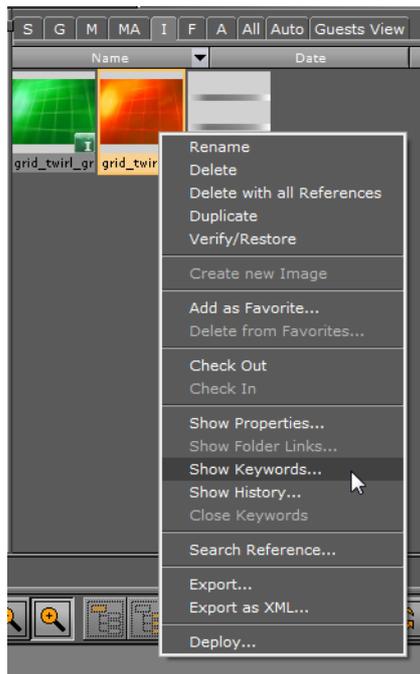
Note: When more than one source Element is selected, and they all have a common keyword, only that common keyword will show in the Keywords panel.

IMPORTANT! Click **Yes** to confirm.



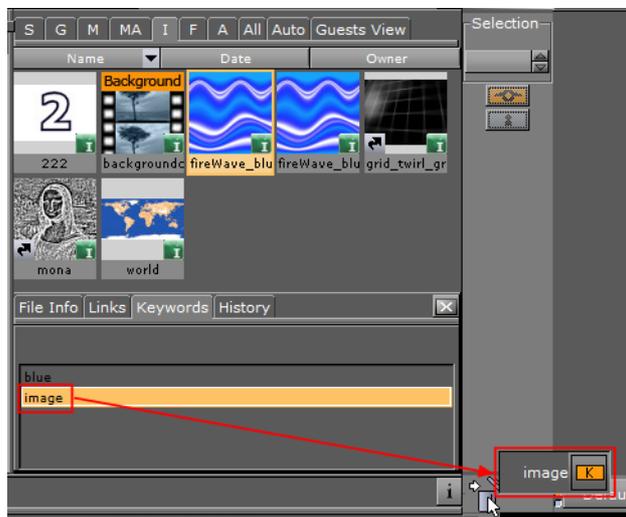
To Remove a Keyword from an Element

1. In the Server area, right-click the Element with the Keyword to be deleted.
2. Click **Show Keywords...**

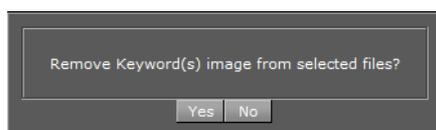


IMPORTANT! Click the Keyword(s) to be deleted.

3. Drag to the Trash Can.



IMPORTANT! Click **Yes** to confirm:

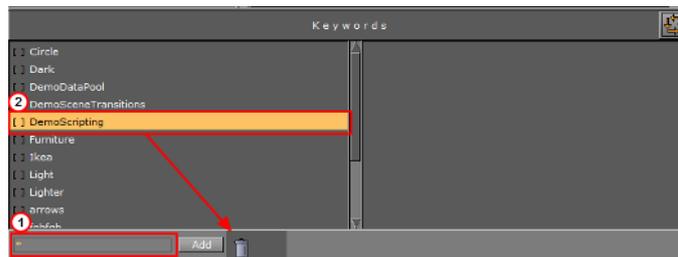


To Delete a Keyword

1. Select **Import** from the [Main Menu Right](#).

Note: The Keywords panel is not available if **Archives to Import** is selected in the Import file drop-down menu.

2. In the Keywords panel search for a Keyword(s) (1):
 - To search for all Keywords, type * (an asterisk) in the text box, and press <Enter>, or
 - To search for a specific Keyword, type the name of the Keyword in the text box, and press <Enter>.
3. Select a Keyword(s) (2), and drag to the Trash Can.
4. Click **Yes** to confirm.

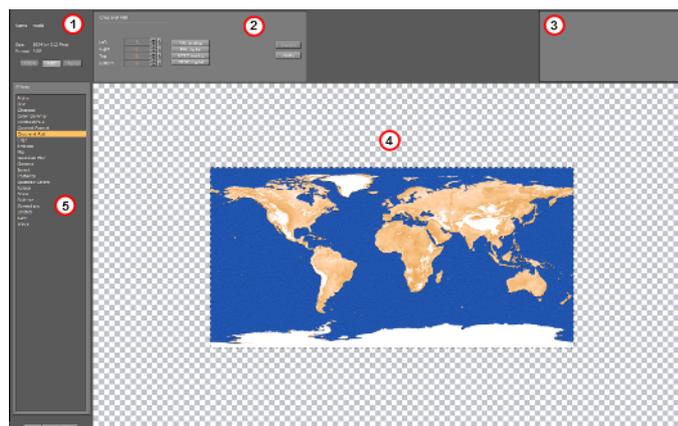


Note: Keywords can only be removed from the Graphic Hub if they are not assigned to any files.

5.9 Image Editor

The Image Editor makes it possible to modify Image Elements in the Graphic Hub database. Individual editors are also available for Materials ([Material Editor](#)) and Fontstyles ([Fontstyle Editor](#)).

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



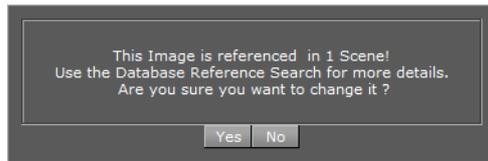
The Image Editor consists of:

- Image General Information (1)
- Effect Modifications (2)

- Effect Preview (3)
- Main Image Area (4)
- Effects (5)

To Open an Image in the Image Editor

- Double-click the image in the [Server Area](#) to open the image in the Image editor.
If an image is referenced in a Scene, a dialog box will show to confirm the action or not (see [Referenced Elements](#)).



Note: An Element that is opened in an editor, is locked in the database (see [Active Elements](#)).

This section contains information on the following topics:

- [General Image Information](#)
- [Image Effects](#)
- [Apply and Preview Effects](#)

5.9.1 General Image Information

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



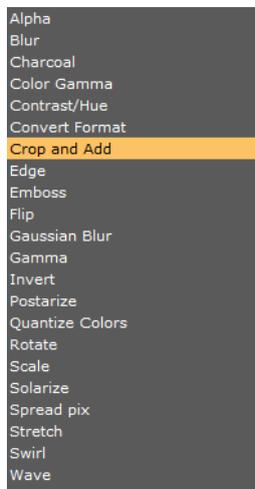
- **Name:** Shows the image name.
- **Size:** Shows the image size in pixels.
- **Format:** Shows the image format (RGBA, RGB, or Alpha).

Click the format buttons to view the image 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.

Image Effects

The following effects can be applied to an image. To view the effect properties and see changes made, see [Apply and Preview Effects](#):



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

Note: This option only applies for alpha format images.

- **Blur:** Creates a blurred image.
- **Charcoal:** Creates a posterized, smudged effect by redrawing the image. Major edges are boldly drawn, while mid tones are sketched using a diagonal stroke.
 - **Radius:** The smaller the radius the more details will be seen.
 - **Sigma:** Effects the sharpness of the image.
- **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.



- **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.
- **Edge:** The Edge filter searches for borders between different colors and detects contours of objects:
 - **Radius:** A low value results in black, high-contrasted image with thin edges. A high value results in thick edges with low contrast and many colors in dark areas.
- **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.
 - **Angle:** Defines the direction of the embossment.
 - **Value:** Defines amount of effect.
- **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.
 - **Range:** Affects the blur spread (not much visible since same effect may be achieved by configuring 'value' parameter)
 - **Blur Value:** Amount of blur
- **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.

- **Postarize:** Effects the saturation of the image
 - **Level (%):** Amount of the effect, at high values image becomes completely black and white
- **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.
- **Spread Pix:** Creates a dissolving effect for the image pixels
- **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.

5.9.2 Apply and Preview Effects

Various effects can be selected from the [Image Effects](#). The selected effect can then be modified in the Effect Modification area (1) and the result viewed in the main image area, or previewed (2).



The Effect Modification area has two common buttons, which apply to most effect settings:

- **Preview:** Shows a preview of the changes in the image preview area (2).
- **Apply:** Applies the changes to the image.

Note: Some effect modifications apply directly to the image in the main image area.

To Apply an Effect to an Image

1. Select an effect from the [Image Effects](#) list.
2. Make modifications to the effect as required in the Effect Modification area. In most cases use the Preview area to view the effect on the image.
3. Add other effects as required.
4. Click Save or Save as...

5.10 Fontstyle Editor

The Fontstyle Editor makes it possible to modify Fontstyle Elements in the Graphic Hub database. Individual editors are also available for Materials ([Material Editor](#)) and Images ([Image Editor](#)).

Viz Artist uses Fontstyles to show text in Scenes. When a Font is imported into the Graphic Hub database, Viz Artist imports the base font and creates up to eight different Fontstyles:



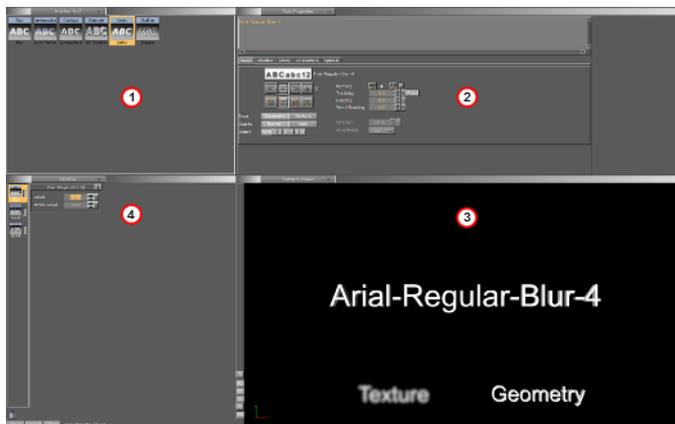
- One Fontstyle which looks like the base font (1)
- Up to four Fontstyles in different blur levels (2)
- Up to three Fontstyles using only the outline of the base font (3)

Tip: The number of blur levels and outlines for each Fontstyle can be defined in the **Font Options** section of the [Viz Configuration](#) (see the [Viz Engine Administrator Guide](#)).

Tip: The Outline and Blurred Fontstyles can be hidden or shown in the [Server Area](#) (see [To Show/Hide Blur or Outline Fontstyles](#)).

In the Fontstyle Editor, Fontstyles can be modified through the use of [Modifiers](#). For example, a Fontstyle can be modified to reflect a corporate identity.

As Viz Artist uses an open interface for the plugins used as modifiers, almost any modifications can be applied to a Fontstyle. New Fontstyles can be created based on existing Fontstyles through the **Save As...** operation.



The Fontstyle Editor consists of:

- Available Modifiers (1)
- Properties of previewed text (2)
- Preview of the modifications (3)
- Modifier parameters (4)

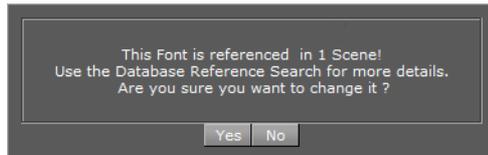
Note: The properties help to illustrate how the modifications will look like in a Scene. Changes made on the right side of the editor, for example kerning and material, will not be saved with the Fontstyle. However, there is one exception

from this rule. If the kerning between two fixed characters are modified, this is saved with the Fontstyle. To create a kerning between two fixed characters, use the keyboard shortcuts <Alt+Left Arrow>/<Alt+Right Arrow>.

To Open a Fontstyle in the Fontstyle Editor

- Double-click the Fontstyle in the [Server Area](#) to open it in the Fontstyle Editor.

If a Fontstyle is referenced in a Scene, a dialog box will show to confirm the action or not (see [Referenced Elements](#)).

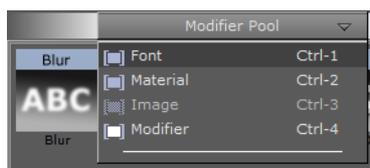


Note: An Element that is opened in an editor, is locked in the database (see [Active Elements](#)).

This section contains information on the following topics:

- [Available Modifiers](#)
- [Modifiers](#)
- [Modifier Parameters](#)
- [Preview Menu](#)
- [Fontstyle Viewer](#)
- [Fontstyle Viewer Menu](#)
- [Fontstyle Viewer Bar](#)
- [To Open a Fontstyle in the Fontstyle Editor](#)

5.10.1 Available Modifiers



The available modifiers selected in from the Pool menu, which is positioned at the top left of the Fontstyle Editor.

- **Font:** Show all the Fontstyles in the database.
- **Material:** Show all the Materials in the database.
- **Image:** Not available
- **Modifier:** Show all available modifiers that can be applied to the Fontstyle.

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

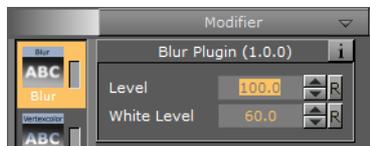
provides an open connection, so that additional modifiers can be obtained or created and added as plugins.

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.

Modifier Parameters

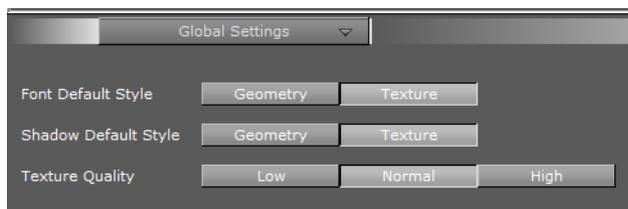


The Modifier parameters menu is positioned in the middle left part of the Fontstyle Editor.



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

Global settings in Fontstyle Editor



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

Note: The texture quality cannot be changed in a Text Editor.

5.10.2 Working with Modifiers

This section contains information on the following procedures and topics:

- [To Add a Modifier to a Fontstyle](#)
- [To Edit a Modifier](#)
- [To Enable/Disable a Modifier](#)
- [To Remove a Modifier from a Fontstyle](#)

- [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 Fontstyle

1. Select **Modifier** from the [Available Modifiers](#).
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.



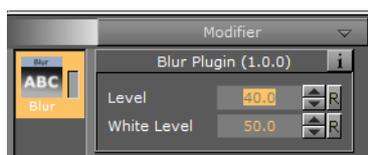
To Remove a Modifier from a Fontstyle

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 Fontstyle in texture mode.



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

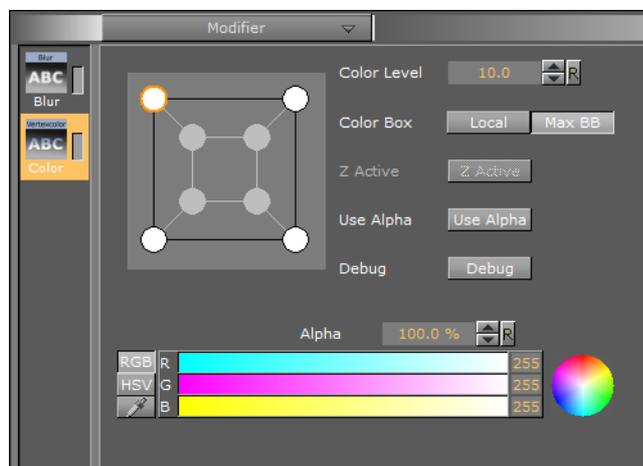
Color Per Vertex



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

The circles in the Color Per Vertex modifier represent the four corners of the Fontstyle text:

- Click a circle to edit its color values.
- Click on a line that connects two circles to select both circles
- Hold <Ctrl> and click to select multiple circles.
- Colors can also be dragged from one circle to another.

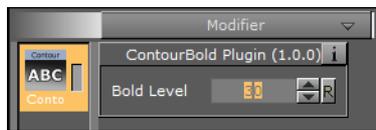


- **Color Level:** Defines how much color should be applied.
- **Color Box:**
 - **Local:** Applies the color to the vertices of every single character.
 - **Max BB:** Applies the color to the vertices of the bounding box.
- **Z Active:** If the [Extrude](#) modifier is used in addition to Color Per Vertex, the colors can be changed both in the front and back. If no such modifier is applied, only the colors in the front can be edited.
- **Use Alpha:** Show where the color should include. For example if Alpha is enabled and the bottom right alpha is set to 0 the characters will fade out the bottom right.
- **Debug:** Draws the actual color points to see and understand how color changes effect the characters.

Contour Bold



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



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

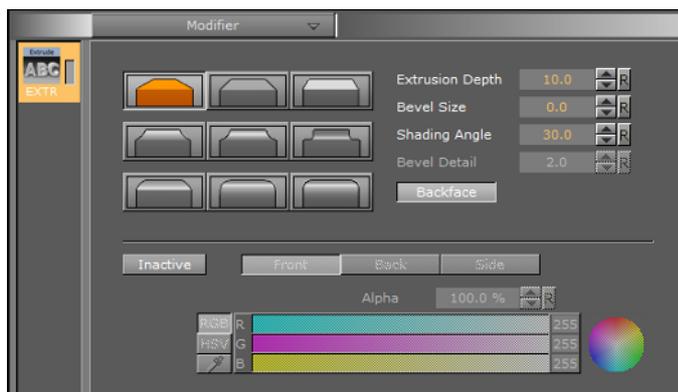
Extrude



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

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

Extrude editor



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.

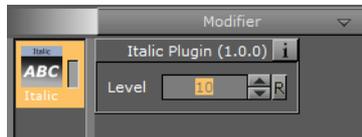
Note: The Bevel Detail option is only available for certain bevel types.

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

Italic



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

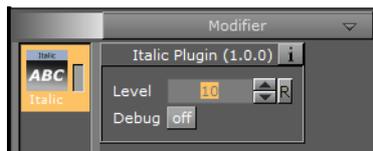


- **Level:** Defines the cursive level of a Fontstyle. The higher the value, the more tilted is the text

Outline



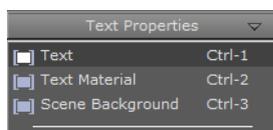
Use the Outline modifier to create an outline of a Fontstyle.



- **Outline Level:** Defines the width of the outline level. The higher the value, the bolder the outline of a Fontstyle.
- **Debug:** Draws the actual color points to see and understand how color changes effect the characters.

5.10.3 Preview Menu

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



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

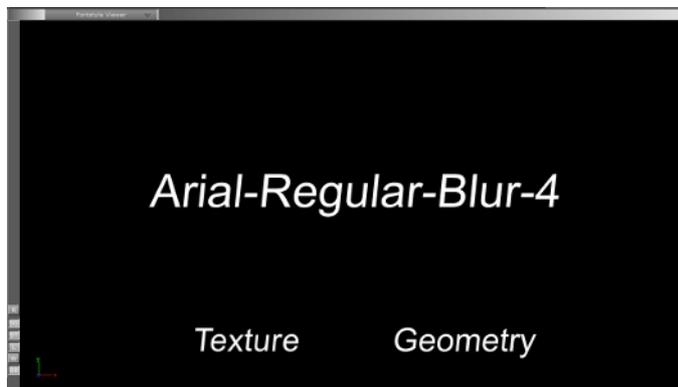
- **Text Properties:** To preview how the Fontstyle will look like with different settings, for example kerning, orientation, and content, select Text Properties from the Preview menu and modify the properties. For more information, see [Text Editor](#) .
- **Text Material:** To preview how a Material would look if used on the Fontstyle, select Text Material from the Preview menu. In the Material Editor that opens, select a color (see [Working with Material Elements](#)).

Tip: To use a Material already from the database, select **Material** from the [Available Modifiers](#) in the Pool menu, select a Material and drag to the color icon in the Material Editor of the Fontstyle.

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

5.10.4 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 Fontstyle.



Additionally to the customizable text in the Fontstyle viewer, the two words Texture and Geometry will be shown. 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 Parameters](#).

Fontstyle Viewer Menu

Right-clicking the [Fontstyle Viewer](#) opens a menu, similar to the menu in the Scene Editor, 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 text with geometry style.
- **ST:** Shows/hides text with texture style.
- **L:** Enables/disables additional light.
- **W:** Switches on/off wire frame mode.
- **BB:** Switches on/off the bounding box visualization.

5.11 Material Editor

The Material editor is used to create and modify materials with customized colors and [Lights](#) effects.

The icon in the upper right corner of the Materials Editor (1) 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, see [Light Editor](#).



When a Material has been added to a Container, the Container-specific Material can be modified in the [Material Editor](#) that opens in the Properties Panel. Only the Container-specific Material will be modified, the original Material in the Graphic Hub database will not be affected (see [Working with Material Elements](#)).

This section contains information on the following topics:

- [Create a Material](#)
- [Material Color Properties and Schemes](#)

See Also

- [Lights](#)
- [Working with Material Elements](#)

5.11.1 Create a Material

To create a new material and save it use the procedure detailed below.

To Create a New Material

1. Click **Server**.
2. Click the **M** (Materials) tab.
3. Right-click and select **Create new material**.
4. Enter a name for the new Material

- Click **OK**.



- In the Server area double-click the new Material to open the Material Editor.
- Adjust the [Material Color Properties and Schemes](#) as necessary.



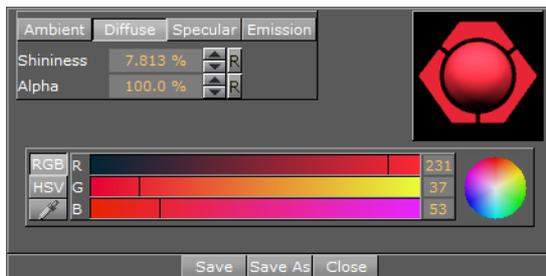
- Save any changes:
 - Save:** Saves the material in the Graphic Hub.
 - Save As:** Creates a new Material in the Graphic Hub. In the Save As dialog box that opens, type a name for the Material and select a destination folder.
 - Close:** Closes the Material editor without saving the changes.

5.11.2 Material Color Properties and Schemes

This section contains information on the following topics:

- [Material Color Properties](#)
- [Material Base Color](#)
- [To Change a Material Color](#)

Material Color Properties



- 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.

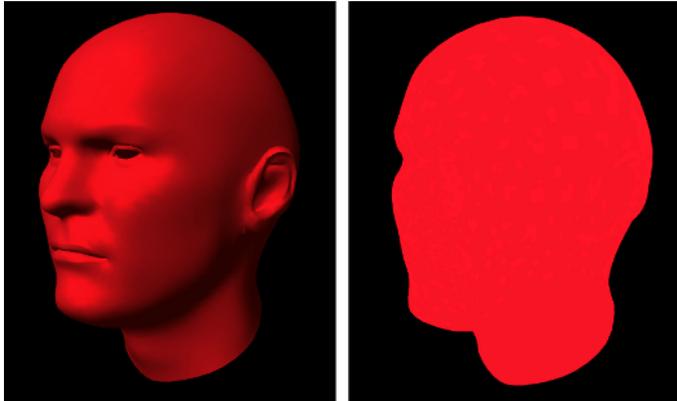
- **Ambient** The area which is not directly lit by any light source. Ambient light surrounds the object and the color looks the same on every side. Since the light does not come from any direction, but instead from “all” directions, the color/light spreads evenly over the object.
- **Diffuse** The object's “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 shows 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.

Material Base Color

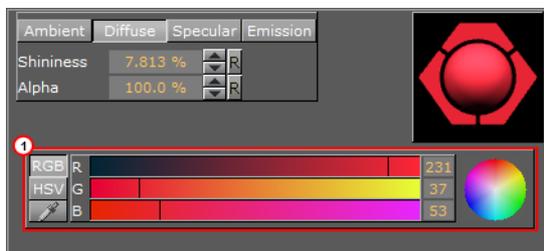
If the Material icon in the Property area (1) is disabled it causes the object's base color to be shown. This is a color without any shading and it is not influenced by light, which means that it is visible even if there are no active light sources. Its initial color is created through a conversion of the material RGB components, but the values can be modified afterwards.



Material Enabled and Disabled



To Change a Material Color



1. In the color selection area (1) select RGB or HSV.
2. In the color selection area, click a color bar, or the color circle, and drag to change the color.

Tip: For RGB only: Right-click on the color sliders and drag to lock all three sliders, so that all values can be changed at once.

or

1. Click on the eye dropper icon in the color field.
2. Move the cursor to pick a color from anywhere on the screen.
3. Click to accept the new color.

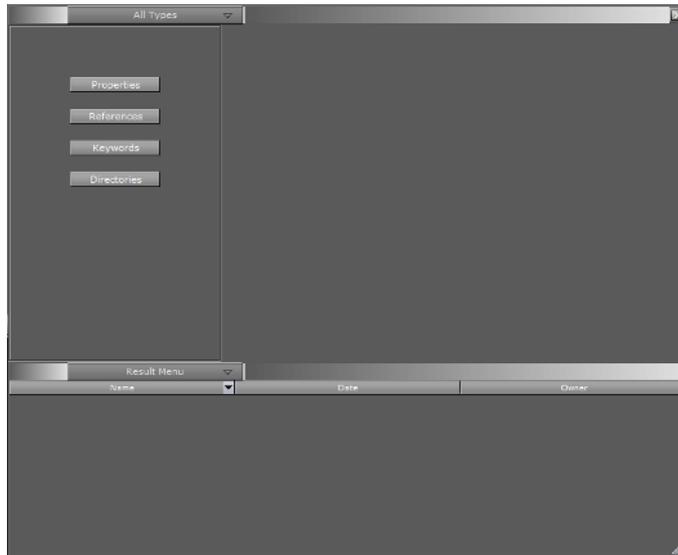
or

- Click a numeric input box and enter a color numeric value.

Tip: Click and hold in the input box and drag left or right to adjust the value.

Tip: Click in the input box and use the up and down arrow keys to adjust the value by a factor of 1. Press and hold the <Alt> key and use the up and down arrow keys to adjust the value by a factor of 10.

5.12 Element Search



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 on four different criteria:

- [Property Search](#)
- [Reference Search](#)
- [Keyword Search](#)
- [Directory Search](#)

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.

- [Combination Search](#)

For information on keywords, see [Assign Keywords to Elements](#).

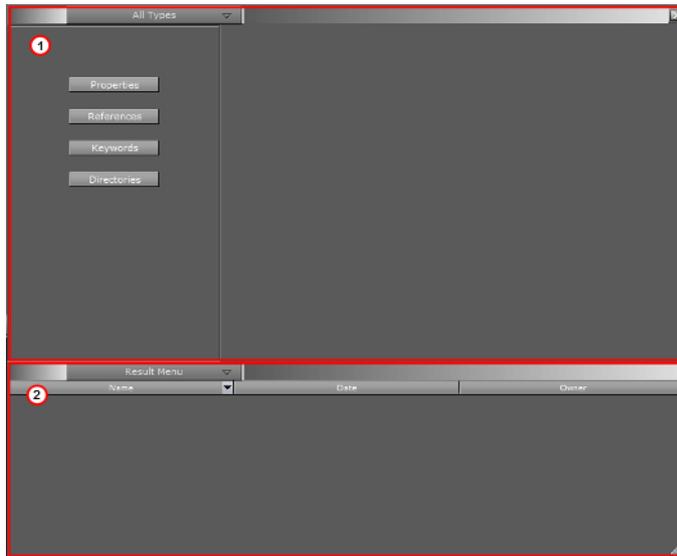
This section contains information on the following topics:

- [Search Panel](#)
- [Search Parameters Area](#)
- [Search Menu](#)
- [Property Search](#)
- [Reference Search](#)
- [Keyword Search](#)
- [Directory Search](#)
- [Combination Search](#)
- [Search Result Area](#)

See Also

- [Working with Elements](#)
- [SmartView Search](#)

5.12.1 Search Panel

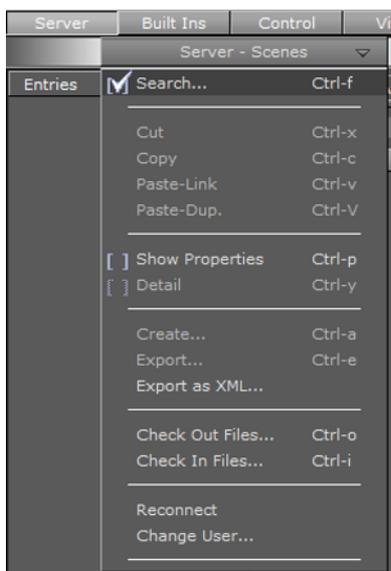


The Search window is divided in two areas:

- [Search Parameters Area](#) (1)
- [Search Result Area](#) (2)

To open the search panel

1. Click **Server** in the [Main Menu Right](#).
2. Open the Search panel:
 - In the [Server Menu](#), click **Search**, or
 - Hold the mouse pointer over the [Server Area](#) and use <Ctrl+F>.



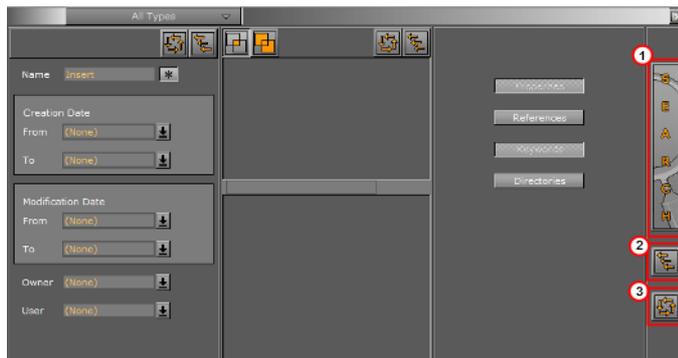
See Also

- [Search Parameters Area](#)

- [Property Search](#)
- [Reference Search](#)
- [Keyword Search](#)
- [Directory Search](#)
- [Combination Search](#)
- [Search Result Area](#)

5.12.2 Search Parameters Area

The Search parameters area is positioned at the top of the Search panel.

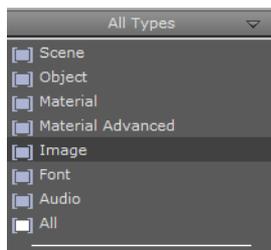


- **Search (1):** Click to start a search.
- **Close All Editors (2):** Close all search editors.
- **New Search (3):** Clear all settings to start a new search.

See Also

- [Search Panel](#)
- [Property Search](#)
- [Reference Search](#)
- [Keyword Search](#)
- [Directory Search](#)
- [Combination Search](#)
- [Search Result Area](#)

5.12.3 Search Menu



The Search menu is positioned at the top of the Search parameters area. Select an Element type from the menu to narrow the Element 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.

See Also

- [Search Panel](#)
- [Search Parameters Area](#)
- [Property Search](#)
- [Reference Search](#)
- [Keyword Search](#)
- [Directory Search](#)
- [Combination Search](#)
- [Search Result Area](#)

5.12.4 Property Search

This editor searches the database for Elements based on the property values entered.

- **Properties Reset:** Clear the settings in the Properties editor to start a new search.
- **Close Property Search:** Close the Properties editor.
- **Name:** Search for Elements with a specified name.
 - Click the asterisk to search with an asterisk wildcard at the start and end of the text string. With the asterisk disabled, only the exact search string is used.
- **Creation Date:** Search for Elements that were created in a specified time period.
- **Modification Date:** Search for Elements that were created in a 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:** Search for Elements that were created by a specified user.

- **User:** Search for Elements that were last modified by a specified user.

Wildcard Text Description

If the full Name is not known then text Wildcards can be used to make a search:

- *: Match any characters. For example 'Element'
- ?: Match any single character. For example 'El?me?t'
- []: Match a range or a list of characters:
 - 'Elem[b-g]nt', or
 - 'Elem[shje]nt'.

See Also

- [Search Panel](#)
- [Search Parameters Area](#)
- [Search Menu](#)
- [Reference Search](#)
- [Keyword Search](#)
- [Directory Search](#)
- [Combination Search](#)
- [Search Result Area](#)

5.12.5 Reference Search

Note: All graphic Elements stored in the Graphic Hub have a Universally Unique ID (UUID). To view an Element's UUID see the [Properties Panel](#).

Any Element can reference to any other Element in the Graphic Hub (see [Referenced Elements](#)). To find out where any given Graphic Hub Element is used, use search by Reference.

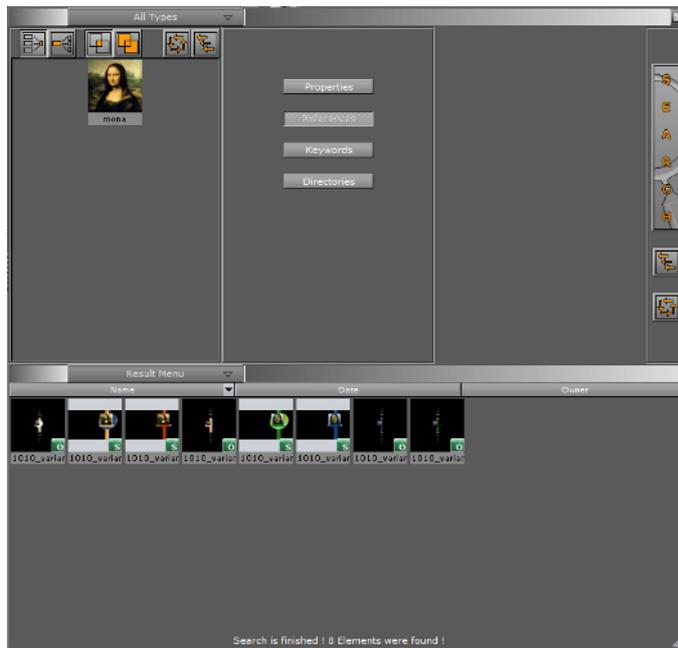
Search by reference can be used to search for:

- Which objects use the selected object(s)
- Which objects are used in the selected object(s)

This section contains information on the following topics and procedures:

- [Reference Search Menu](#)
- [To Search for all Referenced Elements on one Element](#)
- [To Search for all Referenced Elements on Two or More Elements](#)
- [To Delete an Element from a Search by Reference](#)
- [To Search all References on one Element \(Context Menu\)](#)

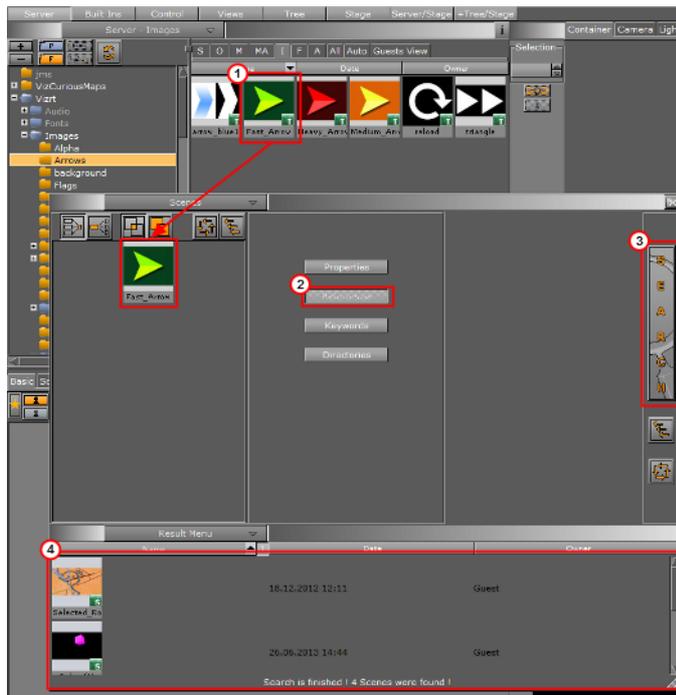
Reference Search Menu



Note: It is possible to add multiple Elements to the References editor.

-  Searches for Elements which reference to the selected Elements, for example, Images, Geometries, Fonts, etc.
-  Searches for Elements which are referenced by the selected Elements, for example, Images, Geometries, Fonts, etc.
-  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.

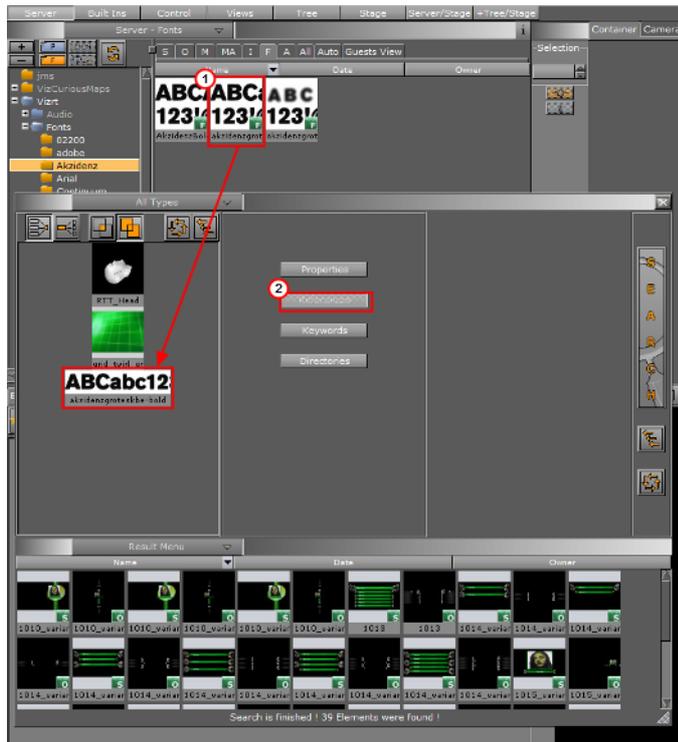
To Search for all Referenced Elements on one Element



1. Open the Search panel:
 - Click on **Search...** in the Server Area drop down menu, or
 - With the mouse pointer over the Server Area, press <Ctrl+F>
2. Click on References in the [Search Panel](#) (2)
3. Select an Element in the [Server Area](#) (1).
4. Drag the Element to the search area.
5. Select to search by:
 - Which objects use the selected object(s) 
 - Which objects are used in the selected object(s) 
6. Click on the Search button (3).

To Search for all Referenced Elements on Two or More Elements

A Reference search can also be completed on more than one Element or Element type.

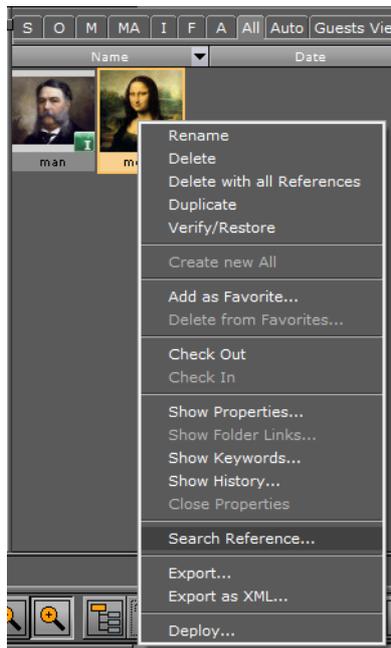


1. Open the Search panel:
 - Click on **Search...** in the Server Area drop down menu, or
 - With the mouse pointer over the Server Area, press <Ctrl+F>
2. Click on References in the [Search Panel](#) (2)
3. Select an Element, or Elements, in the [Server Area](#) (1).

Note: Select from all available Element types, except Materials.

4. Drag the Element(s) to the search area.
5. Specify the search criteria:
 - Search only for references common to all selected Elements
 - Search for references in all selected Elements
6. Select to search by:
 - Which objects use the selected object(s)
 - Which objects are used in the selected object(s)
7. Click on the Search button (3).

To Search all References on one Element (Context Menu)



1. Click on the Element.
2. Do a right-click on the Element to open its context menu.
3. Click **Search Reference...**
4. The [Search Panel](#) opens and shows any Elements which reference the selected Element.

To Delete an Element from a Search by Reference

1. In the [Search Panel](#) right-click the element to be removed.
2. Select **Delete from Reference Search**.



See Also

- [Search Panel](#)
- [Search Parameters Area](#)
- [Search Menu](#)
- [Property Search](#)
- [Keyword Search](#)

- [Directory Search](#)
- [Combination Search](#)
- [Search Result Area](#)

5.12.6 Keyword Search

This editor searches the database for Elements based on keywords.



-  Do a logical AND search of the keywords that are added. The result must be in all searched Elements.
-  Do a logical OR search of the keywords that are added. The result must be in at least one of the searched Elements.
-  Clear the settings in the Keywords editor, to start a new search.
-  Close 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 [Working with Elements](#) .

To Search for a Keyword

1. Type the keyword in the input box.
2. Press <Enter>.



3. Put the required Keyword in the bottom panel:
 - Click and drag the required keyword to the bottom panel, or
 - Click inside the parentheses ([]), to the left of the required keyword.

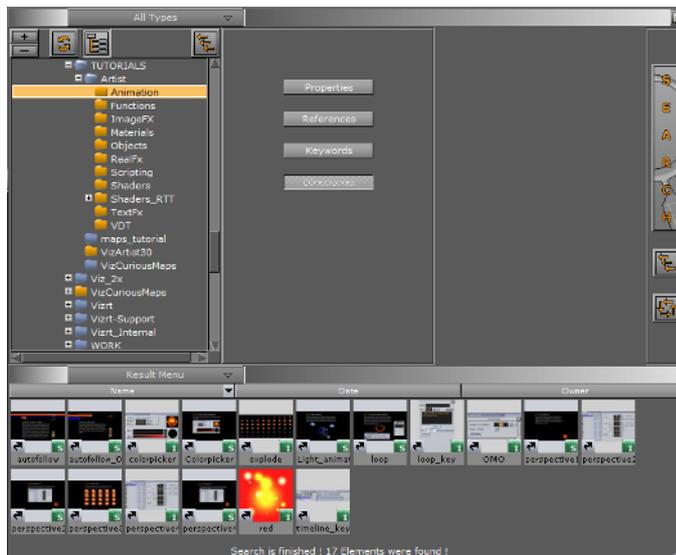
- Click the Search button. The keyword search is based on the keywords that have been moved into the bottom panel.

See Also

- [Search Panel](#)
- [Search Parameters Area](#)
- [Search Menu](#)
- [Property Search](#)
- [Reference Search](#)
- [Directory Search](#)
- [Combination Search](#)
- [Search Result Area](#)

5.12.7 Directory Search

The directory search editor narrows down a search to a selected folder.



-  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 name.

Tip: A branch can also be expanded by clicking the Plus icon  to the left of the folder name, but this will not expand the sub-branches.

-  Collapses the currently selected branch and all its sub-branches.

Note: A branch that can be collapsed has a Minus icon to the left of the folder name.

Tip: A branch can also be collapsed by clicking the Minus icon  to the left of the folder name.

-  Refreshes the tree.
-  Searches for Elements in the selected folder, and recursively in all sub-folders. When disabled, the search is performed in the selected folder only.
-  Closes the Directories editor.

To Search for Elements in Directories

1. Click on the Directories button.
2. Select a folder from the tree.
3. Click the Search button.

Note: If no other search criteria has been entered, when the Search button is clicked the search will show all Elements in the selected folder (including sub-folders).

See Also

- [Search Panel](#)
- [Search Parameters Area](#)
- [Search Menu](#)
- [Property Search](#)
- [Reference Search](#)
- [Keyword Search](#)
- [Combination Search](#)
- [Search Result Area](#)

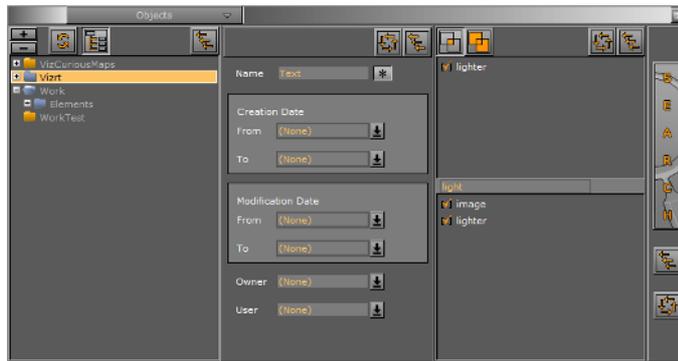
5.12.8 Combination Search

With a Combination search, searches can be further refined.

A combination search uses up to three search parameters to search the Graphic Hub database. Select from:

- [Property Search](#)
- [Reference Search](#)
- [Keyword Search](#)
- [Directory Search](#)

Click on each required search parameter in the [Search Panel](#).



In the example above, the search will be for files that contain the word **Text** which is in the **Vizrt** project folder and sub-folders, and also has a keywords **image** and **lighter** assigned.

See Also

- [Search Panel](#)
- [Search Parameters Area](#)
- [Search Menu](#)
- [Property Search](#)
- [Reference Search](#)
- [Keyword Search](#)
- [Search Result Area](#)

5.12.9 Search Result Area

The Result area is positioned at the bottom of the Search window and shows all search results.



Note: A double-click an Element in the Result area will also select the Element in the Server area.

To Use Searched Elements

Note: To select more than one Element press and hold <Ctrl> and click on each required Element.

Note: The search panel will stay in the foreground until it is closed.

Click on an Element(s) in the Results panel and drag directly to various areas where the Element can be used, i.e.:

- The [Scene Tree](#), or
- The [Favorites Bar](#) , or
- The [Scene Editor](#), or
- The [Container Editor](#)

IMPORTANT! Be careful when moving a linked element, as the element will be moved from the first folder found to the new destination.

To Copy Elements

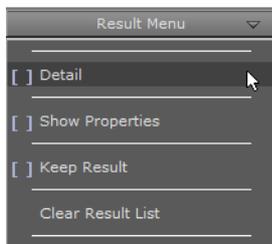
1. Click on an Element(s) in the Results panel
2. Press <Ctrl> and drag the Element(s) from the Result area to the target folder in the Server area.

To Create a Folder Link

1. Click on an Element(s) in the Results panel.
2. Press <Alt> and drag the Element(s) from the Result area to the target folder in the Server area.

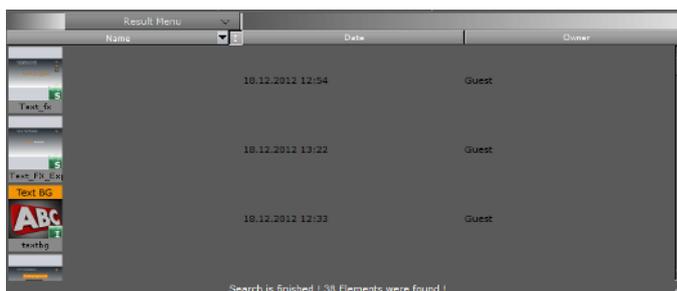
Result Menu

The Result area menu is positioned at the top of the Result area and it defines the appearance of the search results.



- **Detail:** By default, Elements are presented in the Result list with thumbnail and Element name. A vertical list shows, with the additional information of date of creation and user name of element owner.

Note: Click on the 'I' column to enable or disable thumbnails.



- **Show Properties:** Shows the [Properties Panel](#) 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 a new search is done. The results from the new search will be added to the list.
- **Clear Result List:** Clears the Result list.

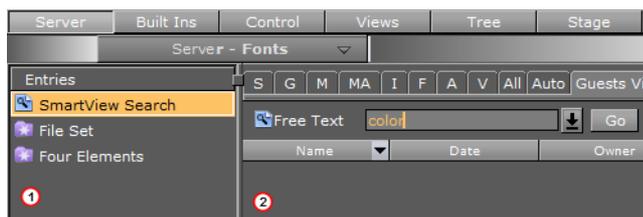
5.13 SmartView Search

A more comprehensive search can be done with the SmartView feature.

Note: This feature is only available with Graphic Hub version 2.2 and up.

A SmartView search includes any or all of the following:

- File names
- Folder names
- File owners
- File UUID (if enabled in the Graphic Hub)
- File type (Geometry, Fontstyle, Images, etc.)
- Assigned keywords



In the Entries column (1) SmartView Search text strings can be saved to use again, and the results of a Smart Search can be saved in [File Sets](#) which are a collection of Elements saved under an assigned name:

-  Saved SmartView Searches
-  File Sets

The SmartView search results are shown in the Server area (2). From the Server area:

- Objects can be dragged directly to where they are needed.
- Double-click on folders to view sub-folders and contents.
- Right-click an object and select **Jump to Server View** to change the Entries column into the Graphic Hub hierarchy. The object is then seen in its home directory.

This section contains information on the following topics:

- [Search String Syntax](#)
- [SmartView Searches](#)
- [File Sets](#)

See Also

- [Element Search](#)

5.13.1 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* project folders 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 build search strings.



Click the Expand button (1), to show the builder search buttons. Click on a button to show or not show a builder search option:

- Scenes
- Geometries
- Materials
- Mat. Adv. (Materials Advanced)
- Images
- Fonts
- Audio
- Project
- Folder

Builder search buttons can also be fast selected. Click on an builder search button with:

- **<Ctrl+Click>**: Selects all
- **<Shift+Click>**: Deselects all
- **<Alt+Click>**: Only show the clicked builder search button

The buttons selected will refine the SmartView search for those particular items.

See Also

- [SmartView Searches](#)
- [File Sets](#)

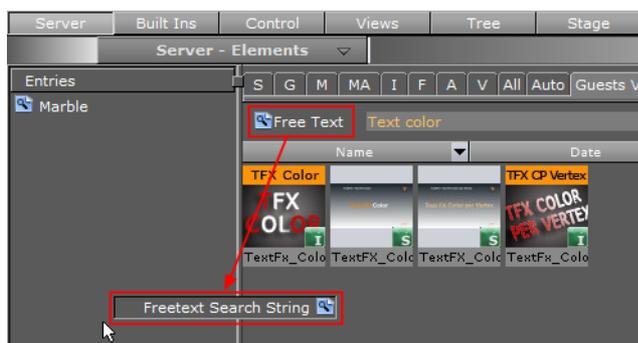
5.13.2 SmartView Searches

To use the SmartView search use the procedures detailed below:

- [To Do a SmartView Search](#)
- [To Save a SmartView Search](#)
- [To Use a Saved SmartView Search String](#)
- [To View Previous SmartView Searches](#)

To Do a SmartView Search

1. Click **Server**.
2. Click the **<Username>s 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>.

**To Save a SmartView Search**

1. Drag the Free Text icon into the Entries column to the left of the SmartView area.
2. When prompted, give the search string a name.
3. Press <Enter>.

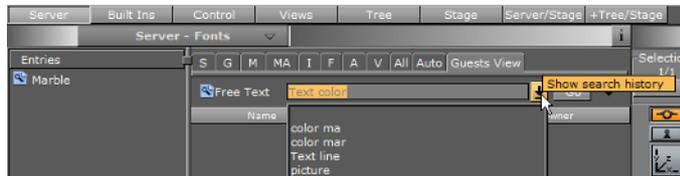
To Use a Saved SmartView Search String

1. Click on a saved search in the Entries column. The saved search is entered in the Free Text field.
2. Click on **Go**.

or

- Double click a saved search string the Entries column. The saved search is entered in the Free Text field and search beings automatically.

To View Previous SmartView Searches



- Click the arrow to the right of the Free Text field to view all SmartView searches since Viz Artist was launched.

See Also

- [Search String Syntax](#)
- [File Sets](#)

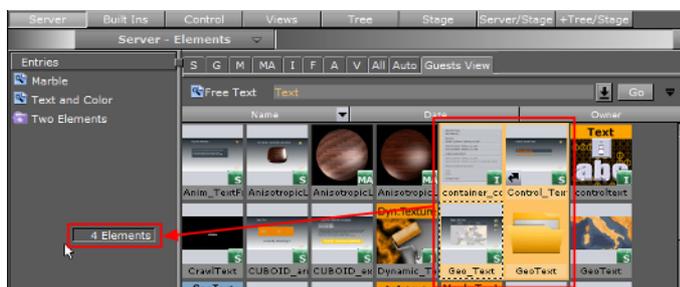
5.13.3 File Sets

File sets are a collection of objects saved under an assigned name. File sets are unique to each user on each instance of the Graphic Hub. Placing objects and folders in a file set does not move them from their location in the Graphic Hub hierarchy; rather it just makes them all accessible from the SmartView tab.

This section contains information on the following topics and procedures:

- [To Save a SmartView File Set](#)
- [To Add Files to a SmartView File Set](#)
- [To Remove Files from a SmartView File Set](#)
- [To Add an Empty SmartView File Set](#)
- [Entries Column Context Menu](#)

To Save a SmartView File Set

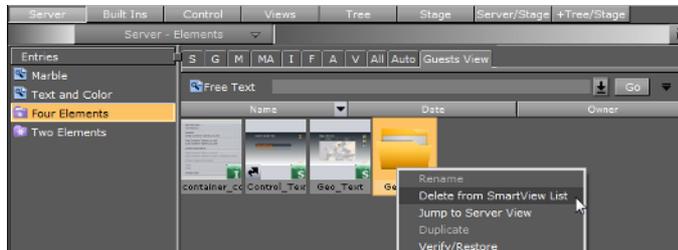


1. Select all required objects from a search result and drag them into the Entries column.
2. When prompted, enter a name for the file set
3. Click **OK**.

To Add Files to a SmartView File Set

- Select all required objects from a search result and drag them into the target file set in the Entries column.

To Remove Files from a SmartView File Set



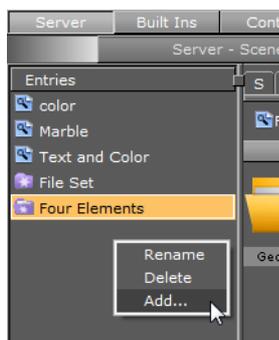
1. In the Entries column, click the relevant SmartView file set.
2. In the result area, select the required objects
3. Remove the selected results from the SmartView file set:
 - Drag the objects to the Trash Can, or
 - Right-click the objects and select **Delete from SmartView list**.

To Add an Empty SmartView File Set

1. Do a right-click in the Entries column.
2. Select **Add...**
3. Enter a name for the new Smart Search file set.
4. Click **OK**.

Entries Column Context Menu

Highlight a Saved search or a file set and right-click to open a context menu.



- **Rename:** Rename the highlighted Saved search or file set.
- **Delete:** Delete the highlighted Saved search or file set.
- **Add...:** Add a new File Set.

See Also

- [Search String Syntax](#)
- [SmartView Searches](#)

5.14 Background Loading

The background loading of Geometries, Materials and Images is supported in Viz Artist (with limitations). Background Loading can either be an automatic or a manual operation.

This section contains the following topics:

- [Automatic Background Loading](#)
- [Manual Background Loading](#)
- [Background Loading Limitations](#)

5.14.1 Automatic Background Loading

When automatic background loading is set-up, Viz Engine automatically uses background loading whenever an image, and/or an object, is loaded (with some limitations (see [Background Loading Limitations](#)).

This section contains the following topics:

- [Automatic Background Loading Configuration](#)

Automatic Background Loading Configuration

In the Viz Config file the option **Background_Loading_Level** is available to set up automatic background loading.

.....
Example: Path to the Viz Config file: <viz data folder>/VIZ-Z420-DCO-0-0.cfg

.....
IMPORTANT! When background loading of Geometries VBO must be enabled in **Render Options** of the [Viz Configuration](#) (see the [Viz Engine Administrator Guide](#)).

Available values for Background_Loading_Level are:

Value	Description
BGL_INACTIVE	Disables automatic background loading
BGL_IMAGES	Enables automatic background loading for images
BGL_OBJECTS_AND_IMAGES	Enables automatic background loading for images, geometries and materials

5.14.2 Manual Background Loading

Manual background loading can be done on an on demand basis. To do this commands are available to use Container objects:

Command	Description
GEOM SETBGL	Load (and set) a geometry in the background. The usage of the command is exactly the same as the existing GEOM SET command.
IMAGE SETBGL	Load (and set) an image in the background. The usage of the command is exactly the same as the existing IMAGE SET command.
MATERIAL SETBGL	Load (and set) a material in the background. The usage of the command is exactly the same as the existing MATERIAL SET command.

This section contains the following topics:

- [Manual Background Loading Configuration](#)

Manual Background Loading Configuration

To use manual background loading, the **Background_Loading_Level** config option must be set to **BGL_INACTIVE**.

Example: Path to the Viz Config file: *<viz data folder>/VIZ-Z420-DCO-0-0.cfg*

IMPORTANT! When background loading of Geometries VBO must be enabled in **Render Options** of the [Viz Configuration](#) (see the [Viz Engine Administrator Guide](#)).

5.14.3 Background Loading Limitations

This section details the limitations of background loading.

This section contains the following topics:

- [Scenes](#)
- [Compressed and Un-compressed Images](#)
- [Objects](#)
- [Function Plug-ins](#)
- [Image combining](#)
- [Black/Transparent flashing image textures](#)

Scenes

Background loading of whole Viz Artist Scenes is not supported.

Compressed and Un-compressed Images

Only compressed images (DXT) are supported for background loading. Their mipmap-levels have been pre-generated and stored in the Graphic Hub, which is required for background loading to work properly. The generation of mipmap levels on-demand (as required for un-compressed images) can interrupt the renderer. This primarily depends

on the size of the un-compressed images. The larger the image, the more impact on the renderer during background loading.

Objects

Only native geometry and imported third-party geometry objects containing vertex-data, materials and textures are supported targets for background loading. Containers hosting any sort of other plugin may fail to load in the background. Containers hosting Function plug-ins will NOT be loaded in the background but rather fall back to a normal foreground loading. Containers hosting RTT shaders will interrupt the renderer if the particular RTT shader is used for the first time (shader is being compiled), but will no longer interrupt the renderer when loading the same shader in the background again.

Function Plug-ins

If a geometry, which is to be loaded in the background, contains one or more Function Plug-ins, the console will show a warning and the geometry will not be loaded in the background but in the foreground instead, this can interrupt the renderer.

Image combining

If a geometry uses the second image (image2) in a texture, this might have an impact on the performance of the renderer. Mostly if one of the images are too large and/or un-compressed.

Black/Transparent flashing image textures

When setting an image with background loading, for technical reasons the renderer will get the handle to a black (or transparent) texture until the first mipmap level of the image is loaded. This can result in a short flash of black (or transparent) texture while background loading textures on a visible object.

5.15 Built Ins

In Built Ins there are a standard set of Viz Artist installation plugins.



Most plugins can be added to [Containers](#), except, for example, Scene plugins. Because Viz Artist uses an open interface for the plugins, new geometries can either be obtained or created and then installed with Viz Artist. Also, most plugins have a properties editor.

When a plugin is applied to a Container, in the Scene Tree, a plugin icon will show in the target Container (see [Container Icons](#)).

The Built-Ins button on the [Main Menu Left](#) gives access to the various plugins:

- [Geometry Plugins](#)
- [Container Plugins](#)

- [Shader Plugins](#)
- [Scene Plugins](#)
- [Media Assets](#)

This section contains information on the following procedures:

- [Content Tabs](#)
- [Plugin Context Menu](#)
- [View and Search for Plugins](#)
- [Add Plugins to Containers](#)

5.15.1 Content Tabs

The Built Ins are contained under various tabs and in folders.



- [Geometry Plugins \(GP\)](#)
- [Container Plugins \(CP\)](#)
- [Shader Plugins \(SH\)](#)
- [Scene Plugins \(SP\)](#)
- [Media Assets \(MA\)](#)

Click on a Built In tab to open its panel and folder list.

Double-click a Built In tab to open its panel and go to the default folder, in the folder list. Default folders are:

- **Geometry Plugins (GP):** Default
- **Container Plugins (CP):** Global
- **Shader Plugins (SH):** Default
- **Scene Plugins (SP):** Default
- **Media Assets (MA):** Channel

5.15.2 Plugin Context Menu

A right-click on a plugin icon to open the plugin context menu.

This section contains information on the following procedures:

- [Geometry, Container and Shader Menu](#)
- [Scene Menu](#)
- [Media Asset Menu](#)

Geometry, Container and Shader Menu



- **Add to Container(s):** Add the selected plugin to a Container(s)
- **Add as Favorite:** Add the selected Texture to the [Favorites Bar](#).
- **Delete from Favorites:** Delete the selected Texture from the [Favorites Bar](#).

Scene Menu



- **Add to Scene:** Add the selected plugin to a Scene
- **Remove from Scene:** Remove the selected plugin from a Scene

Media Asset Menu



- **Add <Media Asset> as Texture:** Add the selected Media Asset to a Scene (automatically adds a new Container). See [Media Assets as Texture](#)
- **Add <Media Asset> as DVE:** Add the selected source as DVE (see [Media Assets as DVE](#)).
- **Show settings of <Media Asset>:** Show the settings of the Media Asset (see [Media Asset Properties](#)).

Note: Show settings of...: Only available when the selected Media Asset has been added to a Scene, and the Scene Tree is in Texture or DVE view.

- **Add as Favorite:** Add the selected source to the [Favorites Bar](#).
- **Delete from Favorites:** Delete the selected source from the [Favorites Bar](#).

The [Background and Foreground Media Assets](#) Media Assets have a limited context menu:

- **Add as Favorite:** Add the selected source to the [Favorites Bar](#).
- **Delete from Favorites:** Delete the selected source from the [Favorites Bar](#).

5.15.3 View and Search for Plugins

This section contains information on the following procedures:

- [To View Plugins](#)
- [Add Plugins to Containers](#)
- [Plugin Context Menu](#)

To View Plugins

1. Click **Built Ins**.
2. Select the appropriate plugin category, either:
 - From the drop-down menu that shows above the plugins view, or
 - Click on of the tabs, or
 - <Ctrl+1>: Geometry Plugins
 - <Ctrl+2>: Container Plugins
 - <Ctrl+3>: Shaders
 - <Ctrl+4>: Scene Plugins
 - <Ctrl+5>: Media Assets

Tip: Click the **All** folder in left panel shows all plugins, without any categorization.

Tip: Press a letter on the keyboard to jump to a plugin which starts with that letter.

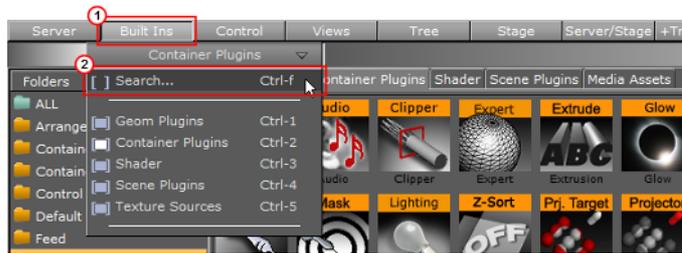
Tip: Double-click a folder tab to go to the **Default** folder for that plugin category.

Tip: All plugins can be dragged straight into the Scene Tree (except Scene plugins).

To Search for a Plugin

1. Click on **Built Ins** (1).
2. Open the plugin search panel:
 - Click on the **Built Ins** drop-down menu and click on **Search...** (2), or

- Press <Ctrl+F>.



IMPORTANT! In the plugin search panel enter a text search string (3).

3. Click on **Go** (4), or press <Enter>.



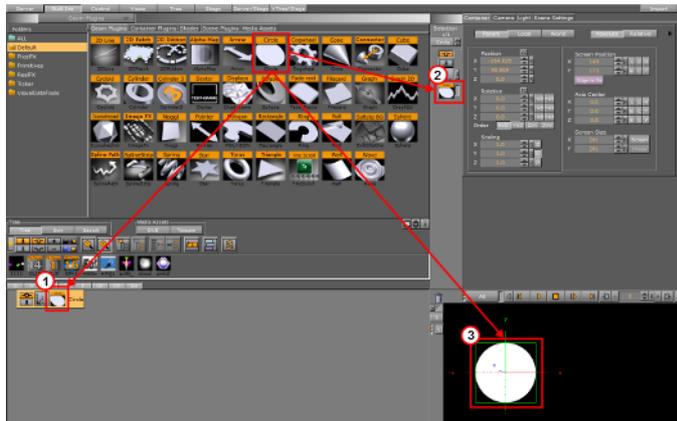
Note: The searched plugins have the same Add and Context menu functions as the plugin icons in the plugin tabs (see [Built Ins](#)).

Note: The plugin can be dragged to a Container in the Properties Panel.

Note: Only some Container plugins will drag to the [Scene Editor](#), when they will the target object becomes active when the icon is moved over the object

Note: When a Media Asset or a Geometry plugin is added to the [Scene Editor](#) a Container is created in the [Scene Tree](#) for the Media Asset or the Geometry plugin.

5.15.4 Add Plugins to Containers



Most plugins can be dragged to a target Container(s) in the [Scene Tree](#) (1), the Properties Panel (2) or the [Scene Editor](#) (3). Each plugin also has a context menu where a plugin can also be added to a Scene.

This section contains information on the following procedures:

- [To Add a Geometry plugin](#)
- [To Add a Container Plugin](#)
- [To Add a Shader](#)
- [To Add a Scene Plugin](#)
- [To Add a Media Asset](#)
- [To Add a Plugin to Multiple Containers](#)
- [To Add a Plugin to Favorites](#)
- [To Open a Plugin Properties Editor](#)

To Add a Geometry plugin

1. Drag the Geometry plugin icon to either:
 - A Container or multi selected Containers
 - The [Scene Tree](#) area (creates its own container)
 - The [Scene Editor](#) area
 - The [Container Editor](#) area

or

1. Highlight the target Container(s)
2. Right-click the plugin icon and select **Add to Container(s)** from the [Plugin Context Menu](#)

Note: A Container can only contain one Geometry plugin at a time. If a new Geometry plugin is added the first one will be removed from the Container.

To Add a Container Plugin

1. Drag the Container plugin icon to either:

- A Container or multi selected Containers
- An object in the Scene Editor (only some container plugins will do this, when they will the object becomes active when the icon is moved over the object.
- The [Container Editor](#) area

or

1. Highlight the target Container(s)
2. Right-click the plugin icon and select **Add to Container(s)** from the [Plugin Context Menu](#)

To Add a Shader

1. Drag the icon to either:
 - A Container or multi selected Containers
 - An object in the Scene Editor
 - The [Container Editor](#) area

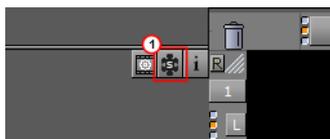
or

1. Highlight the target Container(s)
2. Right-click the plugin icon and select **Add to Container(s)** from the [Plugin Context Menu](#)

Note: A Container can only contain one Shader at a time. If a new Shader is added the first one will be removed from the Container.

To Add a Scene Plugin

1. Open the Plugin tab in [Scene Settings](#):
 - Click on *Scene Settings* -> *Plugin*
 - Click on the Scene plugin icon (1)



2. Drag the Scene Plugin icon to the Scene drop area.

or

1. Right-click the Scene Plugin icon to open the [Plugin Context Menu](#).
2. Click on **Add to Scene...**

Tip: If a Scene Plugin has been applied to a Scene, the Scene Plugin icon will show orange .

To Add a Media Asset

See [To Add a Media Asset as DVE](#)

See [To Add a Media Asset as Texture](#)

See [Video Playback](#) for more information about Media Assets

To Add a Plugin to Multiple Containers

1. Highlight the required Containers.
2. Drag a plugin icon to a highlighted Container
3. Release the Media Asset icon.

or

1. Drag a plugin icon.
2. Hold the plugin icon over a Container.
3. Continue to hold the left-click and do a right-click to drop the plugin icon on the Container.
4. Continue to hold the left-click and repeat for each required Container

To Add a Plugin to Favorites

1. Right-click the icon to open the [Plugin Context Menu](#).
2. Click **Add as favorite...**

Note: Plugins can also be added as a Favorite from the search panel ([To Search for a Plugin](#)).

To Open a Plugin Properties Editor

- Click on the Media Asset icon, in the Container or the in the Properties Editor.

See Also

- [Built-in View Shortcuts](#)

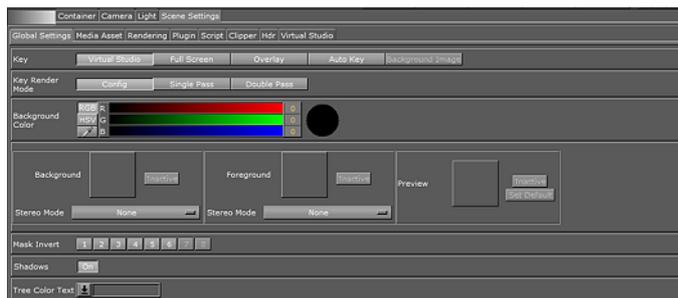
6 Global Tasks

This section contains information on the following topics:

- [Scene Settings](#)
- [Video Clip Playback Considerations](#)
- [Video Playback](#)
- [Supported Codecs](#)
- [Import Files and Archives](#)
- [Archiving](#)
- [Deploy Elements](#)
- [Post Rendering](#)
- [Integration with 3rd Party Software](#)

6.1 Scene Settings

To open the Scene Settings editor, select the Scene Settings tab in the Property panel. 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:

- [Global Settings](#)
- [Media Asset Properties](#)
- [Rendering](#)
- [Plugin](#)
- [Script](#)
- [Clipper](#)
- [HDR \(High Dynamic Range\)](#)
- [Virtual Studio](#)

See Also

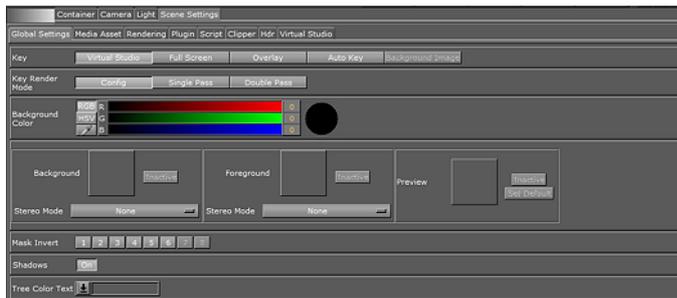
- [Video Playback](#)

6.1.1 Global Settings

This section contains information on the following topics and procedures:

- [Global Settings Properties](#)
- [Images as Background or Foreground](#)

Global Settings Properties



In the Global Settings panel the basic parameters for a Scene can be modified:

- **Key:** Defines how the alpha key signal should be generated. The default is Overlay.



- **Virtual Studio:** This option is relevant only in combination with the Viz Virtual Studio 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](#)). In a Virtual Studio set, to do a camera flight with the virtual camera when the real camera is stationary, the graphics must be set to **Full Screen** to avoid a situation where the virtual and the real camera have different position values. When switching the camera from Remote to Editor to do the flight, the keyed graphics automatically switches from Overlay to Full Screen
- **Full Screen:** Outputs non-transparent key signal
- **Overlay:** Sets as key signal only those Containers that have a [Key](#) function applied. Only Containers in the Scene Tree with a Key function applied 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, it does not require the application of [Key](#) functions to the Container. This function provides 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 applied will be rendered on top
- **Key Render Mode:**

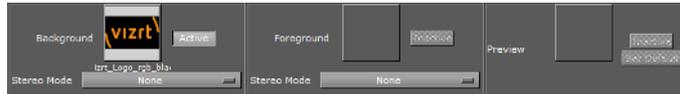


- **Config:** Renders the key according to the settings in the [Viz Configuration](#).
- **Single Pass:** Uses one rendering step. Use this for all new scenes and for scenes imported from version 2.8, as it is much faster.
- **Double Pass:** Uses two rendering steps. Use for backward compatibility in scenes created in Viz Artist versions 3.0 to 3.3.

- **Background Color:** Sets the color of the background. For information about how to select a color (see [Material Color Properties and Schemes](#)).



- **Image and Media Asset Placement** (See also [Images as Background or Foreground](#) and [Media Asset Workflow](#)):



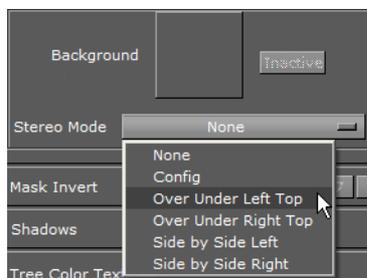
- **Background:** Add an image or Media Asset as the Scene background
- **Foreground:** Add an image or Media Asset as the Scene foreground

Note: If the foreground image or Media Asset has alpha, the Scene can be shown through the image or Media Asset.

- **Preview:** Add a fixed layout to the Scene. This visualizes how the total preview (camera picture and/or graphics/Media Asset) will look like. To set the image as global default, click the **Set Default** button.

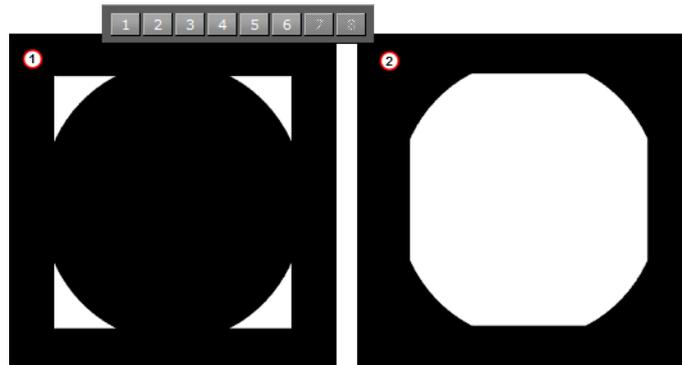
Tip: It is possible to drag results from a database search onto the Background, Foreground, and Preview drop zones.

- **Stereo Mode:** (license required) Select the correct stereo mode for the inserted image (not available if **Stereo Mode** is set to **Off** in the Viz Configuration (see [Camera in Viz Configuration](#)):
 - **None:** No Stereo mode
 - **Config:** Images are drawn as set in the Viz Configuration (see [Camera in Viz Configuration](#))
 - **Over Under Left Top** or **Over Under Right Top:** Two images are stored, one over (top) and the other under (bottom), **left or right eye (top) first**. The image height will be halved, providing half resolution
 - **Side by Side Left** or **Side by Side Right:** **Two images are stored, side by side, left or right eye first. The image width will be halved, providing half resolution**



- **Mask Invert:** Makes it possible to set one or more of the mask layers in a Scene to be inverted. With no mask inverted (1), the mask cuts out a hole where it covers

the mask target object. With mask invert enabled (2), the mask target is only visible where the mask covers it:



- **Shadows:** Enables or disables the shadow function in a 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: 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. Add text to the global scene tree Colors in User Interface of Viz Configuration (see the Viz Engine Administrator Guide). To create customized colors for the scene tree, see [Favorites Bar](#)

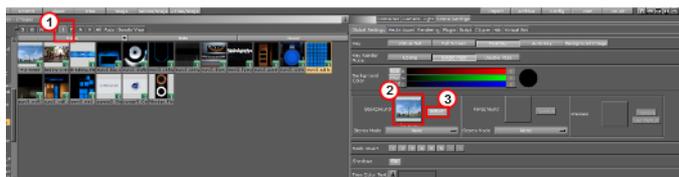


Note: This text is Scene specific.

Images as Background or Foreground

This section details procedures with images use as background, foreground and preview:

- [To Add an Image](#)
- [To Remove an Image](#)
- [To View an Image Source Project or Folder](#)
- [To Show/Hide an Image in the Scene Editor](#)



To Add an Image

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

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 Source Project or Folder

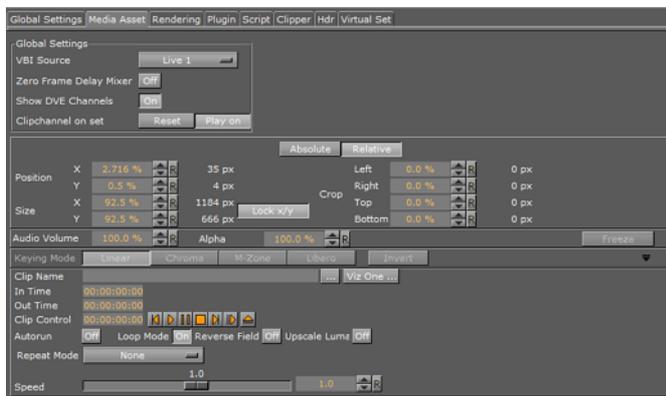
- Click the image on the drop zone.
The project or folder it is placed in is visible. If the image is linked, the first project or folder will be selected.

To Show/Hide an Image in the Scene Editor

- Click the Active/Inactive button next to the drop zone (3).
If the image has a different aspect ratio than the screen resolution, the image will be stretched to fit.

6.1.2 Media Asset Properties

Most of the settings for video playback are set in the Properties panel for the selected Media Asset (*Scene Settings -> Media Asset*).



This section contains information on the following topics:

- [To View Media Asset \(DVE or Texture\) Properties](#)
- [To View Media Asset \(Texture\) Properties](#)
- [Common Media Assets Properties](#)
- [Clip Properties](#)
- [Live Properties](#)
- [Stream Properties](#)
- [GFX Properties](#)

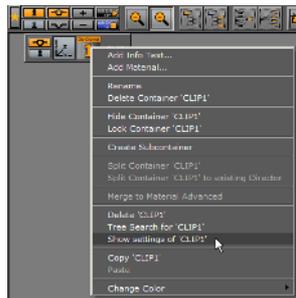
To View Media Asset (DVE or Texture) Properties

1. Add a Media Asset as [DVE](#) or [Texture](#) to a Scene.
2. In the Media Asset Manager select **DVE** or **Texture** (to view the added Media Asset)
3. Click on the required Media Asset.

Note: DVE or Texture view: If a Media Asset has been added to a Scene, its properties can also be selected by a <right-click> on the Media Asset icon in the Media Asset list (Built-ins), and select **Show settings of <clip channel>**.

To View Media Asset (Texture) Properties

1. <Right-click> on the Media Asset icon in a Container.



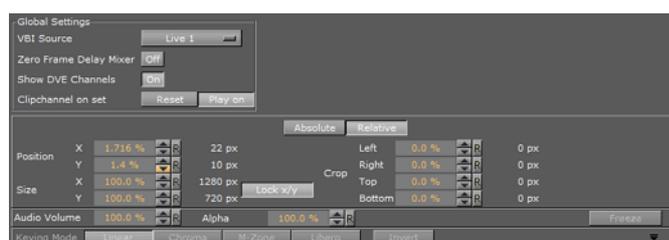
2. Click on **Show settings of [Media Asset]**.

Or:

1. Click on the Media Asset icon in the Container.
2. In the [Texture Editor](#) click on the Media Asset image icon.

In the properties panel the Media Assets tab is automatically selected, and the Media Asset Manager (Scene Tree panel) opens

Common Media Assets Properties



- **Global Settings:**
 - **VBI Source:** Select which channel to use for output. Types of data sent include closed captions, teletext, time codes and other digital data
 - **Zero Frame Delay Mixer:** Set to **On** to give the shortest input to output delay, with graphics on top of it

Tip: To work correctly, specific values for H-/V- phase must be set (see [H-/V- Phase Values for Zero Frame Delay Mixer](#)).

Note: Inputs and outputs are tight bound to: input A to output A and input B to output B.

Note: The resolution of the input must be the same as the output.

Note: A Dual Channel setup is only supported if input A and input B are a 100% in sync, and both channels use the same resolution.

- **Show DVE Channels:** Enable or disable the DVE output
- **Clip Channel on set: Play on:** By default a clip channel continues to play the clip, even the Scene does not contain that specific channel. When setting a Scene with this channel active, the clip will be shown from its current position onwards, when no clip or the same clip is set in the Scene. **Reset:** Play will start from the beginning of the clip, when it's the same clip name, and will show black, when no clip is set in the channel at all.
- **Absolute or Relative:** Switch. If you select to work in *Relative* mode you provide the parameters for Position, Size and Crop in **percent**. The corresponding value in pixels (px) will be shown in the information field next to the input value field. Similarly, if you select to work in *Absolute* you provide the values in **pixels** and the corresponding value in percent will be shown in the information field.
- **Position Controls:** (*DVE only*) Use the available controls to position the DVE
 - **Position**
 - **Size**
 - **Crop**
- **Audio:** Set the Audio volume
- **Alpha:** For each layer adjust the level of opacity (100% is fully opaque and 0% is fully transparent)
- **Freeze:** (*Texture only*) Freeze the video (helps to make the selection of colors easier, especially around edges, etc.)
- **Keying Mode** (*Texture only*: see [Keying Mode for Video](#)):
 - **Linear:** No chroma keying
 - **Chroma:** Uses the internal chroma keyer
 - **M-Zone:** Uses the multi-zone keyer
 - **Libero:** Uses the Viz Libero keyer
 - **Invert:** Only works with Chroma and Libero

H-/V- Phase Values for Zero Frame Delay Mixer

For X.mio2, DSX.LE2 and DSX.LE3

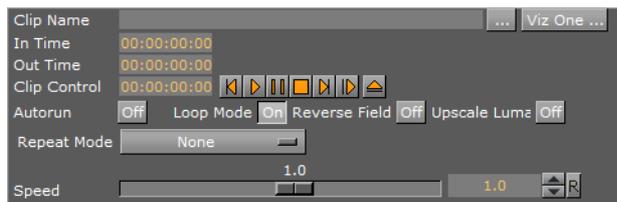
- **PAL:** Leave the values at the default
- **NTSC:** Leave the values at the default
- **720p50:** Valid H-Phase range from -900 to -1300
- **720p59.94:** Valid H-Phase range from -600 to -1000
- **1080i25:** Valid H-Phase range from -1600 to -2000
- **1080i29.97:** Valid H-Phase range from -1100 to -1600

See Also

- [Video Playback](#)
- [Clip Properties](#)
- [Stream Properties](#)
- [GFX Properties](#)

Clip Properties

In addition to the [Common Media Assets Properties](#), Video Clip also has these properties:



- **Clip Name:** Select the video clip to be shown:
 - Click on the **...** to open the Video Clip directory.
 - Click on **Viz One ...** to import video from Viz One.

Note: With the Clip Channel Media Asset it is possible to play images as DVE or Texture with the Matrox clip player. Supported image containers are jpg and png. The images have to be available on a physical disk for this feature to work (see [Video Clip Playback Use Cases](#)).

- **State Active Clip:** Show the status of the active clip:
 - Green 'traffic light' shows that the clip has loaded successfully.
 - Grey 'traffic light' shows that no clip has been loaded.
 - Red 'traffic light' shows that there was an error during the clip loading process.
- **State Pending Clip:** Show the status of a pending clip:
 - Green 'traffic light' shows that the clip has loaded successfully.
 - Grey 'traffic light' shows that no clip has been loaded.
 - Red 'traffic light' shows that there was an error during the clip loading process.
- **In Time:** Set the video clip 'in time'
- **Out Time:** Set the video 'out time'
- **Clip control:** Player controls for the selected clip
- **Autorun:**
 - **Off:** As soon as the video clip is loaded, the first frame is presented
 - **On:** As soon as the video clip is loaded, it begins to play
- **Loop Mode:**
 - **Off** The video clip will play only once.
 - **On** The video clip will play in a loop.
- **Reverse Field:**

- **Off:** When backward playback is used (speed is negative), the fields in interlaced mode are played out 'as is'
- **On:** The first as second fields in interlaced mode are swapped
- **Upscale Luma:**
 - **Off:** The key will be used 'as is'.
 - **On:** The key information will be upscaled from 16-235 to 0-255
- **Repeat Mode:**
 - **None:** The playback of black frames when the clip is in an erroneous state
 - **Field:** The first field of the last frame will be played out when the clip is paused
 - **Frame:** 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.

Live Properties

The Live video properties are listed in the [Common Media Assets Properties](#) (see also [Live Video Media Asset](#)).

Stream Properties

In addition to the [Common Media Assets Properties](#), Video Stream also has these properties (see also [Video Stream Media Asset](#)):



- **URL:** Enter the URL of the Stream to be shown (Video Stream only):
 - **Load:** Click to load the video Stream
 - **Clear:** Click to remove the video Stream.
- **Repeat Mode:**
 - **None:** The playback of black frames when the clip is in an erroneous state
 - **Field:** The first field of the last frame will be played out when the clip is paused
 - **Frame:** The last frame will be played out when the clip is paused

GFX Properties

In addition to the [Common Media Assets Properties](#), GFX Channels also have these properties (see also [GFX Channel Media Asset](#)):



- **Scene:** Drag a scene, which should be rendered in the GFX channel
- **Play Controls:** Control the animation of the set scene
- **Controlled:** If activated the GFX Channel animation can be controlled by the main or parent Scene animation controls, for example, if the animation in the main Scene starts, it triggers the start of the GFX Channel Scene

- **Force Key:** If a GFX channel is rendered as Texture, it will be in RGBA. Therefore the Scene needs to have keying information for an alpha channel (the key scene setting is set to auto key or a key plugin is used in the Scene Tree). If the Scene does not have any keying information, enable **Force Key** to render the whole Scene keyed
- **Keep Aspect:** If **On** the GFX Scene is rendered with its own aspect ratio (undistorted). If Keep Aspect is **Off** then the Scene will be scaled to fit the GFX channel size.
- **Camera:** Select the camera view of the GFX Channel Scene (see [Camera Views](#))

Note: By default the GFX Channel automatically takes the camera settings of the Scene it is applied to.

See Also

- [Video Playback](#)

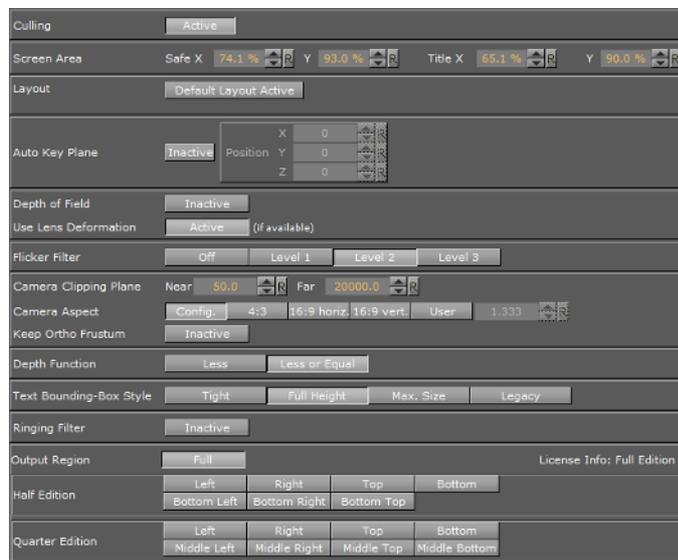
6.1.3 Rendering

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

This section contains information on the following topics and procedures:

- [Rendering Properties](#)
- [To Design Graphics with a User Defined Aspect Ratio](#)

Rendering Properties



- **Culling:** With Culling **Active** for a Scene it saves performance. Objects whose bounding box is outside the camera view are not rendered. This is of great importance to performance in large scenes and virtual studio. Normally Culling should be **Active**.
- **Screen Area:**
 - **Safe X/Y:** Set the size of the Safe area X and Y values in percent of the screen resolution. Single click the **R** button on the X or Y to reset to its default value. Double-click the **R** button to reset both values to default values.

- **Title X/Y:** Set the size of the Title area X and Y values in percent of the screen resolution. Single click the **R** button on the X or Y to reset to its default value. Double-click the **R** button to reset both values to default values.
- **Layout:** The Video Wall Distributor defines a layout of the Viz Engine on air to be part of a cluster of Viz Engines:
 - **Default Layout Active:** Click to use the video wall layout, as set in the Video Wall Distributer.
 - **Use Layout:** Click to make the Video Wall layout inactive

Note: Both these parameters are set in the Video Wall Distributer.

- **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. 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 Studio 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 Viz Virtual Studio documentation.

- **Depth of Field:** Emulates the behavior of a real camera where objects further away from the focal plane show blurred and objects close to the focal plane show sharp. The respective parameters can be set in the camera settings under the Focus settings (see [Camera Editor](#)).
- **Use Lens Deformation:** By default lens deformation is applied to a Scene, if these conditions apply:
 - Render scale in the Viz Configuration is set to greater than one, and a camera is used with a lens file, with distortion loaded. This setting forces a Scene to be rendered without lens deformation

Note: When Lens Deformation is active, the [Glow](#) plugin and Cube Mapped images will not work correctly.

- **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.
- **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 [Viz Configuration](#) (see the [Viz Engine Administrator Guide](#)).
 - **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 **Video Wall**, set the user-defined camera aspect to the same value as the **Aspect** shown in the [Viz Configuration](#) (**Output Format** -> **User Defined** -> **Aspect** (which is set dynamically based on the other values)). When building a Scene with a customized aspect ratio for other, e.g. hand-held/mobile devices or online publishing, set the **Keep Editing Aspect (Render Options** in [Viz Configuration](#)) parameter to **Editor** as this will not have an affect on the On-Air or Viz Engine output (see also [To Design Graphics with a User Defined Aspect Ratio](#)).
- **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:** The bounding box is drawn tight around all text used in this Scene. This means just the height of the actual text used
 - **Full Height:** The bounding box is higher. The bounding box is calculated 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:** For Scenes imported from version 2.8 so that the various plugins which depend on font bounding boxes are compatible in version 3.5 and on, 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 (see **Font Options** in [Viz Configuration](#) (see the [Viz Engine Administrator Guide](#))).

- **Ringling Filter:** Reduces the ringing artifacts on the SDI output and applies only to SD formats. The ringing filter can be set to Off (see **Render Options** (set **Force Ringling Filter Off** to **Active**) in [Viz Configuration](#) (see the [Viz Engine Administrator Guide](#))).

Note: The default Ringling Filter is located in **Scene Default Values** ([Viz Configuration](#)).

- **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. 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. 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](#) (or press <F11>).
3. If you plan to render graphics in many different aspects (e.g. still images for online publishing) you should select the **Render Options** section and set *Keep Editing Aspect* to **Editor**.
 - This will make sure 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 the screen resolution and aspect (e.g. 16:9).
 - 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. Close Viz Artist.
6. Start Viz Artist.
7. Click the **Server** button.
8. From the Properties Panel click the [Scene Settings](#) tab.
9. Under the [Rendering](#) tab set **Camera Aspect** to **User**.
10. 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

- [Camera Controls and Shortcuts](#)

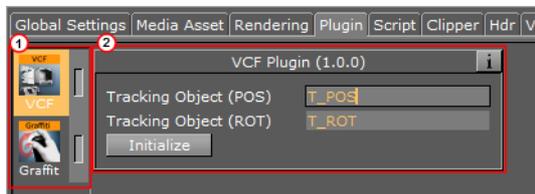
6.1.4 Plugin

Scene plugins can be applied to a Scene in the Plugin panel. 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, and then installed into Viz Artist.

Installed Scene Plugins can be found under [Built Ins](#) in the [Main Menu](#).

To view the Scene plugins, click on the [Built Ins Button](#) and select **Scene Plugins** (in the drop-down menu) or click  in the [Element Area](#).

For a detailed list and description of Scene plugins, see [Scene Plugins](#) .



The Scene Plugin panel is divided in two sections:

- Scene plugins applied to a Scene, as icons (1)
- Scene Plugin editor (2 (opens when the Scene plugin is highlighted)).

This section contains information on the following topics and procedures:

- [To Add a Scene Plugin](#)
- [To Edit a Scene Plugin Properties](#)
- [To Show Scene Plugin Information](#)
- [To Enable/Disable a Scene Plugin](#)
- [To Remove a Scene Plugin](#)

To Add a Scene Plugin



1. Click **Built Ins**.
2. Select **Scene Plugins** (SP). Browse for the Scene plugin to apply to a Scene.
3. Drag the Scene plugin to the Scene Plugin panel:
 - Drag the selected Scene Plugin to the drop zone, or
 - Right-click on the Scene plugin icon and select **Add to Scene...**

Tip: If a Scene Plugin has been applied to a Scene, the Scene Plugin icon will show orange.



To Edit a Scene Plugin Properties

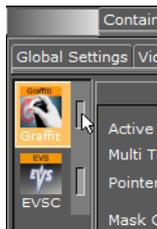
1. In the Scene Setting's Plugin area, select the plugin.
2. Edit the plugin's properties in the editor to the right.

To Show Scene Plugin Information

1. In the Scene Setting's Plugin area, select the plugin.
2. In the plugin editor, click the info icon.

To Enable/Disable a Scene 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 a Scene. This gives overview when adjusting the plugin properties to the values you need.

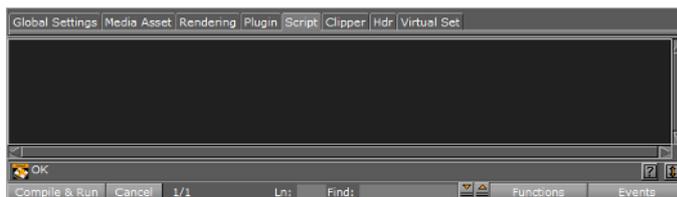
Note: After you add a plugin to the scene, it will automatically be enabled.

To Remove a Scene Plugin

- Drag the plugin to the trash can, or
- Right-click on the plugin (in the Scene Plugin tab) and select **Remove from Scene**.

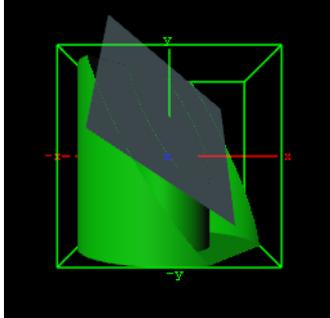
6.1.5 Script

In the Script panel, scripts can be added to the Scene. For more information on scripting see the [Script Editor](#) section.



6.1.6 Clipper

In the Clipper panel, settings for scenes that have the [Clipper](#) plugin attached can be modified.

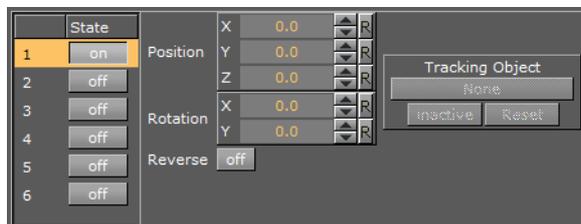


The [Clipper](#) plugin is an alternative to the mask function; the clipper works in 3D and the mask in 2D. Objects using the clipper plugin, when placed behind the clipper plane, will be masked/clipped. If an object is only partially behind a clipper plane, only the part of the object that is behind the plane will be clipped out since the clipping is done in true 3D space. The plugin uses the OpenGL clipping planes that supports up to six planes.

This section contains the following topics:

- [Clipper Properties](#)

Clipper Properties



- **Number:** Shows the clipper plane number, from 1 to 6
- **State:** Set the selected clipper plane **On** or **Off**
- **Position:** Modify the position of the clipper plane along the X, Y, and Z axis
- **Rotation:** Modify the rotation along the X and Y axis
- **Reverse:** Set to **Off**, everything in front of the clipper plane is shown. 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

- [Create Animations](#)
- [Clipper](#)

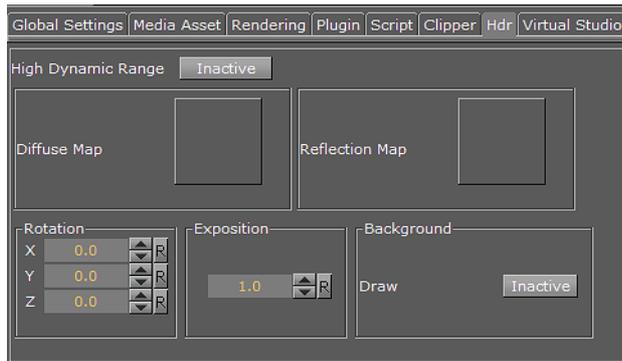
6.1.7 HDR (High Dynamic Range)

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 contains the following topics:

- [HDR Properties](#)

HDR Properties



- **High Dynamic Range (Active/Inactive):** Enables HDR lighting instead of the default OpenGL lighting on Containers tagged for HDR lighting. To tag Containers for HDR lighting the HDR plugin must be placed in a Container to enable HDR lighting on the respective container and its sub-tree.
- **Diffuse Map:** This is a HDR Cube map (must be a vertical cross and should have been imported from an EXR file). It defines the diffuse light map.
- **Reflection Map: Same as Diffuse map but for the reflection part of the lighting.**
- **Rotation:** Rotate the environment map with the set parameters.

Note: Diffuse map and reflection map together form the 'environment map'.

- **Exposition:** Controls the brightness of the maps (similar to the exposure time of an ordinary camera).
- **Background (Active/Inactive): Draws the Reflection map as background.**

See Also

- [HDR](#)

6.1.8 Virtual Studio

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.



- **Studio to Virtual:** Use to offset camera view
- **Virtual Blue Screen:** Use to check and view settings
 - **Hold-out Matte:** If set to **On** chroma keying and color correction only happens in the cyc area as defined by VizIO
 - **Draw cyc/co-cyc:** Trash Matte selection
 - **Display Mode:** Trash Matte visibility
 - **Send Sample Geometry:**
 - **Center Shift Cross:** Show the Center Shift Cross in the Scene Editor
 - **Head Up Display:**

6.2 Video Clip Playback Considerations

There are two methods to playback Video Clips:

- Playback directly in a Clip Channel (see [Video Clip Playback](#))
- Playback as a Key Frame in the Stage (see [Clip Animation in the Stage](#))

In both methods the Media Assets all share the same available Clip Channels (1 to 16).

A Clip Channel can only have one video clip applied to it. If Clip Channel1 is applied to a Container, and is also applied to a Director in the Stage, the same video clip will be shown in the Container and Stage, and the whichever is set last will have priority.

When the same Clip Channel is used in both the Scene Settings and the Stage, the main problem is that, whatever is changed in one place is reflected in the other.

For example, when the out time of the clip is changed in the Scene Settings (which means the content of the player itself are changed) these changes affect the clip playback in the Stage as well, because it is the same player.

In most if not all cases you do not want this to happen.

To have Clip Channels in both the Scene Tree and the Stage use two different Clip channels (i.e. Clip1 and Clip2), or do not use both methods in one Scene.

Layers

All Scenes, in all layers, also share the same Clip Channels.

If a Scene is loaded in the Main Layer with a video clip on Clip Channel 3, and a Scene is also loaded to the Back Layer with a different video clip, but also on Clip3, both Scenes/Layers will play the last loaded video clip. The one from the Back Layer in this example.

Clip Players

Every Clip Channel holds at least one Clip Player. If pending Clip Player support is enabled for the Clip Channel, an additional Clip Player is used for play back.

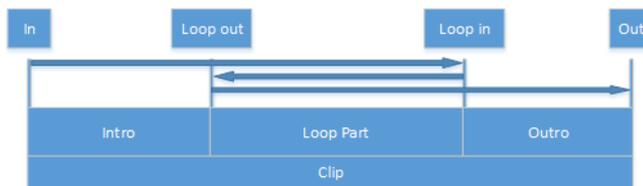
Every Clip Player uses up additional system memory. But seamless transition from one video clip to another Video clip is only possible with pending Clip Player support enabled.

This section contains the following topics:

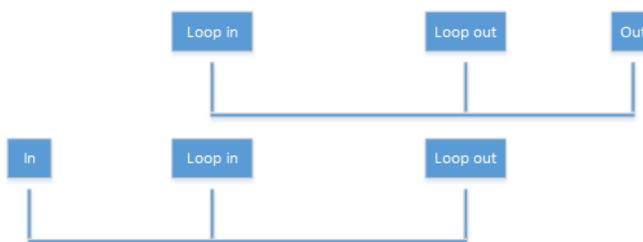
- [3 and 4-Point Loops in Clip Player](#)
- [General, Mixed and Special Modes](#)
- [Clip Handling](#)
- [IO Matrix](#)

6.2.1 3 and 4-Point Loops in Clip Player

A 4-Point Loop describes a clip that has at least a loop part and an intro or an outro or both. To define a 4-Point Loop, the loop in and out time code and the number of times the loop part should be repeated, possibly infinite, are required.



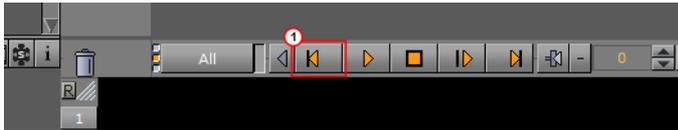
Note: A 3-Point Loop is a clip which has a loop part and only an intro or an outro (see below).



The 4-Point Loop features are only supported on video version of Viz Artist and require Matrox hardware to work, i.e. X.mio, X.open or DSX LE.

Note: Other supported video hardware use the DirectShow clip player and do not support 4-point-loop clips.

IMPORTANT! The two clip players need to be configured correctly for the 4-Point Loop to work correctly (the animation in the Stage must be set to the beginning). Click on the **Jump to beginning of animation** button (1) to set the two clip players, to the beginning of the animation.



This section contains the following topics:

- [Example: 4-Point Loop Use](#)
- [Parameters for 4-Point Loops in the Stage Editor](#)

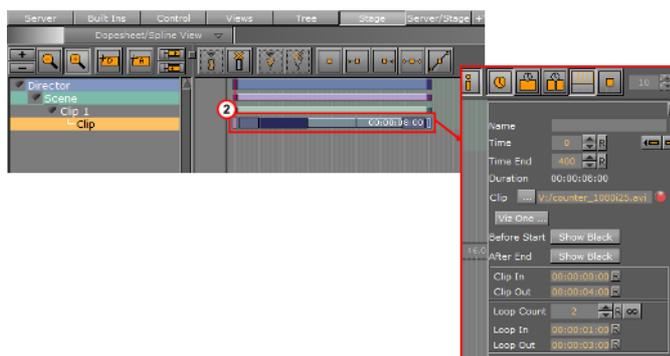
Example: 4-Point Loop Use

4-Point Loops can be useful in a wide variety of cases where you need to loop a video segment. Perhaps you do not know in advance exactly how many times you will want to loop the video. For example, let us assume the video clip is a spinning globe animation. You know you have an intro, but how long the globe should spin, is dependent of the time another scene element is shown, say a video of the anchor speaking. When the anchor-video is done, you will want the spinning globe to exit the scene.

Parameters for 4-Point Loops in the Stage Editor

Additional fields have been added to the clip setting pane where loop in, loop out and loop count can be set as shown in the image below. No additional command is needed to start or stop the loop.

Click on the Clip channel (2) to view the 4-Point Loop parameters.



During scene-design the values of the 4-Point Loop are recalculated automatically when the Key Frames are changed by dragging them in the director tree. Any changes made in the entry fields will be reflected in the time line pane to show the real-time length of the clip.

When the start or the end frame of the clip is dragged it will snap to the nearest possible time code to have an integer loop count.

See Also

- [Animation Stage](#)
- [Key Frame Editor](#)
- [Channel Editor](#)

6.2.2 General, Mixed and Special Modes

This section contains the following topics:

- [Play Clips with Resolutions Different from Channel](#)
- [Frame Accurate Clip Playback](#)

Play Clips with Resolutions Different from Channel

It is possible to play clips even if the resolution is different from the one set for the channel. However, there are some restrictions to that:

- Can only play clips from the same genlock family
- The clip's resolution must be of a regular broadcast format

For detailed information on which combinations are possible check the appendix.

Frame Accurate Clip Playback

If the clip, in the stage, needs to be played frame accurately, these features need to be set to active:

- First, the pending clip playback support has to active. If not the clips will not be ready when needed (see **Video Input: Clip Input** in the [Viz Configuration](#) (see the [Viz Engine Administrator Guide](#))).
- Additionally the Frame Accurate Viz Commands (FAVC) need to be set to active in **Global Properties** (see **Communication** in [Viz Configuration](#) (see the [Viz Engine Administrator Guide](#)))
- And the Command Execution Field Dominance should be set to **Odd Retrace Counter** (see Communication -> Global Properties in [Viz Configuration](#))

When using external control applications it may be necessary to tweak the FAVC Bias.

For a scene containing a clip in the stage to play frame accurately it is necessary to use the SCENE*CUE command, which pre loads the first clip Key Frame.

6.2.3 Clip Handling

This section contains the following topics:

- [Slow Motion Playback](#)
- [Fast Forward and Fast Backward](#)
- [Flushing Clip Channels](#)

Slow Motion Playback

A shuttle rate between -1.0 and 1.0 will result in the clip playing slower than normal, where negative values result in a backwards playback.

The actual frame to play is determined by the internal frame counter which is increased on every grabbed frame.

For a frame rate of 0.3 this would be:

Internal 0	Frame 0	Internal 0.3	Frame 0
Internal 0.6	Frame 0	Internal 0.9	Frame 0
Internal 1.2	Frame 1	Internal 1.5	Frame 1
Internal 1.8	Frame 1	Internal 2.1	Frame 2

Fast Forward and Fast Backward

A shuttle rate <-1.0 or >1.0 results in the clip playing faster than normal, where negative values result in a backwards playback.

The calculation of the next Frame to be delivered is done the same way as described above using the set shuttle rate.

Flushing Clip Channels

The **FLUSH** command can be used to clear the active clip player and reset all parameters to their default value. All valid clip frames still in the pipeline will be cleared as well and black will be sent.

If the pending clip player also needs to be cleared, send **FLUSH** to the PENDING command channel.

6.2.4 Settings

This section contains the following topics:

- [Repeat Mode](#)
- [Reverse Fields](#)
- [Loop Mode](#)
- [Auto Run](#)

Repeat Mode

Determines how clip playback should behave when paused:

- **None:** Black will be played out
- **Frame:** The last frame is shown, which shows both fields
- **Field:** Only the last field is shown, copied to first and second field

This setting is only relevant if the clip's scan mode is interlaced.

Reverse Fields

This setting is used to guarantee a smooth playback when playing the clip backwards by exchanging first and second field.

This setting is only relevant if the clip's scan mode is interlaced.

Loop Mode

If on, the clip loops infinitely between clip in and out times.

Auto Run

If on, the clips start immediately after being loaded and delivering the first frame.

6.2.5 Scripting and Plugin interface

This section contains the following topics:

- [Scripting](#)
- [Plugins](#)

Scripting

The scripting interface refers to the plugin interface and has therefore the same limitations as the plugins have.

To use a clip channel in scripting, first retrieve a channel from the system:

```
oClipChannel as ClipChannel = system.GetClipChannel( iChannel )
```

This object can now be used to control the clip channel:

```
oClipChannel.SetClipName( "clip.avi" )
oClipChannel.LoopMode = true
oClipChannel.Play(0)
```

The scripting interface to clip channel allows for the same use cases as the plugin interface does.

Plugins

The plugin interface provides almost all functions available in the Command Interface. The only limitation here is, all functions refer to the current scene. The use cases are the same as described in [3 and 4-Point Loops in Clip Player](#). Stage use cases cannot be addressed with the plugin interface.

6.2.6 IO Matrix

This section contains the following topics:

- [Source PAL](#)
- [Source 720p](#)
- [Source 1080i](#)

Source PAL

Output DVE	Video In				Output Texture	Video In		

Output DVE	Video In				Output Texture	Video In		
PAL	+	-	-		PAL	+	-	-
720p	-	-	-		720p	-	-	-
1080i	+	-	-		1080i	+	-	-

Output DVE	Clip In				Output Texture	Clip In		
PAL	+	-	-		PAL	+	-	-
720p	-	+	-		720p	+	+	-
1080i	+	+	+		1080i	+	+	+

Source 720p

Output DVE	Video In				Output Texture	Video In		
PAL	-	-	-		PAL	-	+	-
720p	-	+	-		720p	-	+	-
1080i	-	+	-		1080i	-	+	-

Output DVE	Clip In				Output Texture	Clip In		
PAL	+	-	-		PAL	+	+	-
720p	-	+	-		720p	+	+	-
1080i	+	+	+		1080i	+	+	+

Source 1080i

Output DVE	Video In				Output Texture	Video In		
PAL	-	-	-		PAL	-	-	+
720p	-	-	-		720p	-	-	+
1080i	-	-	+		1080i	-	-	+

Output DVE	Clip In				Output Texture	Clip In		
PAL	+	-	-		PAL	+	-	+
720p	-	+	-		720p	+	+	+
1080i	+	+	+		1080i	+	+	+

6.3 Video Playback

Video playback in Viz Artist depends on the machine and video board configuration. There are three video playback methods:

- [Video Clip Playback](#), or
- [Live Video Feeds](#) or
- [Live Feed from a Video Stream](#)

Video is applied to a Scene either as [DVE](#) or as a [Texture](#).

Before video can be played, the **Video Input** settings must be configured in the [Viz Configuration](#) (see the [Viz Engine Administrator Guide](#)). These settings determine the frame rate (PAL/NTSC) of the whole video system.

Video clips imported from Viz One can be played if the clip's container/codec combination is supported by the currently installed video card and the corresponding clip player (Matrox clip player for Matrox video card, DirectShow player for all other boards).

These video cards are supported for video playback:

- DVS
- Bluefish
- NVIDIA Quadro Digital Video Pipeline
- Matrox

There are a few varieties of Matrox cards (X.mio, X.mio2 and X.mio2 Plus), which mainly differ in the number of possible video inputs and outputs.

Example: A X.mio2 Plus/44 card has four inputs and four outputs. The four outputs are paired as Fill and Key, which gives two output channels maximum.

Example: A X.mio2/24 card has two inputs and four outputs. The four outputs are paired as Fill and Key, which gives two output channels maximum.

The Matrox DSX LE2<n> (see the release notes for supported <n> versions) does not have any input, codec support or compositor. Furthermore, for DSX LE3, X.mio, X.mio2 and X.mio2 Plus Matrox cards video inputs and clip channels can be disabled in the **Matrox** section of the [Viz Configuration](#) (see the [Viz Engine Administrator Guide](#)).

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 the correct genlock setup there is no guarantee that the Engine will function correctly (see **Video Output** in [Viz Configuration](#) ([Viz Engine Administrator Guide](#))).

Video parameters are modified in the Scene Settings [Media Asset Properties](#) Tab.

For video control from an external application see the **Viz Command Interface Documentation** (go to, *Start -> All Programs -> vizrt -> Viz 3.x -> Documentation*).

This section contains the following topics:

- [Video Application](#)
- [Video Clip Storage](#)
- [Video Clip Playback Use Cases](#)
- [Video Clip Playback](#)
- [Live Video Feeds](#)
- [Live Feed from a Video Stream](#)
- [Keying Mode for Video](#)
- [Keying Mode Parameters](#)
- [Seamless Input Channel Switcher](#)

6.3.1 Video Application

Video is applied in Viz Artist either as Digital Video Effect (DVE) or as a Texture:

DVE

The playback of external video through **DVE** does not have any impact on rendering performance, since it is handled solely on the board.

When DVE is used, the graphics scene and the channel are composited on the video card (Matrox X.mio) and an SDI monitor is needed to show the video channel. There is also a limited set of 2D control options available in the design and animation of the Viz Artist scene (position, size, crop and alpha).

Chroma keying is not supported for channels set to DVE.

Texture

The playback of external video through **Texture** will have an impact on rendering performance, since the live video needs to be transferred to the graphics board.

When a video is applied as a Texture, the video is part of the graphics.

When more control of a video is needed, for example, 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.

6.3.2 Video Clip Storage

All Viz Artist/Engine machines, as well as Vizrt clients (with preview - e.g. Viz Trio or Viz Weather), come with a pre-mapped video (D:) drive. This drive is configured for faster access and speeds. This is where all video clips should be stored.

IMPORTANT! It is not recommended to store video clips on the C:\ drive, as this is where the operating system is installed.

A second or third video clip directory can be added in **Local Settings** of the [Viz Configuration](#) (see the [Viz Engine Administrator Guide](#)), which can be used for:

- A [Dual Channel and Trio One Box Configuration](#), or
- With any Viz Artist/Engine configuration for the selection of a secondary video clip directory if a directory fails.

If the file path for a video clip is set to '<clip_root>', and the first directory falls over the same video file will be searched for and used, from the next directory in the Clip directory list.

If, when multiple Clip directories are selected, the **Clip Name** box (see [Clip Properties](#)) is used to search for a video, the search will default to the last used directory.

Viz One

Video clips can also be imported from Viz One. To do this Viz One must be integrated with Viz Engine. See [Viz One](#) in [Viz Configuration](#) of the [Viz Engine Administrators Guide](#) and [Viz One Deployment Guide](#).

Dual Channel and Trio One Box Configuration

When more than one directory path is selected, for example, D:\AEClips; D:\DecoClips, etc., in the **Clip Name** box change the file path to '<clip_root>'. The file name is then searched for, in the selected directories (selected in the Viz Configuration), one after the other, for stored video clips with the same name.

Example: Search for a video clip named 'next_tuesday.avi' in D:\AEClips\AFL\Promos. In the **Clip Name** box change 'D:\AEClips\AFL\Promos' to '<clip_root>'. The filename shown in Viz Artist should be <clip_root>\next_tuesday.avi.

See Also

- **Local Settings** in [Viz Configuration](#) (see the [Viz Engine Administrator Guide](#))

6.3.3 Video 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 \(Full Screen and 3D Space Transformation\)](#)

Tip: With the Clip Channel Media Asset it is possible to play images as DVE or Texture with the Matrox clip player. Supported image containers are jpg and png. The images have to be available on a physical disk for this feature to work. One advantage of an image played as a clip, is that the same features available for a video clip can be used on the image.

Playing Overlay Clips (Not Full Screen)

Playing a clip which is not full screen in size, like a spinning logo, an insert snipe or a special effect without audio. These are known as overlay clips.

To play overlay clips it is recommended to use the [SoftClip](#) 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.

[SoftClip](#) 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 to 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 do not need to be played back in full screen mode, use [MoViz](#). MoViz is very similar to [SoftClip](#), with the main difference being that it supports audio. MoViz is based on the DirectX clip playback workflow on Windows, which means 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 clips with 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 to 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 the Matrox clip channel to DVE. In DVE mode you get the maximal clip quality achievable.

When using DVE mode, do not forget to apply a key function to your Viz Elements or switch the renderer to autokey, otherwise they will not show in front of the clip (see [Keying Mode for Video](#)).

If you want to use an alpha channel on this clip, use the Matrox Mpg2iFrame&alpha codec and configure the clip channels in the [Viz Configuration](#) ([Viz Engine Administrator Guide](#)) 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 and 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 Configuration](#) ([Viz Engine Administrator Guide](#)) 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.

6.3.4 Video Clip Playback

Video clip playback should be used sparingly. The size and formats of some video clips can affect the Viz Engine render performance.

Note: To build a Scene with video clips (in Viz Artist that supports clip playback) there are also three plugin options that can also be used:

- [Image Clip](#)
- [MoViz](#)
- [SoftClip](#)

This section contains the following topics:

- [Video Clip Playback General Guidelines](#)
- [To Play a Video Clip as Texture](#)
- [To Play a Video Clip as DVE](#)
- [Clip Animation in the Stage](#)

See Also

- [Live Video Feeds](#)
- [Live Feed from a Video Stream](#)

Video Clip Playback General Guidelines

For video clip playback methods, consider these general guidelines:

Use...	Playback Method
If no 3D effects or animation control - and only limited 2D effects - on the clips are needed	To Play a Video Clip as DVE
For full screen (e.g. background) clips	
If video must be mapped onto a texture in the scene or requires minor adjustments	Clip Animation in the Stage
For non-standard image format/resolution/codecs	The SoftClip plugin, or The MoViz plugin.
If more than two small clips are required	
For low resolution and short animations	The Image Clip plugin.

Always test the performance of the different solutions and work with the solution that gives the best quality versus performance ratio.

A performance test should always take place on a machine with the same specifications as those used for production playback.

Video clip playback can be applied through either [DVE](#) or [Texture](#).

IMPORTANT! The video clip selected must have the same resolution as the channel in question. Make sure that the **Output Format** and the **Video Input** resolutions are the same, i.e., Video Input: Clip Channel 1: HD 1080i / Output Format: 1080i (see [Viz Configuration](#) in the [Viz Engine Administrator Guide](#))

IMPORTANT! Make sure that the correct container and codec is selected (see [Supported Codecs](#)).

IMPORTANT! To be able to use a Live input channel, a Clip channel or a Stream channel, they first have to be set to active in **Video Input** (see [Viz Configuration](#) in the [Viz Engine Administrator Guide](#)).

Tip: Enable **w/Alpha** when alpha is required in video clips. Either two clips, or one clip that contains both the fill and key, is required. If there is only one clip, which contains the fill (without a key), the key will be created dynamically and be fully opaque. For the naming convention of 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](#).

Note: The Matrox Clip Player supports TDIR (Time Delay Instant Replay), which enables Viz Engine to play back clips while they are still being written by Viz One. When a clip is played, with TDIR support enabled, looping is not enabled for this clip. Viz Engine determines automatically if it needs to set TDIR support **on**, or if the clip is already loaded.

Note: If alpha is required with a video clip, click the [Config Button](#) and configure the alpha in **Video Input** (see [Viz Configuration](#) in the [Viz Engine Administrator Guide](#)).

Note: For Dual Channel and Trio OneBox set up see [Dual Channel and Trio One Box Configuration](#).

Note: BG Graphics are always the lowest in the stack, even below zero, and FG graphics are assigned a default of 100. A priority cannot be assigned to BG Graphics.

To Play a Video Clip as Texture

1. Click on **Built Ins**.

2. Click on **Media Assets** (MA).
3. <Right-click> on a Clip Channel, and select **Add Clip<1> as Texture** (1).
 - The **Texture** button in the Scene Tree panel (Media Asset Manager) is automatically selected (2)
 - The selected Clip Channel is added to the Scene Tree and a new Container is created (click **Tree** to view)



Tip: The **Clip<1>** Channel can be made active or inactive. Toggle the active/inactive button (3).

Tip: The **Clip<1>** Channel can also be dragged to the Scene Tree, where it is added as a Texture in a new Container.

4. Click on **Clip<1>** Channel, in the Scene Tree panel (Media Asset Manager) Texture view, to open its properties panel (see [Media Asset Properties](#)).

Tip: <Right-click> the **Clip<1>** icon in its Container. Select **Show settings of Clip<1>**.

Tip: Click the  icon in the Scene Tree to view Media Asset properties at any time

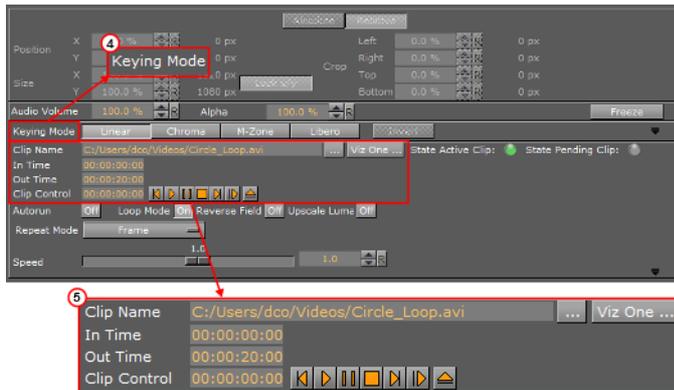
5. Select the required **Keying Mode** (4 (see [Keying Mode for Video](#))).
6. In the **Clip Name** (5) line, click , or , and select a video clip to import.

Note: To import from Viz One, Viz One must be integrated and configured with Viz Artist/Engine (see [Viz One](#)).

7. Configure the video clip playback as required (see [Media Asset Properties](#)).
8. Click  (5 (Play)) to test the video clip.

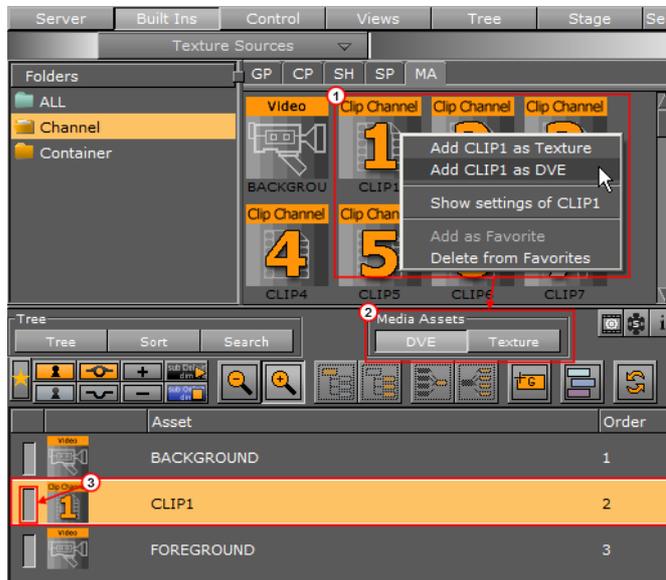
Note: Matrox Card Only: A green 'traffic light' shows that the video clip has loaded successfully. A gray 'traffic light' shows that no video clip has been loaded. A red 'traffic light' shows that there was an error when the video clip was loaded.

Note: Click the expand button to show addition information. If the clip loaded successfully, clip details are shown. If the clip did not load successfully, additional errors are shown.



To Play a Video Clip as DVE

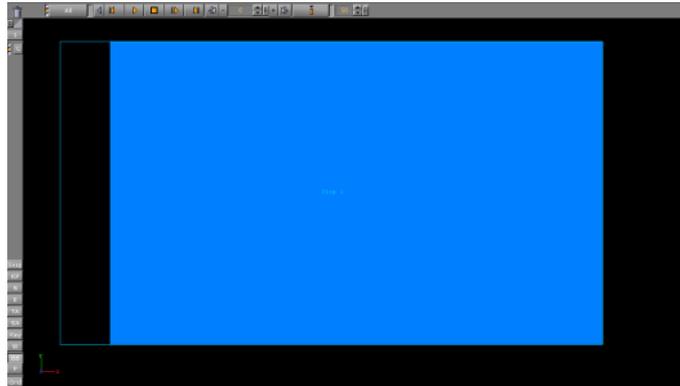
1. Click on **Built Ins**.
2. Click on **Media Assets (MA)**.
3. <Right-click> on the required Clip Channel.



4. Select **Add Clip<1> as DVE** (1).
 - The **DVE** button in the Scene Tree panel (Media Asset Manager) is automatically selected (2).

Tip: The **Clip<1>** Channel can be made active or inactive. Toggle the active/inactive button (3).

The [Scene Editor](#) shows a blue rectangle, which shows that the live input channel is correctly configured for live video.



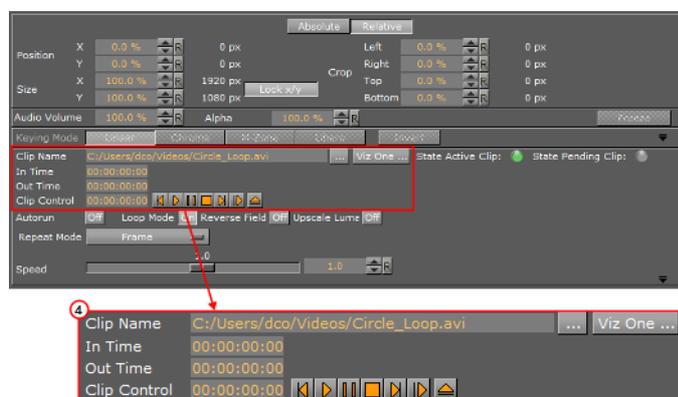
5. Click on the **Clip<1>** Channel to open its properties panel (see [Media Asset Properties](#)).
6. Adjust the parameters in the DVE control panel, as required (see [Media Asset Properties](#)).
7. In the **Clip Name** (4) line, click **...** (ellipsis), or **Viz One ...**, and select a video clip to import.

Note: To import from Viz One, Viz One must be integrated and configured with Viz Artist/Engine (see [Viz One](#)).

8. Configure the video clip playback as required (see [Media Asset Properties](#)).
9. Click **▶** (4 (Play)) to test the video clip.

Note: Matrox Card Only: A green 'traffic light' shows that the video clip has loaded successfully. A gray 'traffic light' shows that no video clip has been loaded. A red 'traffic light' shows that there was an error when the video clip was loaded.

Note: Click the expand button to show addition information. If the clip loaded successfully, clip details are shown. If the clip did not load successfully, additional errors are shown.

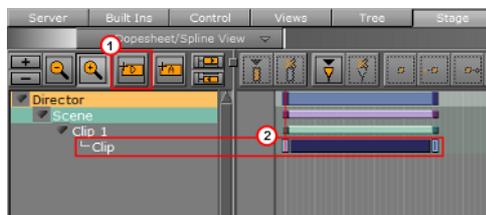


Clip Animation in the Stage

Video can also be played out as part of an animation. This is useful when video clips are required to interact with animation.

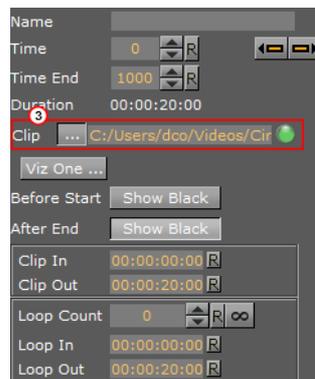
To Animate a Clip

1. Before a video clip can be animated, it must first be applied as a [Texture](#) or [DVE](#). See:
 - [To Play a Video Clip as Texture](#)
 - [To Play a Video Clip as DVE](#)
2. Click **Stage**.
3. Click the New Director button (1).
4. Right-click on the Director.
5. Select **Clip1 > Clip**.



6. In the Clip1 channel, click in the dark blue area (2). The Clip Key Frame Editor opens on the far right.
7. In the Clip Key Frame Editor click **...**, or **Viz One ...** (3), and select a video clip to import.

Note: To import from Viz One, Viz One must be integrated and configured with Viz Artist/Engine (see [Viz One](#)).



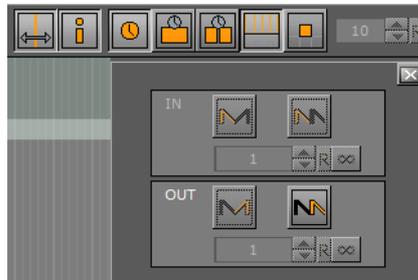
Note: This does not necessarily have to be the clip enabled in the previous procedures. It can be any video clip that Viz Artist can play.

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.

8. Configure the Clip Key Frame Editor as required (see [Clip Key Frame Editor Properties](#)).

To Play a Single Clip in the Stage with Stage Loop

1. Set a clip in the stage as described in [To Animate a Clip](#).
2. Click **Clip**, in the Stage Tree.
3. Set the clip to loop a specific number of times or infinite (see [Common Channel Editor Properties](#)).



To Play Multiple Clips in One Channel

1. Add a clip to the stage as described in [To Play a Single Clip in the Stage with Stage Loop](#).
2. Set the time indicator to a position after the first clip Key Frame
3. Click  to add a Key Frame.
4. Select a clip for the new Key Frame (see [To Animate a Clip](#)).

Show black rules for pauses between clips:

Both clips NS	Frame 1 of Clip 2 will be shown
Clip 1 AE, Clip 2 NS	Frame 1 of Clip 2 will be shown
Clip 1 NS, Clip 2 BS	Last frame of Clip 1 will be shown
Clip 1 AE, Clip 2 BS	Black will be shown
Key: NS: Nothing set BS: Show black before start set AE: Show black after end set	

If **Pending** is Active (which is recommended) subsequent clips are loaded into the pending clip player and are swapped at the correct time. This feature enables the clips to concatenate seamlessly. However, when pending support is set to Inactive, black frames will be shown until the clip is ready to be played (see [Pending](#) in [Viz Configuration](#) ([Viz Administrator Guide](#))).

To Play Multiple Clips in One Channel with Stage Loop

1. Set multiple clips in the stage as described in [To Play Multiple Clips in One Channel](#).
2. Set the clips to loop. Click the corresponding entry in the Stage Tree.

All Key Frames including the pauses in between and before will be taken into account during play back, for example, pause - clip 1- pause - clip 2 will loop as such.

To Play Single and/or Multiple Clips in Multiple Channels

- This is a combination of the use cases described above, with the clips used on several channels.

6.3.5 Live Video Feeds

A live video feed can be can be applied through either [DVE](#) or [Texture](#).

Tip: Matrox Card Only. Enable **w/Alpha** when external video with alpha is required. 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.

Note: If alpha is required with the external video, click the [Config Button](#) and configure the alpha in **Video Input** (see [Viz Configuration](#) in the [Viz Engine Administrator Guide](#)).

Note: To be able to use a Live input channel, a Clip channel or a Stream channel, they first have to be set to active in **Video Input** (see [Viz Configuration](#) in the [Viz Engine Administrator Guide](#)).

This section details the following procedures:

- [To Feed Live Video as Texture](#)
- [To Feed Live Video as DVE](#)

See Also

- [Video Clip Playback](#)
- [Live Feed from a Video Stream](#)

To Feed Live Video as Texture

1. Click on **Built Ins**.
2. Click on **Media Assets** (MA).
3. <Right-click> on the required Live Channel, and select **Add Live<1> as Texture** (1).
 - The **Texture** button in the Scene Tree panel (Media Asset Manager) is automatically selected (2)

- The selected Live Channel is added to the Scene Tree and a new Container is created (click **Tree** to view)



Tip: The **Live<1>** Channel can also be dragged to the Scene Tree, where it is added as a Texture in a new Container.

Tip: The **Live<1>** Channel can be made active or inactive. Toggle the active/inactive button (3).

4. Click on **Live<1>** Channel, in the Scene Tree panel (Media Asset Manager) **Texture** view, to open its properties panel (see [Media Asset Properties](#)).

Tip: Also <right-click> the **Clip<1>** icon in its Container. Select **Show settings of Live<1>** to show its properties at any time.

Tip: Click the  icon in the Scene Tree to view Media Asset properties at any time

5. Select the required **Keying Mode** (see [Keying Mode for Video](#)).



6. Click **Tree**.
7. Add a Geometry to the Container (see [Add Plugins to Containers](#)).

To Feed Live Video as DVE

Note: If a DVS board is installed, DVE is simulated through the software.

1. Click on **Built Ins**.
2. Click on **Media Assets (MA)**.
3. <Right-click> on the required Live Channel.
4. Select **Add Live<1> as DVE**.

- The **DVE** button in the Media Asset Manager (Scene Tree panel) is automatically selected

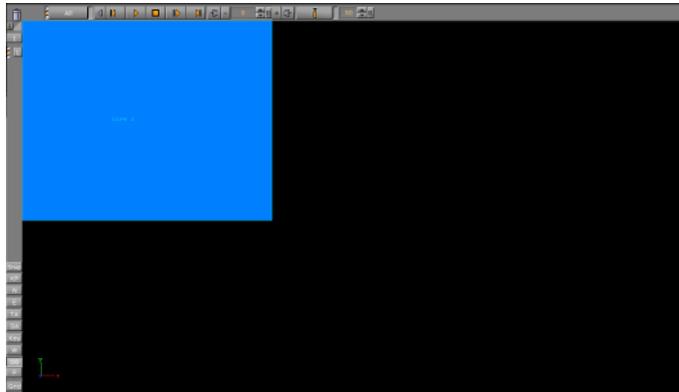


Tip: The **Live<1>** Channel can be made active or inactive. Toggle the active/inactive button (1).

5. Click on **Live<1>** Channel, in the Media Asset Manager (Scene Tree panel) **DVE** view, to open its properties panel (see [Media Asset Properties](#)).

Tip: Click the  icon on the top of the Scene Tree to view Video settings, at any time.

6. Adjust the parameters in the DVE control panel, as required (see [Media Asset Properties](#)).
7. The [Scene Editor](#) shows a blue rectangle, which shows that the live input channel is correctly configured for live video.



6.3.6 Live Feed from a Video Stream

Video from a video stream can be applied through either [DVE](#) or [Texture](#).

To play a Video Stream, make sure that a Stream channel is active (see **Video Input** in [Viz Configuration](#) in the [Viz Engine Administrator Guide](#)).

Note: To be able to use a Live input channel, a Clip channel or a Stream channel, they first have to be set to active in **Video Input** in the [Viz Configuration](#).

This section details the following procedures:

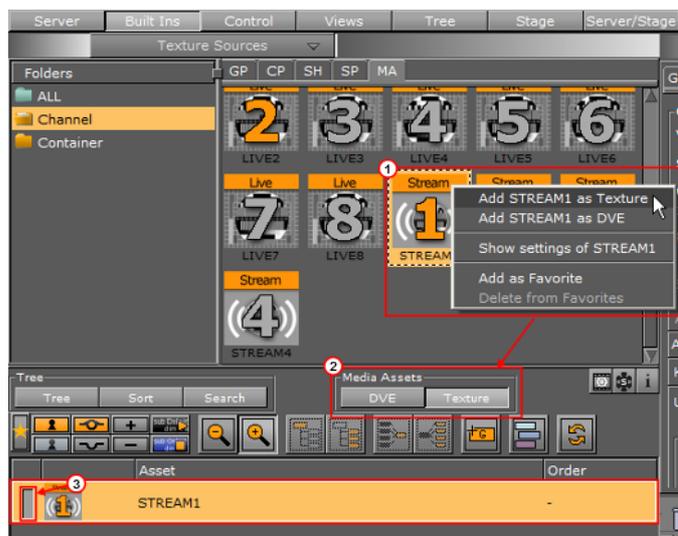
- [To Feed a Video Stream as Texture](#)
- [To Feed a Video Stream as DVE](#)
- [To Remove a Video Stream](#)

See Also

- [Video Clip Playback](#)
- [Live Video Feeds](#)

To Feed a Video Stream as Texture

1. Click on **Built Ins**.
2. Click on **Media Assets (MA)**.
3. <Right-click> on a Stream Channel, and select **Add Stream<1> as Texture (1)**.
 - A new container is created in the Scene Tree and the Stream Channel is added to it as a property
 - The **Texture** button in the Media Asset Manager (Scene Tree panel) is automatically selected (2)



Tip: The **Stream<1>** Channel can also be dragged to the Scene Tree, where it is added as a Texture in a new Container.

Tip: The **Stream<1>** Channel can be made active or inactive. Toggle the active/ inactive button (3).

4. Click on **Stream<1>** Channel, in the Media Asset Manager (Scene Tree panel) **Texture** view, to open its properties panel (see [Media Asset Properties](#)).

Tip: Also <right-click> the **Stream<1>** icon in its Container. Select **Show settings of Stream<1>** to show its properties at any time.

Tip: Click the  icon on the top of the Scene Tree to view Media Asset properties at any time

5. Select the required **Keying Mode** (see [Keying Mode for Video](#)).



6. Enter the stream **Url**.
7. Click on **Load**.



Note: To load a new Video Stream, enter the new Url in the **Url** box. This will automatically clear the old Video Stream.

To Feed a Video Stream as DVE

1. Click on **Built Ins**.
2. Click on **Media Assets (MA)**.
3. Right click on the required Stream Channel.
4. Select **Add Stream<1> as DVE**.
 - The **DVE** button in the Media Asset Manager (Scene Tree panel) is automatically selected.



Tip: The **Stream<1>** Channel can be made active or inactive. Toggle the active/ inactive button (1).

The [Scene Editor](#) shows a blue rectangle, which shows that the live input channel is correctly configured for live video.



5. Click on **Stream<1>** Channel, in the Media Asset Manager (Scene Tree panel) **DVE** view, to open its properties panel (see [Media Asset Properties](#)).

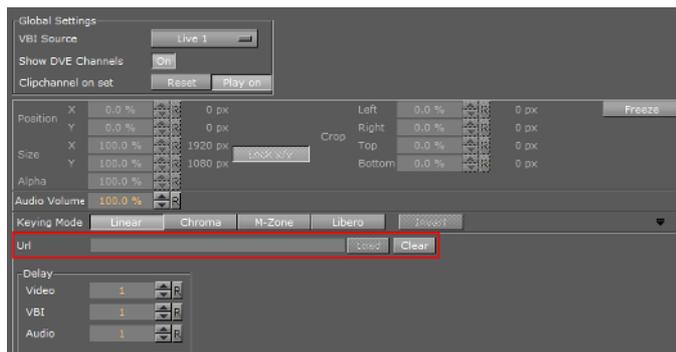
Tip: Click the  icon on the top of the Scene Tree to view Video settings, at any time

6. Adjust the parameters in the DVE control panel, as required (see [Media Asset Properties](#)).
7. Enter the stream **Url**.
8. Click on **Load**.



To Remove a Video Stream

- Click on the **Clear** button.



6.3.7 Keying Mode for Video

There are four types of keying modes built into Viz Artist:

- [Linear Key](#)
- [Chroma Key](#)
- [M-zone Key](#)
- [Libero Key](#)

This section contains information on the following topics and procedures:

- [Linear Key](#)
- [Chroma Key](#)
- [M-zone Key](#)
- [Libero Key](#)
- [To Configure Keying Mode](#)
- [Keying Mode Parameters](#)

See Also

- [Configure the Keyer: Best Practices](#)

Linear Key

Linear key is the standard keying technique, 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)

Chroma Key

Chroma key is mostly used with a virtual studio.

In 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 were 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.

Note: Each input channel has a freely configurable chroma keyer.

M-zone Key

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.

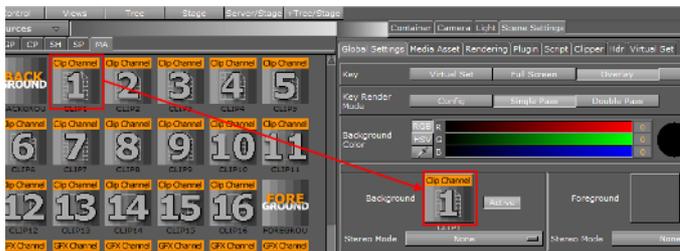
Libero Key

If Viz Libero Clips are to be played, the Keyer must be set to Libero. This way the chroma key settings from the Viz Libero Clip will be used. Make sure that the clip comes with the '.cid' (Camera ID), '.trk' (for tracking data) and '.col' file (chroma key settings).

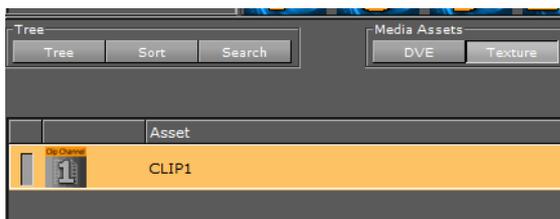
To Configure Keying Mode

1. Click **Built Ins**.
2. Click the **Media Assets** tab.
3. In the Properties panel click the **Scene Settings** tab
4. Click the **Global Settings** tab.

5. Select a Media Asset and drag it onto the **Background Image** or **Foreground Image** drop-zone.



6. Click on the Media Asset in the Media Asset Manager (this opens the **Media Asset** tab in the **Scene Settings** panel).

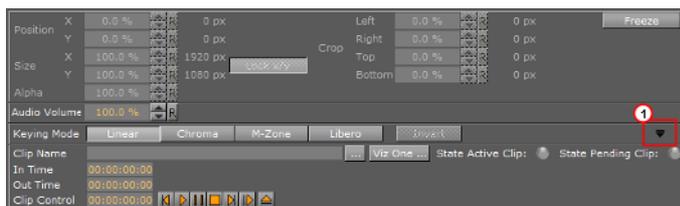


7. In the **Keying Mode** row, click the relevant button to enable a keying method. Options are:

- [Linear Key](#)
- [Chroma Key](#)
- [M-zone Key](#)
- [Libero Key](#)

Note: To use the M-zone key, the [Chroma Keyer](#) plugin must also be used. Drag the [Chroma Keyer](#) plugin to the container which holds the video texture in the Scene Tree.

8. In the Keying Mode row, click the arrow (1) to open up the [Keying Mode Parameters](#).



Note: When in **DVE** mode, only the Linear keying mode is available.

9. Configure the **U-Color** and **V-Color** (set the background color in the YUV color space). The point set by these two parameters is the reference for the next parameters, which determine the colors to remove.
All colors between the point defined by the **U-Color** and **V-Color** and the **Inner UV-Diameter** will be removed completely, and all colors between the **Inner UV-Diameter** and the **Outer UV-Diameter** the transparency will be set to a value between 0% and 100%.

10. Configure the **Sample X/Y** coordinates and the **Sample Width/Height (Upper Radius/Lower Radius)** with [UV Lookup](#). These parameters define a section of the background in the video-input. Press **Sample Now** to start the self-configuration (see [Sample](#)).
11. Configure **Color Correction**. This can help to completely remove the blue or green colors from the image (see [Color Correction](#)).
12. Configure the Luminance Control:
 - **Restore Dark Colors/Shadows:** All pixels with a luminance lower than complete restoration to the key/alpha will be make the pixel 100% opaque (see [Restore Dark Colors/Shadows](#)).
 - **Restore Bright Colors:** All pixels with a luminance higher than **Complete Restoration to** the key/alpha will be making the pixel 100% opaque (see [Restore Bright Colors](#)).

6.3.8 Keying Mode Parameters

The Keying mode parameters are split into six panels:

- [UV Lookup](#)
- [U-Color/V-Color](#)
- [Color Correction](#)
- [Restore Dark Colors/Shadows](#)
- [Restore Bright Colors](#)
- [Sample](#)

UV Lookup

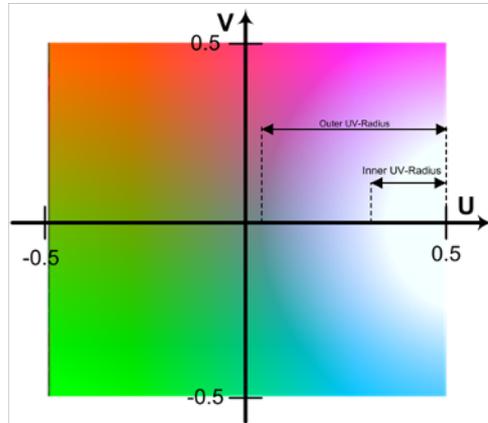


- **UV Lookup:** Drag a UV Look up image to the place holder. Click **R** to reset the UV Lookup image box

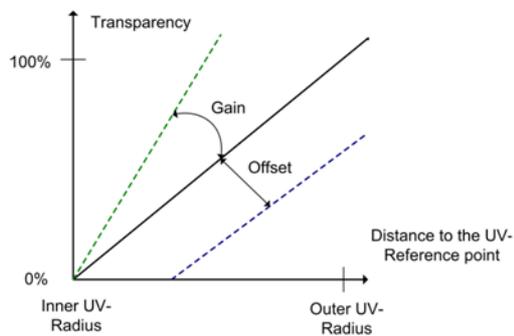
Note: A UV lookup image is a luminance image of size 256x256.

- **Mouse Sampling:** Collect the chroma information out of a video signal, at the position of the mouse pointer:
Hold the left mouse button and move the mouse around to add the colors which are underneath the mouse, to the chroma key table. Hold down <Alt> at the same time, the colors will be removed from the chroma key table again.
- **Save:** Save a modified UV Lookup image
- **Clear:** Remove the current UV LookUp texture from the current scene (no colors are removed from the image)

U-Color/V-Color



- **U-Color** and **V-Color**: Set the background color in the YUV color space. The point set by these two parameters is the reference for the next parameters, which determine the colors to remove.
- **Inner UV-Diameter/ Outer UV-Diameter**: All colors between the point defined by the **U** and **V-Color** and the **Inner UV-Diameter** are removed completely and for all colors between the **Inner UV-Diameter** and the **Outer UV-Diameter** the transparency will be set to a value between 0% and 100%.



Color Correction



- **Color Correction**: blue or green colors can be removed completely from the image.
 - **Off**: Color Correction is not in use
 - **Blue**: The parameters for the two neighboring colors (**Cyan**, **Magenta**) are active
 - **Green**: The neighboring colors are **Yellow** and **Cyan**

Color Correction is especially helpful to remove chroma bleed. If **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 stay unchanged. Low values signify that this tone will be removed completely from the image.

High **Gain** values for all these parameters mean that nearly all tones of this color will stay unchanged. Low **Gain** values signify that this tone will be removed completely from the image.

Restore Dark Colors/Shadows



- **Restore Dark Colors/Shadows:** When set to **Active** all pixels with a luminance lower than complete restoration to the key/alpha will be make 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 stay unchanged. **Gain** can be used to darken dark colors even more, if it is set below 1.0. **Gain** is only applied to pixels with luminance between 0 and **Zero Restoration To**.

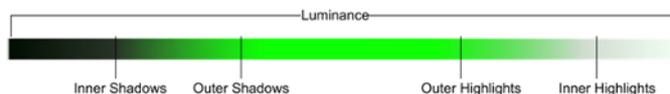
Tip: Gain is useful in a virtual studio, where the set design uses dark colors in a shadow, but when seen through the camera the shadow is lighter, in color, than the floor. A darker shadow will make the final composition more believable.

Restore Bright Colors



- **Restore Bright Colors:** When set to **Active** 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 stay unchanged. **Gain** can be used to make bright colors show even brighter. **Gain** is only applied to pixels with a luminance between 1 and **Zero Restoration To**.



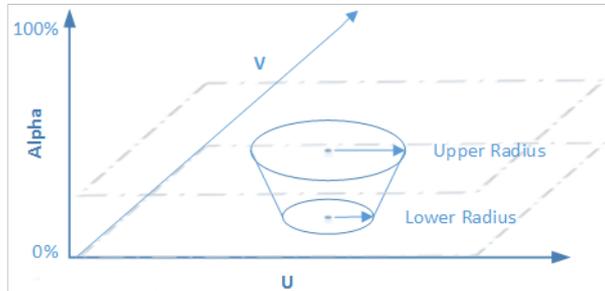
Sample

The **Sample** functions define a section in the video input, which can make the Chroma Key results more accurate.

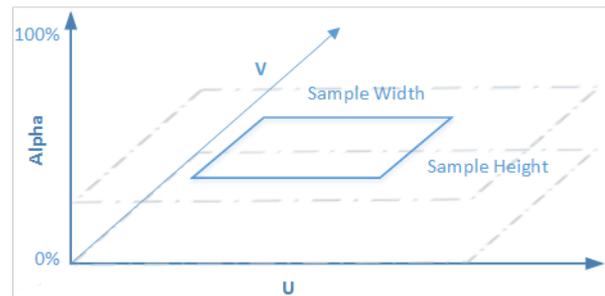
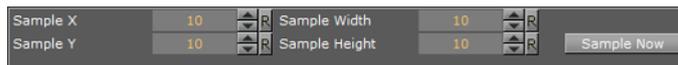
Press **Sample Now** to start the self-configuration.

- **Sample With** [UV Lookup](#):

- Set the **Sample X/Y** coordinates: The x/y position on the screen to select the picking color. Not used
- Set the **Upper Radius/Lower Radius**



- **Sample Without UV Lookup:**
 - Set the **Sample X/Y** coordinates: The x/y position on the screen to select the picking color
 - Set the **Sample Width/Height**.



Configure the Keyer: Best Practices

- 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 show, that you have 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.

6.3.9 Seamless Input Channel Switcher

This feature works as a software-enabled video and audio crossbar (a coordinated switching system). In earlier versions of Viz Engine every Viz Engine video channel was bound to one physical video in channel. The usage of another physical input was only possible if you changed the setting in the Viz Configuration.

With seamless input channel switching you can change this binding seamlessly even during play out. It is also possible to use more inputs than the number of licensed channels, for example, if a video card is equipped with an X.RIO card. However, you can only use the number of licensed channels simultaneously. At the moment the switching is only possible using commands.

Note: Seamless switching is relevant for X.mio, X.mio2 and X.RIO.

The initial setting is defined by the configuration. During run time you can change the used video input channel by sending the following commands to Viz Engine.

This section contains information on the following commands:

- [Changing the Input Channel](#)
- [Checking the Input Channel](#)
- [Releasing the Input Channel](#)
- [Getting the Free Physical Inputs](#)

Changing the Input Channel

The command, **RENDERER*VIDEO*VIDEOIN*1*LEASE SET B**, sets video channel 1 to use physical input B as its source. If used during play out, the change happens at the next frame. If you try to lease an invalid channel or a channel that is currently used by another Viz Engine input, you will get Invalid as a result.

Checking the Input Channel

The command, **RENDERER*VIDEO*VIDEOIN*2*LEASE GET**, checks which physical input is currently mapped to Viz Engine's channel 2.

Releasing the Input Channel

The command, `RENDERER*VIDEO*VIDEOIN*1*LEASE RELEASE`, releases the currently captured physical input channel.

Getting the Free Physical Inputs

The command, `RENDERER*VIDEO*VIDEOIN*3*FREE_DEVICES GET`, lists the free physical inputs as a comma separated list (e.g. A, C, E, F).

6.4 Supported Codecs

This section details the supported Codecs for all supported Video Cards used by Vizrt:

- [Matrox Codecs](#)
- [DVS and Bluefish Codecs](#)

See Also

- [Matrox](#) (see the [Viz Engine Administrator Guide](#))
- [Bluefish444](#) (see the [Viz Engine Administrator Guide](#))
- [Digital Video Systems](#) (see the [Viz Engine Administrator Guide](#))

6.4.1 Matrox Codecs

The Matrox X.mio2 Plus 24/6000 card, which can play the SD codecs, except D10/ D12, is Vizrt's standard card. This model can be upgraded to 8500 class by a license upgrade.

Note: The 8500 class is only available for X.Mio2 and X.Mio2 Plus, not for X.Mio. The 8000 class upgrade is available for X.Mio.

All codecs are implemented in the software.

Matrox cards support these codec types and formats:

- [PAL Codecs](#)
- [NTSC Codecs](#)
- [720p50 and 720p59.94 \(60M\) Codecs](#)
- [1080i25 and 1080i29.97 \(30M\) Codecs](#)
- [1080p50 and 1080p59.94 \(60M\) Codecs](#)
- [QuickTime Writer Plugin Codecs](#) (see also: [QuickTime Codecs for the Viz Artist/ Engine 64-bit Platform](#))

The Matrox X.mio, X.mio2 and X.mio2 Plus cards support a selection of different codecs for both SD and HD.

The Matrox codecs use the Matrox card itself as a kind of a license dongle, therefore certain codecs require certain card classes as follows (these classes apply to X.mio, X.mio2 and X.mio2 Plus):

- 6000 class (SD codecs only. No D10)
- 8000 class (Same as 6000 + HD codecs and D10, D12)

- 8500 class (Capable of Apple ProRes playback)

IMPORTANT! Codec DNxHD: The use of codec DNxHD requires its own license, issued by Vizrt.

QuickTime Codecs for the Viz Artist/Engine 64-bit Platform

The only QuickTime codecs supported on a 64-bit machine, with Matrox hardware for clip playback, as well post rendering with MatroxFileWriter and ClipOut channel, are:

- RLE (animation): (Playback only)
- H264
- Apple ProRes
- HDV
- XDCam
- DVCPRO
- DNxHD (Matrox X.mio2 Video board required)

IMPORTANT! Codec DNxHD: The use of codec DNxHD requires its own license, issued by Vizrt.

Without Matrox hardware, QuickTime is currently not supported for the 64-bit Viz Engine.

Note: QuickTime is natively supported by the Matrox clip player, which is available with any Matrox video card or Matrox X.Open dongle. The QuickTime player cannot be used as fallback on a 64-bit Viz Engine (there is no 64-bit SDK from Apple)

6.4.2 PAL Codecs

This section contains the following topics:

MPEG2 IFrame	DVCPRO
MPEG2 IBP	D10
MJPEG Lossless	H264
DVCAM	Uncompressed

MPEG2 IFrame

Container	Codec	Features	Audio
.avi	MPEG2-IFrame422 720x576	Alpha: no Bitrate: 10-50 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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

Container	Codec	Features	Audio
.avi	MPEG2-IFrame4224 720x576	Alpha: yes Bitrate: 10-50 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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

See Also

- [Matrox Codecs](#)

MPEG2 IBP

Container	Codec	Features	Audio
.avi	MPEG2 IBP SD 4:2:0, Elementary, Program, Transport 720x576	Alpha: no Bitrate: 1-15 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.avi	MPEG2 IBP SD 4:2:2, Elementary, Program, Transport 720x576	Alpha: no Bitrate: 5-50 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.mxf	MPEG2 IBP SD 4:2:0, Elementary, Program, Transport 720x576	Alpha: no Bitrate: 1-15 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.mxf	MPEG2 IBP SD 4:2:2, Elementary, Program, Transport 720x576	Alpha: no Bitrate: 5-50 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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

See Also

- [Matrox Codecs](#)

MJPEG Lossless

Container	Codec	Features	Audio
.avi	MJPEG Lossless 720x576	Alpha: no Bitrate: VBR Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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

See Also

- [Matrox Codecs](#)

DVCAM

Container	Codec	Features	Audio
.avi	DV/DVCAM 4:2:0 720x576	Alpha: no Bitrate: 25 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.mxf (OP1a)	DVCAM 4:2:0 720x576	Alpha: no Bitrate: 25 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio PCM: - 4 ch: 16 in 16bit; 48kHz
.mov	DVCAM 4:2:0 720x576	Alpha: no Bitrate: 25 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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	DVCAM 4:2:0 720x576	Alpha: no Bitrate: 25 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio PCM: - 2 ch: 16 in 16bit; 48kHz

See Also

- [Matrox Codecs](#)

DVCPRO

Container	Codec	Features	Audio
.avi	DVCPRO 4:1:1 720x576	Alpha: no Bitrate: 25 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.avi	DVCPRO 50 4:2:2 720x576	Alpha: no Bitrate: 50 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.mxf (OP1a)	DVCPRO 4:1:1 720x576	Alpha: no Bitrate: 25 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.mxf (OP1a)	DVCPRO 50 4:2:2 720x576	Alpha: no Bitrate: 50 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.mxf Panasonic P2 (OP-Atom)	DVCPRO 4:1:1 720x576	Alpha: no Bitrate: 25 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio
.mxf Panasonic P2 (OP-Atom)	DVCPRO 50 4:2:2 720x576	Alpha: no Bitrate: 50 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio
.mov	DVCPRO 4:1:1 720x576	Alpha: no Bitrate: 25 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.mov	DVCPRO 50 4:2:2 720x576	Alpha: no Bitrate: 50	No Audio PCM:

Container	Codec	Features	Audio
		Bitdepth: 8 VBI: N/A Timecode: N/A	- 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	DVCPRO 4:1:1 720x576	Alpha: no Bitrate: 25 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio PCM: - 2 ch: 16 in 16bit; 48kHz
.dv	DVCPRO 50 4:2:2 720x576	Alpha: no Bitrate: 50 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio PCM: - 2 ch: 16 in 16bit; 48kHz

See Also

- [Matrox Codecs](#)

D10

Container	Codec	Features	Audio
.mxf XDCAM (OP1a)	D10 (IMX) 4:2:2 720x608	Alpha: no Bitrate: 30, 40, 50 Bitdepth: 10 VBI: required Timecode: required	AES3 - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz

See Also

- [Matrox Codecs](#)

H264

Container	Codec	Features	Audio
.mov	H.264/AVC 720x576	Alpha: no Bitrate: VBR Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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

See Also

- [Matrox Codecs](#)

Uncompressed

Container	Codec	Features	Audio
.avi	Uncompressed YUVU 4:2:2 720x576	Alpha: no Bitrate: 160 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.avi	Uncompressed YUAVUA 4:2:2:4 720x576	Alpha: yes Bitrate: 240 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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

See Also

- [Matrox Codecs](#)

6.4.3 NTSC Codecs

This section contains the following topics:

MPEG2 IFrame	DVCPRO
MPEG2 IBP	D10
MJPEG Lossless	H264
DVCAM	Uncompressed

MPEG2 IFrame

Container	Codec	Features	Audio
.avi	MPEG2-IFrame422 720x480	Alpha: no Bitrate: 10-50 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.avi	MPEG2-IFrame4224 720x480	Alpha: yes Bitrate: 10-50 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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

See Also

- [Matrox Codecs](#)

MPEG2 IBP

Container	Codec	Features	Audio
.avi	MPEG2 IBP SD 4:2:0, Elementary, Program, Transport 720x480	Alpha: no Bitrate: 1-15 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.avi	MPEG2 IBP SD 4:2:2, Elementary, Program, Transport 720x480	Alpha: no Bitrate: 5-50 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.mxf (OP1a)	MPEG2 IBP SD 4:2:0, Elementary, Program, Transport 720x480	Alpha: no Bitrate: 1-15 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.mxf (OP1a)	MPEG2 IBP SD 4:2:2, Elementary, Program, Transport 720x480	Alpha: no Bitrate: 5-50 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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

See Also

- [Matrox Codecs](#)

MJPEG Lossless

Container	Codec	Features	Audio
.avi	MJPEG Lossless 720x480	Alpha: no Bitrate: VBR Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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

See Also

- [Matrox Codecs](#)

DVCAM

Container	Codec	Features	Audio
.avi	DV/DVCAM 4:1:1 720x480	Alpha: no Bitrate: 25 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.mxf (OP1a)	DVCAM 4:1:1 720x480	Alpha: no Bitrate: 25 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio PCM: - 4 ch: 16 in 16bit; 48kHz
.mxf Panasonic P2 (OP-Atom)	DV/DVCAM 4:1:1 720x480	Alpha: no Bitrate: 25 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio
.mov	DVCAM 4:1:1 720x480	Alpha: no Bitrate: 25 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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 4:1:1 720x480	Alpha: no Bitrate: 25 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio PCM: - 2 ch: 16 in 16bit; 48kHz

See Also

- [Matrox Codecs](#)

DVCPRO

Container	Codec	Features	Audio
.avi	DVCPRO 4:1:1 720x480	Alpha: no Bitrate: 25 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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

Container	Codec	Features	Audio
.avi	DVCPRO 50 4:2:2 720x480	Alpha: no Bitrate: 50 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.mxf	DVCPRO 4:1:1 720x480	Alpha: no Bitrate: 25 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.mxf	DVCPRO 50 4:2:2 720x480	Alpha: no Bitrate: 50 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.mxf Panasonic P2 (OP-Atom)	DVCPRO 4:1:1 720x480	Alpha: no Bitrate: 25 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio
.mxf Panasonic P2 (OP-Atom)	DVCPRO 50 4:2:2 720x480	Alpha: no Bitrate: 50 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio
.mov	DVCPRO 4:1:1 720x480	Alpha: no Bitrate: 25 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.mov	DVCPRO 50 4:2:2 720x480	Alpha: no Bitrate: 50 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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	DVCPRO 4:1:1 720x480	Alpha: no Bitrate: 25	No Audio PCM:

Container	Codec	Features	Audio
		Bitdepth: 8 VBI: N/A Timecode: N/A	- 2 ch: 16 in 16bit; 48kHz
.dv	DVCPRO 50 4:2:2 720x480	Alpha: no Bitrate: 50 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio PCM: - 2 ch: 16 in 16bit; 48kHz

See Also

- [Matrox Codecs](#)

D10

Container	Codec	Features	Audio
.mxf XDCAM (OP1a)	D10 (IMX) 4:2:2 720x512	Alpha: no Bitrate: 30, 40, 50 Bitdepth: 10 VBI: required Timecode: required	AES3 - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz

See Also

- [Matrox Codecs](#)

H264

Container	Codec	Features	Audio
.mov	H.264/AVC 720x480	Alpha: no Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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

See Also

- [Matrox Codecs](#)

Uncompressed

Container	Codec	Features	Audio
.avi	Uncompressed YUVU 4:2:2 720x480	Alpha: no Bitrate: 160 Bitdepth: 8	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz

Container	Codec	Features	Audio
		VBI: N/A Timecode: N/A	- 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz
.avi	Uncompressed YUAVUA 4:2:2:4 720x480	Alpha: yes Bitrate: 240 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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

See Also

- [Matrox Codecs](#)

6.4.4 720p50 and 720p59.94 (60M) Codecs

This section contains the following topics:

MPEG2 IFrame MPEG2 IBP DVCPRO HD HDV XDCAM HD	DNxHD AVC-Intra H264 ProRes Uncompressed
---	--

MPEG2 IFrame

Container	Codec	Features	Audio
.avi	MPEG2-IFrame422 1280x720	Alpha: no Bitrate: 50-300 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.avi	MPEG2-IFrame4224 1280x720	Alpha: yes Bitrate: 50-300 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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

See Also

- [Matrox Codecs](#)

MPEG2 IBP

Container	Codec	Features	Audio
.avi	MPEG2 IBP HD 4:2:0, Elementary, Program, Transport 1280x720	Alpha: no Bitrate: 50-300 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.avi	MPEG2 IBP HD 4:2:2, Elementary, Program, Transport 1280x720	Alpha: no Bitrate: 50-300 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.mxf (OP1a)	MPEG2, IBP HD 4:2:0, Elementary, Program, Transport 1280x720	Alpha: no Bitrate: 50-300 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.mxf (OP1a)	MPEG2, IBP HD 4:2:2, Elementary, Program, Transport 1280x720	Alpha: no Bitrate: 50-300 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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

See Also

- [Matrox Codecs](#)

DVCPRO HD

Container	Codec	Features	Audio
.avi	DVCPRO HD 1280x720	Alpha: no Bitrate: 100 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.mxf (OP1a)	DVCPRO HD 1280x720	Alpha: no Bitrate: 100 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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

Container	Codec	Features	Audio
.mxf Panasonic P2 (OP-Atom)	DVCPRO HD 1280x720	Alpha: no Bitrate: 100 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio
.mov	DVCPRO HD 1280x720	Alpha: no Bitrate: 100 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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	DVCPRO HD 1280x720	Alpha: no Bitrate: 100 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio PCM: - 2 ch: 16 in 16bit; 48kHz

See Also

- [Matrox Codecs](#)

HDV

Container	Codec	Features	Audio
.mov	HDV 1280x720	Alpha: no Bitrate: 25 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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

See Also

- [Matrox Codecs](#)

XDCAM HD

Container	Codec	Features	Audio
.mxf XDCAM (OP1a)	XDCAM HD, IBP HD 4:2:0, Elementary, Program, Transport 1280x720	Alpha: no Bitrate: 25 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio PCM: - 2 ch: 16 in 16bit; 48kHz - 4 ch: 16 in 16bit; 48kHz

Container	Codec	Features	Audio
.mxf XDCAM (OP1a)	XDCAM HD422, IBP HD 4:2:2, 1280x720	Alpha: no Bitrate: 50 Bitdepth: 8 VBI: N/A Timecode: N/A	PCM: - 8 ch: 24 in 32bit; 48kHz
.mov	XDCam EX 1280x720	Alpha: no Bitrate: 25CBR, 35VBR Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.mov	XDCam HD 422 1280x720	Alpha: no Bitrate: 18-35 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.mp4	XDCam EX 1280x720	Alpha: no Bitrate: 25CBR, 35VBR Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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

See Also

- [Matrox Codecs](#)

DNxHD

Container	Codec	Features	Audio
.mxf (OP-Atom)	DNxHD 145, 220, 220x 1280x720	Alpha: no Bitrate: 100 Bitdepth: 8, 10 VBI: N/A Timecode: N/A	No Audio
.mxf (OP1a)	DNxHD 145, 220, 220x 1280x720	Alpha: no Bitrate: 100 Bitdepth: 8, 10 VBI: N/A Timecode: N/A	No Audio 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
.mov	DNxHD 4:2:2	Alpha: no	No Audio

Container	Codec	Features	Audio
	1280x720	Bitrate: 36, 145, 220 Bitdepth: 8; 10 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

See Also

- [Matrox Codecs](#)

AVC-Intra

Container	Codec	Features	Audio
.mxf Panasonic P2 (OP-Atom)	AVCIntra Class 50 1280x720	Alpha: no Bitrate: 50 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio
.mxf Panasonic P2 (OP-Atom)	AVCIntra Class 100 1280x720	Alpha: no Bitrate: 100 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio
.mxf (OP1a)	AVCIntra Class 50 1280x720	Alpha: no Bitrate: 50 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.mxf (OP1a)	AVCIntra Class 100 1280x720	Alpha: no Bitrate: 100 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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

See Also

- [Matrox Codecs](#)

H264

Container	Codec	Features	Audio
.mov	H.264/AVC 1280x720	Alpha: no Bitrate: VBR	No Audio PCM:

Container	Codec	Features	Audio
		Bitdepth: 8; 10 VBI: N/A Timecode: N/A	- 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz
.mp4	XDCAM EX 1280x720	Alpha: no Bitrate: VBR Bitdepth: 8; 10 VBI: N/A Timecode: N/A	No Audio 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

See Also

- [Matrox Codecs](#)

ProRes

Container	Codec	Features	Audio
.mov	I-Frame 4:2:2 1280x720	Alpha: no Bitrate: 147, 220 Bitdepth: 10 VBI: N/A Timecode: N/A	No Audio 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
.mov	I-Frame 4:4:4:4 1280x720	Alpha: yes Bitrate: VBR Bitdepth: 8, 10 VBI: N/A Timecode: N/A	No Audio 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

See Also

- [Matrox Codecs](#)

Uncompressed

Container	Codec	Features	Audio
.avi	Uncompressed YUVU 4:2:2 1280x720	Alpha: no Bitrate: 700 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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

Container	Codec	Features	Audio
.avi	Uncompressed YUAVUA 4:2:2:4 1280x720	Alpha: yes Bitrate: 1050 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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

See Also

- [Matrox Codecs](#)

6.4.5 1080i25 and 1080i29.97 (30M) Codecs

This section contains the following topics:

MPEG2 IFrame MPEG2 IBP DVCPRO HD HDV XDCAM HD	DNxHD AVC-Intra H264 ProRes Uncompressed
---	--

MPEG2 IFrame

Container	Codec	Features	Audio
.avi	MPEG2-IFrame422 1920x1080	Alpha: no Bitrate: 50-300 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.avi	MPEG2-IFrame4224 1920x1080	Alpha: yes Bitrate: 50-300 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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

See Also

- [Matrox Codecs](#)

MPEG2 IBP

Container	Codec	Features	Audio
.avi	MPEG2 IBP HD 4:2:0, Elementary, Program, Transport 1920x1080	Alpha: no Bitrate: 50-300 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.avi	MPEG2 IBP HD 4:2:2, Elementary, Program, Transport 1920x1080	Alpha: no Bitrate: 50-300 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.mxf (OP1a)	MPEG2, IBP HD 4:2:0, Elementary, Program, Transport 1920x1080	Alpha: no Bitrate: 50-300 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.mxf (OP1a)	MPEG2, IBP HD 4:2:2, Elementary, Program, Transport 1920x1080	Alpha: no Bitrate: 50-300 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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

See Also

- [Matrox Codecs](#)

DVCPRO HD

Container	Codec	Features	Audio
.avi	DVCPRO HD 1920x1080	Alpha: no Bitrate: 100 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.mxf (OP1a)	DVCPRO HD 1920x1080	Alpha: no Bitrate: 100 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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 z

Container	Codec	Features	Audio
.mxf (OP-Atom)	DVCPRO HD 1920x1080	Alpha: no Bitrate: 100 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio
.mov	DVCPRO HD 1920x1080	Alpha: no Bitrate: 100 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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	DVCPRO HD 1920x1080	Alpha: no Bitrate: 100 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio PCM: - 2 ch: 16 in 16bit; 48kHz

See Also

- [Matrox Codecs](#)

HDV

Container	Codec	Features	Audio
.mov	HDV 1440x1080	Alpha: no Bitrate: 25 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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

See Also

- [Matrox Codecs](#)

XDCAM HD

Container	Codec	Features	Audio
.mxf (OP1a)	XDCAM HD, IBP HD 4:2:0, Elementary, Program, Transport 1440x1080	Alpha: no Bitrate: 25 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio PCM: - 2 ch: 16 in 16bit; 48kHz - 4 ch: 16 in 16bit; 48kHz

Container	Codec	Features	Audio
.mov	XDCam EX 1920x1080	Alpha: no Bitrate: 25CBR, 35VBR Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.mov	XDCam HD 1440x1080	Alpha: no Bitrate: 35 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.mov	XDCam HD 422 1920x1080	Alpha: no Bitrate: 18-35 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
.mp4	XDCam EX 1920x1080	Alpha: no Bitrate: 25CBR, 35VBR Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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

See Also

- [Matrox Codecs](#)

DNxHD

Container	Codec	Features	Audio
.mxf (OP-Atom)	DNxHD 145, 220, 220x 1920x1080	Alpha: no Bitrate: 100 Bitdepth: 8, 10 VBI: N/A Timecode: N/A	No Audio
.mxf (OP1a)	DNxHD 145, 220, 220x 1920x1080	Alpha: no Bitrate: 100 Bitdepth: 8, 10 VBI: N/A Timecode: N/A	No Audio 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
.mov	DNxHD 4:2:2	Alpha: no	No Audio

Container	Codec	Features	Audio
	1920x1080	Bitrate: 36, 145, 220 Bitdepth: 8; 10 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

See Also

- [Matrox Codecs](#)

AVC-Intra

Container	Codec	Features	Audio
.mxf (OP-Atom)	AVCIntra Class 50 1920x1080	Alpha: no Bitrate: 50 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio
.mxf (OP-Atom)	AVCIntra Class 100 1920x1080	Alpha: no Bitrate: 100 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio
.mxf (OP1a)	AVCIntra Class 50 1920x1080	Alpha: no Bitrate: 50 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.mxf (OP1a)	AVCIntra Class 100 1920x1080	Alpha: no Bitrate: 100 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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

See Also

- [Matrox Codecs](#)

H264

Container	Codec	Features	Audio
.mov	H.264/AVC 1920x1080	Alpha: no Bitrate: VBR	No Audio PCM:

Container	Codec	Features	Audio
		Bitdepth: 8; 10 VBI: N/A Timecode: N/A	- 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz
.mp4	XDCAM EX 1920x1080	Alpha: no Bitrate: VBR Bitdepth: 8; 10 VBI: N/A Timecode: N/A	No Audio 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

See Also

- [Matrox Codecs](#)

ProRes

Container	Codec	Features	Audio
.mov	I-Frame 4:2:2 1920x1080	Alpha: no Bitrate: 147, 220 Bitdepth: 10 VBI: N/A Timecode: N/A	No Audio 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
.mov	I-Frame 4:4:4:4 1920x1080	Alpha: yes Bitrate: VBR Bitdepth: 8, 10 VBI: N/A Timecode: N/A	No Audio 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

See Also

- [Matrox Codecs](#)

Uncompressed

Container	Codec	Features	Audio
.avi	Uncompressed YUVU 4:2:2 1920x1080	Alpha: no Bitrate: 800 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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

Container	Codec	Features	Audio
.avi	Uncompressed YUAVUA 4:2:2:4 1920x1080	Alpha: yes Bitrate: 1200 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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

See Also

- [Matrox Codecs](#)

6.4.6 1080p50 and 1080p59.94 (60M) Codecs

This section contains the following topics:

MPEG2 IFrame	ProRes
------------------------------	------------------------

MPEG2 IFrame

Container	Codec	Features	Audio
.avi	MPEG2-IFrame422 1920x1080	Alpha: no Bitrate: 50-300 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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
.avi	MPEG2-IFrame4224 1920x1080	Alpha: yes Bitrate: 50-300 Bitdepth: 8 VBI: N/A Timecode: N/A	No Audio 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

See Also

- [Matrox Codecs](#)

ProRes

Container	Codec	Features	Audio
.mov	I-Frame 4:2:2 1920x1080	Alpha: no Bitrate: 147, 220 Bitdepth: 10 VBI: N/A Timecode: N/A	No Audio 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

Container	Codec	Features	Audio
.mov	I-Frame 4:4:4:4 1920x1080	Alpha: yes Bitrate: VBR Bitdepth: 8, 10 VBI: N/A Timecode: N/A	No Audio 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

See Also

- [Matrox Codecs](#)

6.4.7 QuickTime Writer Plugin Codecs

The QuickTime Writer Plugin supports these QuickTime codecs in the following formats:

Note: Depending on additionally installed QuickTime codec there might be more codecs available

Name	Options
Animation	Alpha: optional
BMP	Alpha: no
Cinepak	Alpha: no
Component Video	Alpha: no
DV - PAL	Alpha: no
DV/DVCPRO - NTSC	Alpha: no
DVCPRO - PAL	Alpha: no
Graphics	Alpha: no
H.264	Alpha: no
JPEG2k	Alpha: optional
Motion JPEG A	Alpha: no
Motion JPEG B	Alpha: no
MPEG-4 Video	Alpha: no
Photo - JPEG	Alpha: no
Planar RGB	Alpha: optional
PNG	Alpha: optional
Sorenson Video	Alpha: no

Name	Options
Sorenson Video 3	Alpha: no
TGA	Alpha: optional
TIFF	Alpha: optional
Uncompressed	Alpha: optional
Video	Alpha: no

See Also

- [Supported Codecs](#)

6.4.8 DVS and Bluefish Codecs

The DVS and Bluefish Video cards only support the **Microsoft DirectShow Filtergraph** framework to render video clips. This enables play, for example, of MPEG files or streams from a server.

This means that DVS and Bluefish Video cards can play everything that can be played out by the Windows Media Player, so it depends on what codecs/DirectShow filters are installed on the machine.

For more information on how to use video clips see [Video Clip Playback Considerations](#) and [Video Playback](#), in particular, the [SoftClip](#) and [MoViz](#) plugins.

See Also

- [Supported Codecs](#)
- [Matrox Codecs](#)
- [Matrox](#) (see the [Viz Engine Administrator Guide](#))
- [Bluefish444](#) (see the [Viz Engine Administrator Guide](#))
- [Digital Video Systems](#) (see the [Viz Engine Administrator Guide](#))

6.5 Import Files and Archives

Viz Artist can [Import Files](#) (Fonts, Images, Geometries, Scenes, Audio clips) and [Import Archives](#). By convention, Viz Artist archives have the filename extension **.via**.

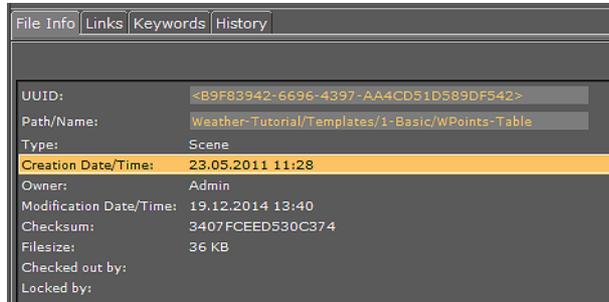
It is important to keep in mind that Viz Archives are, if possible, self contained and complete. This means that if a user archives one or more *scenes*, then **all** referenced elements in the scene, including links, fonts, images and so on, will be archived. On the other hand, a user can choose to archive only individual elements such as some fonts or some images.

All elements Viz Artist uses are kept in the *Graphic Hub* (GH). Viz Artist connects to the GH using a network connection, even if the GH is local.

The Graphic Hub is a **database**, and every element in the GH is uniquely identified by its **UUID** - a unique identifier number.

Elements in the GH can have links. This means that possibly several Path Names can link (or "point") to the same element in the GH database.

You can view the UUID, The Path/Name and Link information by right-clicking on any element in the server-view in Artist and select the option *Show Properties*:



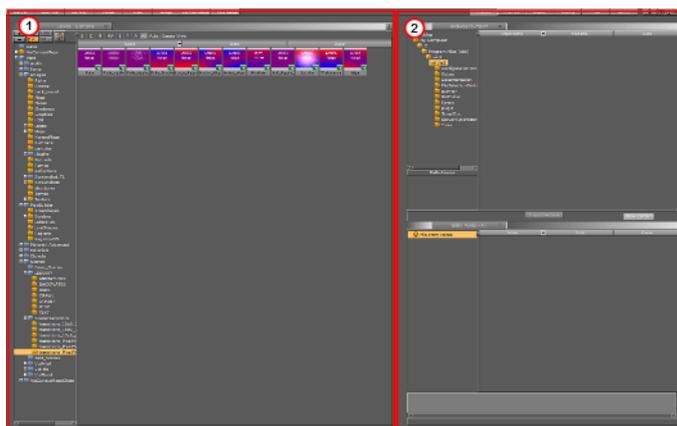
Tip: It is strongly recommended that you have a clear and well communicated plan for organizing the path(s) and naming conventions for all resources in the Graphic Hub. This will make life easier for users and import/export operations will be easier to perform without creating clutter.

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 Window](#)
- [Supported Files for Import](#)
- [Import Files](#)
- [Import Archives](#)

6.5.1 Import Window



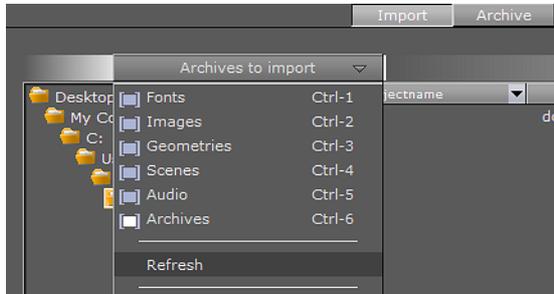
The Import Window consists of two panels:

- **Server view:** Gives access to all folders and projects in the Graphic Hub (1).
- **Import area:** Gives access to the source drives, where assets can be imported from (2).

To Open the Import Window

- Click the [Import Button](#) from the [Main Menu](#), or
- Press <F9>

Import Menu



The Import menu is positioned at the top of the Import Window. When an element type is selected from the menu, only the assets of that particular type will be listed in the file list.

6.5.2 Supported Files for Import

The following file formats are supported for import:

- [Fonts](#)
- [Images](#)
- [Geometries](#)
- [Scene Formats](#)
- [Audio](#)
- [3D Formats](#)

Fonts

Type	Comments
Adobe Type1 fonts	.pfb files. Postscript Type 1 fonts.
Adobe Type2 fonts	Postscript Type 2 fonts.
True type fonts	Recommended, most common format.

Images

Format	Comments
.bw	Black and white raster image (sgi). Legacy.
.gif	Graphics Interchange Format.
.hdr	High Dynamic Range Image File.
.png	Portable Network Graphics. Lossless raster graphics.

Format	Comments
.ppm	Portable pixmap (Netpbm).
.psd	Layers defined in a Photoshop file will be imported as separate images into the database.
.rgb	SGL's native raster image format. Legacy.
.rgba	RGB with alpha.
.tex	TeX
.tga	Truevision TARGA
.tif	Tagged Image File Format
.txc	PS2 txc
.xpm	X11 pixmap

3D Formats

Format	Comments
3ds Max .3ds / .fbx	Autodesk Max 3ds file format. Fbx is a 3d file format supported by many 3d vendors.
Autocad .dxf	Drawing Interchange Format.
Maya .fbx	FBX for Maya
Softimage .fbx	FBX for Softimage
VRML 2.0	Virtual Reality Modeling Language
VRML including animation	Virtual Reality Modeling Language
Wavefront .obj	Geometry definition file format

Scene Formats

Format	Comments
After Effects .aep	See Adobe After Effects Integration See Working with Adobe After Effects Files Note: A license is required for the use of Adobe After Effects and the import of Adobe After Effects Scenes, issued by Vizrt.
CINEMA 4D .c4d	See CINEMA 4D Integration See Working with CINEMA 4D Files Note: A license is required for the import of CINEMA 4D Scene, issued by Vizrt.

Format	Comments
Autodesk .fbx	FBX is a 3D asset exchange format compatible with many 3D tools. See Working with FBX Files

IMPORTANT! All geometries to be imported into Viz Artist must be polygon geometries.

IMPORTANT! Geometries imported from CINEMA 4D must have been saved as a *.c4d file with **Save Polygons for Melange** (Edit -> Preferences -> Files) enabled. This will make sure that all items saved by CINEMA 4D will be stored together with their corresponding polygon data.

Note: A range of products can export to FBX (see [Working with FBX Files](#)).

Audio

There are two ways to capture audio in the Viz Engine: through Matrox or a DirectSound compatible device.

Video clips can have interleaved audio in it. Maximum 16 channels, minimum 2 (stereo) as mono is not supported. The format is limited to 24bit and 48khz.

The following formats are supported for Clips:

Format	Comment
WAVE	Up to 96kHz, 24Bit and 16 Channels. The recommended format.
MP3	All Formats (Stereo only).
OggVorbis	All Formats, up to 16 Channels.

See Also

- For audio details see Viz Engine Guide: [Audio in Viz](#).

Geometries

Format
.obj
.3ds

Archives

Archives can contain all elements Viz support. See [Import Archives](#).

6.5.3 Import Files

Use File Import to import:

- Fonts
- Images
- Geometries
- Scenes
- Audio clips

This section contains information on the following topics and procedures:

- [Path Alias](#)
- [Browse for Files to Import](#)
- [To Import Selected Files](#)

See Also

- [Assign Keywords to Elements](#)
- [Supported Files for Import](#)
- [CINEMA 4D Integration](#)
- [Adobe After Effects Integration](#)

Path Alias

Use a path alias to help streamline the import process. Defined Path Aliases pre-configure file paths that are often used as import operations (see **Path Aliases** in [Viz Configuration](#) (see the [Viz Engine Administrator Guide](#))).

Browse for Files to Import

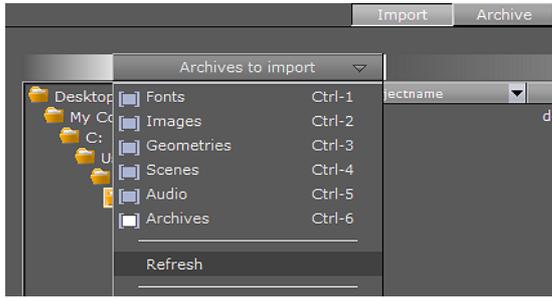
Use the import file list, on the left side of the Import Window, to browse the directories on a disk:

- <Double-click> on a directory to show its sub directories
- <Click> on a directory to show the contents in the right window
- <Click> on a path alias to quickly jump to this directory
- Hold down the <Ctrl> key and click on all required files to select multiple files.

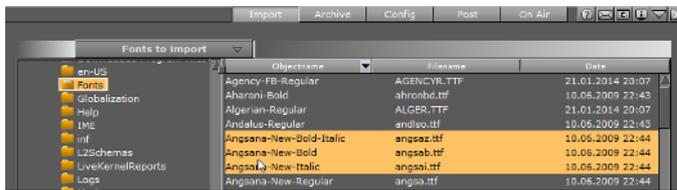
To Import Selected Files

1. Open the target folder in the Server view.
2. Click **Import**.
3. Select the type of file to import:
 - Click on the [Import Menu](#), and click on the type of file to import, or

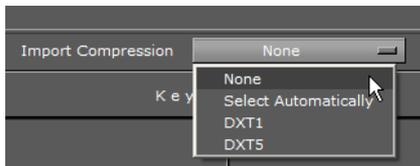
- Press the keyboard shortcut (see [File Shortcuts](#))



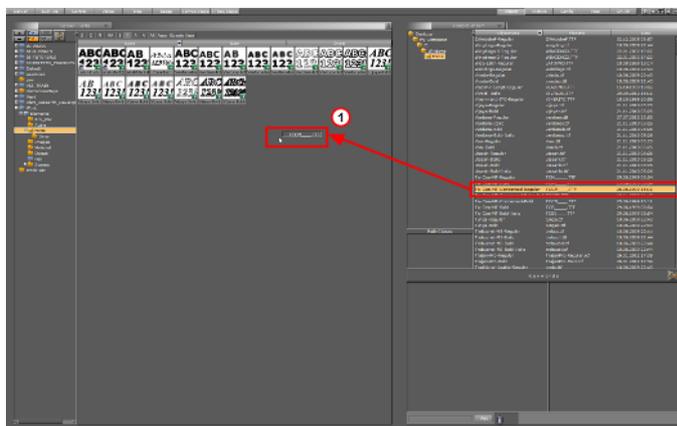
4. In the Import directory tree, search for the files to import (see [Browse for Files to Import](#)).
5. Select the required file or files.



6. **Images only:** Select the import compression for the image or images. Click on **Import Compression**. 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.



7. Import the selected file or files:
 - Drag the file or files to the target folder, in the Server Area (1), or
 - Double-click a single file (the file is added to the target folder)



6.5.4 Import Archives

An Archive (filename extension .via) is a single file that holds various types of Elements and remembers the project or folder they were stored. Archives are typically used for backup and for transferring Scenes from one Graphic Hub (GH) to another GH.

IMPORTANT! 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 Graphic Hub Manager (see [Graphic Hub User Guide](#))

This section contains information on the following topics:

- [Path Alias](#)
- [Archive Content Appearance](#)
- [To Import an Archive to the Root of a Graphic Hub](#)
- [To Import an Archive to a Specific Location in a Graphic Hub](#)
- [To Import Part of an Archive](#)
- [To Search for Import Files](#)

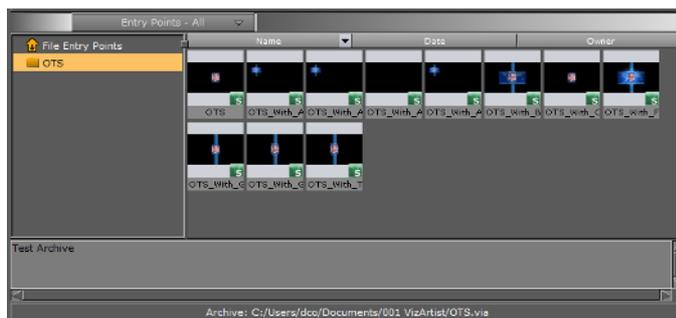
Path Alias

Using a path alias can help to streamline the import process. Defined Path Aliases pre-configure file paths that are often used for import operations (see **Path Aliases** in [Viz Configuration](#) in the [Viz Engine Administrator Guide](#)).

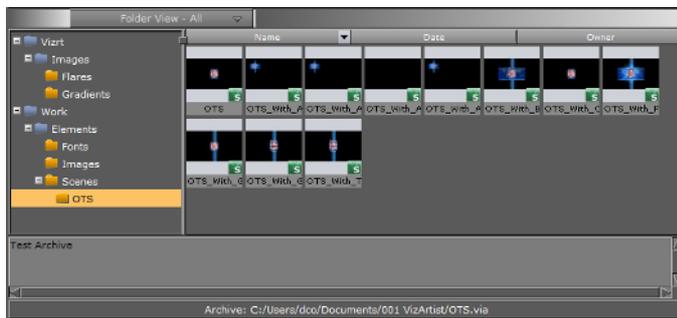
Archive Content Appearance

Use the drop-down menu in the preview panel to see the different aspects of an element's composition.

- **Entry Point View:** Show all Entry Points. An entry point is created for each element or project or folder added to an archive by drag and drop.

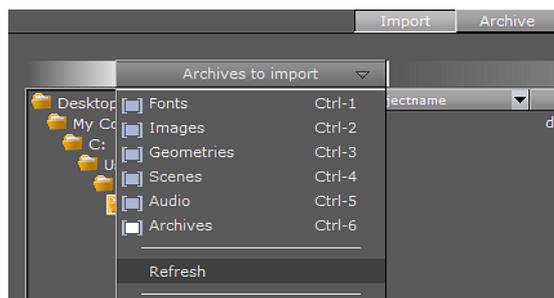


- **Folder View:** Show all the geometries that are contained in the Archive (includes referenced items as well).

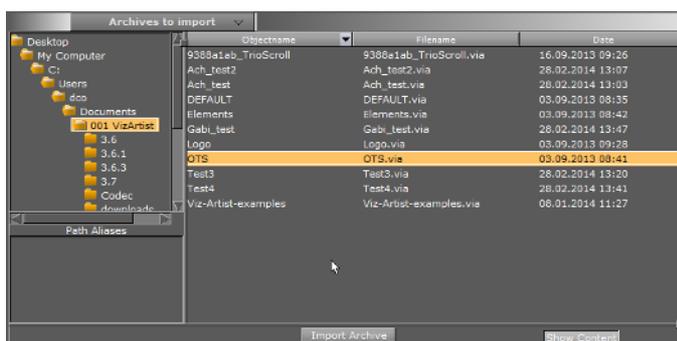


To Import an Archive to the Root of a Graphic Hub

1. Click **Import**.
2. Select **Archives to Import:**
 - Click on the [Import Menu](#), and click on **Archives to Import**, or
 - Press <Ctrl+6>



3. In the Import tree, search for the Archive to import.
4. Select the required Archive (see [Browse for Files to Import](#)).
5. Click on **Import Archive**.



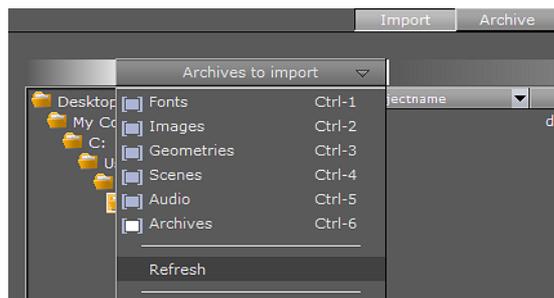
Note: Press the **Show Content** button to show the Archive's content in the panel below. The Show Content button can be deactivated for faster performance (some Archives can take a long time to show).

IMPORTANT! Using this method of import, the archive will be imported to the "root" (/) of the GH while maintaining the Path and Names of the elements in the

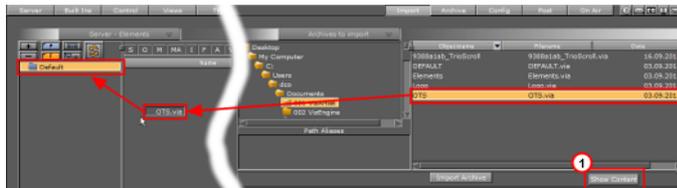
archive. If there are existing elements in the GH with the same UUID as in the archive you are importing from, you will be asked if you want to skip importing or overwrite existing elements.

To Import an Archive to a Specific Location in a Graphic Hub

1. Click **Import**.
2. Select **Archives to Import**:
 - Click on the [Import Menu](#), and click on **Archives to Import**, or
 - Press <Ctrl+6>



3. In the Import directory tree, search for the Archive to import (see [Browse for Files to Import](#)).
4. Select the required Archive in the right side panel.
5. Drag the Archive to either:
 - the target Folder or Project in the Server View folder tree, or
 - the Server View element area.
6. The archive will import.



Tip: Press the Show Content button (1) to show the Archive's content. The Show Content button can be deactivated for faster performance as some big archives can take a long time to display.

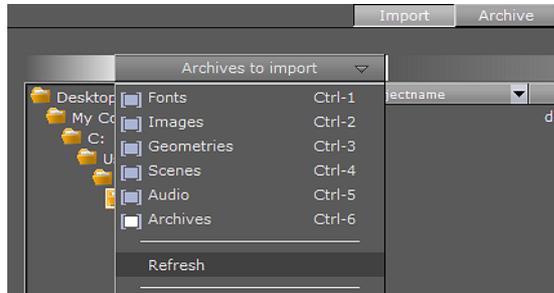
IMPORTANT! You should use this method of import when you want to control which Project or Directory the archive should import into. The Path and Names of the elements in the archive you are importing from will be preserved using the location in the GH you selected by drag and drop as a "start point". If there are existing elements in the GH with the same UUID as in the archive you are importing from, you will be asked if you want to skip importing or overwrite existing elements.

To Import Part of an Archive

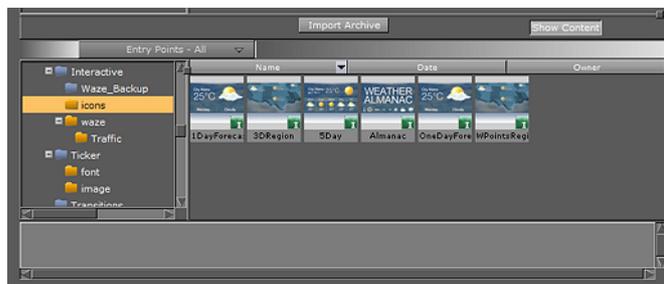
1. Click **Import**.

2. Select **Archives to Import:**

- Click on the [Import Menu](#), and click on **Archives to Import**, or
- Press <Ctrl+6>



3. In the Import directory tree, search for the Archive to import (see [Browse for Files to Import](#)).
4. Select the required Archive in the right side panel.
5. From the top menu, select **Entry Points View**.



6. Select an import option:

- To import all entry point files from the archive, drag the **File Entry Points** file folder and drop it in the left panel.
- To import a folder and its contents from the archive, drag the required folder from the folder entry points and drop it in the left panel.
- To import one or more files, drag and drop them into the left panel.

IMPORTANT! The archive is imported into the folder selected in the Graphic Hub tree. All reference files are included to guarantee full functionality. This method of import will "flatten" the elements you are importing. That means: the path or directory structure of the elements in the archive you are importing from will not be kept - this method of import is for importing parts or elements of an archive.

Note: To import projects or folders, the same procedure applies, with the exception that you select **Folder View** from the pull-down menu, and drag and drop the required folder.

To Search for Import Files

Search for files before or after the import for an Archive. Use this feature to see if similar or the same files are present in the system and to find the file's location after import.

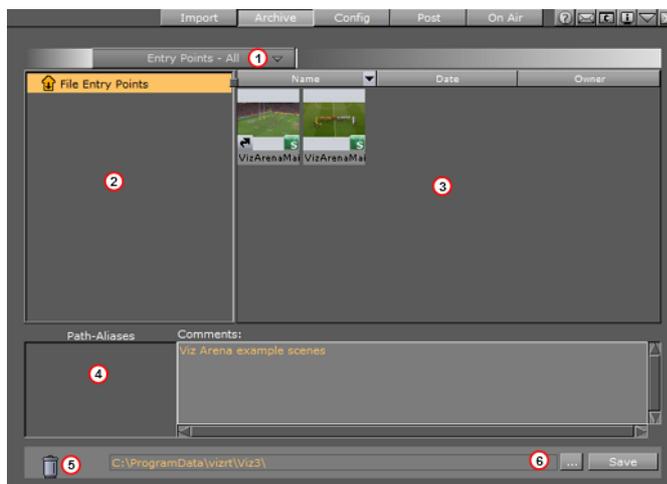
1. <Right-click> any of the Elements in the Archive.

2. Select **Search for this file in current GH.**



6.6 Archiving

Archiving allows you to export graphical resources such as one or several Scenes including fonts, images and so on to a file (archive) on disk. The archive will by convention have a **.via** filename extension. The archive can be useful for safekeeping and for transferring data to another graphics hub.

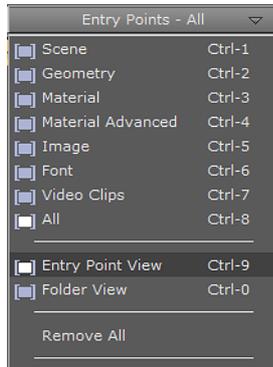


- Entry-Points (1) pull-down menu is located just below the main menu and allows you to select data types. See [Entry-Points Menu](#).
- File Entry Points panel (2) is the target area for the folders you will archive.
- The Data Panel (3) is the target area for resources to archive. Drag items such as Scenes, Fonts, Images etc. to this area to include them in the archive.
- Use Path-Aliases (4) to select aliases that represent actual directories or paths. Path-Aliases must be pre-defined in *Configuration > Path Aliases*. See *Viz Configuration* in *Viz Engine Administrator Guide*. Aliases are typically short and descriptive names used as a convenient shortcut to directories which could have long or cumbersome names.
- The Comments area can be used for assigning comments to the archive.
- Use the trash-can (5) to remove items you do not want to archive.
- Browse and select a file or specify an archive name (6) before pressing Save. The extension **.via** will be attached to the file name if not specified.

Entry points represent the main items in an archive. Other items in the archive are referenced and attached automatically to the archive.

Entry-Points 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:

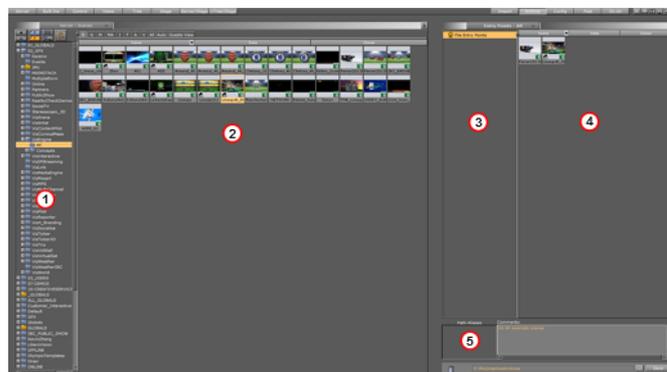


- **Scene to Video Clips** Only shows the Elements of the selected file type.
- **All** Shows Elements of all file types.
- **Entry Point View** Shows only entry points.
- **Folder View** Shows the folder hierarchy.
- **Remove All** Removes all Elements from the archive.

6.6.1 Create an Archive

To Create an Archive

1. In the main menu, click **Archive**.
The Archive view is shown.



2. From the Server side, drag folders into the Archive view's left panel and files into the right panel.
3. If required, enter a comment in the comments area.

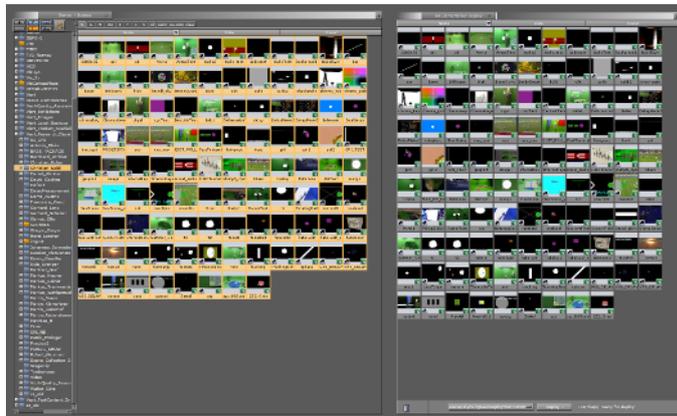
Note: Comments can be a maximum of 255 characters.

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 Geometries 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 Pilot by using Viz Pilot or Viz Template Wizard. If Viz Artist is used, the Geometries can be added manually.

6.7 Deploy Elements

The Deploy panel makes it possible to copy Elements, with all their necessary data, and deploy them from one server to another. For example, if a Scene is deployed, all items referenced by the Scene (Images, Fonts, etc.), will also be deployed.



Click on the Deploy button to show the Deploy pane.

Note: The Deploy button only shows in the GUI if at least one deploy server is set up. See **Deploy Properties** in **Database** of the [Viz Configuration](#) (see the Viz Engine Administrator Guide).

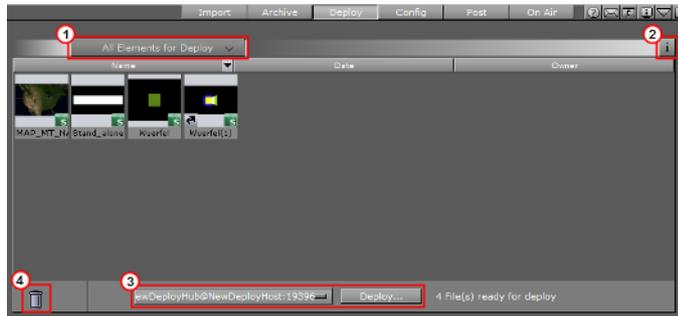


To deploy Elements from one server to another, deploy servers must be set up in the **Deploy Properties** (see **Database** in [Viz Configuration](#)).

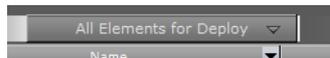
This section contains the following topics:

- [Deploy Panel Overview](#)
- [Deploy Panel Procedures](#)

6.7.1 Deploy Panel Overview



- Click to open an Element filter menu (1).



-  Shows information about the deploy action. Choose to close the Deploy Information panel and keep messages or not. The icon will show orange when information is added (2).
- Select a Deploy server and click **Deploy** to start the deploy action (3).



- **Drag and drop unwanted Elements from the Deploy panel (4).**



Element Context Menu in the Deploy Panel

1. Right-click an Element to open its context menu.
2. Select **Search for this file in current GH**.
The Element will show in the Server area.

See Also

- [Deploy Panel Procedures](#)
- [Element Context Menu](#)

6.7.2 Deploy Panel Procedures

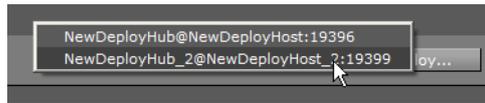
This section contains the following procedures:

- [To Deploy Selected Elements](#)
- [To Deploy Single Elements](#)
- [To Delete Elements from the Deploy Panel](#)

To Deploy Selected Elements

1. Click on **Deploy**.
2. Drag and drop all the required Elements from the Server Area to the Deploy Area.

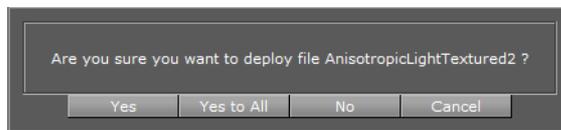
3. Select a deploy server from the drop-down box.



4. Click **Deploy....**



5. Select from:
 - **Yes:** Deploy each Element one at a time
 - **Yes to All:** Deploy all Elements together
 - **No:** Cancel each Element one at a time
 - **Cancel: Cancel the deployment**



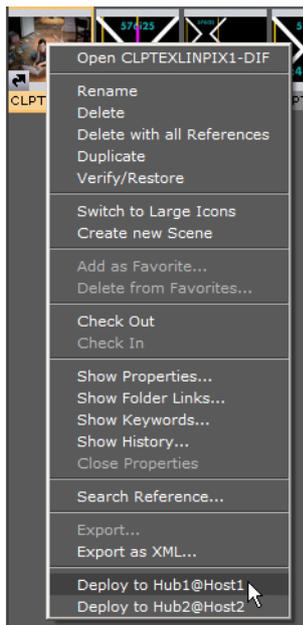
6. After deployment or cancellation select to clear the deploy list or not.

Note: If the list is not cleared all the Elements will stay in the Deploy list until required again. The list will automatically clear when Viz Artist is restarted.

To Deploy Single Elements

1. Right-click on the selected Element in the Server Area.
2. In the context menu select **Deploy to <selected server>**.

Note: If there is more than one deploy server, select from the list.



To Delete Elements from the Deploy Panel

- Drag and drop an Element or selected Elements to the trash

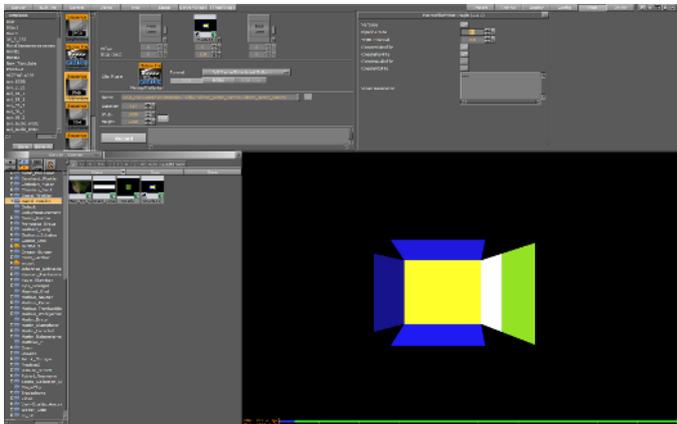


See Also

- [Deploy Panel Overview](#)
- [Element Context Menu](#)

6.8 Post Rendering

Viz Artist Scenes can be exported as video or image files. This process is called post rendering.

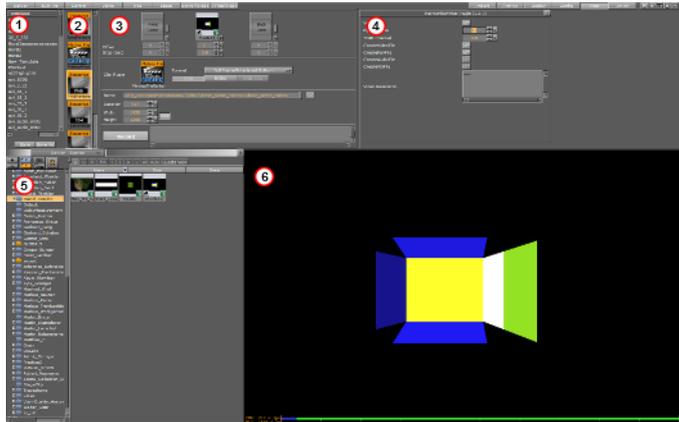


Note: Post rendering is not possible with clips or live video feeds. However, these items can be post rendered through the [SoftClip](#) plugin, if it is animated through the stage.

This section contains information on the following topics:

- [Post Render Screen](#)
- [Post Render a Scene](#)
- [Post Render Templates](#)
- [Post Rendering Plugins](#)
- [Advanced Issues with Video Codecs](#)

6.8.1 Post Render Screen



In the post render video you will find:

- Template (1)
- On the far left is the bar where all installed plugins are listed (2). Additional plugins can be installed at any time
- Post rendering setup (3):
 - **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.
- Plugin property editor (4)
- Server area (5)
- Output (6)

6.8.2 Post Render a Scene

This section contains the following procedure:

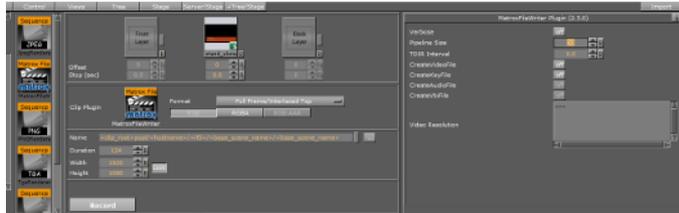
- [To Post Render a Scene](#)

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. Options are:
 - JPEG
 - PNG
 - MOV

- TGA
- TIFF
- VideoRenderer
- MatroxFileWriter

Note: Matrox File Writer requires a Matrox card to be installed.



Note: See [Post Rendering Plugins](#) section for additional Matrox File Writer settings.

4. Set an Offset, if required.
You can see the how the offset affects the scene in the post rendering area.
5. Set a Stop (in seconds), if required.
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. 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. 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 shown 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.

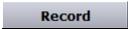
8. In the **Name** field, assign a destination file path and name.

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.



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, 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](#)

6.8.3 Post Render 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 panel for saved templates.

Note: Post rendering templates are saved in the Graphic Hub *and not* on the local machine.

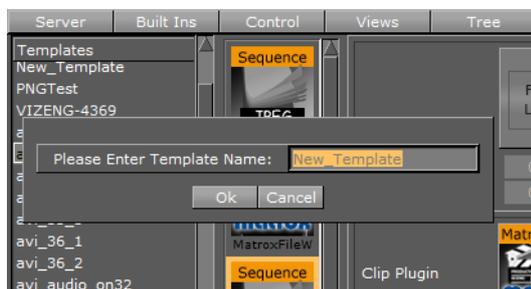
This section contains the following procedures:

- [To Save a Post Render Template](#)
- [To Change a Post Render Template](#)
- [To Load a Post Render Template](#)
- [To Rename a Post Render Template](#)
- [To Delete a Post Render Template](#)

To Save a Post Render Template

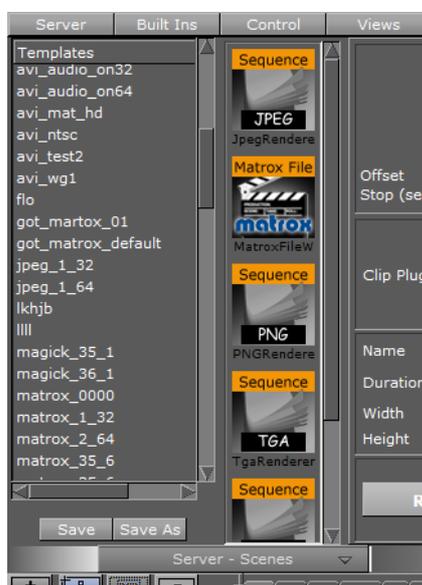
1. Configure all the settings in the procedure [To Post Render a Scene](#).
2. In the Templates panel, click **Save**.
3. Enter a name for the template.
4. Click **OK**.

Saved templates show in the Templates panel.



To Change a Post Render Template

1. Double-click the post rendering template in the Templates panel.
2. Change the required settings.
3. Click **Save**.

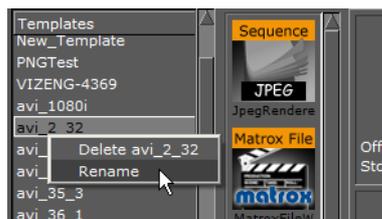


To Load a Post Render Template

- Double-click the post rendering template in the Templates panel.

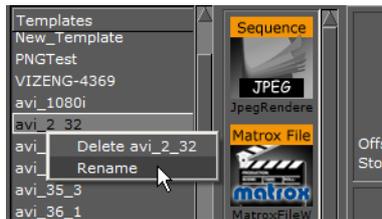
To Rename a Post Render Template

1. In the Templates panel, right-click the template and select **Rename**.
2. Enter a new name and click **OK**.



To Delete a Post Render Template

- In the Templates panel, right-click the template and select **Delete <name>**.



6.8.4 Post Rendering Plugins

Post rendering plugins define the format of the post rendered files. Viz Artist offers a set of image and video rendering plugins:



- [Image Render Plugins](#)
- [Video Clip Render Plugins](#)

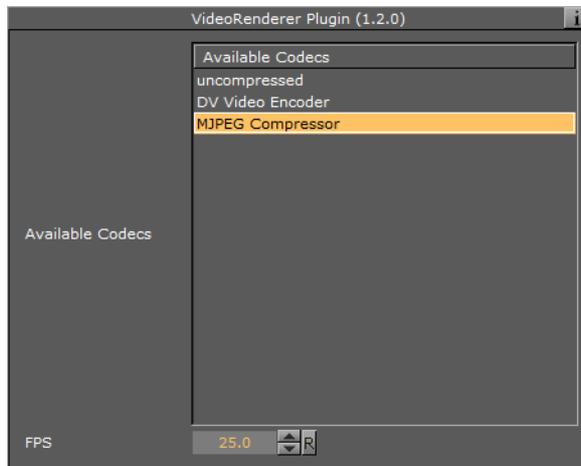
Image Render Plugins

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.

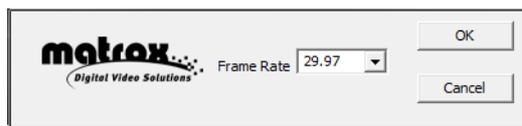
Some of the plugins offer a Quality setting that allows to setup the quality level of the jpeg images to be created.

Video Clip Render Plugins

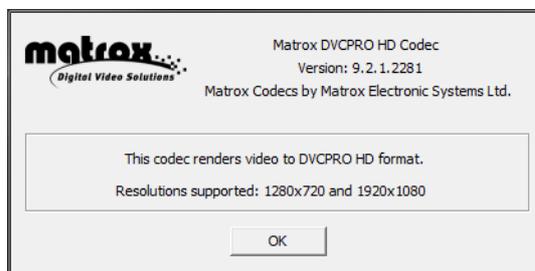
The video rendering plugins render a Scene to a Video clip. If the Video Renderer or QuickTime Renderer plugin are used, a menu showing all installed video codecs will show:



- **Available Codecs:** List all available codecs.
- **Codec Settings:** Shows the available codec settings (if available).



- **Codec Info:** Shows information about the codec (if available).



- **FPS:** Set the frame rate per second
- **Flicker Filter:** Set the Flicker filter rate

Select the codec to be used and specify the parameters for Frame rate and Flicker Filter. If the video codec allows, specify codec parameters.

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.

Note: The **Video Output** DVI out setting must be set to **Inactive** (see [Viz Configuration](#)).

When a Scene is to be Post Rendered ([To Post Render a Scene](#)),

Note: These additional settings for the **Matrox File Writer** plugin:



- **Pipeline Size:** 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.
- **TDIR Interval:** Set the interval time, in seconds, to update the video file while it is recorded. The value must be 1 or more (less than one will not activate the TDIR feature).

Note: TDIR is only supported on '.avi', '.p2.mxf', 'xdcam.mxf', 'xdcam_422.mxf', '.mov', and '.wav' containers. The MATROX_MOV .mov containers write a separate file (<basename>.Ref<.extension>) that is updated to read the .mov file. This reference file needs to be used while the file is growing. The QT_MOV containers update the .mov file directly.

- **CreateVideoFile:** Set to On to create a fill clip.
- **CreateKeyFile:** Set to On to create a key file.
- **CreateAudioFile:** Not in use.
- **CreateVbiFile:** Not in use.
- **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 show when this is set to On.
- **Video Resolution:** Shows clip resolution information.

See Also

- [Supported Codecs.](#)

6.8.5 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 32-bit uncompressed video.
- Not all codecs support both 24bit and 32-bit 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 show. Also, some codecs will crash unless they get a handle.
- Files created in raw uncompressed mode can display artifacts when played back 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.
- The free Matrox Video for Windows (VFW) software codecs let you render and play back Matrox .avi files without having Matrox video hardware installed. The current release as of this writing (2015/1 codec version 2.0) supports both Windows 7 and Windows 8.1. Visit the Matrox Support site for more information and to download the codec installer: http://www.matrox.com/video/en/support/windows/vfw_software_codecs/ http://www.matrox.com/video/en/support/windows/vfw_software_codecs/ [Matrox VFW codecs](http://www.matrox.com/video/en/support/windows/vfw_software_codecs/). 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 Matrox VFW Codecs 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 MPEG2 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 (playback only)
 - Matrox M-JPEG and HDV (playback only)

See Also

- [Video Clip Playback Use Cases](#)
- [Grabbit Properties](#)

6.9 Integration with 3rd Party Software

Viz supports the import of Scenes from third party file formats.

This section contains the following topics:

- [Adobe After Effects Integration](#)
- [CINEMA 4D Integration](#)
- [TriCaster Integration](#)

6.9.1 Adobe After Effects Integration

Adobe After Effects project files can be imported directly into Viz Artist (see [Working with Adobe After Effects Files](#)).

This section contains the following topics:

- [System Requirements](#)
- [To Use After Effects Import with 32-Bit Viz Engine](#)

See Also

- [Working with Adobe After Effects Files](#)
- [Import Files and Archives](#)

System Requirements

These are the minimum requirements to import After Effects Scenes into Viz Artist:

- Viz Artist 3.7.0 or higher
- Graphic Hub 2.4.0 or higher
- Adobe After Effects CS6 (or higher with Viz 3.7.1)
- Matrox X.mio2, X.mio2 Plus or X.open (for clip playback)

When Viz Artist/Engine is installed the file AEEExport.aex is copied to the folder with highest Adobe After Effects version number. If no Adobe After Effects version (CS6, CC) is installed, then the AEEExport.aex file is not copied.

If After Effects is installed after Viz Artist/Engine, then starting the Viz Artist/Engine installer with the **Repair** option will install the file.

To Use After Effects Import with 32-Bit Viz Engine

It is recommended to use Adobe After Effects (AE) import with the 64-bit version of Viz Engine, but if required AE Import can be used with 32-bit Viz Engine by following these steps:

1. Even if usage is on a 32-bit Viz Engine the 64-bit Viz Engine must be installed. The 64-bit installer will copy the required plugin *AEEExport.aex* to the AE install directory: *<viz install folder> \import\AE*
2. Copy the **AE** plugin directory from 64-bits Viz from step (1) above to the 32-bits Viz Engine location: *%ProgramFiles(x86)%\vizrt\Viz3\import* - this will normally be: *C:\Program Files (x86)\vizrt\Viz3\import*
3. Restart the Viz Engine

6.9.2 CINEMA 4D Integration

Scenes imported into Viz Artist from CINEMA 4D can be connected through a CINEMA 4D LiveLink tool (see [Working with CINEMA 4D Files](#)).

IMPORTANT! CINEMA 4D: a license is required for the import of CINEMA 4D Scene, issued by Vizrt.

IMPORTANT! The CINEMA 4D LiveLink tool is only available on Viz Artist 3.7 (or higher) 64-bit version only.

This section contains the following topics:

- [System Requirements](#)
- [CINEMA 4D Set-up](#)
- [CINEMA 4D LiveLink Tool Installation](#)

System Requirements

These are the minimum requirements to import CINEMA 4D Scenes into Viz Artist:

- **CINEMA 4D:** Version R15

IMPORTANT! If the CINEMA 4D version is R15 <= SP2, do the procedure detailed in: <viz install folder>\r15_sp2_fix, after the installation.

- **Viz Artist:** Version 3.7 or higher (64-bit only)
- **License:** Maxon CINEMA 4D license
- **To Import:**
 - **Viz Artist:** Version 3.7 or higher (64-bit only)
 - **License:** CINEMA 4D import 1 license

CINEMA 4D Set-up

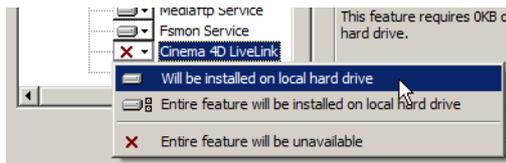
This section details how to set up CINEMA 4D for Viz Artist.

1. Open CINEMA 4D.
2. Click *Edit -> Preferences -> Files*
3. Enable **Save Polygons for Melange**.
4. If required, also set **Save Animation for Melange** to On.

IMPORTANT! If **Save Animation for Melange** is set to On, make sure to read about the [Current Limitations](#) of Viz Artist and CINEMA 4D integration.

CINEMA 4D LiveLink Tool Installation

The CINEMA 4D LiveLink tool is part of the Viz Artist installation, but it is not installed by default. When Viz Artist is installed make sure to select CINEMA 4D LiveLink and add it as part of the install (see the [Viz Engine Administrator Guide](#)).



IMPORTANT! CINEMA 4D must be installed on the system for the installer to correctly install the LiveLink tool.

6.9.3 TriCaster Integration

The integration of Viz Artist and TriCaster provides a fill and key IP stream, which is automatically detected by TriCaster as a source for graphics overlay (see [Working with TriCaster](#))

System Requirements

- Viz Artist/Engine: 3.6.3 and higher (Video version)
- TriCaster: All versions of TriCaster

See Also

- [Working with TriCaster](#)

7 Lights

Lights are used to illuminate objects in a Scene.

How lights and materials are used in a Scene makes a big difference in the appearance of a rendered Scene. [Materials](#) define how light reflects off a surface, while direct light and ambient light levels define the light that is reflected.

If Materials are used in a Scene then lights are required as well. A Material needs to reflect light to be visible. Without light just the ambient part of the material is visible, which means that the object is rendered almost black with default settings.

This section contains information on the following topics:

- [Types Of Light](#)
- [Light Editor](#)
- [Light Visualization](#)
- [Light Source Animation](#)

See Also

- [Material Editor](#)
- [Materials](#)

7.1 Types Of Light

To understand what effect light has on the surface of rendered 3D objects, three types of light can be defined:

- [Local Light](#)
- [Infinite Light](#)
- [Spot Light](#)

When lights are used it is important to understand what effect each type of light creates on the surface of a rendered 3D object.

See Also

- [Light Editor Right Panel](#)

Local Light

Local lights have color and position within a Scene, but no single direction. They give off light equally in all directions (isotropic light source). A light bulb is a good example of a local light.

Infinite Light

Infinite lights have color and direction. They emit parallel light. This means that all light generated by infinite lights travels through a Scene in the same direction. Imagine a directional light as a light source at near infinite distance, such as the sun.

Spot Light

Spotlights have color, position, and direction in which they emit light. Light emitted from a spot light is made up of a bright inner cone and a larger outer cone, with the light intensity diminishing between the two.

7.2 Light Editor

The Light editor makes it possible to modify the light settings in a Scene. A Scene can have up to eight active, simultaneous, lights.



The Light Editor consists of two panels:

- **Left panel (1):** Shows a list of all eight editable lights.
- **Right panel (2):** Shows the parameters for the currently selected light.

This section contains information on the following topics:

- [Light Editor Left Panel](#)
- [Light Editor Right Panel](#)
- [Light Editor Procedures](#)

See Also

- [Types Of Light](#)
- [Light Visualization](#)
- [Light Source Animation](#)

7.2.1 Light Editor Left Panel

The Light Editor left panel shows a list of all eight editable lights, and their current status.

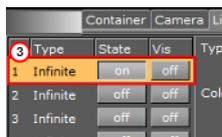
	Type	State	Vis
1	Local	on	off
2	Spot	on	off
3	Infinite	on	off
4	Infinite	off	off
5	Infinite	off	off
6	Infinite	off	off
7	Infinite	off	off
8	Infinite	off	off

- **Number:** The number of each available light, from 1 to 8
- **Type:** The type of light used, which is selected in the [Light Editor Right Panel](#)
- **State:** Set the light On or Off

- **Vis:** Set the visualization of a light On or Off (see [Light Visualization](#)):
 - **On:** The visualization of a light can be seen in the [Scene Editor](#) (and the [Scene Editor Views](#))
 - **Off:** The visualization of a light cannot be seen in the output window

Note: The Bounding Box (BB (see [Scene Editor Buttons](#))) must be set to **On**. If it is set to **Off** no light visualizations can be seen in the [Scene Editor](#) or [Scene Editor Views](#).

Note: If the Bounding Box is set to **On**, the visualization of a light, which **State** is set to **On** and is highlighted in the GUI (3), is always visible in the Scene Editor.



See Also

- [Light Editor Right Panel](#)
- [Light Editor Procedures](#)

7.2.2 Light Editor Right Panel

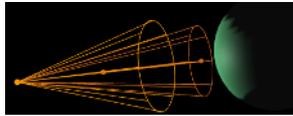
The Light Editor right panel shows the editable parameters for the currently selected light:



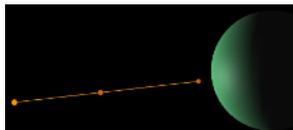
- **Type:** Select which Light type to use:
 - **Local:** See [Local Light](#) for light type details. The Local light has properties for **Color** and **Position**.



- **Spot:** See [Spot Light](#) for light type details. The Spot light has properties for **Color, Position, Direction/Rotation,** and **Spot (Penumbra and Cone Angle).**

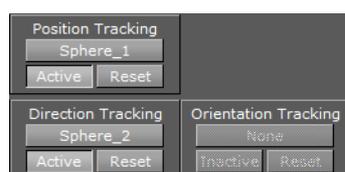


- **Infinite:** See [Infinite Light](#) for light type details. The Infinite light has properties for **Color, Position,** and **Direction/Rotation.** Infinite is selected by default.



- **Color:** Set the color of the light (see [Material Color Properties and Schemes](#)).
- **Position:** Set the position of the light along the X, Y, and Z axis.
- **Direction:** Define the direction vector for Spot or Infinite light
- **Rotation:** Set the values to rotate the **Spot** or **Infinite** light sources
- **Tracking:** A light can be configured to track the Position, and Direction or Orientation of a Container (see [To Set a Light to Track Containers](#)):
 - **Position Tracking:** The position of the light will be the same as the target Containers position
 - **Direction Tracking:** The direction of the light will be updated to always point towards the current position of the target container
 - **Orientation Tracking:** The rotation (orientation in space) of the light will be the same as the target Containers rotation (orientation in space)

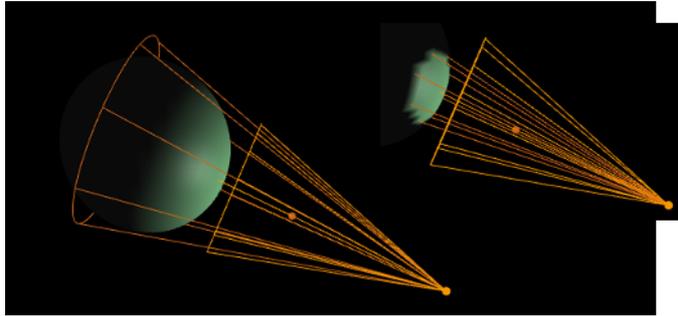
Note: Direction and Orientation tracking cannot be used at the same time. The results would contradict each other (both of them result in a potentially different rotation value for the container).



The following parameters can also be adjusted for tracking with light:

- **Active/Inactive:** Drag a Container to the drop zone, to automatically activate light tracking. To stop tracking, click the Active/Inactive button
- **Reset:** Disable light tracking
- **Spot** (available when **Spot** is selected):
 - **Penumbra:** Sets the concentration of the light (penumbra) within the light cone of the Spot light. When set to zero, the whole light cone has the same intensity. If set greater than zero, the intensity decreases away from the center

- **Cone Angle:** Adjust the cone angle to adjust the shape of the light



Note: A Material Element must be used to show light effects (see [Working with Material Elements](#)).

See Also

- [Light Editor Left Panel](#)
- [Light Editor Procedures](#)

7.2.3 Light Editor Procedures

This section contains information on the following topics:

- [To Apply Light in a Scene](#)
- [To Set a Light to Track Containers](#)
- [To Copy the Properties of One Light to Another Light](#)

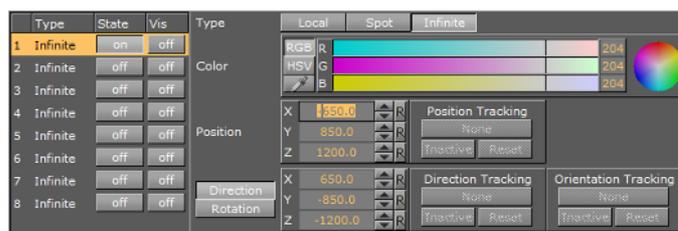
See Also

- [Light Editor Left Panel](#)
- [Light Editor Right Panel](#)

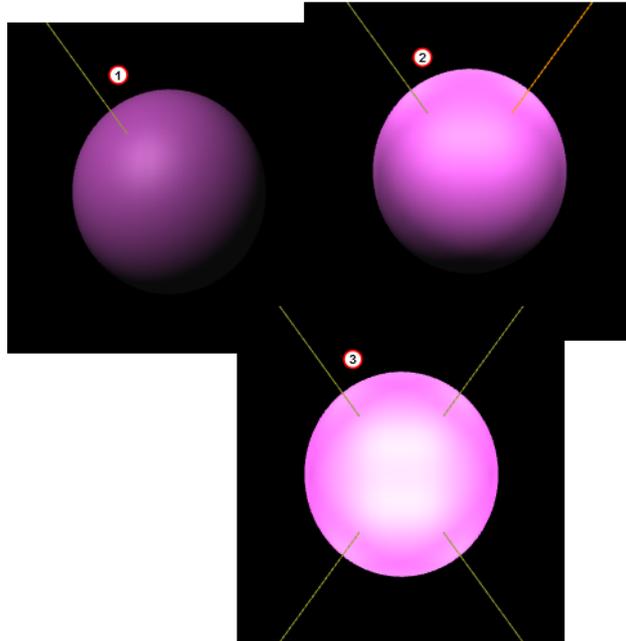
To Apply Light in a Scene

1. Click **Built Ins**.
2. Click the **Geom Plugins** tab.
3. Drag an object into the Scene Tree.
4. Add a material to the same Container.
5. Click the **Light** tab to view the Light Editor.

Note: Light number 1 is ON by default.



6. In the left panel, click any or all of the lights in the **State** column to show different light perspectives:
 - One light (1)
 - First four lights (2)
 - All lights (3)



See Also

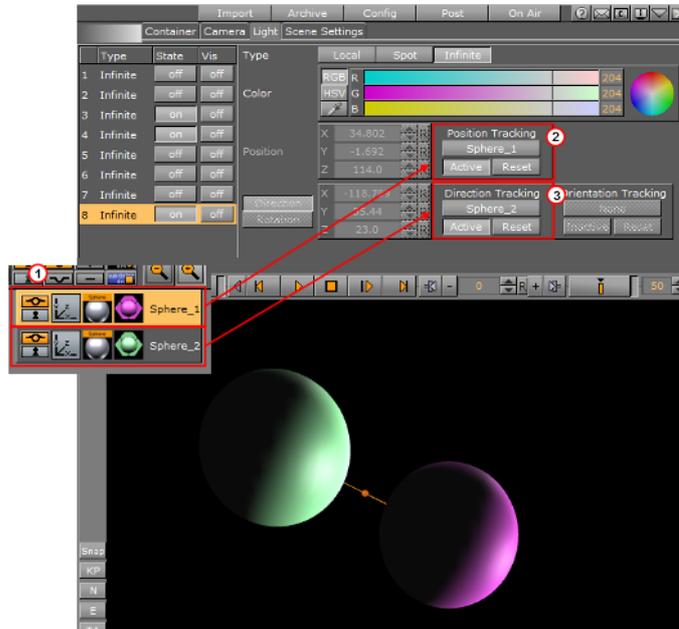
- [Lighting](#) Container Plugin

To Set a Light to Track Containers

A light can be configured to track containers in both position and direction (see **Tracking** in the [Light Editor Right Panel](#))

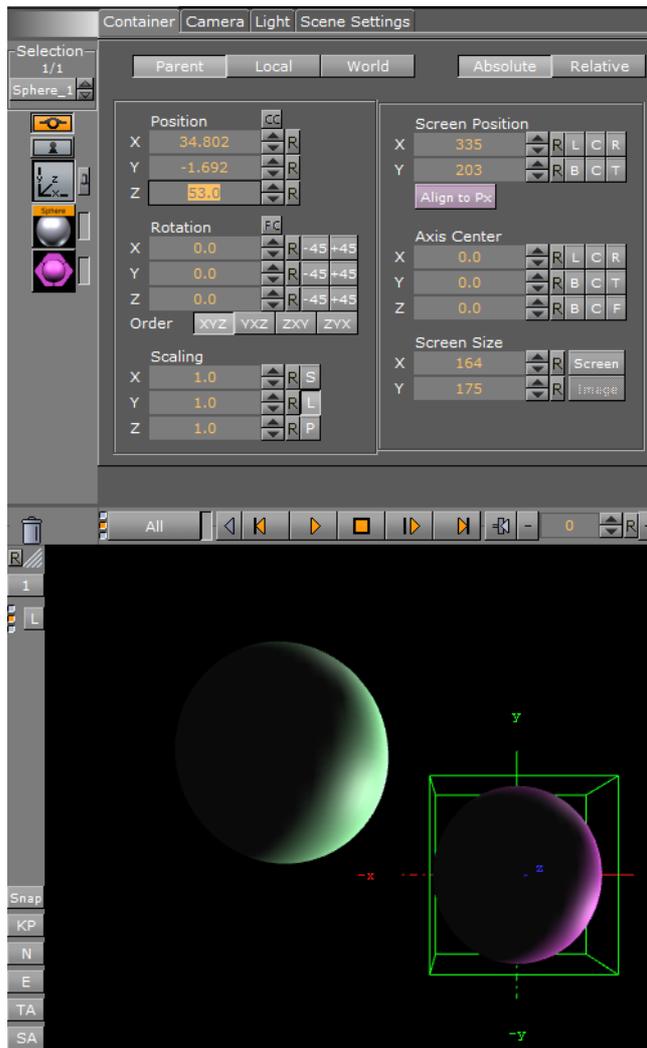
1. In the Scene Tree create two Objects (1), one to track position (Sphere_1) and one to track rotation (Sphere_2).
2. In the right panel select the **Light** tab.
3. Drag Sphere_1 to the Position Tracking drop zone (2).

4. Drag Sphere_2 to the Direction Tracking drop zone (3).



5. In the Properties panel click the **Container** tab.

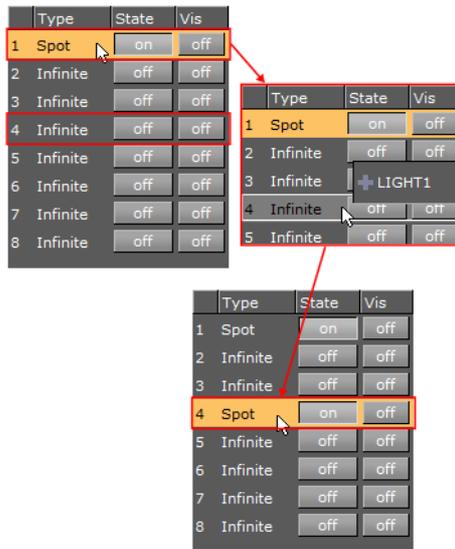
- Adjust the Position settings to track the objects.



To Copy the Properties of One Light to Another Light

- Select the light for the source properties.

2. Drag it onto the target light.



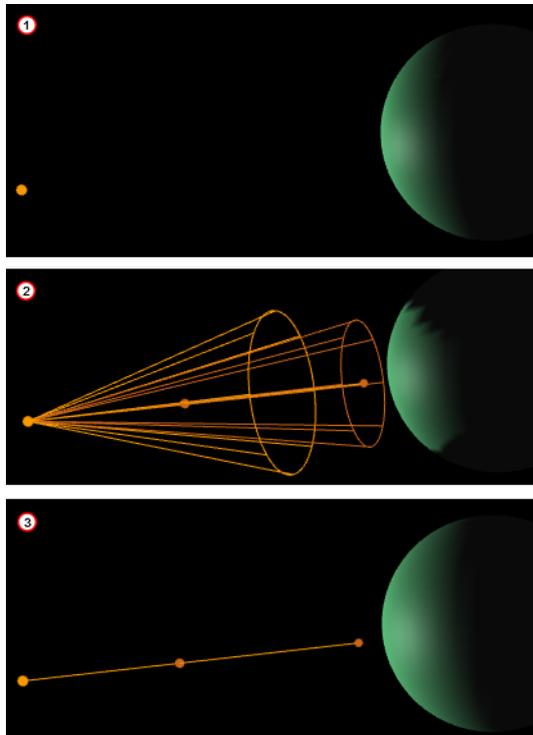
The target light will inherit all properties belonging to the source light.

7.3 Light Visualization

The position, direction or rotation of a light source can be set and modified in the [Light Editor](#) or directly in the [Scene Editor](#) or the [Scene Editor Views](#), by selection of handles, which are part of the light visualization.

In the [Light Editor](#) a light can be set to **On** or **Off** (its **State**), and its visualization (editing parameters) can also be set **On** or **Off**.

A [Local Light](#) is visualized as a single point (1), and the [Spot Light](#) (2) and [Infinite LightVizBold](#) (3) are visualized as a line with three handles. The **Spot** light also has a wire-frame box that shows the penumbra and light cone angle (the scope of the light).



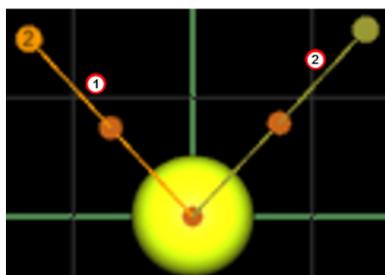
If the **State** of a light is set to **On**, and it is also highlighted in the Light Editor, its visualization will be shown in the [Scene Editor](#) (if **BB** is clicked (see [Scene Editor Buttons](#))) and the [Scene Editor Views](#).

To view the visualization of all lights, whose **State** is set to **On**, set **Vis** for each light to **On** (see [Light Editor Left Panel](#)).

	Type	State	Vis
1	Infinite	on	on
2	Infinite	on	off
3	Infinite	off	on

If all available light visualizations are visible, click on a light in the [Light Editor](#) to select its visualization in the [Scene Editor](#) and the [Scene Editor Views](#).

A selected light visualization shows orange (with its light source number) (1). An unselected light visualization shows green (2), with two handles shown in orange.



Each light visualization has handles, which are used to modify the light direction in the [Scene Editor](#) and [Scene Editor Views](#) (see [To Modify the Light Direction with Light Handles](#)).

To Modify the Light Direction with Light Handles

1. Open the [Light Editor](#).
2. Select a light, which has its **State** set to **On**.

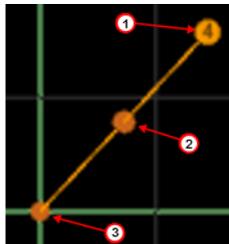
The visualization of the selected light will show in the Scene Editor.

Note: To view the visualization of all active lights set **Vis** to **On** for each light.

	Type	State	Vis
1	Local	on	off
2	Spot	on	off
3	Infinite	on	off
4	Infinite	off	off
5	Infinite	off	off
6	Infinite	off	off
7	Infinite	off	off
8	Infinite	off	off

3. Click on a light visualization handle and drag where required.

Each light visualization has handles, which are used to modify the light direction in the [Scene Editor](#) and [Scene Editor Views](#) (see [To Modify the Light Direction with Light Handles](#)).



- **Handle (1):** The light source. Drag the handle to set its position
- **Handle (2):** Drag the whole light without altering the proportionality of position and direction
- **Handle (3):** The end point. Drag the handle to set the direction and distance of the light

7.4 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 Key Frame.
3. Change the value.
4. Add a Key Frame again.

Animations can also easily be created in one of the orthogonal views. There, the light sources show as symbols, and by repositioning them and adding Key Frames, animation sequences are created. For more information, see [Light Visualization](#) .

More advanced light source animations are normally created with the light source tracking feature, see [To Set a Light to Track Containers](#) .

For more general animation information, see [Create Animations](#) .

8 Cameras

The camera which shows the Viz Artist Scene can be controlled much the same way that a real camera is controlled. Viz Artist 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:

- [Camera Editor](#)
- [Stereo Settings](#)
- [Stereoscopy Best Practices](#)
- [Stereoscopic Output Using Shutter Glasses](#)
- [Change Camera Parameters in Orthogonal Views](#)
- [Track Objects with a Camera](#)
- [Receive Tracking Data from a Real Camera](#)
- [Copy Properties from one Camera to Another](#)
- [Camera Selection](#)
- [Camera Animation](#)

8.1 Camera Editor



The Camera editor makes it possible to modify the properties for the 16 editable cameras that can be used in Viz Artist.

- **Camera Editor Left Panel (1):** Shows a list of all 16 editable cameras and the current view of each camera ([Left Panel Parameters](#))
- **Camera Editor Right Panel (2):** Shows the parameters for the currently selected camera ([Right Panel Parameters](#))

Tip: The selected camera will automatically be set as viewing **Perspective** in the Scene Editor.

To Open the Camera Editor

- In the Properties Panel click the **Camera** tab.

This section has information on the following topics:

- [Left Panel Parameters](#)
- [Right Panel Parameters](#)

See Also

- [Stereo Settings](#)
- [Stereoscopy Best Practices](#)
- [Stereoscopic Output Using Shutter Glasses](#)
- [Change Camera Parameters in Orthogonal Views](#)
- [Track Objects with a Camera](#)
- [Receive Tracking Data from a Real Camera](#)
- [Copy Properties from one Camera to Another](#)
- [Camera Selection](#)
- [Camera Animation](#)

8.1.1 Left Panel Parameters

Nr	Name	Control	Vis	
1		Editor	off	P
2		Editor	off	P
3		Editor	off	P
4		Editor	off	P
5		Editor	off	P
6		Editor	off	P
7		Editor	off	P
8		Editor	off	P
9		Editor	off	P
10		Editor	off	P
11		Editor	off	P
12		Editor	off	P
13		Editor	off	P
14		Editor	off	P
15		Editor	off	P
16		Editor	off	P

-   **Lock/Unlock:** Click the Keyhole to lock/unlock a Camera (shows orange when locked). It is not possible to use Scene Editor area keyboard shortcuts on a locked camera (see [Camera Controls and Shortcuts](#)).

Note: Keyboard shortcuts, which switch to another camera will work, for example 'Switch to Front Camera', 'Switch to Left Camera', etc.

Tip: The locked cameras can still be edited. Click the [Views Button](#) in the [Main Menu](#). In the Views panel drag the camera to change properties of the locked camera (see [Change Camera Parameters in Orthogonal Views](#)). Or change the camera parameters in the camera editor right panel (see [Right Panel Parameters](#)).

- **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:** Toggle between:
 - **Remote:** Enable to receive tracking data from a real camera.
 - **Editor:** Enable to configure camera settings.
- **Vis:** Shows the status of the camera:
 - **Off:** The camera will not be shown in the six predefined orthogonal views. To see 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 shown with infinite lines. This is a good aid for visualizing the scope of a camera that is located far away.
- **View:** Shows the current [Camera Views](#) of each camera. Options are:
 - **P:** Perspective
 - **O:** Orthogonal
 - **W:** Window

See Also

- [Right Panel Parameters](#)

8.1.2 Right Panel Parameters

The parameters in the right panel change for each type of camera view. There are three [Camera Views](#).



This section has information on the following topics:

- [Camera Views](#)
- [Parameters for Perspective View](#)
- [Parameters for Orthogonal View](#)
- [Parameters for Window View](#)

See Also

- [Left Panel Parameters](#)

8.1.3 Camera Views

There are three camera views available:

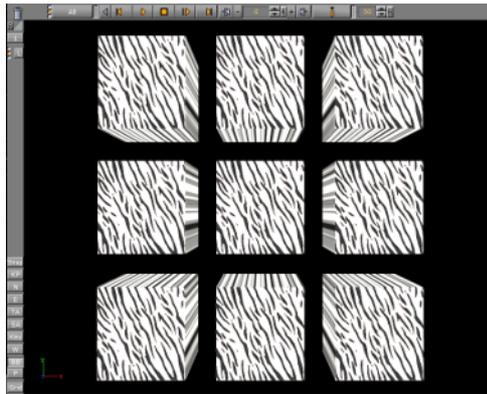
- [Perspective View](#)
- [Orthogonal View](#)
- [Window View](#)

The parameter options depend on the View selected:

- [Parameters for Perspective View](#)
- [Parameters for Orthogonal View](#)
- [Parameters for Window View](#)

Perspective View

The basic view of a virtual camera is a normal depth perspective, showing three dimensions (see [Parameters for Perspective View](#)).

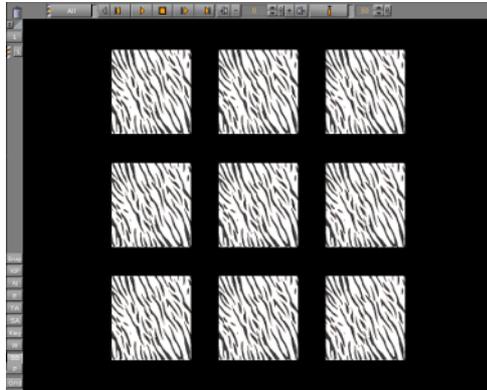


See Also

- [Parameters for Perspective View](#)

Orthogonal View

Shows a camera view without perspective, so that objects will not change their size when being moved towards or away from the camera (see [Parameters for Orthogonal View](#)).

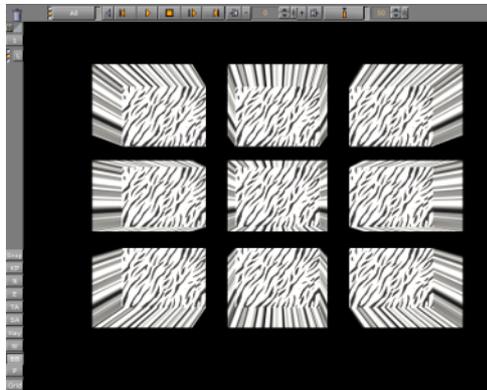


See Also

- [Change Camera Parameters in Orthogonal Views](#)
- [Parameters for Orthogonal View](#)

Window View

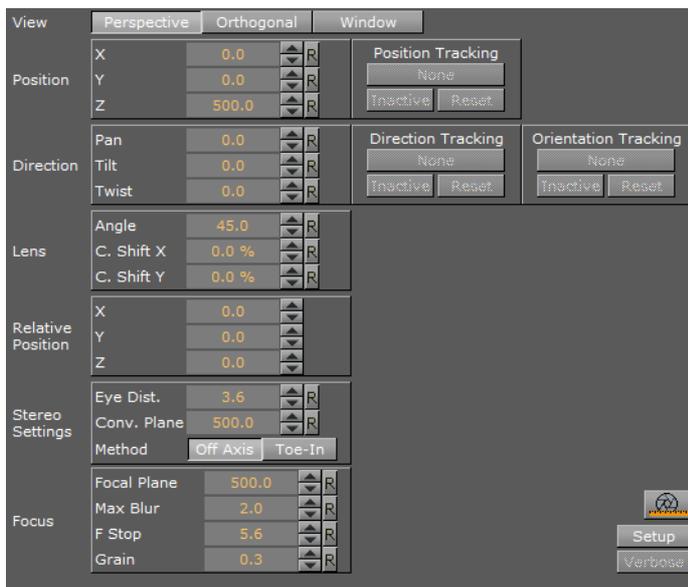
Shows the camera view like looking through a real window. The graphics, through a virtual camera, are adapted according to the real camera movement to produce this behavior. Set parameters for the physical position and rotation of the display unit (see [Parameters for Window View](#)).



See Also

- [Parameters for Window View](#)

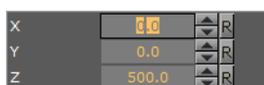
8.1.4 Parameters for Perspective View



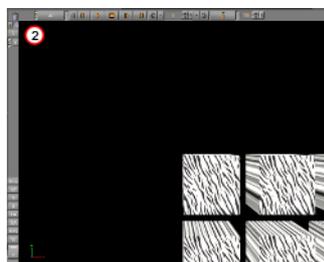
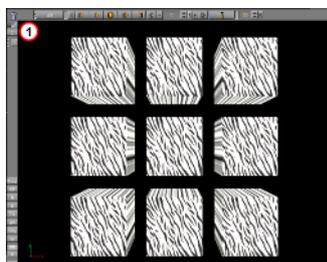
This section contains the following topics:

Position Direction Lens Relative Position Stereo Settings	Focus Tracking Miscellaneous Buttons Avoid Artifacts
---	---

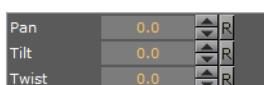
Position



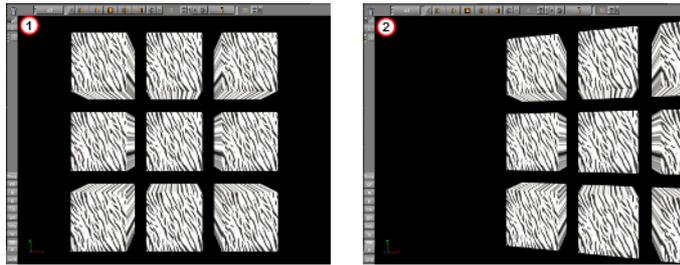
Set the position of the camera along the X, Y, and Z axis. Default setting (1) and modified setting (2)



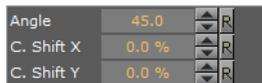
Direction



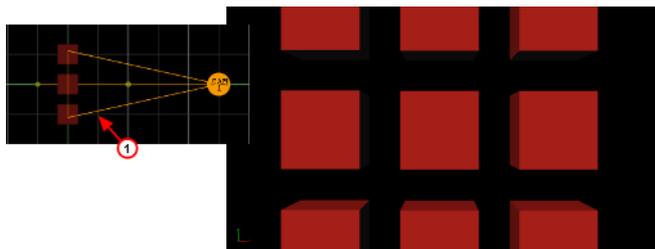
Set the values for Pan, Tilt, and Twist. Pan set to 0 (1). Pan set to 15 (2)



Lens

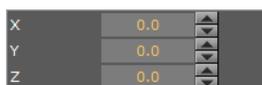


- **Angle:** Set the Camera 'field of view' angle (1)
- **C. Shift X:** Shift the center of the lens on the X axis.
- **C. Shift Y:** Shift the center of the lens on the Y axis.



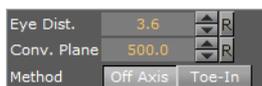
Relative Position

Set 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.



Stereo Settings

Available if set in **Camera Properties** (see [Viz Configuration](#) in the [Viz Engine Administrator Guide](#))



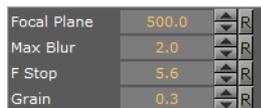
- **Eye Dist.:** Fine-tunes for individual users. A higher value for will result in more depth whereas a lower value will flatten the image.
- **Conv. Plane:** A higher value for the zero parallax distance means that objects will come closer to the viewer, a lower one will push the graphics to the back.
- **Method:**
 - **Off Axis:** Creates a distortion-free stereo pair. This is the preferred option.

- **Method 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 show 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.

Focus

Focus settings are available if the **Depth of Field** parameter is set to **Active** (see [Rendering Properties](#)).



- **Focal Plane:** Define the distance between the camera and objects which are in focus.
- **Max Blur:** Set the size of maximum blur furthest from the camera. For example a value of 20 signifies that a pixel is enlarged to a size of 20 pixels (see [Avoid Artifacts](#)).
- **F Stop:** Set the F-Stop value (relative aperture): The lens model used by the depth of field algorithm mimics a real lens behavior
- **Grain:** Define the blur level

Tracking

A camera can be set up to track the Position, and Direction or Orientation of a Container.



- **Position Tracking:** The position of the camera will be the same as the target containers position
- **Direction Tracking:** The rotation of the camera will be updated to always point towards the current position of the target container
- **Orientation Tracking:** The rotation (orientation in space) of the camera will be the same as the target containers rotation (orientation in space)

Note: Direction and Orientation tracking cannot be used at the same time. The results would contradict each other (both of them result in a potentially different rotation value for the container).

- **Active/Inactive:** Drag a container to the drop zone, camera tracking is automatically activated. To inactivate the tracking, click the **Active/Inactive** button
- **Reset:** Disables camera tracking

See Also

- [Track Objects with a Camera](#)

Miscellaneous Buttons



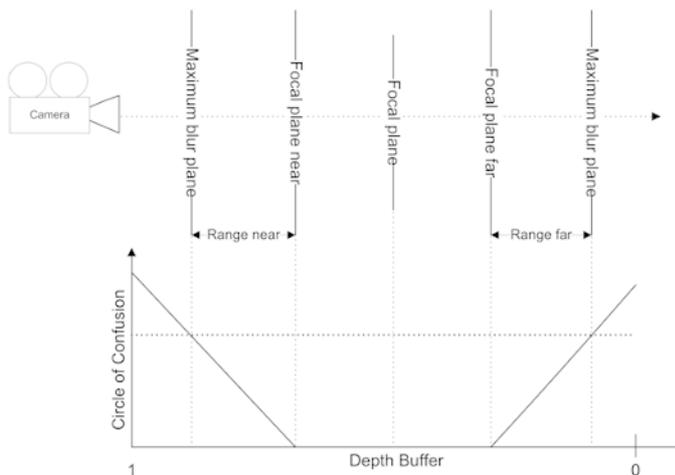
- **Lens File Calibration Editor:** Opens the Lens File Calibration Editor
- **Verbose:** Linked to cameras in virtual studio. For more information about the Virtual Studio option see the Virtual Studio documentation

See Also

- [Parameters for Orthogonal View](#)
- [Parameters for Window View](#)

Avoid Artifacts

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.

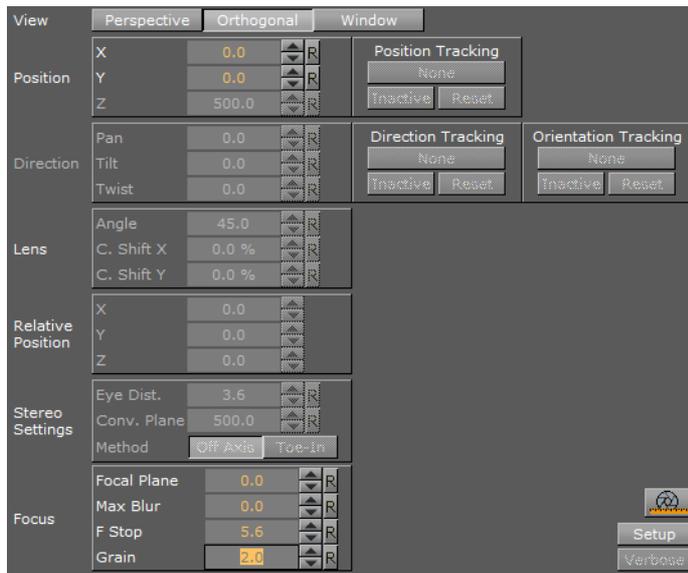


Set the clipping planes in the range you really need for the scene. So if your scene has only a depth of 200, set the far clipping plane to 250 or a similar value. Since the depth of field shader takes the depth from the depth buffer, this causes more accurate results.

Try to avoid high blur levels for objects before the focal plane, as this can cause unwanted effects.

You can increase performance, but artifacts would show for blur levels lower than 30px.

8.1.5 Parameters for Orthogonal View



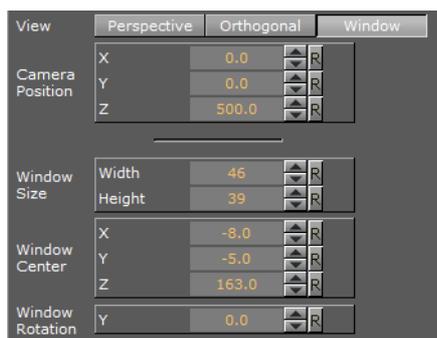
The parameters for Orthogonal view are:

- **Position:** Sets the position of the camera along the X and Y axis.
- **Focus:** See [Parameters for Perspective View](#).

See Also

- [Parameters for Perspective View](#)
- [Parameters for Window View](#)

8.1.6 Parameters for Window View



- **Camera Position:** Sets the position of the camera along the X and Y axis.

Note: When Control ([Left Panel Parameters](#)) is set to Remote, Camera Position is not available and the Tracking system controls the Camera Position.

- **Window size:** Set the window size (set to the size of the display unit).
- **Window center:** Set the position of the window from the zero position of the Tracking system.
- **Window rotation:** Rotate the window in the Y axis.

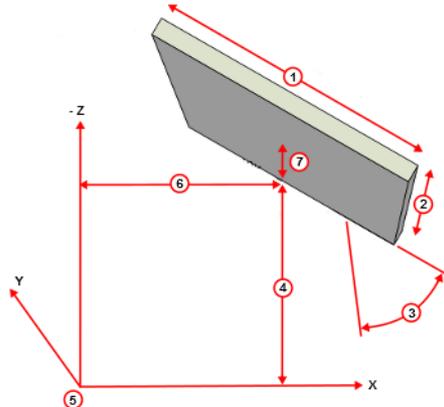
See Also

- [Parameters for Perspective View](#)
- [Parameters for Orthogonal View](#)

8.1.7 Configure a Virtual Window



1. Set the **View** to **Window**.
2. Set the **Control** to **Remote**.
3. Set **Window Size** (1 and 2). The width and height of the video wall, screen or monitor.
4. Set **Window Center** (4, 6 and 7). The distance of the screens center point to the zero point of the studio, defined in the studio setup.
5. Set **Window Rotation** (3). The rotation (y axis) angle of the video wall, screen or monitor, in relation to the coordinate system of the studio.



Item	Description
1	Window Width
2	Window Height
3	Window Rotation
4	Window Center
5	Zero Point of VSS
6	Window Center X
7	Window Center Y

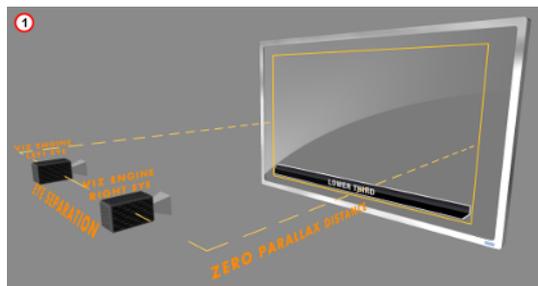
8.2 Stereo Settings

Any Viz Artist animation that was created for 2D broadcast can also be rendered as a stereo pair.

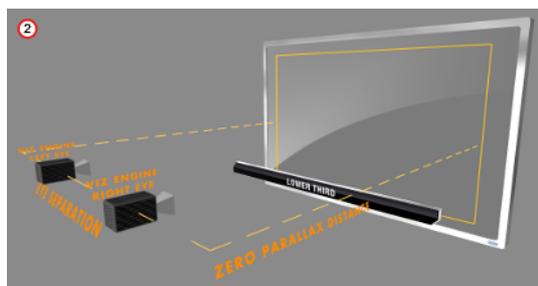
There are two parameters which define how each image is rendered. Both parameters are part of the camera configuration, but they can be changed remotely on the fly as well.

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.

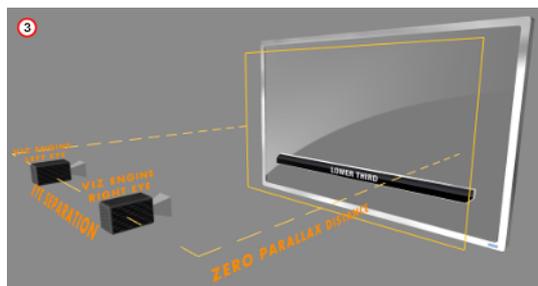
If the scene is built in the same scale as the monitor, with the camera at the same distance from the object as the viewer, and the eyes separated at the same amount an object positioned at the zero parallax, the distance shows to be flat with the screen's surface (1):



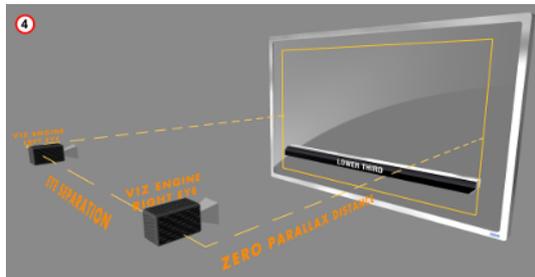
If the zero parallax distance is increased the object moves to the front (2):



If the distance is decreased the object moves away (3):



The eye separation defines how flat or deep an object shows. If the eye separation is increased, the scene shows deeper (4):



A very large eye separation parameter results in an unreadable 3D representation, as the brain cannot mix the images for the left and right eye anymore. Zero eye separation results in a flat representation, like a 2D image.

See Also

- Stereo Settings in [Parameters for Perspective View](#)
- [Stereoscopy Best Practices](#)

8.3 Stereoscopy Best Practices

The following is a list of what to do and not to do 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 show 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 artifacts.
- 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](#)

8.4 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 \geq 100Hz

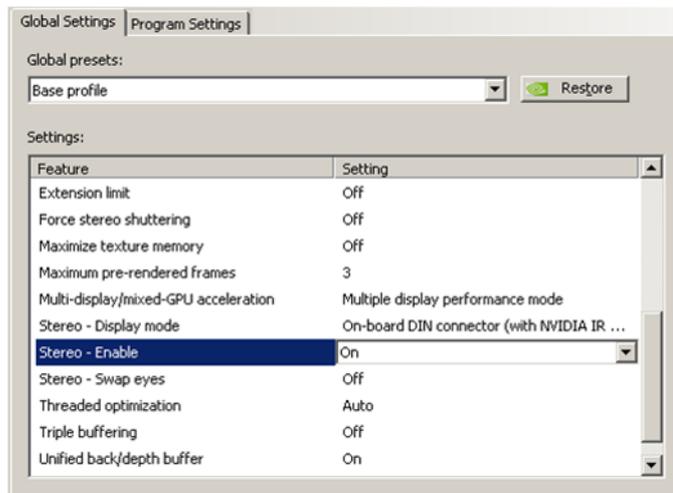
Software Requirements

- Viz Artist/Engine 3.3 or later
- The display must be set to a refresh rate \geq 100Hz

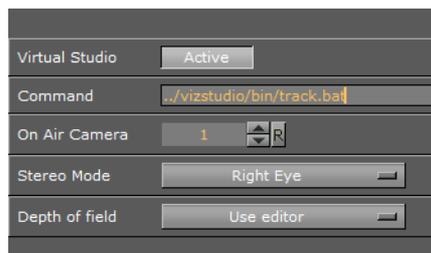
To Set Up a 3D Working Environment

1. In the Windows video board 3D settings, Set **Stereo - Enable** to **On**
2. Select the appropriate **Stereo - Display mode** for each stereo device.

Note: For NVIDIA's 3D Vision this is 'On-board DIN connector'.



- In **Camera** settings (see [Viz Configuration](#) in the [Viz Engine Administrator Guide](#)) set **Stereo Mode** to **Quad Buffered**.



Also 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.

See Also

- [Camera Editor](#)
- [Stereo Settings](#)
- [Stereoscopy Best Practices](#)
- [Change Camera Parameters in Orthogonal Views](#)
- [Track Objects with a Camera](#)
- [Receive Tracking Data from a Real Camera](#)
- [Copy Properties from one Camera to Another](#)
- [Camera Selection](#)
- [Camera Animation](#)

8.5 Change Camera Parameters in Orthogonal Views

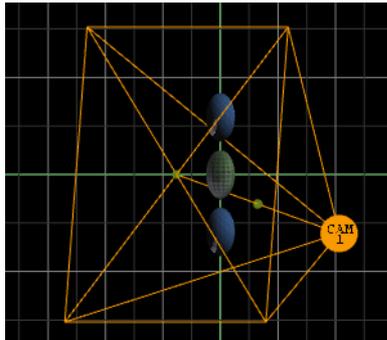
Camera parameters can be modified in the predefined orthogonal views.

This section contains information on the following procedures:

- [To View the Predefined Orthogonal Views](#)
- [To Change the Camera Views](#)

To View the Predefined Orthogonal Views

- Click **Views**.
The camera is shown as an orange circle (3) in the predefined orthogonal views. Two small, green, circular handles (1 and 2) make it possible to modify the position along the X, Y and Z axis of the camera, and also the values for pan and tilt.



To Change the Camera Views

- Drag handle (1): Change the pan and tilt while the position of the camera stays fixed.
- Drag handle (2): The camera (3) and handle (1) will be dragged along, so that pan and tilt will not be altered.
- Drag the camera (3): Change the position of the camera while handle (1) stays fixed. The values for pan and tilt will also change.

See Also

- [Camera Editor](#)
- [Stereo Settings](#)
- [Stereoscopy Best Practices](#)
- [Stereoscopic Output Using Shutter Glasses](#)
- [Track Objects with a Camera](#)
- [Receive Tracking Data from a Real Camera](#)
- [Copy Properties from one Camera to Another](#)
- [Camera Selection](#)
- [Camera Animation](#)

8.6 Track Objects with a Camera

A virtual camera can be set up to track Containers both regarding position and direction. Position Tracking means that the position of the camera will be in the center of the container. Direction Tracking direction means that the camera will look at the center of the container, for example throughout an animation.

There are three options to configure the parameters view, The options depend on the number of tracking objects:

- **Zero Tracking Objects:** Only Position X and Y can be modified
- **One Tracking Object:** When a Container is dragged to the Position Tracking drop zone, all the parameters are grayed out. In the Properties Panel, switch from **Camera** to **Container** to modify Positions X, Y and Z
- **Two Tracking Objects:** Only the Lens and Relative Position parameters can be modified

See Also

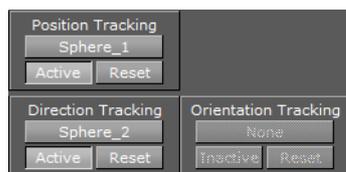
- [Tracking](#) in [Right Panel Parameters](#)

To Track an Object

1. In the Scene Tree create two Objects, one to track position (Sphere_1) and one to track rotation (Sphere_2).

Tip: It is better to use objects with a low polygon count, like a rectangle, as a tracking object. High polygon tracking objects, such as a sphere, make the scene heavy and slow.

2. Drag the first container into the Position Tracking drop zone.



3. Drag the second container into the Direction or Orientation Tracking drop zone (see [Tracking](#)).

Note: Direction and Orientation tracking cannot be used at the same time. The results would contradict each other (both of them result in a potentially different rotation value for the container).

4. Modify the 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 Properties Panel).

See Also

- [Camera Editor](#)
- [Stereo Settings](#)
- [Stereoscopy Best Practices](#)
- [Stereoscopic Output Using Shutter Glasses](#)
- [Change Camera Parameters in Orthogonal Views](#)
- [Receive Tracking Data from a Real Camera](#)
- [Copy Properties from one Camera to Another](#)
- [Camera Selection](#)
- [Camera Animation](#)

8.7 Receive Tracking Data from a Real Camera

In virtual studio 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 Panel Parameters](#), in the Control column, click the **Editor** button to set it to **Remote**.

Nr	Name	Control	Vis	
1		Remote	on	P
2		Remote	off	P
3		Editor	off	P
4		Editor	off	P
5		Editor	off	P
6		Editor	off	P
7		Editor	off	P

For more information about the Virtual Studio option, see the Virtual Studio documentation.

See Also

- [Camera Editor](#)
- [Stereo Settings](#)
- [Stereoscopy Best Practices](#)
- [Stereoscopic Output Using Shutter Glasses](#)
- [Change Camera Parameters in Orthogonal Views](#)
- [Track Objects with a Camera](#)
- [Copy Properties from one Camera to Another](#)
- [Camera Selection](#)
- [Camera Animation](#)

8.8 Copy Properties from one Camera to Another

The properties applied to one camera can be copied and applied to another camera.

To Copy Camera Properties

- In the [Left Panel Parameters](#), click the source camera and drop it on the target camera.

The properties of the source camera will be copied onto the target camera.

Note: Click on or drop the source camera in the areas shown below.

Nr	Name	Control	Vis	
1		Editor	on	P
2		Remote	on	P
3	CAMERA1		on	P
4		Editor	on	P
5		Editor	on	P
6		Editor	on	P

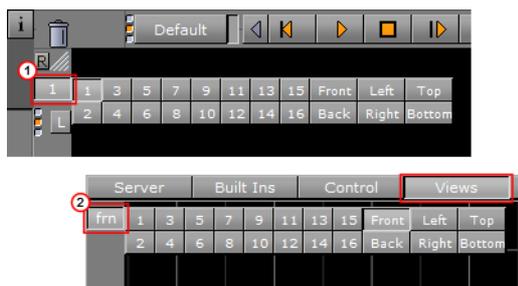
See Also

- [Camera Editor](#)
- [Stereo Settings](#)
- [Stereoscopy Best Practices](#)
- [Stereoscopic Output Using Shutter Glasses](#)
- [Change Camera Parameters in Orthogonal Views](#)
- [Track Objects with a Camera](#)
- [Receive Tracking Data from a Real Camera](#)
- [Camera Selection](#)
- [Camera Animation](#)

8.9 Camera Selection

A Scene can be viewed from various camera angles and views, with 16 editable cameras and six predefined orthogonal views available.

In the [Scene Editor](#) and [Scene Editor Views](#) the Camera Selection button (1 and 2) shows which camera or view is currently selected and used. Click the button to open the Camera Selection panel.



Each Camera can also be selected or deselected in the [Camera Editor](#).

This section contains the following topics:

- [Camera Selection Panel Properties](#)
- [Camera Selection in the Scene Editor](#)
- [Camera Selection in the Scene Editor Views](#)

See Also

- [Scene Editor Views](#)

- [Scene Editor Buttons](#)
- [Camera Editor](#)
- [Stereo Settings](#)
- [Stereoscopy Best Practices](#)
- [Stereoscopic Output Using Shutter Glasses](#)
- [Change Camera Parameters in Orthogonal Views](#)
- [Track Objects with a Camera](#)
- [Receive Tracking Data from a Real Camera](#)
- [Copy Properties from one Camera to Another](#)
- [Camera Animation](#)

Camera Selection Panel Properties

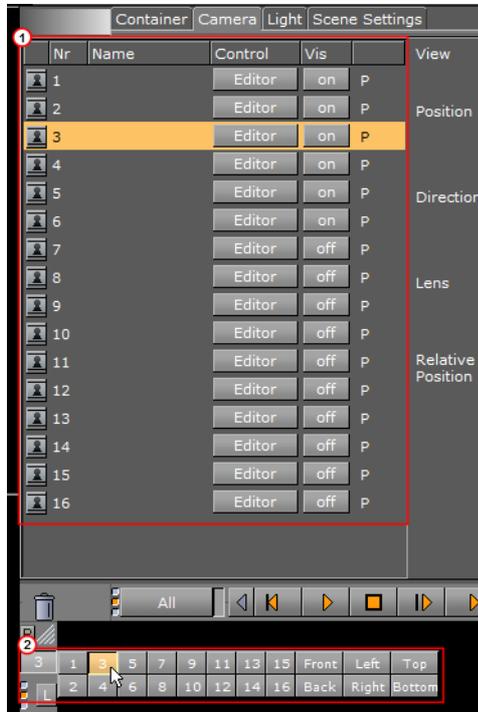
1	3	5	7	9	11	13	15	Front	Left	Top
2	4	6	8	10	12	14	16	Back	Right	Bottom

Note: The position and direction of the 16 editable cameras can be defined in the [Camera Editor](#) Panel.

Cameras and Views	Keyboard	Numpad	Mouse
1 to 10	Keys 1 to 0		Mouse click
11 to 16			Mouse click
Front: Shows output through orthogonal camera in front of the scene		7	Mouse click
Back: Shows output through orthogonal camera behind the scene		0	Mouse click
Left: Shows output through orthogonal camera at the left side		4	Mouse click
Right: Shows output through orthogonal camera at the right side		6	Mouse click
Top: Shows output through orthogonal camera above the scene		8	Mouse click
Bottom: Shows output through orthogonal camera below the scene		2	Mouse click

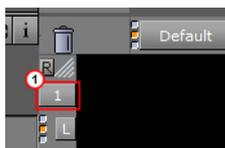
8.9.1 Camera Selection in the Scene Editor

In the Scene Editor a Scene can be viewed from different camera angles and views. Use the Camera Selection panel (2 (which also includes camera views)), or the Camera Editor (1) to select the different angles or views.

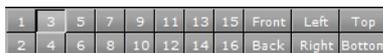


To Select a Different Camera Angle or View

1. Click the Camera Selection button (1), or open the [Camera Editor](#).



2. Select a Camera angle or view:
 - In the Camera Selection panel select a camera (1 to 16) or a camera view (see [Camera Selection Panel Properties](#)) To close the Camera Selection panel, click the Camera Selection button again.



or

- In the Camera Editor, 1) click on the required camera (2), 2) use the keyboard up/down arrows, or 3) use the keyboard PageUp/PageDown buttons (go to the top or bottom camera), to select a camera

2	Name	Control	Vis	2
1		Editor	on	P
2		Editor	on	P
3		Editor	on	P
4		Editor	on	P
5		Editor	on	P

or

- With the mouse cursor over the Scene Editor use the keyboard or Numpad (see [Camera Selection Panel Properties](#))

See Also

- [Camera Controls and Shortcuts](#)

8.9.2 Camera Selection in the Scene Editor Views

A Scene can also be viewed from different camera angles and views, in the [Scene Editor Views](#) .



To Select a Different Camera Angle or View

1. In each view click the Camera Selection button (1), as required.



2. Select a Camera angle or view:
 - In the Camera Selection panel select a camera (1 to 16) or a camera view (see [Camera Selection Panel Properties](#))

1	3	5	7	9	11	13	15	Front	Left	Top
2	4	6	8	10	12	14	16	Back	Right	Bottom

or

- With the mouse cursor over a Scene View, use the keyboard or Numpad (see [Camera Selection Panel Properties](#))

8.10 Camera Animation

All camera parameters can be animated in the same way as [Create Animations](#) in general. The process is roughly as follows:

- Set a value in the Camera editor
- Add a Key Frame
- Change the value
- Add a Key Frame again

Animations can also easily be created in one of the orthogonal views. There the cameras show as symbols and by repositioning them and adding Key Frames, animation sequences are created. For more information, see [Change Camera Parameters in Orthogonal Views](#) .

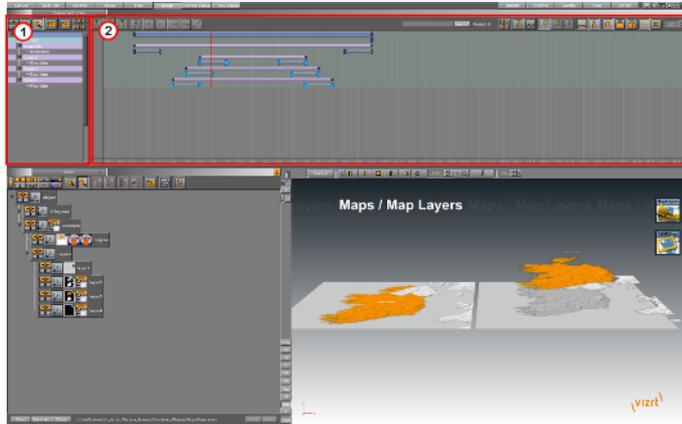
More advanced camera animations are normally created with the camera tracking feature, as in [Track Objects with a Camera](#) .

See Also

- [Camera Editor](#)
- [Stereo Settings](#)
- [Stereoscopy Best Practices](#)
- [Stereoscopic Output Using Shutter Glasses](#)
- [Change Camera Parameters in Orthogonal Views](#)
- [Track Objects with a Camera](#)
- [Receive Tracking Data from a Real Camera](#)
- [Copy Properties from one Camera to Another](#)
- [Camera Selection](#)

9 Animation Stage

The Stage is the interactive user interface to create animations.



The Stage is divided in two:

- The [Stage Tree Area](#) (1): Shows a representation of all animated elements in the current scene.
- The [Stage Editor](#) (2): Shows the animated elements as bars along the timeline.

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

The Stage can be viewed from three different views. In the [Main Menu Left](#) click on either:

- [Stage Button](#)
- [Server/Stage Button](#)
- [+Tree/Stage Button](#)

Note: For the [+Tree/Stage Button](#) view to be enabled, the screen resolution must have a width of greater than 1500.

This section contains the following topics.

- [Stage Tree Area](#)
- [Stage Editor](#)
- [Timeline Editor](#)
- [Timeline Marker](#)
- [Director Control Panel](#)
- [Director Editor](#)
- [Actor Editor](#)
- [Channel Editor](#)
- [Dopesheet Editor](#)
- [Spline Editor](#)

- [Stage Object Editor](#)
- [Key Frame Editor](#)
- [Event Editor](#)

See Also

- [Animation Controls and Shortcuts](#)
- [Stage Views](#)

9.1 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](#)

9.1.1 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.



Expand or Collapse Stage Tree Branch

Click on the arrow to the left of the item to expand or collapse a branch.

- **Expand a Stage Tree Branch:** Click the  (Expand) icon
- **Expand all Stage Tree Branches:** Click the top  (Expand) button or press <Alt> and click the Expand icon to the left of the branch name
- **Collapse a Stage Tree Branch:** Click the  (Collapse) icon to the left of the branch name
- **Collapse all Stage Tree Branches:** Click the top  (Collapse) button

9.1.2 Stage Tree Bar

The Stage Tree bar is located above the [Stage Tree](#).



The Stage Tree bar contains the these options:

-  **Expand / Collapse:** Expand or collapse all branches in the Stage Tree (see also [Expand or Collapse Stage Tree Branch](#)).
-  **Zoom Out / In:** Scales the Stage to provide a better overview of complex animations. Scales the Stage back to standard display.
-  **Add New Director:** Adds a new Director to the Stage Tree. The icon can also be dragged to add a new Director on a specific position (see [To Create a Director](#)).
-  **Add New Action:** Adds a new action to a Director (see [To Create an Action Channel](#)).
-  **Jump to Next / Previous Animation:** If an Actor (Container) has multiple animation channels placed on various Directors, select the next or previous channel in the Stage Tree.

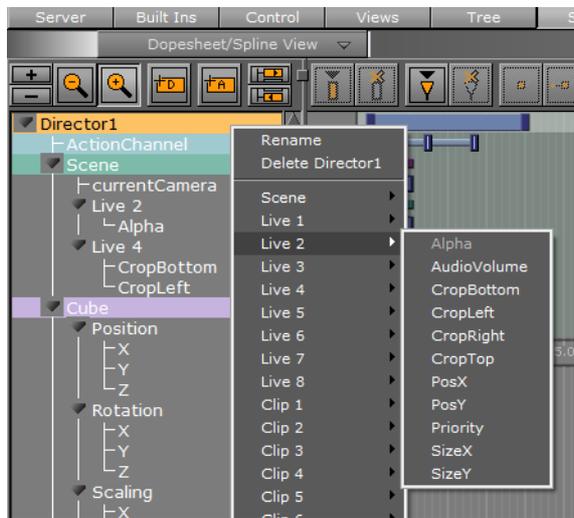
9.1.3 Stage Tree Menu

This section contains information on the these topics:

- [Directors](#)
- [Actors](#)
- [Channels](#)
- [Action Channels](#)
- [Scene Context Menu](#)

Directors

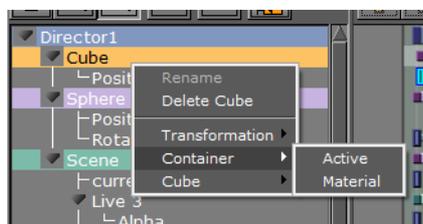
Right-click a Director to open a context menu. For more information about Directors, see [Working with Directors in the Stage Tree](#).



- **Rename:** Click to rename the Director, see [To Rename a Director](#).
- **Delete [Director]:** Click to delete the Director, see [To Delete a Director](#).
- **Scene/Live 1/Live 2 [etc.]:** Click to add a Scene or a Scene with an actor and a editable parameter.

Actors

Right-click an Actor in the Stage Tree to open its context menu (see also [Working with Actors in the Stage Tree](#)).



- **Rename:** Not available.
- **Delete [Actor]:** Deletes the actor (see [To Delete an Actor](#)).
- **Transformation:**
 - **Position:** Creates a position channel for the actor, and automatically adds two default Key Frames.
 - **Rotation:** Creates a rotation channel for the actor, and automatically adds two default Key Frames.
 - **Scaling:** Creates a scaling channel for the actor, and automatically adds two default Key Frames.

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 Key Frames. The active status can be modified in the [Key Frame Editor](#). If an active Key Frame is set to Value=**Off**, the whole actor (container) including all animations will be hidden in the Scene Editor. The action remains hidden until another Key Frame shows in the timeline with Value=**On**.
- **Material:** Creates a material channel for the actor, and automatically adds two default Key Frames. The material can be modified in the [Key Frame Editor](#). For more information (see [Working with Material Elements](#)).

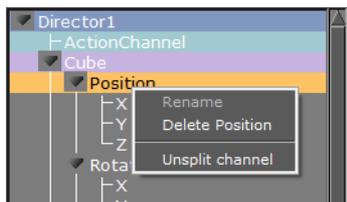
Note: The material option is only available if a material is added to the container.

- **[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](#).

Channels

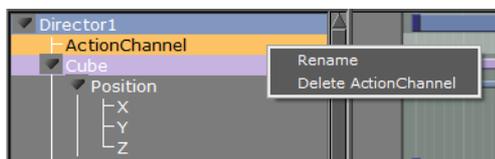
Right-click a channel, in the Stage Tree, to open the channel context menu (see also [Working with Channels in the Stage Tree](#)).



- **Rename:** Not available.
- **Delete [Channel]:** Deletes a channel (see [To Delete a Channel](#)).
- **Split/Unsplit Channel:** View or close a Transformation Channel parameters.

Action Channels

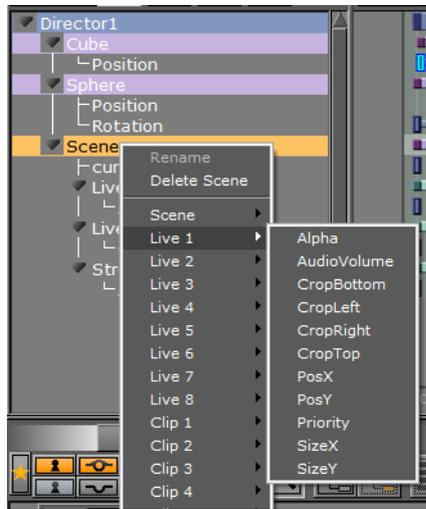
Right-click an Action Channel, in the Stage Tree, to open the Action Channel context menu (see also [Working with Action Channels in the Stage Tree](#)).



- **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](#)).

Scene Context Menu

Right-click a Scene, in the Stage Tree, to open the Scene context menu.



- **Rename:** Not available.
- **Delete Scene:** Deletes a Scene Channel inside the Stage.
- **Scene:** The Scene options refer to various Scene settings.
- **Live, Clip and Stream (Inputs):** Select input parameters such as Alpha, Audio etc.

See Also

- [Clip Key Frame Editor Properties](#)

9.2 Stage Editor

The Stage Editor area shows the animated elements as bars and splines along the timeline.



This section contains information on the following topics:

- [Stage Editor Menu](#)
- [Stage Editor Bar](#)

See Also

- [Timeline Editor](#)
- [Timeline Marker](#)
- [Spline Editor](#)
- [Stage Object Editor](#)

9.2.1 Stage Editor Menu

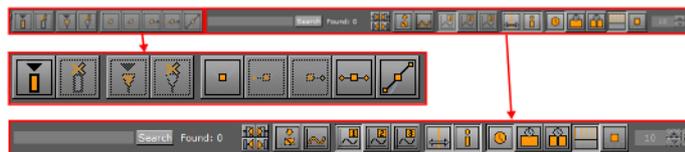
Above the [Stage Tree Area](#) is a menu that defines the appearance of the [Stage Editor](#).



- **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](#).

9.2.2 Stage Editor Bar

The Stage Editor bar is located above the [Dopesheet Editor/Spline Editor](#).



-  **Add Key Frame:** Adds a Key Frame at the current time/frame/field (see [Working with Key Frames](#)).
-  **Delete Key Frame:** Deletes the selected Key Frame (see [To Delete a Key Frame](#)).
-  **Add Stop/Tag:** Adds a stop/tag Event to the selected Director at the current time/frame/field (see [To Create a New Event](#)).
-  **Delete Stop/Tag:** Deletes the selected stop/tag Event (see [To Delete an Event](#)).
-  **Linear Left and Right:** Sets the left and right spline handles in the selected Key Frame to linear.
-  **Smooth Left and Linear Right:** Sets the spline handles in the selected Key Frame to smooth left and linear right.
-  **Linear Left and Smooth Right:** Sets the spline handles in the selected Key Frame to linear left and smooth right.
-  **Smooth Left and Right:** Sets the left and right spline handles in the selected Key Frame to smooth.
-  **Handles Tangential:** Sets the left and right spline handles in the selected Key Frame to tangential.
-  **Search:** Searches the name of a Key Frame.
-  **Found: 0 Items Found:** Show how many items were found

-  **Search Results:** The arrow buttons can be used to search for the first, next, previous and last Key Frame matching the search criteria:



- 1: **Previous** Key Frame
- 2: **Next** Key Frame
- 3: **First** Key Frame
- 4: **Last** Key Frame
-  **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 Channel.
-  **Show Spline 1, 2, or 3:** Shows 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 Key Frame, will show 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, Key Frames 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 Key Frames to field 10, 20, 30, and so on. If a Key Frame should be placed at field 23 while the snap to grid feature is enabled, this must be manually entered in the [Key Frame Editor](#).

-  **Grid Width:** Defines the width of the snap to grid in fields.

See Also

- [Stage Views](#)

9.3 Timeline Editor

The Timeline editor is located above the [Scene Editor](#). It gives various options to control animations.



- **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 or directors (see [To Stop an Animation](#)).

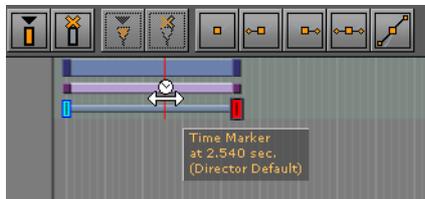
Note: The animation is automatically stopped when it reaches the end of the timeline, or a stop point along the timeline. For more information (see [Locked vs Unlocked Key Frames](#)).

- **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.
- **Key Frame Previous:** Jumps to the previous Key Frame of the selected container in the scene tree.
- **Timeline:** The input box shows the current position on the timeline in fields.
- **Jump to Next Timeline Position:** Jump to the next position in the Timeline.

-  **Key Frame Next:** Jumps to the next Key Frame of the selected container in the scene tree.
-  **Set Key:** Creates a Key Frame 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 Key Frame.

9.4 Timeline Marker

The timeline marker defines the current position along the time axis (X) in the Dopesheet and [Spline Editor](#). The timeline marker is shown 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 Scene Editor accordingly.

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 show 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 Manually Move the Timeline Marker](#)
- [To Move the Timeline Marker a Set Amount of Fields](#)
- [To Move the Timeline Marker a Set Amount of Fields with Shortcuts](#)

To Manually Move the Timeline Marker

- Drag the timeline marker along the timeline in either the [Dopesheet Editor](#) or [Spline Editor](#)

or

- 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 a Set Amount of Fields

- In the [Dopesheet Editor](#), enter the time (in fields) in the **Timeline** input box

or

1. Enable the [Stage Object Editor](#).
2. Select the Director in the [Stage Tree](#) or [Dopesheet Editor](#).
3. In the [Director Editor](#) that shows, enter the time (in fields) in the **Timeline** text box.

or

- Drag the Timeline in the Scene Editor Timeline editor  .

To Move the Timeline Marker a Set Amount of Fields with Shortcuts

- **Move One Field at a Time:**
 - In the [Dopesheet Editor](#) or [Director Editor](#) click the **Timeline** up/down buttons, or
 - Press <Ctrl> and the left or right arrow key on the keyboard

- **Move More Than One Field at a Time:**

.....
Note: The movement interval in fields, is format dependent.

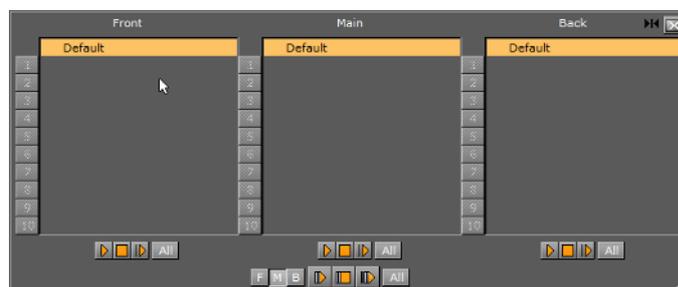
- **Press the left or right arrow key on the keyboard.** This, for example, in 1080i50 format, moves the timeline marker 5 fields, left or right, and in 1080i60 the movement is 6 fields, left or right
- **Press <Alt> and the left or right arrow key on the keyboard.** This, for example, in 1080i50 format, moves the timeline marker 50 fields, left or right, and in 1080i60 the movement is 60 fields, left or right

See Also

- [To Jump to Start of an Animation](#)
- [To Jump to the End of an Animation](#)

9.5 Director Control Panel

Clicking the Director option in the [Timeline Editor](#) opens the Director Control Panel.



The Director Control Panel lists all available Directors of the selected layer. To see all layers (Front, Main and Back) click the expand icon  .

It is possible to select one or more directors in the list to play them individually.

Note: There are two sets of control buttons; one set for controlling individual directors in the selected layer and one set for controlling the all directors in all layers.

- **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/Play Layer:** Plays the current Director(s) of all layers or the selected layer from the beginning (see [Basic Animation Functions](#)).
 - **Stop Animation/Stop Layer:** Clicking manually stops the currently playing animation at the current point on the timeline for all layers or the selected layer. 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 (see [Locked vs Unlocked Key Frames](#)).

- **Continue Animation/Continue Layer:** Clicking manually continues the animation of all layers or the selected layer to the next point on the timeline.
 - **[numbers]:** You can save up to 10 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: These selections are on a per-layer basis and also cannot be saved.

- **F, M, B:** Enables the layers that should be controlled using the Play Stop and Continue Layer buttons.

See Also

- [Director Editor](#)
- [Working with Directors in the Stage Tree](#)
- [Working with Directors in the Dopesheet Editor](#)

9.6 Director Editor

The Director editor shows details about the selected Director.

This section contains the following topics and procedures:

- [Director Editor Properties](#)
- [To Enable the Director Editor](#)

Director Editor Properties



- **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 with 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.
- **Master Clip:** Render an animation, defined in the Stage, based on the actual frame of a played clip. This clip can either be set in [Scene Settings](#) or within a separate Director in the Stage:
 - **Drop-down box:** Select, None, Clip 1 to 16. Select the Clip Channel that contains the clip, which will control the animation
 - **Inherit:** When set to **on**, the set Master Clip Channel will be passed to all sub directors.
 - See [Using the Master Clip Feature](#)
-  **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 (see [Locked vs Unlocked Key Frames](#)).

-  **Continue Animation:** Continues the animation after it has been stopped (see [To Continue an Animation](#)).

To Enable the Director Editor

1. Do 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 Views](#)
- [Using the Master Clip Feature](#)

9.7 Using the Master Clip Feature

The Master Clip feature is particularly useful when you need to tie graphic animations to a specific location in a video clip. Using the director properties to tie a master clip to a director in the [Director Editor](#) will ensure that the stage and the clip are in sync. This is to make sure, that when the clip plays the frame at time 3:12 for example, you will also get the animation of the graphic rendered at 3:12.

It is essential that you have at least 2 top-level directors in a scene using the Master Clip feature. One contains the clip only, and the second all other scene elements. The second director can also contain several sub-directors. For the animations in the second director to be bound to the clip it is necessary to set the Master Clip to the clip channel containing the clip for this director. If there are sub-directors, which should also respect the clip as the animation controller, the button *Inherit* in the [Director Editor](#) must be checked.

A simple scene using the Master Clip feature could be constructed like this:



Note: Please keep in mind that scrubbing in the controlled director is not possible when using the Master Clip feature as the timing is set by the clip. Scrubbing is only supported in the director containing the clip.

9.8 Actor Editor

The Actor Editor shows details about the selected Actor.



- **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. Do 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](#)

9.9 Channel Editor

The Channel editor shows details about the selected Channel.

This section contains information on the following topics:

- [To Enable the Channel editor](#)
- [Common Channel Editor Properties](#)
- [Position Channel Editor Properties](#)
- [Action Channel Editor Properties](#)

See Also

- [Working with Channels in the Stage Tree](#)
- [Working with Channels in the Dopesheet Editor](#)

To Enable the Channel editor

1. Do the procedure: [To Enable/Disable the Stage Object Editor](#).
2. Click a Channel in the [Stage Tree](#) or in the [Dopesheet Editor](#).

9.9.1 Common Channel Editor Properties



Note: Not available for 4 Point Loop clips (see [3 and 4-Point Loops in Clip Player](#)).

IN:

-  **Swing In:** Enables swing in
-  **Loop In:** Enables loop in
-  **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)

OUT:

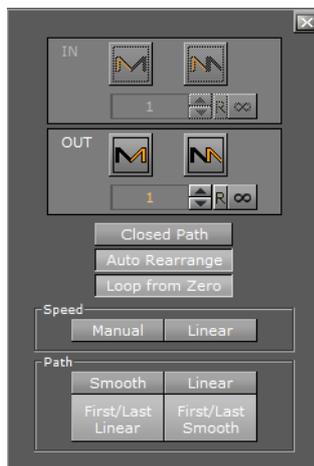
-  **Swing Out:** Enables swing out
-  **Loop Out:** Enables loop out
-  **Number of Swings/Loops:** Defines the number of swings/loops out
-  **Infinite:** Sets the number of swings/loops out to infinite (animation never ends)

See Also

- [Position Channel Editor Properties](#)
- [Action Channel Editor Properties](#)

9.9.2 Position Channel Editor Properties

The **Position Channel** editor has, in addition, these extra parameters that can be edited:

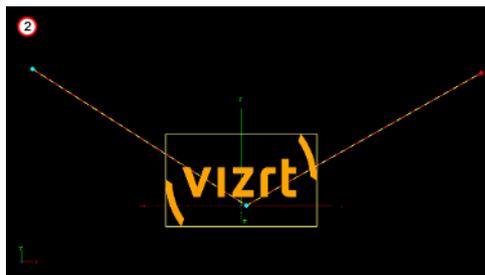
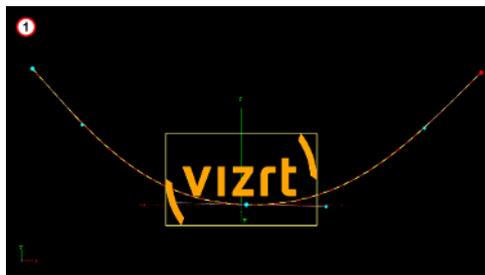


- **Closed Path:** When activated, the animation path will go from the last Key Frame to the first. When deactivated, the animation path will remain open.
- **Auto Rearrange:** By default, this setting is enabled, ensuring that the order of the Key Frames in the Dopesheet editor is the same as in the [Scene Editor](#) so that the animation only ever continues forward. When disabled, if you change the order of the Key Frames in the Dopesheet editor, the spline will remain the same, but the animation may return from the previously last Key Frame to the newly added Key Frame 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:** Locks all Key Frames. There is no linear speed with this option. The duration between each Key Frame can be different, depending on the timing value and the transformation values between the different Key Frames. The result is that you will not have a continuous animation speed.
 - **Linear:** Unlocks all Key Frames. The Key Frames between the first and last Key Frames are locked. If you change the value of either the first or last Key Frame, or any of the Key Frames in between, the timing will change automatically for

each Key Frame. The result is an animation with a continuous speed between all Key Frames.

- **Path:**
 - **Smooth (1):** Sets the animation path to be smooth.
 - **Linear (2):** Sets the animation path to be linear.
 - **First/Last Linear:** Sets the animation to be linear for the first and last Key Frame, and smooth for the rest.
 - **First/Last Smooth:** Sets the animation to be smooth for the first and last Key Frame, and linear for the rest.

Note: These settings are global for the whole animation in question.



See Also

- [Position Key Frame Editor Properties](#)

9.9.3 Action Channel Editor Properties

The **Action Channel** editor has, in addition, these extra parameters that can be edited:



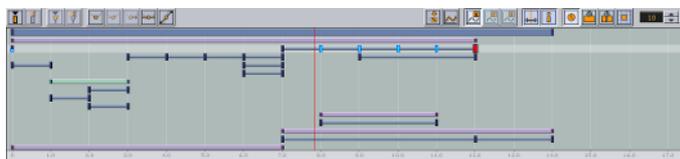
- **Always Execute Action at 0:** If an action is set to execute at 0 on the timeline, press this button to make sure that the action is executed at 0. If this button is not pressed (inactive) the action may not work.

See Also

- [Action Key Frame Editor Properties](#)
- [Working with Action Channels in the Stage Tree](#)
- [Working with Action Channels in the Dopesheet Editor](#)

9.10 Dopesheet Editor

The Dopesheet editor shows the items in the Stage Tree as bars along the timeline.



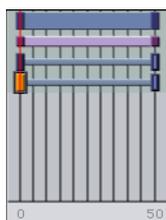
The items are presented vertically in the Dopesheet editor according to the structure in the Stage Tree. The horizontal scaling represents the time value.

Tip: The time value can be set to either seconds, frames, or fields from the [Stage Editor Bar](#).

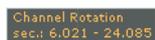
The [Timeline Marker](#) defines the current time, and the Scene Editor shows the animation accordingly.

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.

The image below shows the Dopesheet editor with **Snap to Grid** option enabled with a field width of five:



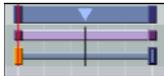
If the Show Info option is enabled in the [Stage Editor Bar](#), a tool tip with time information will show when the pointer is held over an item in the Dopesheet editor:



The Dopesheet editor can contain bars for various stage items: Directors, Actors, Channels, etc. Similar to the [Stage Tree](#), Directors are shown with a gray blue color, Container Actors with light purple, while channels and actions are shown with a gray color.

To build up an animation, Key Frames and stops/tags can be added to the Scene. A Key Frame defines a property at a given time. Multiple Key Frames belonging to the same channel are connected in the Dopesheet editor with one or more bars.

The image below shows an animation where a stop/tag has been added between two Key Frames:



A stop/tag can be added to a Director to end or pause an animation at a given time (see [Working with Events in a Director](#)).

Example: If the scaling of a cube is 1.0 at field 0 and 2.0 at field 50, one Key Frame will be created at field 0 and another at field 50. A bar will connect the two Key Frames in the Dopesheet editor. See the image below:



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 Key Frame will open the [Key Frame Editor](#), and so on.

To Enable/Disable the Dopesheet Editor

- From the [Stage Editor Menu](#) select **Dopesheet View**, or
- Hold the pointer over the Stage and press <Ctrl+2>.

To Enable the Dopesheet Editor Together with the Spline Editor

- From the [Stage Editor Menu](#) select **Dope/Spline View**, or
- Hold the pointer over the Stage and press <Ctrl+4>.

To Resize the Height or Width of the Dopesheet Editor

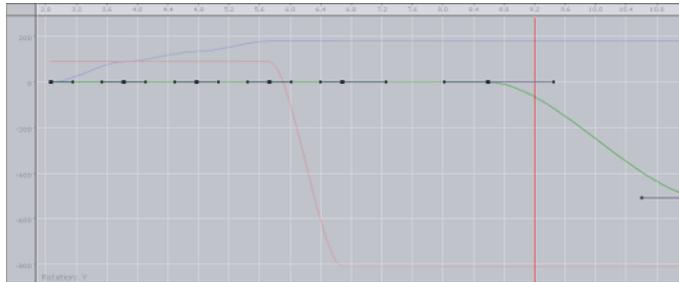
- **Height:** Drag the horizontal separator between the Dopesheet editor and [Spline Editor](#).
- **Width:** Drag the vertical separator between the [Stage Tree Area](#) and [Stage Editor](#).

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 Key Frames](#)

9.11 Spline Editor

The Spline editor shows the Key Frames on a Channel as splines along the timeline.



The Key Frames 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 Scene Editor shows the animation accordingly.

It is possible to drag Key Frames in the Spline Editor both horizontally (changing the time) and vertically (changing the value) at once.

This section contains information on the following topics:

- [Spline Editor Procedures](#)
- [Editing Handles](#)
- [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](#)

9.11.1 Spline Editor Procedures

This section contains information on the following procedures:

- [To Enable/Disable the Spline Editor](#)
- [To Enable the Spline Editor Together with the Dopesheet Editor](#)
- [To Resize the Height or Width of the Spline Editor](#)
- [To Switch Between the Various Splines](#)
- [To Move Key Frames in the Spline Editor](#)

To Enable/Disable the Spline Editor

- From the [Stage Editor Menu](#) select **Spline View**, or
- Hold the pointer over the Stage and press <Ctrl+3>.

To Enable the Spline Editor Together with the Dopesheet Editor

- From the [Stage Editor Menu](#) select **Dope/Spline View**, or
- Hold the pointer over the Stage and press <Ctrl+4>.

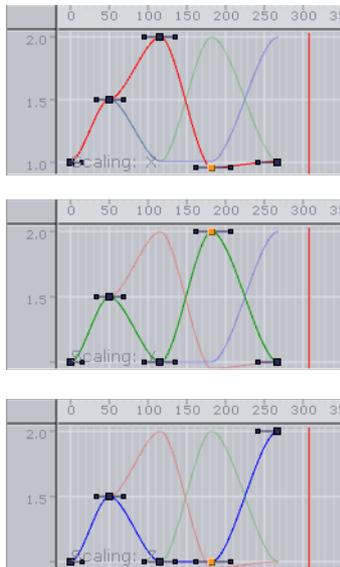
To Resize the Height or Width of the Spline Editor

- **Height:** Drag the horizontal separator between the Dopesheet editor and [Timeline Marker](#).
- **Width:** Drag the vertical separator between the [Stage Tree Area](#) and [Stage Editor](#).

To Switch Between the Various Splines

- Use the Show Spline 1,2, and 3 buttons from the [Stage Editor Bar](#), or
- Use the keyboard shortcuts <Ctrl+Z>, <Ctrl+X> and <Ctrl+C>

Example of axes X, Y, and Z in the Spline editor



To Move Key Frames in the Spline Editor

- Select the required Key Frame 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 Key Frame if you want to snap to grid along the timeline.

You can move a Key Frame in the spline editor, so either the position on the timeline (left and right) or the value stored (up and down) in the Key Frame will be changed.

9.11.2 Editing Handles

As each Key Frame 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 required 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 Key Frame as long as you press <Ctrl>.

Furthermore you can use the buttons in the main menu to edit the handles at a Key Frame. To do so select the Key Frame you want to change the handles in and select the action in the main menu. For ease of use multiple Key Frames can be selected.

-  Set left and right spline handles in selected Key Frame to linear.
-  Set spline handles in selected Key Frame to smooth left and linear right.
-  Set spline handles in selected Key Frame to linear left and smooth right.
-  Set left and right spline handles in selected Key Frame to smooth.
-  Set left and right spline handles in selected Key Frame to tangential.

See Also

- [Animation Controls and Shortcuts](#)

9.11.3 Position in the Spline Editor

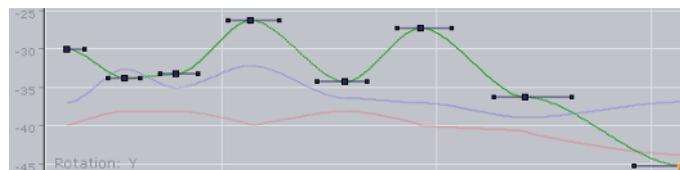
If you animate a position, there are much more considerations to make.

See Also

- [Path Control](#)
- [Spline Control](#)
- [Position Key Frame Editor Properties](#)

9.11.4 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.



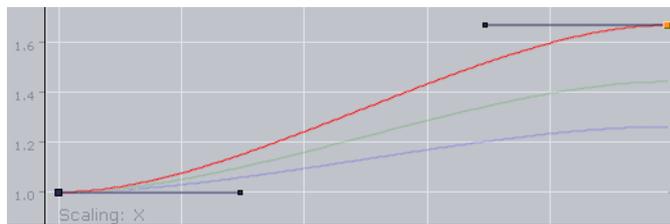
See Also

- [To Create Key Frames for Rotation](#)
- [To Modify the Rotation Settings](#)
- [Rotation Key Frame Editor Properties](#)

9.11.5 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.

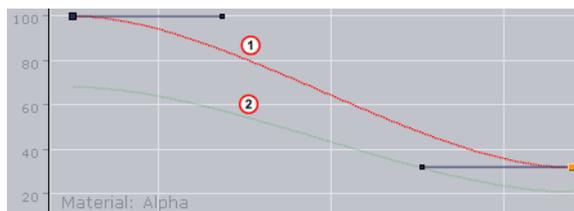


See Also

- [Scale Key Frame Editor Properties](#)

9.11.6 Material in the Spline Editor

If the material has been animated, 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 shown which parameter is currently selected.



See Also

- [To Add and Animate a Material](#)
- [Material Key Frame Editor Properties](#)

9.11.7 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 Key Frames/handles of one spline, you must use the number keys in the [Stage Editor Bar](#) to select the required spline.

Pressing **<Shift>** while moving a Key Frame 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 Key Frame will enable the Snap to Grid function, allowing the Key Frame 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 shown 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 Key Frames with handles.

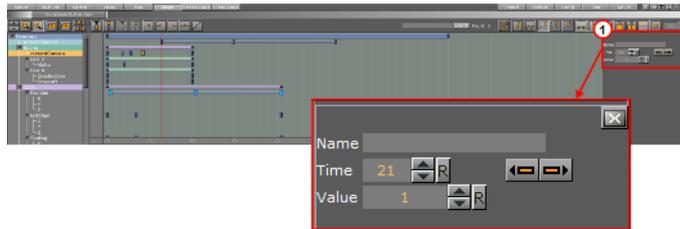
Example of Key Frame and handle tool tips



If the Show Info option is enabled in the [Stage Editor Bar](#), placing the pointer over a Key Frame or handle will show a tool tip with status information about timeline and value.

9.12 Stage Object Editor

The Stage Object editor (1) is positioned at the right side of the [Stage Editor](#). The editor has different views depending on the particular editor to be shown.



Example: The [Director Editor](#) shows information about directors, the [Actor Editor](#) shows information about actors, the [Channel Editor](#) shows information about channels, the [Action Editor](#) shows information about actions, the [Key Frame Editor](#) shows information about Key Frames, the [Event 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**, or
- With the mouse pointer over the Stage area, press <Ctrl+1>.



9.13 Key Frame Editor

The Key Frame editor shows details about the selected Key Frame. To view the Key Frame editor click on a Key Frame.

Make sure that the [Stage Object Editor](#) is enabled.

This section contains information on the following topics:

- [Common Key Frame Editor Properties](#)
- [Locked vs Unlocked Key Frames](#)
- [Action Key Frame Editor Properties](#)
- [Position Key Frame Editor Properties](#)

- [Rotation Key Frame Editor Properties](#)
- [Scale Key Frame Editor Properties](#)
- [Material Key Frame Editor Properties](#)
- [Clip Key Frame Editor Properties](#)

See Also

- [Event Editor](#)

9.13.1 Common Key Frame Editor Properties



- **Name:** Defines the name of the Key Frame. This name is used for external commands or scripts (see how [To Rename a Key Frame](#)).
- **Time:** Defines where along the timeline (in fields) the Key Frame is placed.
-  **Previous/Next buttons:** If the property has more than one Key Frame, the previous or next Key Frame along the timeline is selected.
- **[Value]:** The rest of the Key Frame 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.

See Also

- [Locked vs Unlocked Key Frames](#)
- [Action Key Frame Editor Properties](#)
- [Position Key Frame Editor Properties](#)
- [Rotation Key Frame Editor Properties](#)
- [Scale Key Frame Editor Properties](#)
- [Material Key Frame Editor Properties](#)
- [Clip Key Frame Editor Properties](#)
- [Event Editor](#)

9.13.2 Locked vs Unlocked Key Frames

By default the Key Frames 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 Key Frame will be passed by in the animation.

An unlocked Key Frame stores only information of the position along X-, Y- and Z-axis, without the information at which time the Key Frame should be passed. To reflect this, in the object editor for a locked Key Frame the **Time** is grayed out. Furthermore, as the spline editor shows time and path-percentage, the unlocked Key Frames will not be shown there. In the animation editor unlocked Key Frames will be shown at half height

of locked Key Frames. Viz Artist calculates the time the Key Frame will be passed to fit with the handles set in the next locked Key Frame before and after the unlocked Key Frame.

To better understand the difference between locked and unlocked Key Frames, create a simple position animation. You will see the container moving with different speeds between the Key Frames. Next unlock all Key Frames and play the animation once more. The container will move evenly during the whole animation.

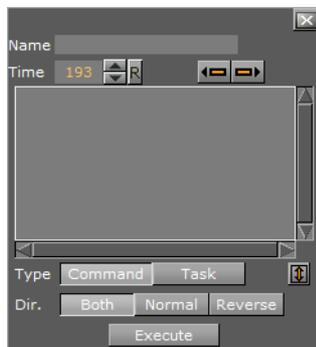
Using unlocked Key Frames you can define the path to fit your needs without editing the time for every Key Frame. This will save you a great amount of time.

See Also

- [Common Key Frame Editor Properties](#)
- [Position Key Frame Editor Properties](#)
- [Rotation Key Frame Editor Properties](#)
- [Scale Key Frame Editor Properties](#)
- [Material Key Frame Editor Properties](#)
- [Clip Key Frame Editor Properties](#)
- [Event Editor](#)

9.13.3 Action Key Frame Editor Properties

In addition to the [Common Key Frame Editor Properties](#) the Action Key Frame Editor defines the Action Channel. The action will work when the Director timeline equals the Action Channel Key Frame.



- **Value:** Enter the command or OS task here
- **Type:** The action Key Frames can initialize two main types of actions:
 - **Command:** Use Viz Artist internal commands to execute a great number of automated tasks in the program. For Viz Artist internal commands documentation go to: <viz install folder> / Documentation / CommandInterface / index.html>
 - **Task:** A normal shell command, like a dos-prompt. Anything which can initialize from the OS command prompt can, in principle, be executed as a task in an Action Key Frame. If the path to the executable file contains spaces, set the complete path in quotation marks, for example, "C:\program files\microsoft office\office\winword.exe"

Note: Multiple engine commands can be executed in the Action Key Frame Editor. The commands must be separated with ; (semi-colon) and on a new line, for example: `RENDERER*TREE*$sometext*GEOM*TEXT SET Hello World;`
`MAIN_SCENE*STAGE*DIRECTOR*$Default GOTO_TRIO $B $A;`

- **Dir:** Sets which way, forwards or backwards, the animation should go to trigger the Action Key Frame:
 - **Both:** The Key Frame will be triggered when the animation goes forward and in reverse.
 - **Normal:** The Key Frame will be triggered when the animation goes forwards.
 - **Reverse:** The Key Frame will be triggered when the animation goes backwards.
- **Execute:** Starts the action

Examples of Internal Commands for Action Key Frames

Here are some examples of Viz Artist internal commands, which can be used in the Action Key Frame Editor to create actions:

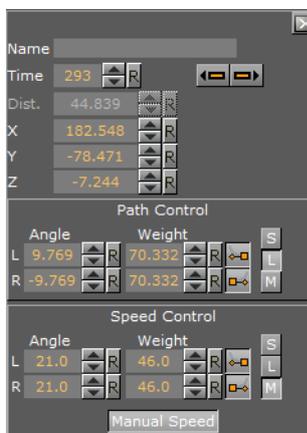
- **THIS_SCENE*:** The created action acts on this Scene only
- **THIS_EDITOR*:** The created action acts on this Editor only
- **THIS_DIRECTOR*:** The created action acts on this Director only
- **THIS_PARENT_DIRECTOR*:** The created action acts on this Parent Director only

See Also

- [Action Channel Editor Properties](#)
- [Working with Action Channels in the Stage Tree](#)
- [Working with Action Channels in the Dopesheet Editor](#)
- **Scene Default Values** (see the [Viz Engine Administrator Guide](#))

9.13.4 Position Key Frame Editor Properties

In addition to the [Common Key Frame Editor Properties](#), a Position Key Frame contains information about the position of a Container at a specific time. The parameters of Position Key Frames are:



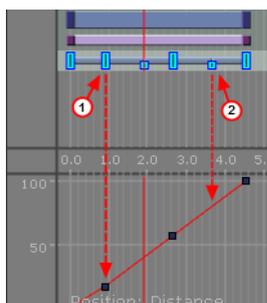
- **Distance:** The distance of a selected Key Frame (value in percent) between the first and last Key Frame of the animation (information only)
- **X:** Defines the position along the X axis.
- **Y:** Defines the position along the Y axis.
- **Z:** Defines the position along the Z axis.

Path Control

- **Angle:** Defines the angle of the handles for the path.
- **Weight:** Defines the weight of the handles for the path.
-  **Smooth Left/Right:** Allows the handles for the path to be smooth or linear (see [Channel Editor](#)).
- **S / L / M:** Defines the way handles for the path should be modified:
 - **S (Single):** Only the left or right parameter can be adjusted.
 - **L (Locked):** Left and right parameters are locked in their current position and adjustment changes both left and right parameters together.
 - **M (Mirror):** The left and right parameters are mirrored when adjusted

Speed Control

- **Angle:** Defines the angle of the handles for the speed.
- **Weight:** Defines the weight of the handles for the speed.
-  **Smooth Left/Right:** Allows the handles for the speed to be smooth or linear.
- **S / L / M:** Defines the way handles for the speed should be modified:
 - **S (Single):** Only the left or right parameter can be adjusted.
 - **L (Locked):** Left and right parameters are locked in their current position and adjustment changes both left and right parameters together.
 - **M (Mirror):** The left and right parameters are mirrored when adjusted
- **Manual Speed:** Click to toggle the Key Frame Active or Inactive:
 - **Active:** Speed Control is unlocked (1). The speed between the previous, selected, and the next Key Frame is calculated by the position and the speed control of this Key Frame in the timeline.
 - **Inactive:** Speed Control is locked (2). The speed between the previous and the next Key Frame is calculated linear between those two and cannot be modified. If the value of either the previous or next Position Key Frame is changed, or any of the Position Key Frames, the timing will be adjusted automatically for each locked Key Frame. The result is an animation with a constant speed between all locked Key Frames.

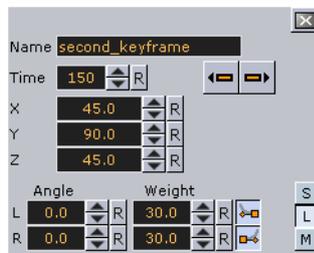


See Also

- [Common Key Frame Editor Properties](#)
- [Locked vs Unlocked Key Frames](#)
- [Action Key Frame Editor Properties](#)
- [Rotation Key Frame Editor Properties](#)
- [Scale Key Frame Editor Properties](#)
- [Material Key Frame Editor Properties](#)
- [Clip Key Frame Editor Properties](#)
- [Event Editor](#)

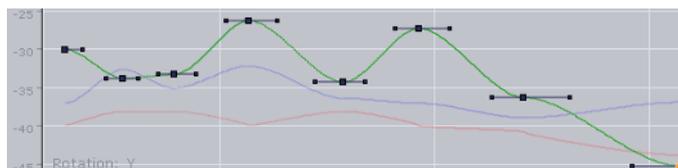
9.13.5 Rotation Key Frame Editor Properties

A rotation Key Frame contains information about the container rotation. In addition to the default options in the [Key Frame Editor](#), for a rotation Key Frame it is possible to define specific rotation options.



- **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.

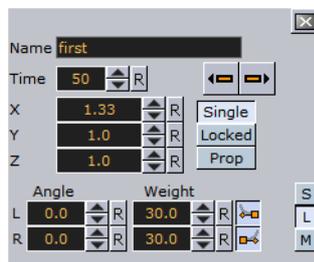
**See Also**

- [Common Key Frame Editor Properties](#)

- [Locked vs Unlocked Key Frames](#)
- [Action Key Frame Editor Properties](#)
- [Position Key Frame Editor Properties](#)
- [Scale Key Frame Editor Properties](#)
- [Material Key Frame Editor Properties](#)
- [Clip Key Frame Editor Properties](#)
- [Event Editor](#)

9.13.6 Scale Key Frame Editor Properties

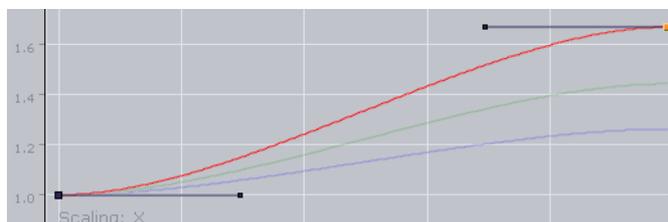
Scale Key Frames have to do with the scaling of the object in the animation. For a scale Key Frame, in addition to the default options in the [Key Frame Editor](#), it is possible to define the scaling along the X, Y, and Z axes in the three input boxes.



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.

Scale axis, X, Y and Z



See Also

- [Common Key Frame Editor Properties](#)
- [Locked vs Unlocked Key Frames](#)
- [Action Key Frame Editor Properties](#)
- [Position Key Frame Editor Properties](#)

- [Rotation Key Frame Editor Properties](#)
- [Material Key Frame Editor Properties](#)
- [Clip Key Frame Editor Properties](#)
- [Event Editor](#)

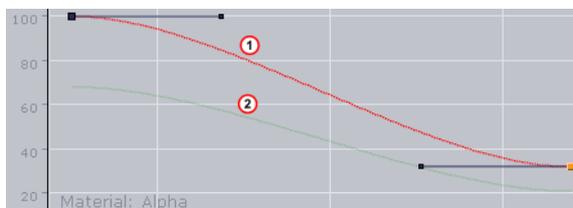
9.13.7 Material Key Frame Editor Properties

With a material Key Frame, in addition to the default options in the [Key Frame Editor](#), it is possible to define color.



If a material has been animated, the object editor for the Key Frames shows a [Material Editor](#). This allows you to conveniently change the material.

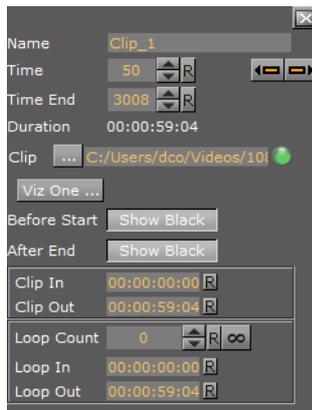
Furthermore in the spline editor two different splines will be shown. The red one is for the alpha value (1), the green one for the shininess (2). As you can only edit the Key Frames/handles of one spline, you must use the number keys in the [Stage Editor Bar](#) to select the required spline. For ease of use in the lower left corner you will find indicated which parameter is currently selected.



See Also

- [Common Key Frame Editor Properties](#)
- [Position Key Frame Editor Properties](#)
- [Rotation Key Frame Editor Properties](#)
- [Scale Key Frame Editor Properties](#)
- [Clip Key Frame Editor Properties](#)
- [Event Editor](#)

9.13.8 Clip Key Frame Editor Properties



- **Time/TimeEnd:** Set the start and end time, in fields, of the Key Frame.
- **Duration:** Time duration of selected clip Key Frame set by **Time/TimeEnd**
- **Clip:** Click the **...** and select a clip to use
- **Viz One:** Select a clip from Viz One
- **Before Start:** Set 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:** Set 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/Out:** Set the time the clip comes in and goes out
- **Loop Count:** Set the loop count for 4 Point Loop clips (see [3 and 4-Point Loops in Clip Player](#))
- **Loop In:** Set the loop in time
- **Loop Out:** Set the loop out time

Example:

- Time = 50
- Time End = 250
- Clip Length = 4 sec (50i clip)
- Clip In = 00:00:01:00
- Clip Out = 00:00:03:00

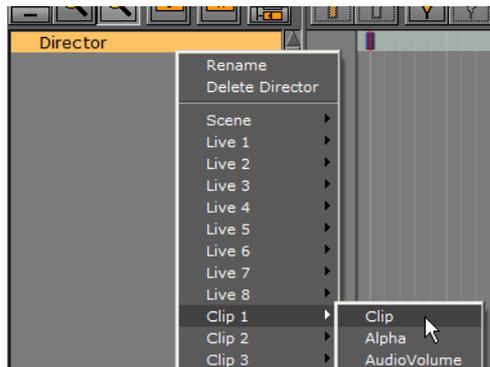
The clip playback starts 1 second after the Director starts. The clip starts at frame 25 and ends at frame 75. Because the Key Frame duration is twice as long the clip part will be played twice.

See Also

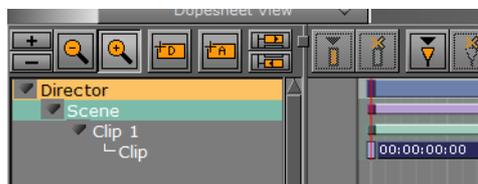
- [To Animate a Clip](#)

To Add a Clip to the Stage

1. Right-click the Director where clip is to be added.



2. Select **Clip <number>** from the context menu.
3. Click **Clip**.
 - To open the clip Key Frame editor click the Clip Key Frame in the Dopesheet



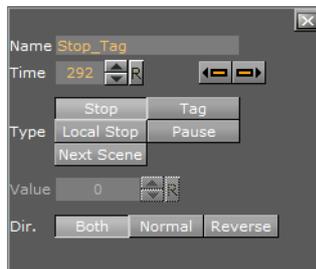
See Also

- [To Animate a Clip](#)
- [Common Key Frame Editor Properties](#)
- [Locked vs Unlocked Key Frames](#)
- [Action Key Frame Editor Properties](#)
- [Position Key Frame Editor Properties](#)
- [Rotation Key Frame Editor Properties](#)
- [Scale Key Frame Editor Properties](#)
- [Material Key Frame Editor Properties](#)
- [Event Editor](#)

9.14 Event Editor

The Event Editor shows details about the selected Stop, Tag, Pause, Local Stop or Next Scene Events.

Event Editor



- **Name:** Set a name for the Event. This name is used for external commands or scripts (see [To Name an Event](#)).
- **Time:** Sets where in the timeline (in frames) the Event is placed (see [To Move an Event along the Timeline](#)).
-  **Previous/Next buttons:** If a Director has more than one Event click the Previous or Next button to move to the previous or next Event along the timeline.
- **Type:** Defines the type of Event used:
 - **Stop (1):** Stops the animation until continue animation is clicked. A stop applies for the Director and all subdirectors.
 - **Tag (3):** Shows a position along the timeline. This Event type does not stop the animation.
 - **Local Stop (2):** Stops the animation until continue animation is clicked. A local stop only applies for the Director it is placed on, not its subdirectors.
 - **Pause (4):** Stops the animation for a specified amount of frames.
 - **Next Scene (5):** Mark a position where a Scene ends. The next Scene will start automatically.

Note: Viz Weather has an automatic play list option where the user does not need to start the next Scene. The Next Scene Event starts the next Scene automatically when the current Scene reaches the Next Scene Tag.



- **Dir: Sets the direction where the Event is valid:**
 - **Both:** The Event is valid to both normal (forwards) and reverse directions.
 - **Normal:** The Event is only valid when the animation direction is normal (forward).
 - **Reverse:** The Event is only valid when the animation direction is reversed.

To Enable the Event Editor

1. Do the procedure [To Enable/Disable the Stage Object Editor](#).
2. Click an Event in the [Dopesheet Editor](#).

See Also

- [Working with Events in a Director](#)

10 Create Animations

When an animation is created, the state of an object (for example its position, rotation, or scaling) is defined at various points in time. Key Frames 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 Events in a Director](#)
- [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 Key Frames](#)
- [Basic Animation Functions](#)
- [Creating a Basic Animation](#)
- [Create an Advanced Animation](#)
- [Further Advanced Animation Topics](#)
- [Create an Over the Shoulder Scene](#)
- [Create a Stand-alone Scene](#)
- [Create Transition Effects](#)

See Also

- [Animation Controls and Shortcuts](#)
- [Stage Views](#)

10.1 Working with Directors in the Stage Tree

To create an animation, there must be at least one Director in the Stage Tree. Directors are used to group a certain set of stage elements, such as Actors and Channels. It is possible to create Directors at the root level of the Stage Tree, or as Sub-Directors.

A Director can hold as many items as required, 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 (see [Basic Animation Functions](#)).

Note: The first Director created in the Stage is always called **Default** Director. Any subsequent Directors are called **Director** (rename as required).

Directors are shown 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

Note: The first Director created in the Stage is always called **Default** Director. Any subsequent Directors are called **Director** (rename as required).



- Click the  (Add Director) button

Note: The new Director is added under the currently selected Director. If no Director is selected then its added at the end of the director tree.

Or

- Drag the  (Add Director) icon to the Stage Tree:

Directors that are dragged to the Stage Tree can be positioned in various ways, and create either root or Sub-Directors.

-  **Sub-Director:** Drop the  icon at the right side of a Director to create a Sub-Director

-  **Root Director Below/Above:** Drop the  icon at the left side of a Director to create a root Director at the same hierarchy level, either below or above the selected Director

To Move a Director

- Drag one or more Directors to a different place in the Stage Tree hierarchy

To Copy a Director

- Press <Ctrl> and drag one or more Directors to a different place in the Stage Tree hierarchy

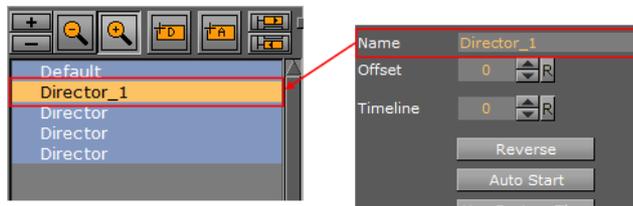
To Rename a Director

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

1. Right-click a Director.
2. In the context menu, select **Rename**.
3. Type a new name.
4. Press <Enter>.

Or:

1. Select the required Director in the Stage Tree.
2. Open the Director Editor (see [To Enable the Director Editor](#)).
3. Type a descriptive name in the **Name** text box.
4. Press <Enter>.



To Delete a Director

1. Right-click a Director in the Stage Tree.
2. In the context menu, select **Delete [Director]**.

Or:

- Drag the Director, or multiple Directors, to the trash can

Or:

- Select a Director, or multiple Directors, and press <Delete>

10.2 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 shown in the [Dopesheet Editor](#) with a gray-blue color.

This section contains information on the following procedures:

- [To Mark a Region Within a Director](#)
- [To Move a Director in the Timeline](#)
- [To Speed Up or Slow Down a Director](#)

See Also

- [Working with Directors in the Stage Tree](#)
- [Director Control Panel](#)
- [Director Editor](#)

To Mark a Region Within a Director

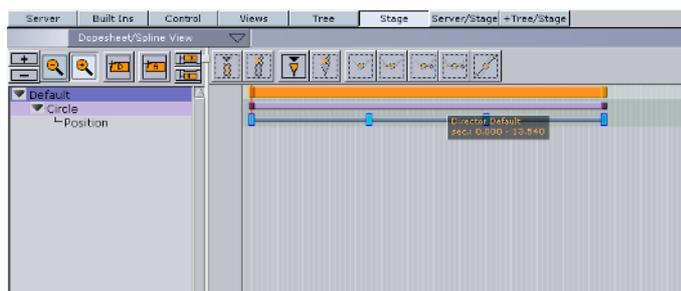
1. In the [Dopesheet Editor](#), press <T+Click> to create a start point in a Director.
2. Press <T+Click> again to create an end point in the Director.



Drag the green start and end points to lengthen or shorten the selected region. In the Director only the Key Frames inside this region are affected.

To Move a Director in the Timeline

- Click on the Director and drag to a different time.



To Speed Up or Slow Down a Director

- Click on the start or the end of the Director.
- Drag the start or end point to change the Director speed:
 - **Speed up:** Lengthen the Director
 - **Slow Down:** Shorten the Director.

10.3 Working with Events in a Director

Events can be added to a Director to add control to an animation. Events that can be added to the Director are:

- **Stop:** Stops the animation until continue animation is clicked. A Stop applies for the Director and all Sub-Directors
- **Tag:** Shows a position along the timeline. This Event type does not stop the animation
- **Local Stop:** Stops the animation until continue animation is clicked. A local stop only applies for the Director it is placed on, not its Sub-Directors.
- **Pause:** Stops the animation for a specified amount of frames.
- **Next Scene:** Mark a position where a Scene ends. The next Scene will start automatically.

Note: Viz Weather has an automatic play list option where the user does not need to start the next Scene. The Next Scene tag starts the next Scene automatically when the current Scene reaches the Next Scene Tag.

This section contains information on the following procedures:

- [To Create a New Event](#)
- [To Change an Event Type](#)
- [To Move an Event along the Timeline](#)
- [To Name an Event](#)
- [To Delete an Event](#)
- [To Create a Tag \(Preview Point\) for External Applications](#)
- [To Create a Tag \(preview point\) for Viz Trio](#)

To Create a New Event

1. Click on a Director in the [Stage Tree](#).
2. Set the [Timeline Marker](#) to the point where the new Event is to be added.
3. Add an Event:
 - Click the  (Add Stop/Tag) button, or
 - Press <S> and click in a Director where the new Event is to be added.

Note: The default new Event is a **Stop** Event.

4. If required, change the Event type (see [To Change an Event Type](#)).

To Change an Event Type

1. Enable the [Event Editor](#) in the Stage.

- In the Director, select the Event to be change.



- In the [Event Editor](#), click an Event type button in the **Type** area to change status.
- If **Pause** is selected: In the Value box, define the number of frames that the animation should pause for, before it continues.

To Move an Event along the Timeline

- Select one or more Events.
- Drag the selected Events along the Director Timeline.

Note: To select more than one Event press and hold <Ctrl> and click on each required Event.

or

- Enable the [Event Editor](#) in the Stage.
- Select an Event in the Director.
- Move the selected Event:
 - In the [Event Editor](#), enter the new position (in frames) in the **Time** input box, or
 - Click in the **Time** input box and drag left or right.

Note: This moves only the last selected Event in the Director.

To Name an Event

- Enable the [Event Editor](#) in the Stage.
- Select an Event in the Director Timeline.
- In the [Event Editor](#), enter a descriptive name in the **Name** text box.
- Press <Enter>.

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

To Delete an Event

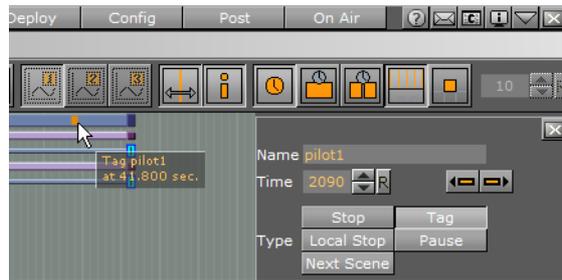
- Select the Event to be deleted.
- Click the  (Delete Stop/Tag) button.

or

- Select one or more Events.
- Press <Delete>.

To Create a Tag (Preview Point) for External Applications

(for example Viz Trio and Viz Pilot)



1. Open the Stage
2. Select the default Director.
3. Position the Time Marker at the point where the Scene is to be previewed.
4. Click the  (Add Stop/Tag) button to add an Event.
5. In the [Event Editor](#) enter **pilot1** (case sensitive) in the Name text box.

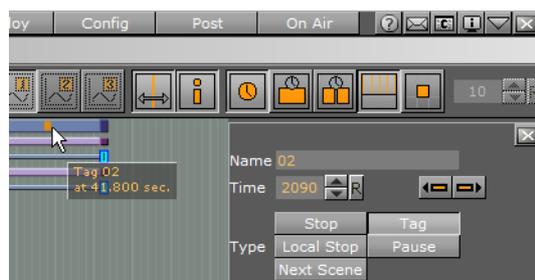
Note: The **pilot1** Event is the default preview point for a foreground Scene in a transition logic Scene.

Note: The name **pilot1** is the Tag name requested by the control applications. For more Tags (preview points) continue with pilot2, pilot3, etc.

6. Click on **Tag** in the Type area.

Note: Custom Tag Events can be used as well; however, those Tags Events are only selectable within the Viz Pilot Newsroom client.

To Create a Tag (preview point) for Viz Trio



1. Open the Stage, and select the default Director.
2. Position the Time Marker at the point where the Scene is to be previewed.
3. Click the  (Add Stop/Tag) button to add an Event.
4. In the [Event Editor](#) enter **the Name** as **01, 02, 03**, etc. according to the value of the corresponding Field identifiers (01, 02, 03, etc.).
 - Setting these points will, when a tab field in Viz Trio is selected (01, 02, 03, etc.), preview the Scene in the local Viz Trio preview window.

See Also

- [Tables and Preview Points](#)

10.4 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 the Container is renamed, the Actor will reflect the change.

If an Actor or Actor channel is clicked on it will also show the relevant Container (in the Scene tree) and object (in the Scene Editor).

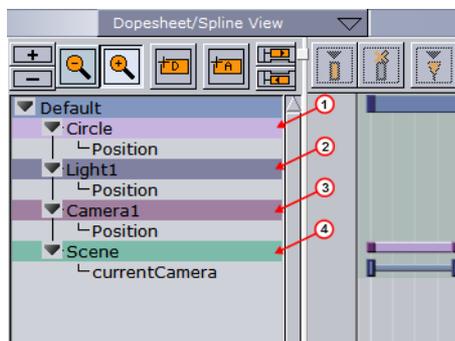
Note: If multiple Actors or Actor Channels are selected this is also shown in the Scene Tree and Scene Editor.

When adding an animation to a Container, the corresponding Actor and channel will automatically show 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.

Each Actor type is represented by a different color:

- **Container Actors:** Light purple (1)
- **Light Actors:** Dark gray (2)
- **Camera Actors:** Dark purple (3)
- **Scene Actors:** Green (4)



This section contains information on the following procedures:

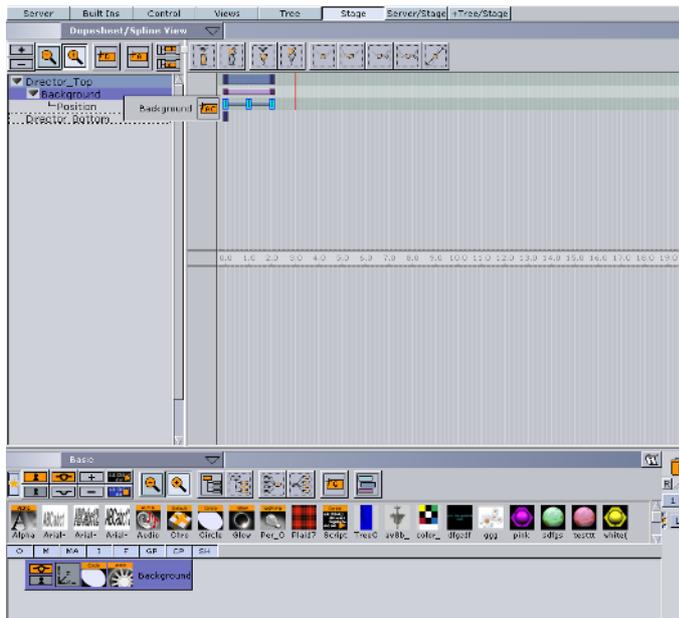
- [To Move an Actor](#)
- [To Copy an Actor](#)
- [To Delete an Actor](#)

See Also

- [Working with Actors in the Dopesheet Editor](#)
- [Working with Channels in the Stage Tree](#)
- [Actor Editor](#)

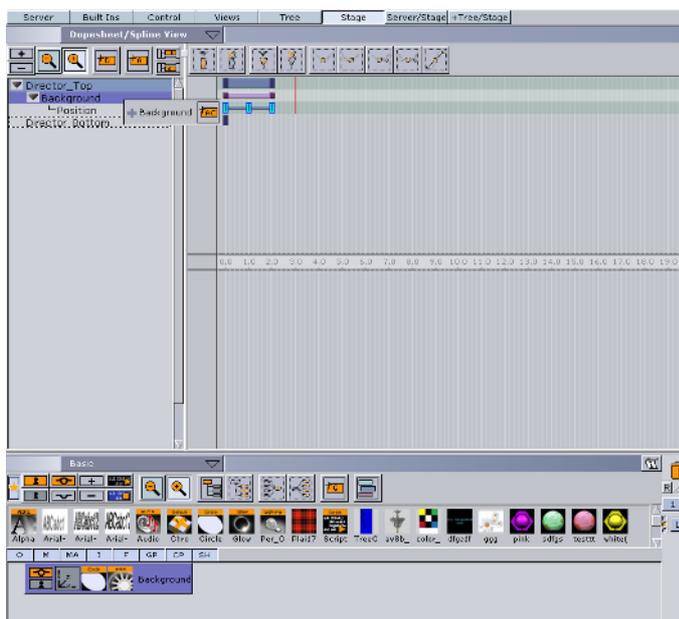
To Move an Actor

- 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 shows, select **Delete [Actor]**.

or

- Drag the actor to the trash can.

10.5 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:

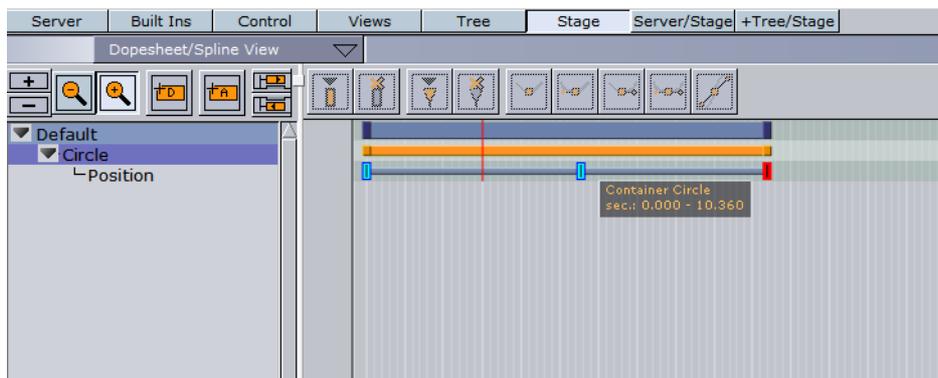
- [To Move an Actor in the Timeline](#)
- [To Speed Up or Slow Down an Actor](#)
- [To Delay all the Animations in an Actor](#)

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 Up or Slow Down an Actor

- Click on the start or the end of the Actor.
- Drag the start or end point to change the Actor speed:
 - **Speed up:** Lengthen the Actor
 - **Slow Down:** Shorten 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).

Also, 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.

10.6 Working with Channels in the Stage Tree

A Channel stores the animated properties of an Actor.

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.

Key Frames are placed along the Channel. Key Frames hold information about the value of the property at the time they are placed on the timeline. If Key Frames 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 Key Frames.

This section contains information on the following procedures:

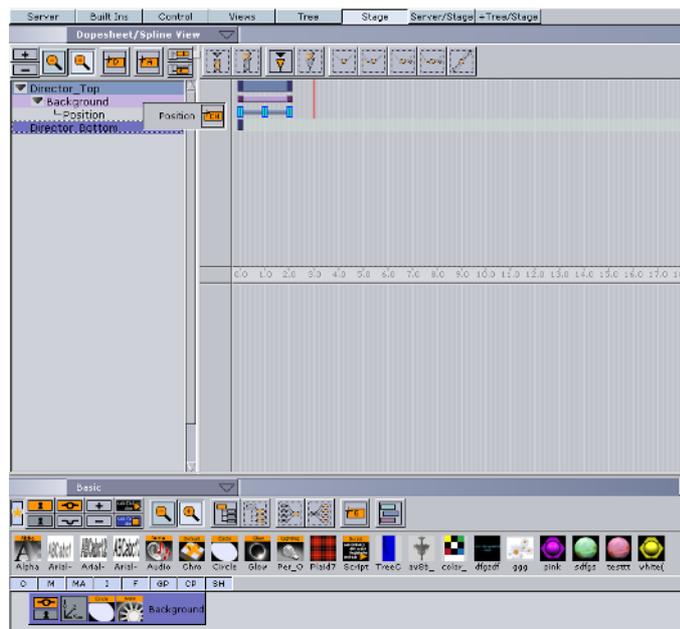
- [To Move a Channel](#)
- [To Copy a Channel](#)
- [To Select a Channel](#)
- [To Delete a Channel](#)

See Also

- [Working with Channels in the Dopesheet Editor](#)
- [Working with Actors in the Stage Tree](#)
- [Working with Key Frames](#)
- [Channel Editor](#)

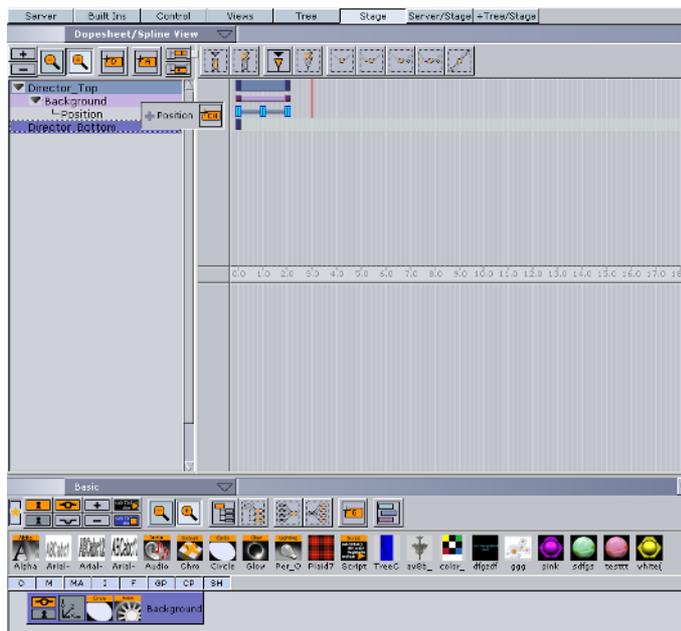
To Move a Channel

- Drag one or more channels from a Director in the Stage Tree to the target Director.



To Copy 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](#), or
- 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 shows, select **Delete [Channel]**.

or:

- Drag the channel to the trash can.

See Also

- [Animating Channels](#)
- [Dopesheet Editor](#)
- [Spline Editor](#)

10.7 Working with Channels in the Dopesheet Editor

Most actions on Channels are done when [Working with Channels in the Stage Tree](#). Actions which affect 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:

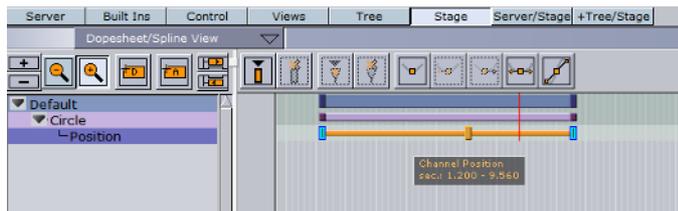
- [To Move a Channel in the Timeline](#)
- [To Speed Up or Slow Down a Channel](#)

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 Up or Slow Down a Channel**

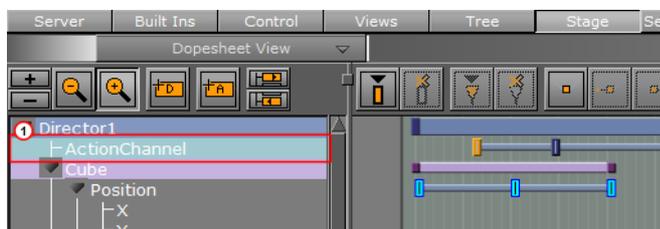
- Click on the start or the end Key Frame of the Channel.
- Drag the start or end Key Frame to change the Channel speed:
 - **Speed up:** Lengthen the Channel
 - **Slow Down:** Shorten the Channel.

10.8 Working with Action Channels in the Stage Tree

Actions Channels are control elements which do 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. Action Channels are colored cyan in the Stage Tree (1).



This section contains information on the these 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 Properties](#)
- [Action Key Frame Editor Properties](#)

To Create an Action Channel

1. Select a Director in the Stage tree.
2. Click  (Add New Action) in the [Stage Tree Bar](#).
3. In the dialog box that opens, type a new descriptive name for the action channel.
4. Press <Enter> or click the <OK> button.

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

or:

1. Drag the  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.
3. 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 shows, select **Rename**.
3. In the text box that shows, 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 shows, select **Delete [Action Channel]**.

or:

- Drag the action channel to the trash can.

10.9 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 done 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:

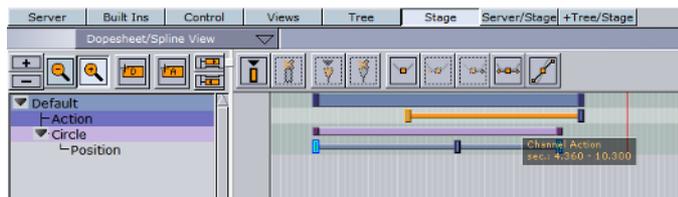
- [To Move an Action Channel in the Timeline](#)
- [To Speed Up or Slow Down an Action Channel](#)

See Also

- [Working with Action Channels in the Stage Tree](#)

To Move an Action Channel in the Timeline

- Drag the action to a different time.



To Speed Up or Slow Down an Action Channel

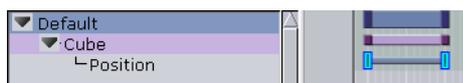
- Click on the start or the end Key Frame of the Section Channel.
- Drag the start or end Key Frame to change the Action Channel speed:
 - **Speed up:** Lengthen the Action Channel
 - **Slow Down:** Shorten the Action Channel.

10.10 Working with Key Frames

A Key Frame stores information about a property at a given time. Key Frames can be added to channels and action channels. If two Key Frames next to each other hold different values, the Key Frames are connected with a bar in the [Dopesheet Editor](#).

When selecting a Key Frame in the Dopesheet editor or the [Spline Editor](#), the corresponding Key Frame in the other editor will be selected too. Working with Key Frames in the Dopesheet enables the [Key Frame Editor](#).

Example of cube with animated position; two Key Frames 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 Key Frames will be created, one at field 0 and one at field 50. Since the position of the cube is different in the two Key Frames, a bar is

added between them, indicating that an animation takes place when the timeline moves from Key Frame A to Key Frame B.

This section contains information on the following procedures:

- [To Add a Key Frame](#)
- [To Select One or More Key Frames](#)
- [To Move One or More Key Frames](#)
- [To Move a Key Frame to a Specific Time](#)
- [To Copy a Key Frame](#)
- [To Rename a Key Frame](#)
- [To Delete a Key Frame](#)
- [To Reverse the Order of Key Frames](#)

See Also

- [Key Frame Editor](#)
- [Stage Editor Bar](#)

To Add a Key Frame

There are four methods detailed to create a Key Frame:

1. Select the channel or action channel in the Stage Tree where the Key Frame should be added.
2. Set the [Timeline Marker](#) to when the Key Frame should be added.
3. Click the  (Add Key Frame) button.

Or:

1. Select the channel or action channel in the Stage Tree where the Key Frame should be added.
2. Set the [Timeline Marker](#) to when the Key Frame should be added .
3. Set the Key Frame:
 - Click the **Set Key Frame** button from the [Timeline Editor](#), or
 - With the mouse pointer over the Scene Editor, press <Enter>.

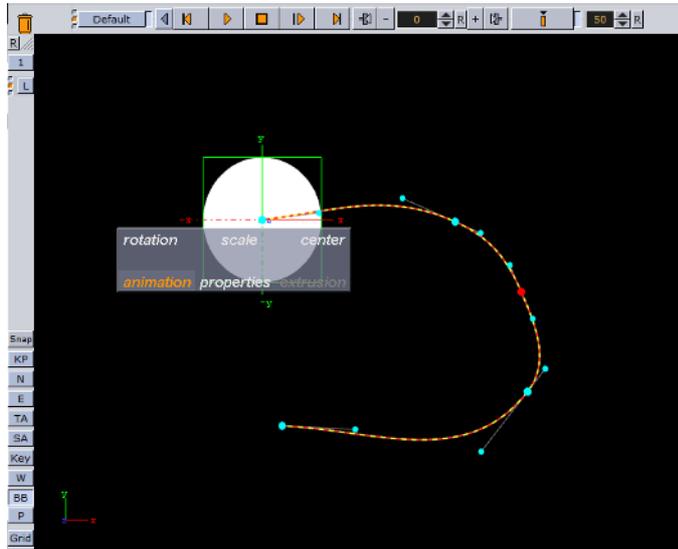
Or:

1. Select the channel or action channel in the Stage Tree where the Key Frame should be added.
2. In the [Spline Editor](#), make sure to hold the pointer above the spline at the required timeline location (the spline should turn orange if doing this correctly).
3. Click the spline.

Or:

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 Key Frame is added.

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 Select One or More Key Frames

- Click on a Key Frame (KF) to select it, this will deselect any previous selected Key Frames.
- Ctrl-Click on a previously selected Key Frame to deselect it.
- Ctrl-Click on Key Frames to have more than one selected.
- Shift-Click on a segment between two Key Frames to select both.
- Ctrl-Click on a segment to select all Key Frames.

Note: When several Key Frames are selected, actions like move or delete will affect all selected Key Frames.

To Move One or More Key Frames

1. Select one or more Key Frames.

Tip: You can also hold down the <S> key and use the rubber band to select multiple Key Frames.

2. Drag the Key Frames along the timeline in the [Dopesheet Editor](#).

Or:

1. Select one or more Key Frames.
2. Drag the Key Frames along the timeline (horizontally) in the [Spline Editor](#).

Note: Dragging a Key Frame vertically in the [Spline Editor](#) will change the value of the Key Frame, not the timeline value.

To Move a Key Frame to a Specific Time

1. Enable the [Key Frame Editor](#).
2. Select a Key Frame.
3. In the Key Frame editor, enter the time (in fields) in the **Time** input box.

To Copy a Key Frame

1. Select one or more Key Frames.
2. Press <C> while dragging the Key Frames to the new position in the [Dopesheet Editor](#).

To Rename a Key Frame

1. Enable the [Key Frame Editor](#).
2. Select a Key Frame.
3. In the Key Frame editor, enter a descriptive Key Frame name in the **Name** text box, and press <Enter>.

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

To Delete a Key Frame

1. Select one or more Key Frames.
2. Click the  (Delete Key Frame) button.

Or:

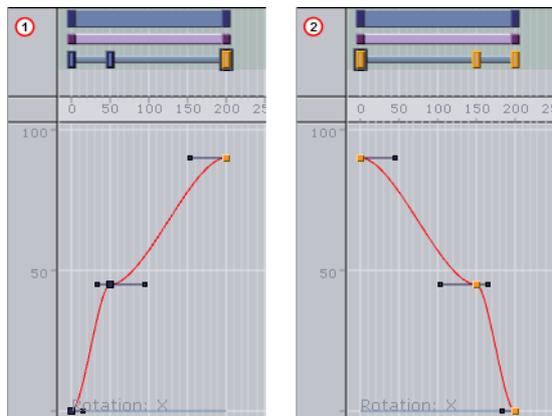
1. Select one or more Key Frames.
2. Press <Delete>.

10.10.1 Reverse the Order of Selected Key Frames

Reversing the order of Key Frames is like mirroring the Key Frames.

Example: Image 1 (below) illustrates a rotation animation, with Key Frames 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 Key Frames have been rotated, image 2

illustrates the animation with Key Frames 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.



To Reverse the Order of Key Frames

1. Select two or more Key Frames in the [Dopesheet Editor](#).
2. While holding the mouse pointer over the Stage, Press <Ctrl+R>.

10.11 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 the End of an Animation](#)
- [To Delay Animations in a Director with 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 required Director.
2. Click the **Start Animation** button in the [Timeline Editor](#).

Or:

1. Enable the [Director Editor](#).
2. Select the required Director.
3. In the Director editor, click the **Start Animation** button.

Or:

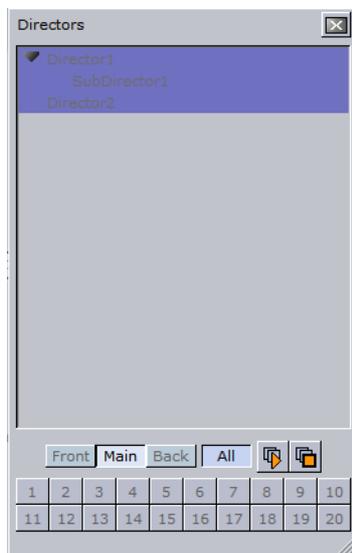
1. Make sure the mouse pointer is in the Dopesheet.
2. Press <Enter>.

Or:

1. Select the required Director.
2. Make sure the mouse pointer is in the Dopesheet and press <Space>.

To Play all Animations in a Scene

1. Enable the [Director Control Panel](#).
2. Click **All**.



3. Do *one* of the following:
 - Make sure the mouse pointer is in the Dopesheet and press <Enter>.
 - Make sure the mouse pointer is in the Dopesheet and press <Space>.
 - 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:
 - Make sure 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).

Or:

1. Select the required Director.

2. Make sure 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.

Or:

1. Enable the [Director Editor](#).
2. In the Director editor, click the **Stop** button.

Or:

1. Select the required Director.
2. Make sure the mouse pointer is in the Dopesheet and press SPACE.

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 Events in a Director](#).

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 required Director.
2. Click the **Continue Animation** button in the [Timeline Editor](#).

Or:

1. Select the required Director.
2. Make sure the mouse pointer is in the Dopesheet and press SPACE.

Or:

1. Enable the [Director Editor](#).
2. Select the required Director.
3. In the Director editor, click the **Continue Animation** button.

To Jump to Start of an Animation

1. Select the required Director.
2. Click the **Go to Start** button in the [Timeline Editor](#).

Or:

1. Select the required Director.
2. Press <Ctrl+Home>.

To Jump to the End of an Animation

1. Select the required Director.
2. Click the **Go to End** button in the [Timeline Editor](#).

Or:

1. Select the required Director.
2. Press <Ctrl+End>.

To Delay Animations in a Director with Offsets

1. Enable the [Director Editor](#).
2. Select the required Director.
3. In the Director editor, set the **Offset** value (in fields).

Also, 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 start or the end point of a Director in the [Dopesheet Editor](#).

Note: If the start of the Director is dragged, 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 Sub-Directors.

To Scale a Region Within a Director

1. Do the procedure [To Mark a Region Within a Director](#).
2. Drag the green start or end point of the marked region.



Scaling is done proportionally and includes all Sub-Directors.

10.12 Creating a Basic Animation

This section shows how to create a basic animation in a Scene.

Animation Task

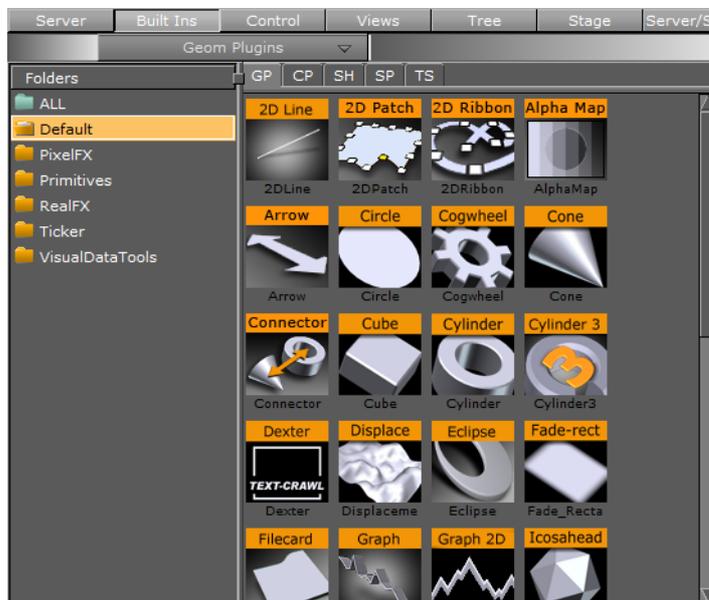
Create a Scene with single a Cube Geometry. Animate the Cube to move right to left.

This use case contains the following procedures:

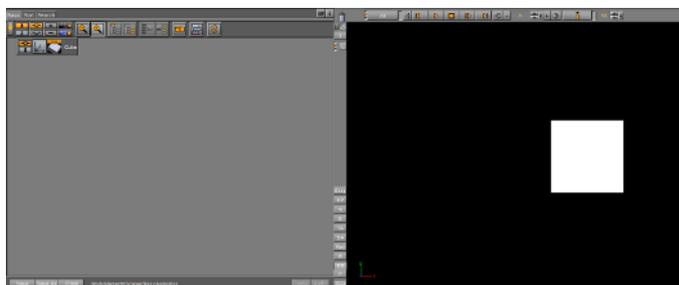
- [To Create the Basic Animation Scene](#)
- [To Create a Basic Animation](#)
- [To Create Key Frames with the Transformation Editor](#)

To Create the Basic Animation Scene

1. Do the procedure [To Create a New Scene](#). Name the scene Basic Animation.
2. Open the Basic Animation Scene.
3. From the [Main Menu](#), click [Built Ins](#)
4. Click the **GP** (Geometry Plugin) tab to show all available Geometry plugins.
5. In the Folders panel, click the **Default** folder.



6. Drag the **Cube** icon from the plugin list into the [Scene Tree](#).
7. Click **Save**.
8. The Scene Tree and Scene Editor should then look like this:



To Create a Basic Animation

1. Open the Basic Animation Scene.
2. Set the timeline, in the Scene Editor (1), to 0 (fields).

3. Create the first Key Frame:



1. Click on the Cube in the Scene Editor.
 2. Click the **Set Key Frame** button (2) or press <Enter>. This sets the first Key Frame.
4. Create the second Key Frame:
 1. In the Scene Editor, drag the Cube to the right, so that it shifts to a different position.
 2. Click the **Set Key Frame** button or press <Enter>. This sets the second Key Frame.

Tip: Use the Transformation editor to move the Cube to precise position (see [To Create Key Frames with the Transformation Editor](#)).

Note: On the Cube Container, in the scene tree, an Anim icon is automatically added, which shows that one or more properties on this Container has been animated.



5. Repeat from step 1 to create more movement, if required.
6. Click the **Start Animation** button (3) to see the created animation.

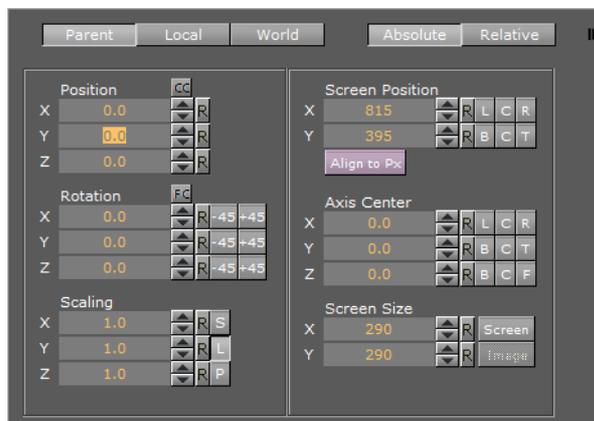
See Also

- [To Manually Move the Timeline Marker](#)
- [To Add a Key Frame](#)

To Create Key Frames with the Transformation Editor

Use the Transformation editor to move the Cube to precise locations then set a Key Frame.

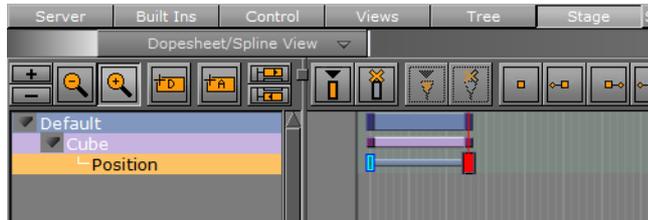
1. After the first Key Frame is set, click the  (Transformation) icon on the Cube container in the scene tree.



2. In the [Transformation Editor](#), set **Position X** to 100.

3. Click the **Set Key Frame** button or press <Enter>. This sets the second Key Frame.
4. Move the Cube as required with the **Position** parameters to a new position. In the new position click the **Set Key Frame** button, or press <Enter>, to set a Key Frame.

A representation of the animation can be seen in the [Stage Tree Area](#).



See Also

- [To Save a Scene](#)

10.13 Create 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.

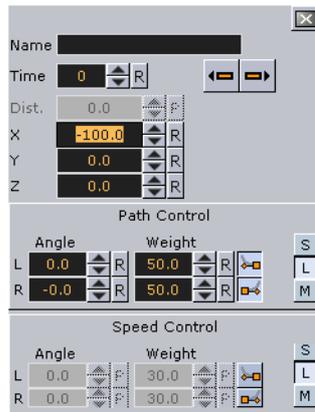
This section contains the following procedures:

- [To Edit the First Key Frame](#)
- [To Edit the Second Key Frame](#)
- [To Create an Additional Key Frame](#)
- [To Add a Stop Point](#)
- [To Create Key Frames for Rotation](#)
- [To Modify the Rotation Settings](#)
- [To Run the Animations Independently](#)
- [To Add and Animate a Material](#)

To Edit the First Key Frame

1. Enable the [Key Frame Editor](#) .

2. Select the Key Frame at the far left in the [Dopesheet Editor](#).



3. In the [Key Frame Editor](#), set the **Position** to X = **-100**.

To Edit the Second Key Frame

1. Select the Key Frame at the right in the [Dopesheet Editor](#).
2. In the [Key Frame 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 Scene Editor to the center.

3. Click the **Play** button in the [Timeline Editor](#).

To Create an Additional Key Frame

1. In the [Timeline Editor](#), set the timeline field to **100**.
2. In the Scene Editor, drag the cube to the right, so that it shifts to a different position.
3. Set an additional Key Frame (see [To Add a Key Frame](#)).
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 Scene Editor, through the center, and all the way to the right part.

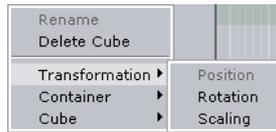
To Add a Stop Point

1. Select the required Director.
2. In the [Timeline Editor](#), set the timeline field to **50**.
3. Create a stop by performing the procedure [To Create a New Event](#).
4. Click the **Play** button in the [Timeline Editor](#).

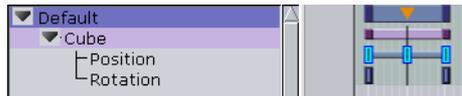
When playing the animation now, the cube will slide rightward until it reaches the center of the Scene Editor, and then come to a halt. To continue playing the rest of the animated journey, see [To Continue an Animation](#).

To Create Key Frames for Rotation

1. Right-click the **Cube** actor in the Stage Tree.



2. In the [Stage Tree Menu](#) that shows, click *Transformation -> Rotation*. A rotation channel and two Key Frames are added.



To Modify the Rotation Settings

1. Select the Key Frame at the right in the [Dopesheet Editor](#).
2. In the [Key Frame 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 Scene Editor, 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.

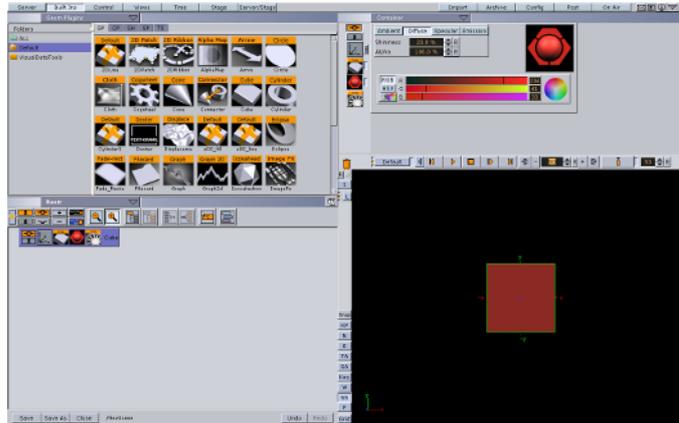
See Also

- [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. Create a Material and define its color (see [To Create a New Material](#)).
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 shown.



4. In the [Timeline Editor](#), set the time when you want the Key Frame to be created.
5. In the [Timeline Editor](#), click the **Set Key** button.
6. Change the material's color (see [Working with Material Elements](#)).
7. To create the next Key Frame, 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.

10.14 Further Advanced Animation Topics

This section details the creation of a Scene with the more advanced animation features in Viz Artist.

This section contains the following topics:

- [Animating Channels](#)
- [Path Control](#)
- [Spline Control](#)
- [Animate Audio](#)
- [Animate Cameras](#)
- [Animate Light](#)
- [Merging Animations](#)
- [Multi Select Animation](#)

As the basis, take the Scene created in [Creating a Basic Animation](#) and improved on in [Create an Advanced Animation](#).

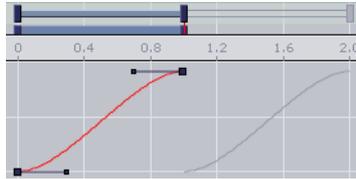
10.14.1 Animating Channels

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)

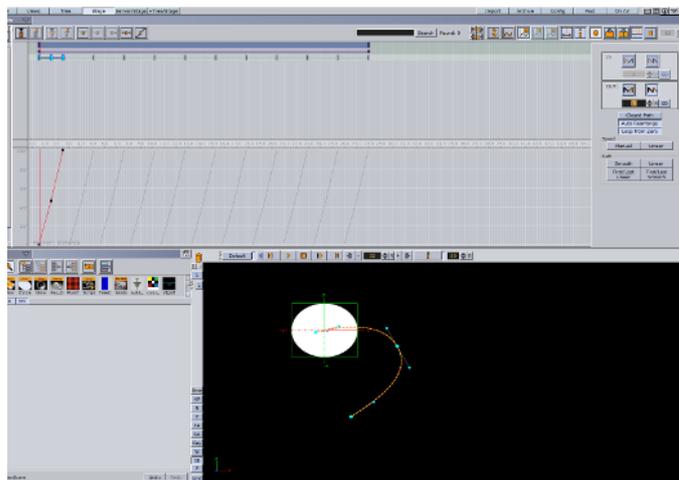
The loop operation makes it possible to repeat the animation of a channel either at the beginning or end of the channel timeline.



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.

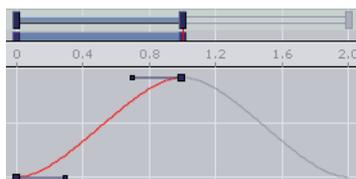


3. Define how many times the loop should run before coming to a halt. Either enter a number in the field or click the **Infinite** button.

When enabling loop, the looped animation shows as gray in the in the [Dopesheet Editor](#)

Repeating Animation Channels in Reverse (Swing)

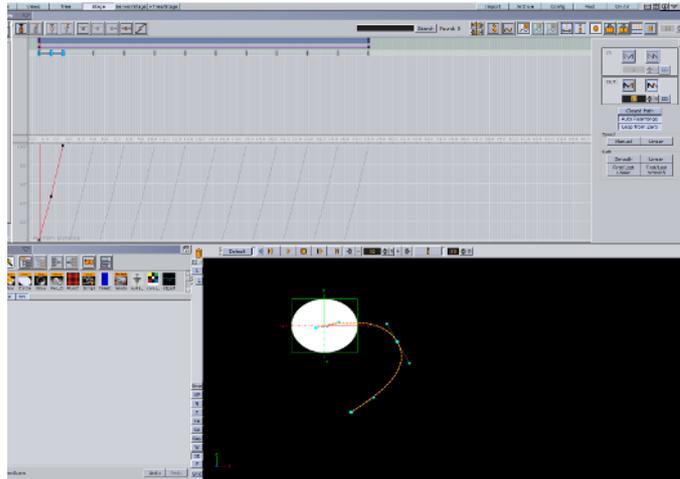
The swing operation makes it possible to repeat the animation of a channel in reverse play either at the beginning or end of the channel timeline.



To Create a Swinging Animation

1. Select a channel

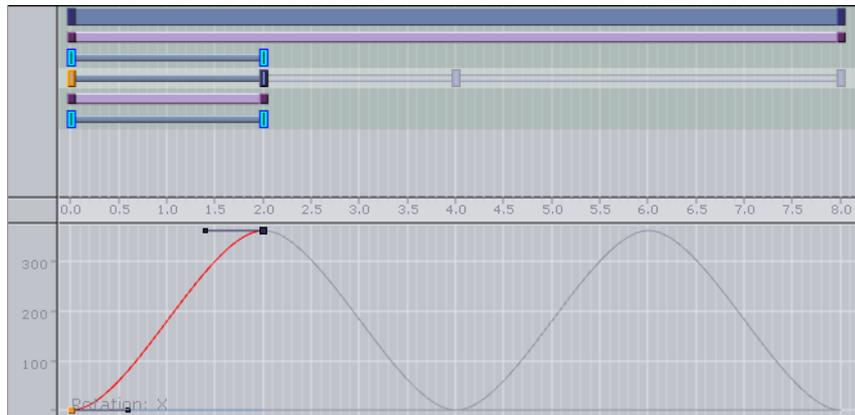
2. In the [Channel Editor](#), enable the **Swing In/Out** buttons.



When enabling swing, the animation shows as gray 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.



To Combine Swing and Loop

1. Do the procedure [To Repeat the Animation of a Channel](#).
2. Do the procedure [To Create a Swinging Animation](#).

Animations with swings and loops show as gray in the [Dopesheet Editor](#).

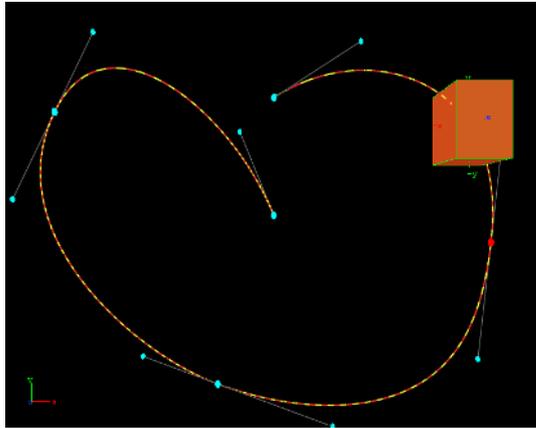
10.14.2 Path Control

The position animation follows a path you define by the Key Frames. A spline will be interpolated from the Key Frames. You can view and edit this spline in the Scene Editor.

To Edit the Spline in the Scene Editor

1. Enable the bounding box:

- Click the **BB** button to the left of the Scene Editor, or
 - With the mouse pointer over the Scene Editor, press .
2. In the scene tree, select the animated container.
 3. Right-click in the Scene Editor and select **Animation**.
The path will be shown as a spline with the Key Frames marked on it.



4. Drag the Key Frames 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 Key Frame has handles (seen in blue) you can use them to modify the Key Frame's trajectory. To do so select a handle and drag it around.

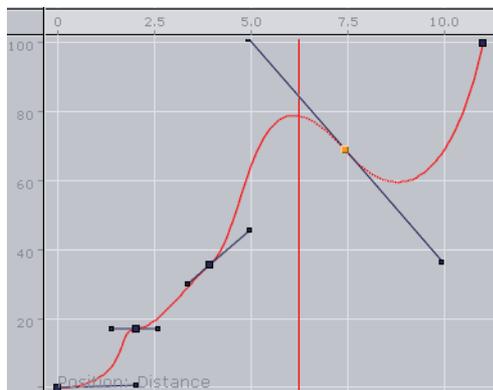
By default the handles are set to be locked and proportional. If you want to switch to a mirrored mode, press <Shift> while you drag the handle around. Pressing <Ctrl> will allow you to edit the single handle only.

An other option to set the handles is the object editor. Select the Key Frame you want to change the handles in and set either **Single**, **Locked** or **Mirrored** and change the values in **Weight L** and **R**.

10.14.3 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 Key Frame along the timeline, but not its percentage of the total trajectory.



By default the handles at every Key Frame on the spline are set to be linear.

This section contains the following procedures:

- [To Edit the Spline](#)
- [To Move Key Frames in the Spline](#)
- [To Insert Key Frames from the Spline](#)
- [To Delete a Key Frame from the Spline](#)

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 Key Frame 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 Key Frame as long as you do not reset it by a click the Key Frame 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 Key Frames in the Spline

- Select the required Key Frame and drag it.
Multiple Key Frames can be selected to move at once.

To Insert Key Frames from the Spline

1. Moving the cursor on the spline exactly where you would like to add the Key Frame.
2. Click on the required location to set the Key Frame.

To Delete a Key Frame from the Spline

1. Select the required Key Frame
2. Press <Delete> or click Delete Key Frame button in the [Stage Editor Bar](#).

Multiple Key Frames can be selected to be deleted at once.

10.14.4 Animate Audio

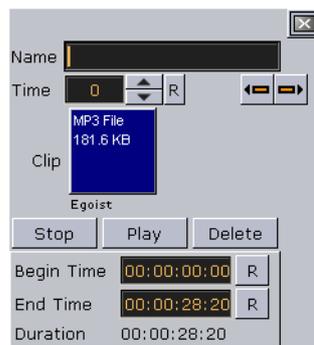
An audio animation plays an audio clip in an animation.

To Enable Audio in an Animation

1. Apply the audio container function to a container in the Scene.
2. Add an actor for this container to the Stage Tree, if it is not present yet.

Tip: The easy way to do is to drag 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 Key Frame 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 Key Frames of the AudioClip channel, configure the following parameters, as necessary.



- **Name:** Sets a name for the Key Frame. This name is for use only for external commands or scripts.
- **Time:** Shows the time value in fields the Key Frame is placed. Editing this value will change to position of the Key Frame.
- **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 Key Frame 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 Key Frame at the end of the clip.
- **Duration:** Shows the duration to be played.

10.14.5 Animate Cameras

You can animate cameras by adding an actor for a camera to the Stage Tree is by setting Key Frames in the Scene Editor.

This section contains the following procedures:

- [To Animate a Camera](#)
- [To Animate the Current Camera](#)
- [To Create a Zoom Animation](#)

To Animate a Camera

1. Set the [Timeline Marker](#) to when the Key Frame should be added to start with the animation of the camera.
2. Click the **Add Key Frame** button in the [Stage Editor Bar](#).
3. Change one of the selected camera properties and set the next Key Frame. This will add an actor in the Stage Tree for the camera whose properties you have changed.

Another way to create camera animations is to enable position and/or direction tracking in the camera editor to animate the camera through a container transformation animation.

See Also

- [Camera Editor](#)
- [Working with Actors in the Dopesheet Editor](#)
- [Working with Key Frames](#)

To Animate the Current Camera

1. Set a Key Frame in the current Camera channel.
2. In the object editor edit the following parameters for the Key Frame as necessary.



- **Name** Set a name for the Key Frame. This name is for use only for external commands or scripts.
- **Time** Shows the time value in fields the Key Frame is placed. Editing this value will change to position of the Key Frame. If you click arrow buttons in the [Stage Editor Bar](#) the previous or next Key Frame 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 shown 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 the 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.

10.14.6 Animate Light

The only way to add an actor for a light to the Stage Tree is by setting Key Frames in the Scene Editor. To do so set the timeline value to the time you want to start with the animation of the light and set the first Key Frame. Now change one of the selected lights properties and set the next Key Frame. 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 [Light Editor](#) .

10.14.7 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 Sub-Director 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 Sub-Directors to the Directors created automatically by merging.

10.14.8 Multi Select Animation

If two or more objects are selected multiple animations can be created in one step. This adds independent animations channels for each container within the selection.

1. Select the objects to be animated.

2. Create the animation, as required (see [Creating a Basic Animation](#)).

An Actor is created for each selected animated object. Once the animation has been created each Actor can be edited separately.

10.15 Create an Over the Shoulder Scene

An Over the Shoulder (OTS) graphic is a graphic which is placed over the shoulder of, for example, the news anchor, and can show on the left or right side of the screen. An OTS graphic can be used to highlight a story with a large image and a headline, and sometimes a sub-headline.

This section contains information on the following procedures, which when completed in sequence create an OTS Scene:

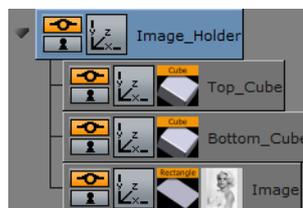
- [To Create a Scene](#)
- [To Add an Image and Basic Geometries](#)
- [To Add Materials](#)
- [To Scale Geometries and Place them in Z Space](#)
- [To Add Text](#)
- [To Add Gradients](#)
- [To Add Background Elements](#)
- [To Add Flares](#)
- [To Animate](#)
- [To Animate Sub Containers](#)
- [To Animate Single Containers](#)
- [To Animate Single Objects](#)
- [To Animate Out](#)
- [To Add Stop Points](#)
- [To Add Controls](#)

To Create a Scene

- Create a new scene and save it as OTS (see [To Create a New Scene](#))

To Add an Image and Basic Geometries

1. Open the OTS scene.



2. Add a [Rectangle](#) geometry to the Scene Tree.
3. Add an image (for example, a headshot) to the Rectangle container.
4. Rename the Rectangle container to Image.
5. Add a [Cube](#) geometry to the scene. Place it above the Image container.

6. Open the Cube Plugin editor and do the following:
 1. Set **Center Y** to Bottom.
 2. Adjust the **Size X** value to fit the width of the image.
 3. Adjust the **Size Y** value to e.g. 10.0.
 4. Adjust the **Size Z** value to e.g. 1.0.
7. Copy the Cube container. Place it below the original Cube container.
 - Press <Ctrl+C> then <Ctrl+V> to place a copy below the original container.
 - Press <Ctrl+Left mouse button> to drag and drop a copy to a new location.
8. In the transformation editor of the Cube container set **Center Y** to Top.
9. Rename the Cube containers to Top_Cube and Bottom_Cube.
10. Open the transformation editor of each Cube container and move the position of the Cubes. Place one above and the other below the image.

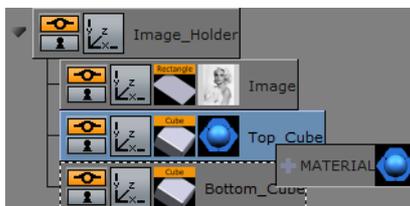


11. Group all containers under a new root container:
 - Select all containers and click the **Group** button, or
 - Select all containers and press <Ctrl+G> (when Caps Lock is disabled).
12. Rename the new group container to Image_Holder.
13. Save the scene.

Note: Geometries can also be scaled with the container's transformation editor. But this will have an effect on the its proportions when, for example, beveled edges are added.

To Add Materials

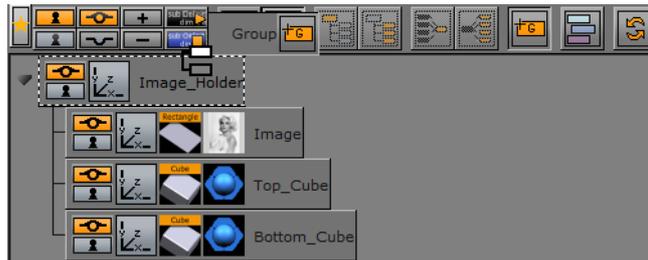
1. Open the OTS scene.
2. Expand the Image_Holder container.
3. Right-click the Top_Cube container and select **Add Material ...** from the context-menu.
4. Open the [Material Editor](#) and define a color (e.g. blue or red).
5. Copy the Material onto the Bottom_Cube container:
 - Drag the Material from one container to the other, or
 - Right-click the Material and from the context menu select Copy 'MATERIAL' and then Paste 'MATERIAL' onto the receiving container.



6. Save the scene.

To Scale Geometries and Place them in Z Space

1. Open the OTS scene.



2. Add a new group container. Place it above the Image_Holder container:
 - Select the Image_Holder container and press <Ctrl+Insert>, or
 - Click the **New Group** button, in the Scene Tree [Tree Button](#), drag it above the Image_Holder container.
3. Rename the New Group container to Body.
4. Add a [Wall](#) geometry to the scene. Place it as a sub-container of the Body container.
5. Rename the new container to Wall.
6. Open the [Wall](#) editor (not the transformation editor) and do the following:
 1. Set the **Center** option to Center.
 2. Adjust the **Height** value to 540 (for example).
 3. Adjust the **Width** value to 75 (for example).



7. Group the Body and Image_Holder containers under a new container:
 - Select the Body and Image_Holder containers and click the **New Group** button, or
 - Select the Body and Image_Holder containers and press <Ctrl+G> (when Caps Lock is disabled).
8. Rename the new group container to Main.
9. Copy the Material from one of the Cube containers to the Wall container.
10. Open the Body container transformation editor.
11. Adjust the **Position Z** value to -30.0 (This will avoid potential conflicts with the geometries in the Image_Holder container).
12. Save the scene.

Note: If the position and/or scaling of the Wall geometry need to be adjusted, make sure to adjust those values on the Body container, as this will keep future additions to the container aligned.

To Add Text

1. Open the OTS scene.
2. Add a New Group container. Place it below the Image_Holder container.
3. Rename the New Group container to Text_Holder.
4. Add a Font to the scene. Place it as a sub-container of the Text_Holder container.
5. Copy the Font container. Place it below the first Font container.

6. Rename the Font containers to headline and sub-headline.
7. Open the [Text Editor](#) of both containers and do the following:
 1. Enter some text (e.g. headline, sub-headline).
 2. Set the **Orientation** option to Center.
8. Open the transformation editor of the headline and sub-headline containers and do the following:
 - Scale and position the containers to fit each other.
9. Add a [Rectangle](#) geometry to the scene. Place it above the Font containers and rename it to Text_Rectangle.
10. Open the transformation editor of the Text_Rectangle container and do the following:
 1. Adjust the **Position Y** value. Center the rectangle between the text objects.
 2. Enable single (S) Scaling.
 3. Adjust the **Scaling X** value to equal the width of the image.
 4. Adjust the **Scaling Y** value to 0.1 (for example).
11. Copy the Material from one of the Cube containers to the Text_Rectangle container.
12. Open the transformation editor of the Text_Holder container and do the following:
 1. Adjust the **Position Y** value to equal the Image_Holder objects.
 2. Adjust the **Scaling** (Locked) if needed.
13. Save the scene.

Note: If the position and/or scaling of the font containers need to be adjusted, make sure to adjust those values on the Text_Holder container, as that will keep future additions to the container aligned.

To Add Gradients

There are several ways to add gradients or ramps. The example below details a very simple way, which uses a gradient image with alpha.

1. Open the OTS scene.
2. Open the Text_Holder container.
3. Add a gradient image to the Text_Rectangle container.
4. Open the transformation editor of the Text_Rectangle container.
5. Adjust the **Scaling** values to make the rectangle equal the intended width of the text.
6. Save the scene.

Note: If the PixelFX [pxGradient](#) plugin is used there are more parameters that can be animated. Gradients from other design tools do not need to be imported.

To Add Background Elements

This procedure details how to add a background image.

1. Open the OTS scene.



2. Add a New Group container. Place it as a sub-container of the Body container.
3. Rename the New Group container to BackgroundImage_Holder.
4. Add an alpha image to the BackgroundImage_Holder container (for example, a map of the world).
5. Copy the Material from the Wall container to the BackgroundImage_Holder container.
6. Open the transformation editor of the Backgroundimage_Holder container and do the following:
 1. Adjust the **Scaling** values as required.
 2. Adjust the **Position Z** value (for example, 2.0) so that it shows in front of the Wall object.
7. Save the scene.

Note: If the position and/or scaling of the background elements as a group need to be adjusted, make sure to adjust those values on the Body container, as that will keep future additions to the container aligned.

To Add Flares

This procedure details how to make the designs look more dynamic by the addition of elements such as flares. A flare is basically an alpha image.

1. Open the OTS scene.



2. Add a New Group container. Place it below the Body container.
3. Rename the New Group container to Flares.
4. Add a New Group container. Place it as a sub-container of the Flares container.
5. Rename the New Group container to Flare_Holder.
6. Add a flare image (with alpha) to the Flare_Holder container.
7. Copy the Material from the Wall container to the Flare_Holder container.
8. Open the transformation editor of the Flare_Holder container.
9. Adjust the **Scale**, **Rotation** and **Position** of the flare so that it makes the intended effect on the object it is applied to.
10. Add the [Expert](#) plugin to the Flare_Holder container.
11. Open the Expert plugin editor.
12. Set **Render Mode** to **Add**.
The source color is now added to the target color.
13. Save the scene.

Note: If more than one flare is required repeat the process, or copy the Flare_Holder container and make the necessary changes (e.g. position and scaling).

To Animate

Before animations are added, move the OTS graphic to its final on-screen position (normally to the top left or right) where it should be seen by the viewer.

1. Open the OTS scene.
2. Open the transformation editor of the Main container.
3. Adjust the **Scaling**, **Rotation**, and **Position** values.
Make sure any conflicts are resolved now. Notice that when the Scene is rotated, objects may “drift” apart as they are placed at different depths (Z space).
4. Save the scene.

This procedure focuses mainly on the animation of a root container, mainly to start with the simplest factor; how do I want my graphics to animate in to view?

1. Open the OTS scene.
2. Set Key Frames to 60/50 (for example) fields.
3. Click the **Set Key Frame** button.
4. Open the transformation editor of the Main container and do the following:
 - Adjust the **Position X** value to 0.0.
5. Set Key Frames to 0 fields.
6. Click the **Set Key Frame** button.

Note: Notice that an Animation object has been added to the Main container.

7. Click the **Start animation** button on the [Timeline Editor](#).
The graphic now moves from Position X at 0 fields to its final on-screen position at 60 fields.
8. Open the [Stage Editor](#).
The Main container and the animated Position property are located under the Default Director.
9. Right-click the Main container and from the context menu select *Transformation - > Scaling*.
This will add a timeline for the **Scaling** property of the Main container.
10. Click the first Key Frame on the Scaling animation's timeline on the [Dopesheet Editor](#).
This will open the Scaling animation [Stage Object Editor](#).
11. Set the **Scaling** values to 0.0.
12. Click the **Start animation** button on the [Timeline Editor](#).
The graphic will now animate from Position X at 0 fields with Scaling at 0.0 to its final size on-screen.
13. Save the scene.

Note: If the Viz Configuration is set to 50 hertz the default animation timeline will be 50 fields. If set to 60 hertz, it will be 60 fields. It is always

important to configure the required output format before work is started on the designs.

To Animate Sub Containers

This procedure focuses on the animation of a sub-containers. In this case the image and text holder containers.

1. Open the OTS scene.
2. Open the [Stage Editor](#).
3. Drag the Image_Holder container onto the Default Director.
4. Right-click the Image_Holder container and select *Transformation -> Scaling* from the context menu.

This will add a timeline for the Image_Holder container's Scaling property.

5. Click the Key Frame for the Scaling animation. In the [Key Frame Editor](#) set it to 30 (for example) fields, to delay the introduction of the image.
6. Copy the animation object from the Image_Holder container to the Text_Holder container.
7. Save the scene.

Note: The Image_Holder and Text_Holder containers are set to grow from the Center. To grow the containers from other angles, open the transformation editor, for each container, and adjust the **Axis Center** properties. If the OTS is placed to the right, try setting the Text_Holder container's Axis Center X to R (Right), and conversely.

Note: Always scrub the timeline in the timeline editor to check for potential conflicts (caused by rotation of objects).

To Animate Single Containers

This procedure focuses on the animation of a single container, and how to add additional Key Frames to the animation. In this case the background image is animated and an additional Key Frame is added to grow the size of the background image before it shrinks back to its final size.

1. Open the OTS scene.
2. Select the BackgroundImage_Holder container.
3. Set the number of fields to 30 in the [Timeline Editor](#).
4. Open the Transformation Editor of the BackgroundImage_Holder container.
5. Set **Scaling** to 0.0.
6. Click the **Set Key Frame** button.
7. Set **Scaling** to the original value.
8. Set the number of fields to 60 in the [Timeline Editor](#).
9. Click the **Set Key Frame** button.
10. Open the [Stage Editor](#).
11. Select the **Scaling animation** timeline for the BackgroundImage_Holder.
12. Move the [Timeline Marker](#) to a position closer to the end of the animation.

13. Click the **Add a Key Frame** button.
14. Click the new Key Frame and set the **Scaling** value to be larger than the final size.
15. Save the scene.

Note: Always scrub the timeline in the timeline editor to check for potential conflicts (caused by rotation of objects).

To Animate Single Objects

This procedure focuses on the animation of single objects. Animating single objects increases the dynamics of the scene, especially when the scene has reached its final position on-screen and all other animations have stopped.

This procedure describes how to make a flare move, continuously, along another object.

1. Open the OTS scene.
2. Open the [Stage Editor](#).



3. Create a new Director and name it Loop:
 - Click the **+D** button, or
 - Right-click the Stage Tree and select **Add Director**.
4. Select the Loop Director.
5. Set the [Timeline Editor](#) to 0 fields.

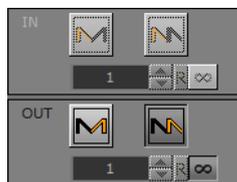


6. Open the [Texture Editor](#) of the Flare image:
 - Set **Position X** to 0 and click the **Set Key Frame** button.
 - Set **Position X** to 10 and click the **Set Key Frame** button.

Note: When **Position X** is set to 10 it makes sure that there are no gaps between the repeating textures.



7. Select both of the **Position X** Key Frames and click the **Linear left and right** button.



8. Click the **Position X** channel to open its [Channel Editor](#). Click the **Loop out** and **Infinite** buttons.

The timeline should now look similar to the image above.



9. Save the scene.

Note: Now that there is an animated Flare it is always good to revisit the **Position**, **Rotation**, and **Scaling** properties of the Flare_Holder container.

To Animate Out

Animations to the scene so far has brought the graphic on-screen for the viewer to see. However, the scene should also be able to animate out instead of simply removing it (hard cut).

This can be achieved in three ways:

- [To Fade out the Graphic with the Alpha Plugin](#), or

- [To Copy the Directors, Flip them Around, then Reverse the Animation](#), or
- [To Create a Separate Out Animation](#)

Note: With one or a combination of the three methods an out animation can be created. The scene will now work in terms animation, however, it will not work well if someone needs to control, for example, when or how the out animation should start. For this we need to add controls (using control plugins) that will allow an operator to control the scene on air.

To Fade out the Graphic with the Alpha Plugin

1. Open the OTS scene.
2. Add the [Alpha](#) plugin to the Main container.
3. Set the timeline to the end of the in animation (at 60 fields for example).
4. Open the Alpha plugin editor.
5. Set the **Alpha** value to 100%.
6. Click the **Set Key Frame** button.
7. Set the **Alpha** value to 0%
8. Click the **Set Key Frame** button.
9. Save the scene.

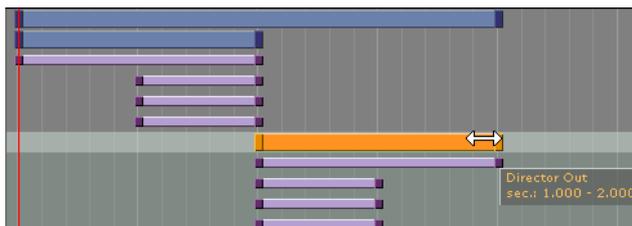
When played the animation should now animate in and fade out to Alpha at 0%.

To Copy the Directors, Flip them Around, then Reverse the Animation

1. Open the OTS scene.
2. Open the [Stage Editor](#).



3. Create a new Director, and name it Default.
4. Place the old Default Director under the new Default Director
5. Rename it to In.
6. Copy the In Director. Place it as a Sub-Director of the Default Director.
7. Rename the copied Director to Out.
8. Select the Out Director.



9. Hold <Ctrl>, select the Out Director's first Key Frame, and drag the Key Frame past the last Key Frame to reverse the animation.
10. Save the scene.

When played the animation should now animate in and out, where the out animation is simply a reverse of the in animation.

To Create a Separate Out Animation

Manually create an 'out' animation from scratch, by the extension of the existing animations to animate out of view.

Also, repeat [To Copy the Directors, Flip them Around, then Reverse the Animation](#) and adjust the different animations as required.

To Add Stop Points

When the scene design is finished Controls can be added that will allow the operator to control the animation while on air. To do this add one or multiple stop points to the Default Director (always the Default Director).

1. Open the OTS scene.
2. Open the [Stage Editor](#).
3. Select the Default Director.
4. Place the [Timeline Marker](#) where the Stop point is to be added (for example, where the In animation stops, and the Out animation starts).
5. Select the Default Director again.
6. Click the **Add a Stop/Tag** button.
This will add a Stop/Tag to the Default Director.
7. Name the first Stop point.
8. Save the scene.

Note: Stop point names can be user defined. However, when the scene is preview with a remote (external) Viz Engine, either from a newsroom system, or when connected through a control application to a Media Sequencer, the pilot1 stop/tag is the default preview point on the Default Director.

Note: The placed pilot1 stop/tag point on the Default Director's timeline, at a given field, enables Vizrt control applications to create a thumbnail and show it in, for instance, the playlist.

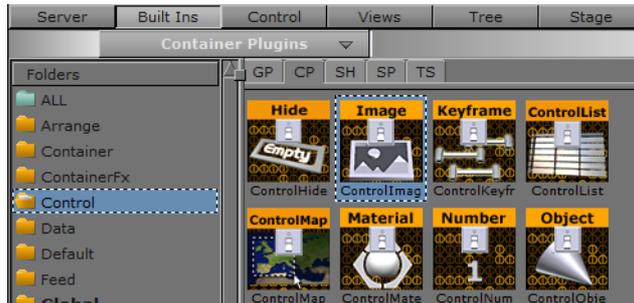
Now the scene is ready for the operator to control the animation. However, the operator still lacks one key element; they need to control the content of the graphics (e.g. text and imagery).

To Add Controls

Stop/tag points are added to the Director to allow the operator to control the animation. However the operator, when required, should be able to control the content of the Scene. For this purpose [Control](#) plugins are used.

1. Open the OTS scene.

- Expand the [Scene Tree](#), or those containers that require any form of control (e.g. text and imagery).



- Click the [Built Ins](#) button on the [Main Menu Left](#).
- Click the Container Plugins (CP) tab and select the Control section.
- Add the [Control Text](#) plugin to the headline and sub-headline containers.
- Add the [Control Image](#) plugin to the Image container.
- Open the headline container [Control Text](#) plugin editor.
- Set **Field Identifier** to 01.
- Repeat the process for the sub-headline and Image containers and set **Field Identifier** to 02 and 03, respectively. Update the **Description** fields as required, by default it will use the container name.
- Save the scene.

Tip: Number the Field Identifiers to help order them in the control application interface.

Note: To check that an operator can use the scene test it with a control application (for example, Viz Pilot with the Template Wizard, Viz Trio, Viz Multichannel, or Viz Weather).

Note: All plugins are installed locally. Plugins used in the design must be available on the production system for the scene to work.

See Also

- [Control Button](#)

10.16 Create a Stand-alone Scene

This section details how to create a basic lower third as a stand-alone Scene. This scene can then be imported to a Viz Trio to create a template for pages which are used for playback.

IMPORTANT! When creating these transition effects also refer to the User Manuals for the required Control Applications: Viz Trio, Viz Pilot, Viz Weather, etc.

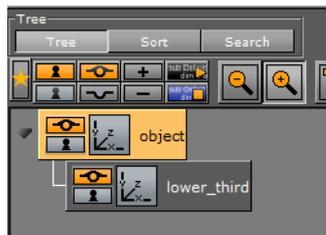
Note: For ease of use, the Scene example has been split up in several procedures. For the example to work, it is recommended to follow each procedure in the order given.

This section contains information on the following topics:

- [To Create a Scene](#)
- [To Add a Background](#)
- [To Add Text](#)
- [To Create an In and Out Animation](#)
- [To Add Stop Tags](#)
- [To Add Key Functions to the Container](#)
- [To Add Expose Properties](#)
- [To Edit Multiple Elements with a Single Value](#)

To Create a Scene

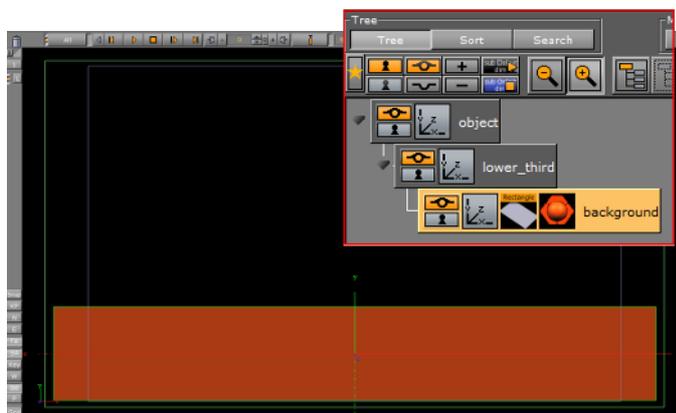
1. Add a group container to the Scene Tree.
2. Rename the group container to **object**.



3. Add second group container as a sub-container to the **object** container.
4. Name the second group container **lower_third**.
5. Save the new scene as **stand_alone_scene** (in this example).

Tip: For ease of use set the Scene Editor Title (TA) and Safe (SA) areas to ON.

To Add a Background



1. Add a Rectangle geometry as a sub-container of the **lower_third** container,

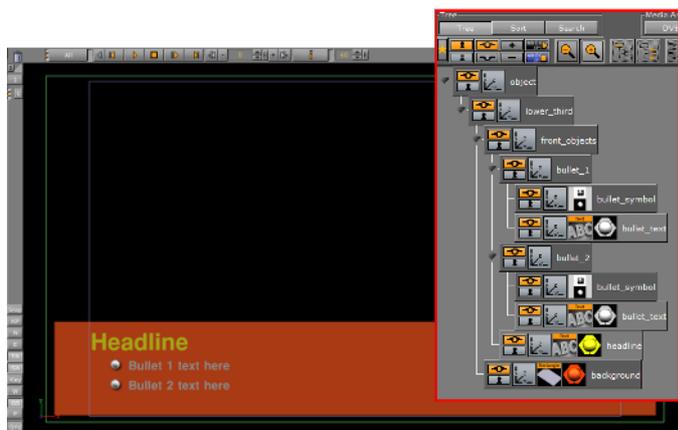
2. Rename it to **background**.
3. Position the rectangle object so that it covers the lower third of the screen:
 - Rectangle editor: **Width 800** and **Height 125**.
 - Transformation editor: **Position Y -160** and **Position Z -100**.

Note: Position Z: Since this element is a background element it should be a little bit behind the other objects on the Z-axis.

4. Add a material (see [To Add Materials](#)) object to give the Rectangle a color.

To Add Text

Add a headline element and two text lines, lead by a bullet symbol.

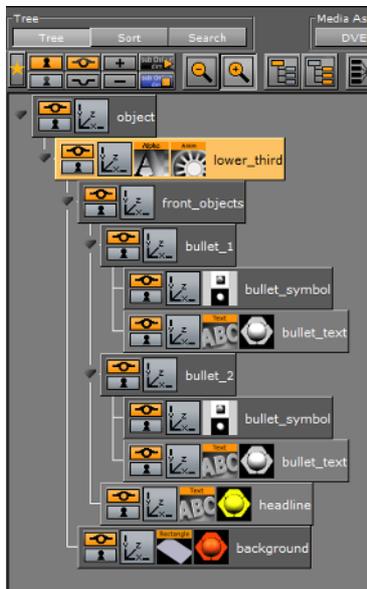


1. Add a container as a sub-container to the **lower_third** container.
2. Name the container **front_objects**.
3. Add a Fontstyle as a sub-container of the **front_objects** container.
4. Name the Fontstyle container **headline**.
5. Open the Fontstyle text editor and:
 - Enter the word **Headline** in the **text input** field
 - Set the Horizontal orientation to **Left**.
6. Add a material to the **headline** container.
7. Open the transformation editor of the **headline** container.
8. **Scale** and **Position** the Fontstyle so that it is placed in the upper left corner of the lower third background.
9. Add a group container as a sub-container of the **front_objects** group.
10. Name the container **bullet_1**.
11. As sub-containers of **bullet_1**, add:
 - A Fontstyle: Name container **bullet_text**
 - Any Image or Object that can function as a bullet symbol: Name the container **bullet_symbol**
12. In **bullet_text** text editor:
 - Enter the text **Bullet 1 text here**
 - Set the horizontal orientation to **Left**.

13. Add a material to the **bullet_text** container.
14. **Scale** and **Position** the **bullet_text** and **bullet_symbol** containers.
15. Position the **bullet_1** container under the **Headline** text.
16. Make a copy of the **bullet_1** container as a sub-container of the **front_objects** group.
17. Rename the copy to **bullet_2**.
18. Adjust the **Position Y** of **bullet_2**.
19. In **bullet_text** text editor enter the text **Bullet 2 text here**

To Create an In and Out Animation

To make the whole lower third rotate in from the left side of the screen and then fade out, the **lower_third** container must be animated.



1. Open the Transformation Editor of the **lower_third** container.



2. Click the **Axis Center X** set **left** button **L** (2).
3. Change the **Rotation Y** value so the container moves out to the left side and is hidden (1).
4. Click **Set Key** (3).



5. Click the **Rotation Y** reset button  (1) to set the **Rotation** value to zero.
6. Click **Set Key**. An animation object should now be visible on the **object** container.



7. Play the animation in the Scene Editor.
8. Add the Alpha plugin to the **lower_third** container.
 - *Built Ins -> Container Plugins -> Global*



9. Open the Alpha editor and:
 1. Set the Alpha value to 100%.
 2. Press the **Set Key** button.
 3. Set the alpha value to 0.0%.
 4. Press the **Set Key** button.

This has created an alpha animation in addition to the rotation animation.

10. Click the [Stage Button](#) to see the animation channels.



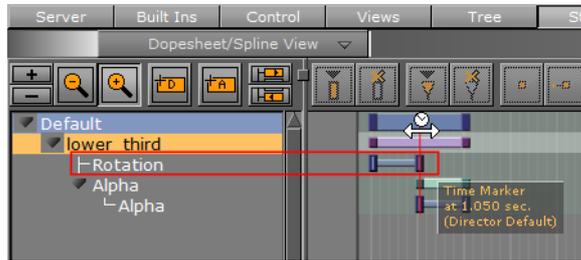
Note: It is possible to move the Key Frames to achieve a different timing. Either drag with the mouse or click on them and alter the time settings in the [Key Frame Editor](#).

To Add Stop Tags

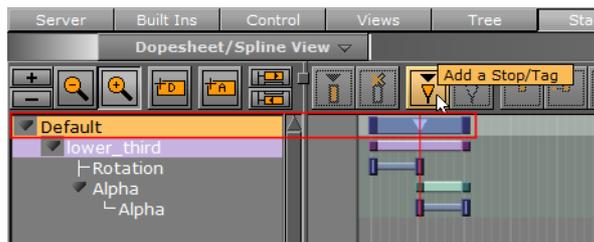
When the Scene is played out, the objects should rotate in from the left and then fade out. To stop the animation, after the rotate in, a stop point must be added before the animation can continue and fade out.

1. Move the timeline (the thin red vertical line) to the second keyframe of the Rotation channel

Note: An exact spot is not needed as it can be adjusted afterwards.



2. Add a Stop keyframe to the **Default** Director.



3. Drag the Stop point (or use the editor that opens when it is selected) so that it has the same time settings as the second keyframe on the Rotation channel.
4. Play the animation to check that the animation stops when the rotate in has finished.
5. Click the **Continue** button to continue with the Alpha fade.



To Add Key Functions to the Container

1. Add the Key plugin to the **background** container.
 - *Built Ins -> Container Plugins -> Global*



2. Click the Key plugin icon to open its editor:
 - Set **Alpha as Key only** to **On**. This is to avoid a dirty key if the background has some level of transparency.
 - Set **Render Mode** to **Blend**.



3. Add a Key plugin to the **front_objects** container.
4. Click the Key plugin icon to open its editor:
 - Set **Alpha as Key only** to **Off**.
 - Set **Render Mode** to **Add**.

This is to avoid the foreground objects to cut a hole in the Key signal of the background object.

To Add Expose Properties

To make a Scene ready for import into Viz Trio, Control plugins must be added. The Control plugins enable the designer to expose scene properties to a Viz Trio user.

1. Add the **Control Text** plugin to the **headline** and both **bullet_text** containers.
 - *Built Ins -> Container Plugins -> Control*

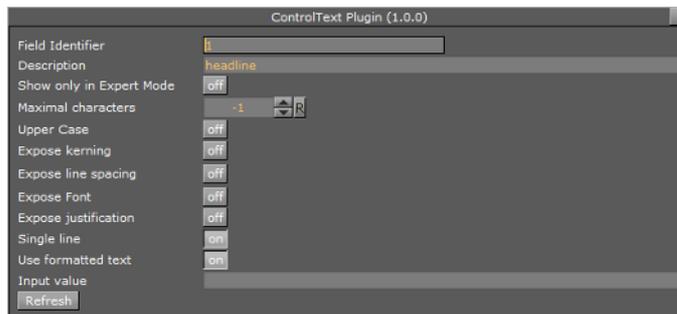


Note: The **Control Object** plugin is automatically added to the top most container in the Scene Tree, which should always be named **object**.



2. Click on a **Control Text** plugin icon to open its editor:
 1. Set the **Field identifier**. The Field identifier must be a numeric value and the value will be used to give the Viz Trio page a tab order.
 2. Add a **Description**.
 3. Repeat for the other two **Control Text** plugins.

Note: In this scene the **headline** container will typically get identifier 1, and the two **bullet_text** containers will get identifier 2 and 3.



Note: More properties can be exposed to the Viz Trio operator. If the **Position**, **Rotation**, and **Scale** properties are exposed the operator can hide or show an object and add material.

3. Click on the **Control Object** icon to open its editor.

4. Type a **Description** for the Viz Trio template (e.g. `lower_third`).



5. Add the **Control Container** plugin to a container to be exposed (e.g. **headline**).



6. Click the **Control Container** plugin icon to open its editor.



7. Type a **Field Identifier**.

Note: If there is another control plugin on that container, like a **Control Text** or **Control Image**, use the same Field identifier.

- X/Y/Z properties for a keyframe in an animation must be specified by a Stop Key Frame name.

To Edit Multiple Elements with a Single Value

- Use the same **Field Identifier** for multiple control plugins to make two or more fields as a single element that will receive the same value.

This can, for example, be used with a bar that scales and a text label that shows its value. The scale value and the text can then be set in a single operation, which makes sure that there is no mismatch between the bar size and the written text value.

10.17 Create Transition Effects

This section details how to design custom transition effects for use with Viz Trio. The scene transition mechanism uses the [GFX Channel Media Asset](#) Media Asset.

IMPORTANT! When creating these transition effects also refer to the User Manuals for the required Control Applications: Viz Trio, Viz Pilot, Viz Weather, etc.

Note: To create transition effects, dynamic textures must be create with the [GFX Channel Media Asset](#) Media Asset.

Tip: The Dynamic textures may also be available as part of an already existing transition effect scenes in the **dynamic** folder (at root level).

The Dynamic textures must be named **layer1** and **layer2**. These are not ordinary images as they have the possibility to load into them an actual scene.

- **layer1:** Represents the first scene.
- **layer2:** Represents the scene to transit to.

It is possible to make any kind of animation, but there are a few design conventions to follow:

- The scene should start with a full screen view of the dynamic texture layer1, and end with a full screen display of the dynamic texture layer2. Use the screen size section in the transformation editor to set the size of the texture container to fit the screen exactly
- A transition scene is a scene that is created to move from one scene to the next in a dynamic way (not by cut)

Note: All scenes must be placed in a **transitions** folder in the scene database to make it visible in the control application.

This section contains the following procedures:

- [To Create Dynamic Textures](#)
- [To Create a Transition Scene](#)

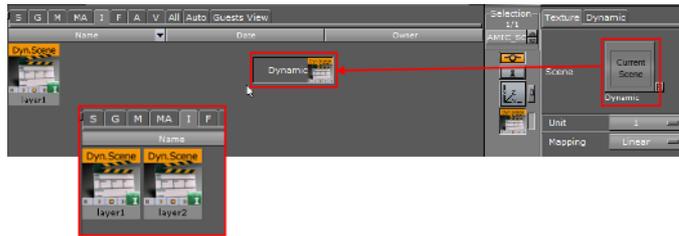
To Create Dynamic Textures

1. Add the **Dynamic Scene** Media Asset to the Scene Tree.



2. Open the Dynamic Scene properties **Dynamic** tab.
3. Configure the **Dynamic** tab properties as required (see [GFX Channel Media Asset](#)).
4. Open the Transformation editor.
5. Set **Screen Size** to **Screen**.
6. Click **Server**.
7. Create a folder at the root level and name it **dynamic**.
8. Select the **I** (Image) tab.
9. Open the Dynamic Scene properties **Texture** tab.
10. Drag the **Current Scene** placeholder icon into the Server Area.

11. Save as 'layer1'.
12. Repeat steps 10 and 11 to create 'layer2'.

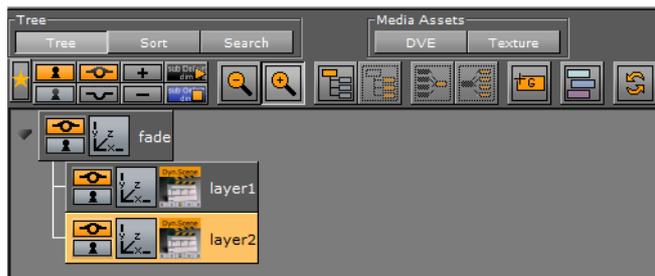


13. Delete the **DYNAMIC_SCENE** container.

To Create a Transition Scene

1. Add a group container to the Scene Tree.
2. Name the container **fade**.
3. Add the dynamic textures 'layer1' and 'layer2' as sub-containers to the **fade** container.

Note: There might not be images in the preview window, but only the bounding box.



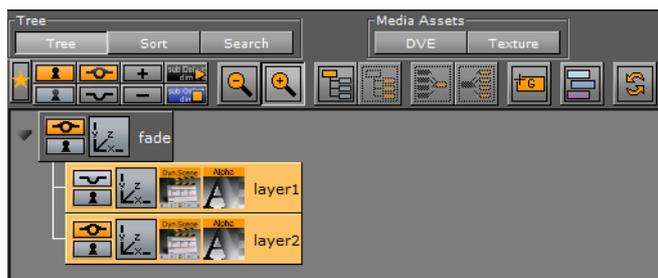
4. Open the **fade** container Transformation editor.
5. Set **Screen Size** to **Screen** (resize the container to fit the whole screen).

Note: The scaling of the main group (**fade**) to screen size will make sure that the dynamic textures (**layer1** and **layer2**) will be the correct full frame size.

6. Add the **Alpha** plugin to the **layer1** and **layer2** containers.



- *Built Ins -> Container Plugins -> Global*



7. Create a fade **out** animation for the **layer1** container:
 1. Click the **Alpha** plugin icon.
 2. Set the **Alpha** to 100%.
 3. Click **Set key**.
 4. Set the **Alpha** value to 0%
 5. Click the **Set key**.



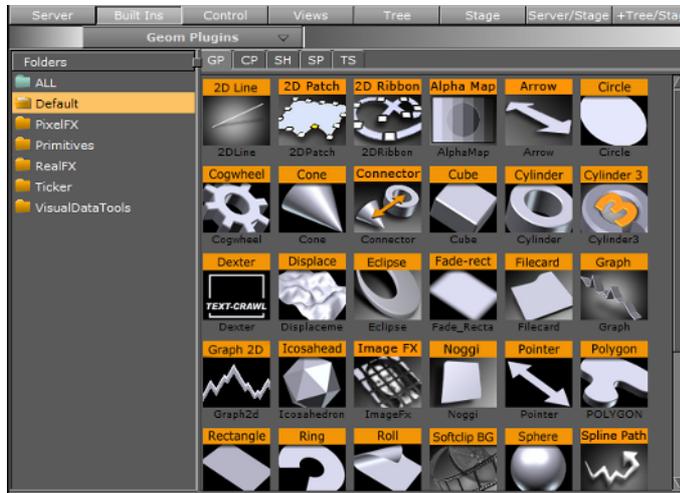
8. Create a fade **in** animation for the **layer2** container:
 1. Click the **Alpha** plugin icon.
 2. Set the **Alpha** to 0%.
 3. Click **Set key**.
 4. Set the **Alpha** value to 100%
 5. Click the **Set key**.

Tip: In the **Stage**, place the timeline somewhere in the middle before the scene is saved as it usually gives the best representation of the transition effect.

9. Click **Server**.
10. Create a folder at the root level and name it **transitions**.
11. Save the scene to the transitions folder.

Tip: Transition scenes are commonly stored in a **transitions** scene folder at the root of the database directory structure.

11 Geometry Plugins



Geometry (Geom) plug-ins are located in these folders:

Default	RealFX Plugins
PixelFX Plugins	Ticker
Primitives	Visual Data Tools

Note: The default path for the Geometry plug-ins is: <viz install folder>\plugin
 \<plugin name.vip>

11.1 Default

The following Geometry plug-ins are located in the Default folder:

2D Line	Graph2D
2D Patch	Icosahedron
2D Ribbon	Image FX
Alpha Map	Noggi
Arrow	Pointer
Circle	Polygon
Cloth	Rectangle
Cog Wheel	Ring
Cone	Roll
Connector	SoftClip Draw Pixels
Cube	Sphere
Cycloid	Spline Path
Cylinder	Spline Strip
Cylinder3	Spring
Dexter	Star
DisplacementMap	Torus

[Eclipse](#)[Fade Rectangle](#)[Filecard](#)[Graph](#)[Trio Scroll](#)[Wall](#)[Wave](#)

11.1.1 2D Line



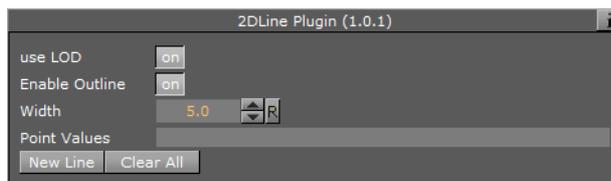
The 2D Line plugin draws a simple 2D line through given 2D coordinates, respectively point values.

Note: This plugin is located in: *Built-ins -> Geom Plugins -> Default*

This section contains the following topics:

- [2d Line Properties](#)
- [To Create 2D Lines](#)

2d Line Properties



- **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:** Press before values are entered (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.

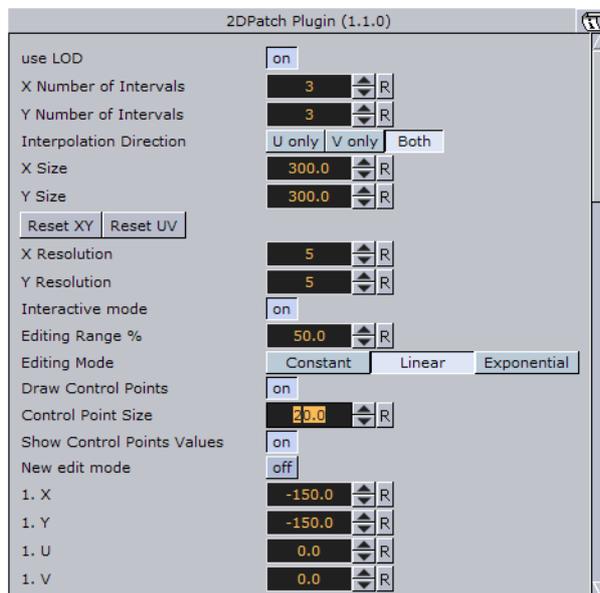
It is possible to specify the interpolation direction to be just in one direction, or in both. Depending on the shape that should be created, it is important to choose the right direction (for example it is not too difficult to create an annulus with this plugin, after choosing the right interpolation direction along the radius).

Note: This plugin is located in: *Built-ins -> Geom Plugins -> Default*

This section contains the following topics:

- [2D Patch Properties](#)
- [2D Patch Workflow](#)
- [2D Patch Tips](#)
- [2D Patch Known Limitations](#)

2D Patch Properties



- **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 <Ctrl>, 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.

2D Patch 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.

Note: 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.

5. With the left button, according to the selection roles above, you can change the X and Y location of each of the control points.
6. Once the shape of the patch is more or less defined, it is 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.
7. 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.
8. 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.
9. 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 [Create 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.

2D Patch Tips

- It is sometimes useful to work with 2D Patch while in wireframe mode. If you have a texture with alpha, to see it correctly, you need to turn the image off with the small enable/disable button near the image icon.
- 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.
- If you are done with the interactive session of defining a 2D Patch, turn the **Interactive** toggle **off**.
- 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 Scene Editor. The way to select another object is first to hit *SPACE*, and thereafter select the other object.
- 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.
- 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.

- Do not change the number of control point along an animation. It is possible, if necessary, to change the resolution number.
- While defining an animation, make sure to remove from it, if necessary, the toggle values of Show Control Points, Interactive Mode and so on.

2D Patch Known Limitations

- 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.
- While copying a 2D Patch, to make it work in the interactive mode, it is necessary to save the scene first.
- If the texture coordinates on the control points are getting outside the [0,1] range artifacts may show. Be aware to keep it within the [0,1] range.

11.1.3 2D Ribbon

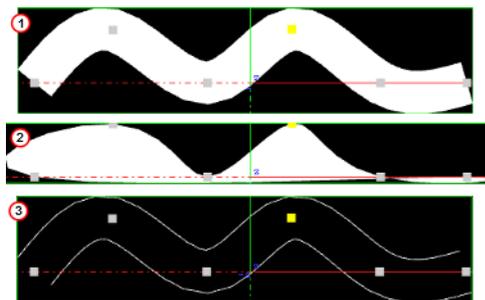


The 2D Ribbon is a curved strip. It is ideal to make customized curves to symbolize, for example, frontiers on a map or similar. The way the object is curved is easily edited in the 2D Ribbon property editor. It is possible to enable inter activity so that counterpoints of the ribbon can be dragged with the cursor.

Note: This plugin is located in: *Built-ins -> Geom Plugins -> Default*



The 2D Ribbon can be used in Ribbon (1), Filled Area (2) or Double Outline (3) modes:



This section contains the following topics and procedures:

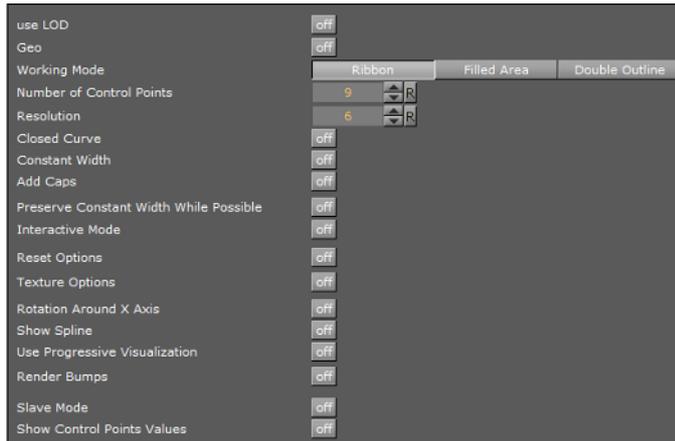
- [2D Ribbon Properties](#)
- [To Create a 2D Ribbon](#)
- [2D Ribbon Tips](#)
- [2D Ribbon Known Limitations](#)

2D Ribbon Properties

The properties of the 2D Ribbon plugin are different for each selected mode:

- [Ribbon Mode Properties](#)
- [Filled Area Mode Properties](#)
- [Double Outline Mode Properties](#)

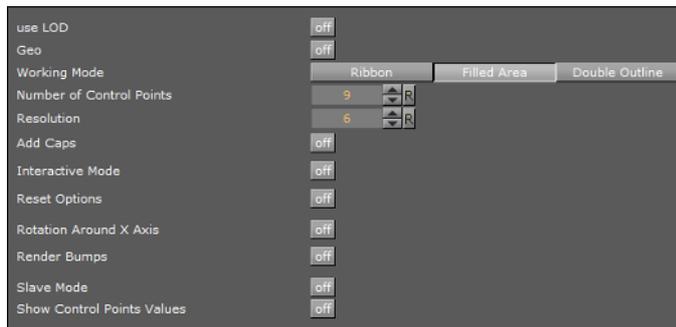
Ribbon Mode Properties



- **Use LOD:** Enable or disable the dynamic Level of Detail (LOD)
 - **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 (when **Slave** mode is set to **off**)
 - **Resolution:** Defines the number of subdivisions to polygons between the control points.
 - **Closed Curve:** Creates a closed a curve like a circle. The forward end touches the backward end.
 - **Constant Width:**
 - Set to **on**: Any width modification is along all of the 2D Ribbon
 - Set to **off**: Any width modification is around the nearest Control Point
-
- Note: Preserve Constant Width While Possible** is not available when this is selected.
-
- **Add Caps:** When set to **on** shows the Begin and End cap parameters. When set to **off** the Begin and End caps are removed, even if they are set to **on** in the two parameters:
 - **Begin Caps:** Set to **on** to apply a cap at the start of the 2D Ribbon. Also set the resolution of the start cap
 - **End Caps:** Set to **on** to apply a cap at the end of the 2D Ribbon. Also set the resolution of the end cap
 - **Preserve Constant Width While Possible:** Set to **on** to make the 2D Ribbon try and maintain its set width when modified (available when **Constant Width** is set to **off**).

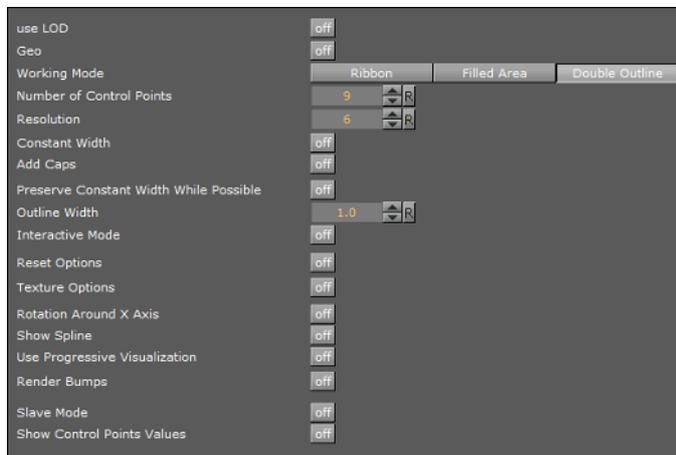
- **Interactive Mode:** Enable or disable the interactive mode of direction manipulation with the cursor:
 - **Editing Range %:** Sets the amount of 'effect' the movement of one point (move with <Shift>) should have on the neighboring points
 - **Editing Mode:** When moving a point, press <Shift> to move the other points simultaneously. How the points move depends on the editing mode that is chosen, **Constant**, **Linear** or **Exponential**
 - **Constant:** The other points will move their positions exactly equal to the one you move
 - **Linear:** The other points move in a linear way, based on the distance from the point that is being moved
 - **Exponential:** The other points move in an exponential way, based on the distance from the point it is being moved
- **Reset Options:** Set the 2D Ribbon to initial values for the Control Points and ribbon size. Changed values of Ribbon **Length** and Ribbon **Width** take affect after  has been clicked
 -  **Reset:** Reset all created curves but does not delete the existing control points.
- **Texture Options:** Set to on for 2D Ribbon Texture options:
 - **Texture Direction:** Shows the texture in different directions. For example, if a pointer right-shows 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 option stretches the texture
 - **Use Texture Factor:** Set to on to enable the texture factor parameter
- **Rotate Around X Axis:** Set to **on** to rotate the created 2D Ribbon around the X axis
- **Show Spline:** Set to **on** to show the spline of the 2D Ribbon
- **Use Progressive Visualization:** When set to **on** the activated parameters will affect the rendering:
 - **Progress Mode:** Select from two options, **By Length** and **By Control Points**. Slices the curve in the x-direction by length or by the control point index
 - **Path Position:** Limits the workspace so there are less control points to prepare
- **Render Bumps:** The curves are top-barbed which is useful for weather broadcasting
 - **Warm:** Gives a rounded edge to the bumps
 - **Cold:** Gives a sharpe edge to the bumps
 - **Occluded:** Gives alternate Warm and Cold bumps
- **Slave Mode:** To have two ribbons, where one acts as the master and the other as the slave. The slave ribbon uses the Number of Control Points and Resolution set for the master.
- **Show Control Points Values:** Enable the manipulation of the Control Points in the render window. X, Y, and W values set the X/Y position and the width of the curve at the different Control Points. These values are typically be changed when edited directly in the Scene Editor.

Filled Area Mode Properties



For details see [Ribbon Mode Properties](#)

Double Outline Mode Properties



For details see [Ribbon Mode Properties](#)

To Create a 2D Ribbon

1. Add the 2D Ribbon plugin to the Scene Tree.
2. Set **Interactive Mode** to **on**.
3. Set the **Number of Control Points**.

Note: This number can be changed at any time. New points are distributed along the already defined ribbon (2).

4. Open **Reset Options**:
 1. Enter (estimate) the **Ribbon Length** and **Ribbon Width** (enter a new figure and press <Enter>).
 2. Click **Reset** to activate the settings.
5. Toggle the **Constant Width**, as required.
6. Position the 2D Ribbon in the Scene.

Note: If the 2D Ribbon is 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

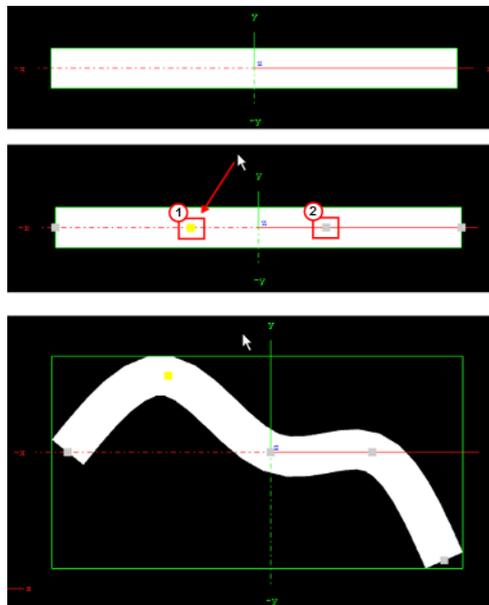
place. There is no problem to continue work on it after the rotation has been made, but because of the perspective, it might be less straightforward.

7. Click the **E** (Events) button in the Scene Editor.
8. Modify the 2D Ribbon shape, either:
 - Use the mouse cursor to modify the Control Points. Click on or near the Control Point. The selected control point will show yellow (1)
 - **Left button**: Click and drag the nearest Control Point
 - **Right button**: Modify the 2D Ribbon width

Note: If the Container, which contains the 2D Ribbon plugin, is selected, and **Interactive Mode** is set to **on**, one Control Point is always active.

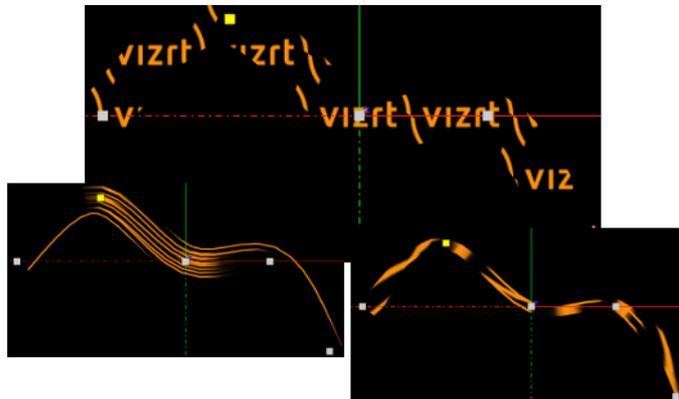
or

- Set **Show Control Points Values** to **on**. Modify the Control Points as required.



9. Set the **Resolution**. For best performance set the resolution as low as possible, but high enough to guarantee a smooth enough look in the corners of the 2D Ribbon, and also a smooth enough mapping of the image/texture on the 2D Ribbon.
10. Choose an image and drag it on the 2D Ribbon. Open the Image texture editor and set:
 - **Mapping** to **Vertex**
 - **Wrap** to **Clamp**
11. Map the image more accurately on the 2D Ribbon. Set **Texture Options** to **on**:
 1. Set the **Texture Length** as required.
 2. Set the **Texture Direction** as required.

3. Set the **Texture Head Location** to see the texture flowing along the 2D Ribbon.



12. This is a good point to review again all set parameters.
13. Define an animation, if required. Typically with the 2D Ribbon the animation will be just on the Texture Head Location.

IMPORTANT! While defining an animation on the parameters of a 2D Ribbon, make sure that the right window is open in Viz Artist. Without it the changes that are made with the cursor will not take effect, with regard to the animation.

2D Ribbon Tips

- It is sometimes useful to work with 2D Ribbon with use LOD set to on.
- If you have a texture with alpha, to see it correctly set the image to off with the small enable/disable button near the image icon.
- When finished with the creation of a 2D Ribbon, make sure that **Interactive Mode** is set to off
- When the 2D Ribbon **Interactive mode** is set to on, and the Scene Editor **E** is active, it is not possible to select another object with the cursor in the Scene Editor. To select another object make sure that **Interactive mode** is set to off and the Scene Editor **E** is inactive
- Animation on the 2D Ribbon is quite heavy on performance. Try to keep the number of Control Points and Polygons as low as possible.
- When a texture is used with a 2D Ribbon, make sure that, along the flow direction, on the texture edges there is a clean line with alpha equal to 0. If not there will be a wake of the image edges in front or behind it.
- Do not change the number of Control Point along an animation. It is possible, if necessary, to change the resolution number.
- While defining an animation, make sure to remove from it, if necessary, the toggle values of Show Control Points, Interactive Mode and so on.

2D Ribbon Known Limitations

- When you have in the scene a 2D Ribbon, by choosing an object in the Viz Artist Scene Editor, the Viz Artist user interface is not updated automatically.
- While copying a 2D Ribbon, to make it work in the interactive mode, it is necessary to save the scene first.

- If the texture coordinates on a Control Point are close to or outside the [0,1] range, artifacts may show. Keep it within the [0,1] range.

See Also

- [2D Follow](#)
- [Create Animations](#)

11.1.4 Alpha Map



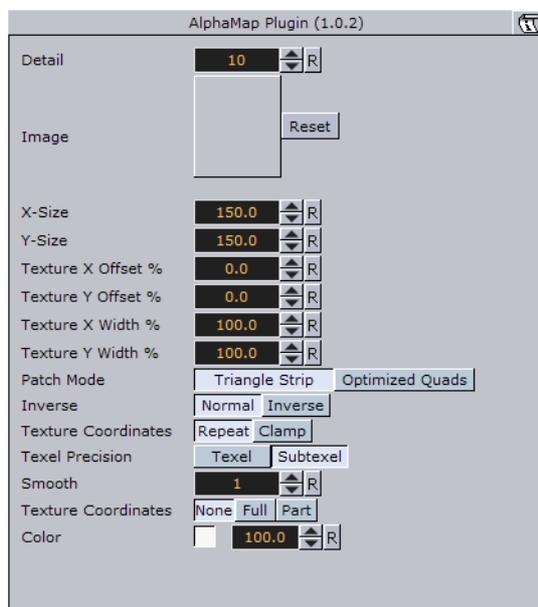
This plugin enables you to create alpha maps by means of a grayscale image which cannot use an alpha image as an alpha channel directly. The intensity value of the grayscale image is translated to an opacity (alpha) value. An image can have another image as alpha channel, but a video source cannot. So if you want a video source to have an overlying alpha mask, you can use the alpha map.

Note: This plugin is located in: *Built-ins -> Geom Plugins -> Default*

This section contains the following topics:

- [Alpha Map Properties](#)
- [To Add an Alpha Map](#)

Alpha Map Properties



- **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 to be repeated or clamped when it is too small to fit onto the alpha map's 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 plugin 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



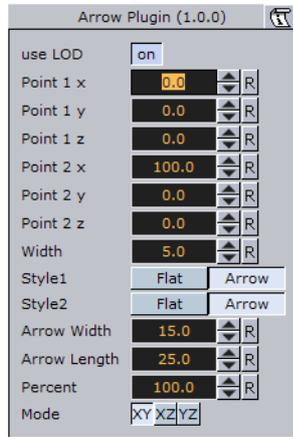
The Arrow built in geometry allows you to create and customize an arrow. The arrow is a 2D object. By clicking on the arrow icon in a container, the arrow editor opens. The default arrow has two heads. At the tip of each arrowhead the reference points are placed. The left point is 1 and the right is 2.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

- [Arrow Properties](#)
- [To create an arrow](#)

Arrow Properties



- **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 plugin to it.
2. Add a material and/or an image to the same container as the arrow plugin to add color and/or texture to it.

11.1.6 Circle



With this plugin you can create a circle with different corner levels.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

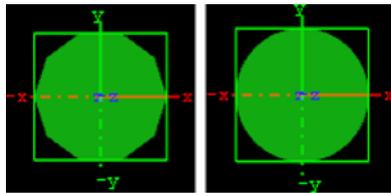
- [Circle Properties](#)
- [To Create a Circle](#)

Circle Properties



- **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 plugin to it.
2. Add a material and/or an image to the same container as the circle plugin to add color and/or texture to it.
3. Set the number of corners.

11.1.7 Cog Wheel



Use this plugin to create a cog wheel. It has a range of with that has a range of [Cog Wheel Properties](#) 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.

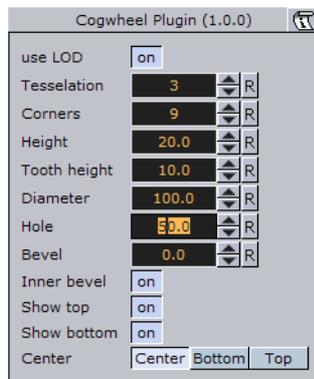
Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

- [Cog Wheel Properties](#)

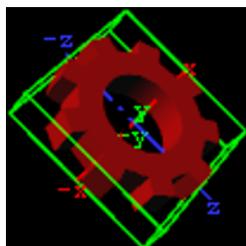
- [To Create a Cogwheel](#)

Cog Wheel Properties



- **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** plugin 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.

11.1.8 Cone



This plugin creates a cone geometry.

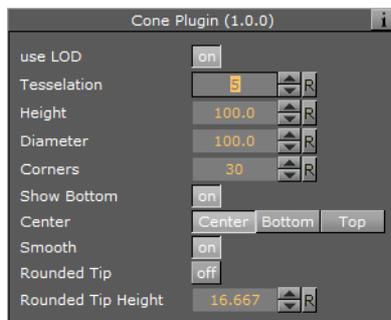
Set the tessellation to a reasonable value since it does not make a great difference in quality.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

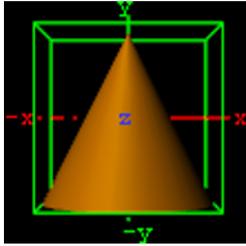
- [Cone Properties](#)
- [To Create a Cone](#)

Cone Properties



- **Use LOD:** Enables/disables dynamic level of detail.
- **Tessellation:** Sets the degree of detail. Parameter name: Tessellation.
- **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** plugin to a container.
2. Add a **material** to the same container.

11.1.9 Connector



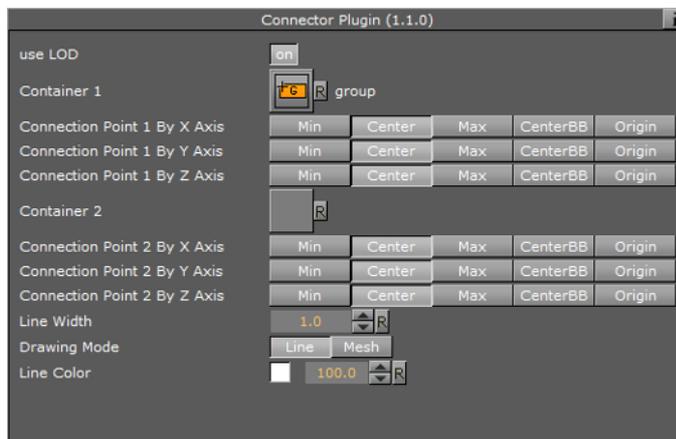
This plugin is useful to connect two objects with a line in a required color and width.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

- [Connector Properties](#)
- [To connect two objects](#)

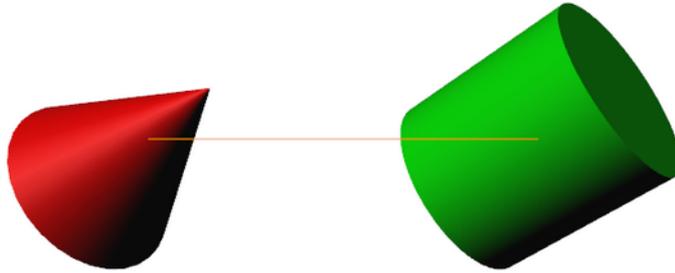
Connector Properties



- **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:** Available 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** plugin 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 plugin 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.

See Also

- Scene Editor [Scene Editor Buttons](#)
- [BoundingBox](#) plugin

11.1.10 Cube



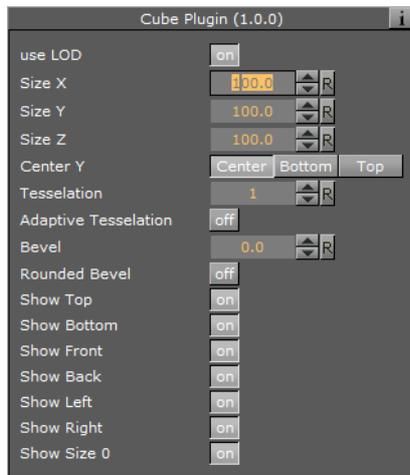
Use this plugin to create cubes with particular widths, heights, depths and other attributes.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

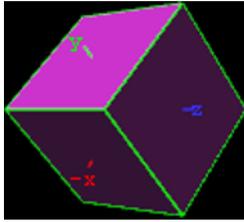
- [Cube Properties](#)
- [To Create a Cube](#)

Cube Properties



- **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: tessellation.
- **Adaptive Tessellation:** Adjusts tessellation 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 tessellation-parameter will be doubled, for a face of size 50 the tessellation is halved. The automatically adapted tessellation still never exceeds its maximum value of 100. Parameter name: adaptiveTessellation.
- **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 tessellation' is enabled: will cause them to become less tessellated accordingly. Tessellation of the bevel itself is affected by 'adaptive tessellation' too. Parameter name: bevel.
- **Rounded Bevel:** If enabled the beveled edges and corners will show 'rounded'. How much the round bevel is tessellated will be affected by 'adaptive tessellation'. 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 plugin 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.

11.1.11 Cycloid



Use this plugin to create a Cycloid.

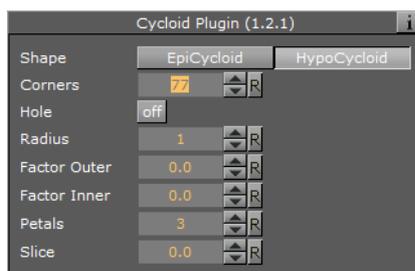
Note: Cycloid: The path traced by a point on a wheel as the wheel rolls, without slipping, along a flat surface.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

- [Cycloid Properties](#)
- [To Create a Cycloid](#)

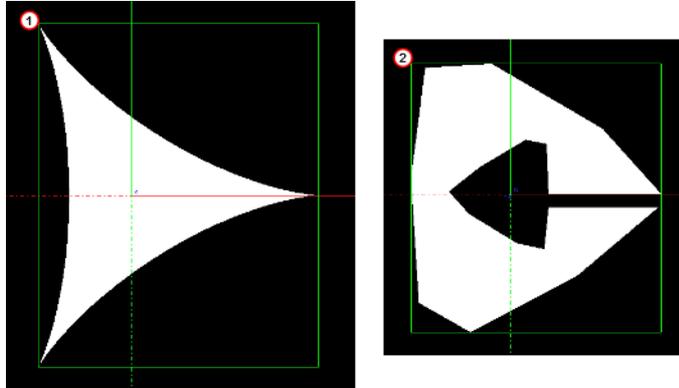
Cycloid Properties



- **Shape**
 - **EpiCycloid:** A plane curve produced by tracing the path of a chosen point of a circle which rolls without slipping around a fixed circle
 - **Hypo Cycloid:** A special plane curve generated by the trace of a fixed point on a small circle that rolls within a larger circle.
- **Corners:** Sets the number of corners the Cycloid is to be made up from.
- **Hole:** Creates a hole in the Cycloid
- **Radius:** Set the radius of the hole

- **Factor Outer:** Set the outer factor of the Cycloid.
- **Factor Inner:** Set the inner factor of the Cycloid (when **Hole** is set to On)

To Create a Cycloid



1. Create a new container.
2. Add the Cycloid plugin to it (1 (default Hypo Cycloid)).
3. Open the Cycloid editor.
4. Modify the Cycloid parameters, for example (2).
 - **Corners:** 8
 - **Hole:** On
 - **Radius:** 20
 - **Factor Outer:** 10
 - **Factor Inner:** 10
 - **Slice:** 5
5. If required, add a material and/or a texture.

11.1.12 Cylinder



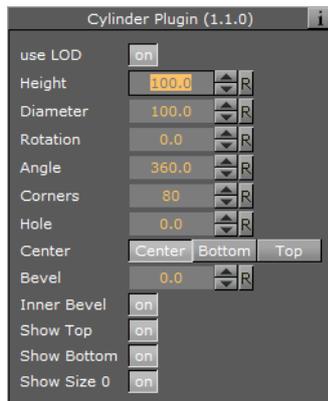
Use this plugin to create cylinders with different heights, widths and depths and other attributes.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

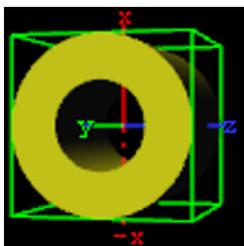
- [Cylinder Properties](#)
- [To Create a Cylinder](#)

Cylinder Properties



- **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.
2. Add the Cylinder or Cylinder3 plugin.
3. Open the Cylinder editor.
4. Set Hole to 50.0.

5. Add a material and/or an image to the same Container.
6. Open the cylinder editor.
7. Set Rotation Y to 75.0 and Z to 90.0.

11.1.13 Cylinder3



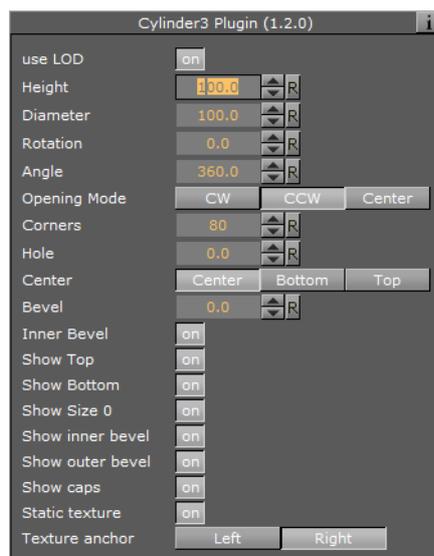
Use this plugin to create cylinders. In relation to the ordinary [Cylinder](#) plugin you have more options and settings available with the Cylinder 3 plugin.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

- [Cylinder3 Properties](#)
- [To Create a Cylinder](#)

Cylinder3 Properties



- **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 plugin, two sub-containers are also added; Animation (visible) and Templates (hidden). In addition it has a set of scripting possibilities.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

- [Common Parameters](#)
- [Control Parameters](#)
- [Design Parameters](#)
- [Text Parameters](#)
- [Trigger Parameters](#)
- [Stopper Parameters](#)
- [Dexter Parameters](#)
- [Script Programming: Parameters](#)
- [Script Programming: Tokens](#)
- [Script Programming: Syntax](#)
- [Script Programming: Templates](#)

- [Script Programming: Newline Logic](#)
- [Script Programming: Triggers](#)

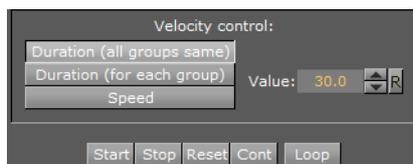
Common Parameters

These buttons are found top right of the Dexter editor and are not active in control mode:



- **Control:** See [Control Parameters](#)
- **Design:** See [Design Parameters](#)
- **Text:** See [Text Parameters](#)
- **Trigger:** See [Trigger Parameters](#)
- **Stopper:** See [Stopper Parameters](#)
- **Info:** Shows the delimited borders defined by the mark settings.
- **Update:** Applies the changes on the already existing containers.
- **Build:** Rebuild all objects.
- **Clear:** Clear the text objects.

Control Parameters



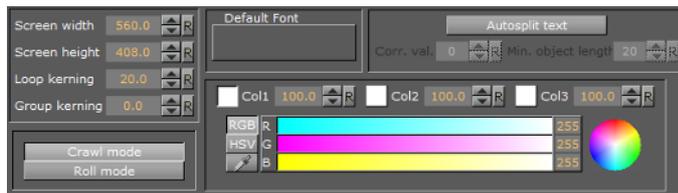
The Control section allows you to start, stop, continue and reset the scroll/roll. In addition it contains buttons that allows you to select the velocity mode and speed of the ticker.

- **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.

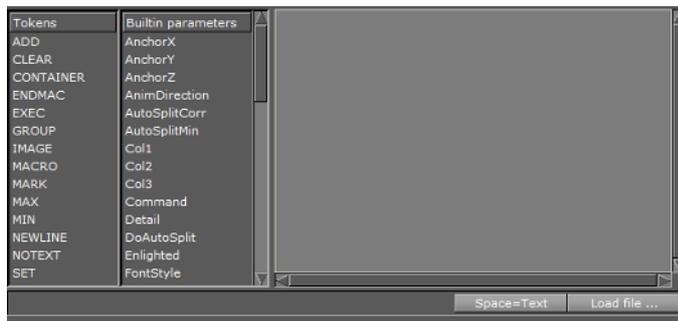
Design Parameters



In the design section of the Dexter editor set a range of parameters to how the text will show:

- **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 Parameters



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 Parameters



- **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 required Viz commands.

The concept of marks and triggers allows you to send commands to Viz Artist/Engine 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.

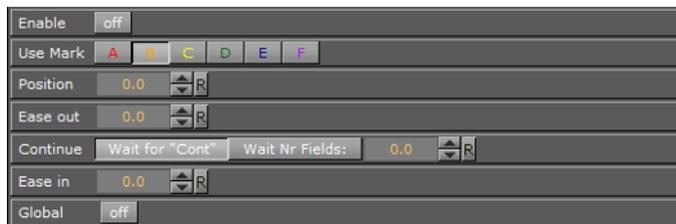
After setting marks and building the crawl/roll, you must specify trigger actions. All eight marks are listed in the trigger section of the editor. The **on/off** switch left to the mark name is used to set that mark active/inactive. The position value right to the mark name sets the position where the 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.

Note: 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 Parameters



- **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 command 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.

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.

Dexter 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).

Name	Type	Function
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.
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.

Name	Type	Function
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

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	sets the color value from the gui/external control (parameter Col1)
Command	-	Sends a command to Viz Artist/Engine.
Detail	Auto	Sets detail/fontstyle of text objects (see above).
DoAutoSplit	0	Enables/disables autosplit feature.
Enlighted	0	Set to 1 if you want enlightened 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 Y position will be stored here.
MarkOffsX, MarkOffsY	0	X and Y offset for marks (see Script Programming: Tokens).
Material	255/255/255/255	sets material used for creating text objects.
ObjectKerning	5	Sets spacing between two text objects.

Name	Default value	Function
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.
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 material pool, e.g. "Vizrt/Artdeco/artdeco.0" (see **Col1, Col2, Col3** in [Script Programming: Parameters](#)).

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** in [Script Programming: Parameters](#)).

Script Programming: Tokens

Name	Syntax	Function
ADD	<ADD Parameter Value>	Adds Value to Parameter (see example below).
CLEAR	<CLEAR Parameter>	Clears Parameter (sets it to "").

Name	Syntax	Function
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, 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, plugins, 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> (see [Script Programming: Tokens](#)) describes a new text object, which is separated from the previous one. Every newline moves AnchorX (see **AnchorX, AnchorY, AnchorZ** in [Script Programming: Parameters](#)) 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:

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.
Camera:	New line.
\$Camera	
<NOTEXT>	The <NOTEXT> token will not add new lines before the next text element occurs (see Script Programming: Tokens).
x	New line.

Lines	Explanation
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 (see [Script Programming: Tokens](#)).

Script Programming: Triggers

The concept of marks and triggers allows you to send commands to Viz Artist/Engine 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 Artist/Engine.

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 Artist/Engine. 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



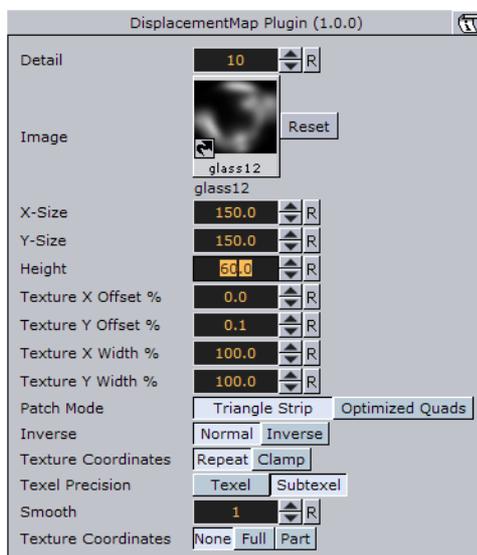
This plugin enables you to create topographical (height) maps by means of a grayscale image. The intensity value of the gray scale image is translated to a height value.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

- [DisplacementMap Properties](#)
- [To Create a Displacement Map](#)

DisplacementMap Properties

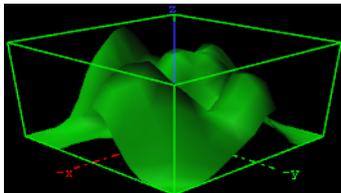


- **Detail:** Sets the detail level of the map, the lower the value the higher the level of detail.
- **Image:** Drag the image you want to use creating the map, onto the placeholder. The image must be a valid luminance or alpha image.
- **X-size:** Sets the size of the map along the X-axis.
- **Y-size:** 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 to be repeated or clamped if it is too small to fit onto the displacement map's 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 lighting.
- **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 the number of polygons in detail too much, as this is a common mistake.

To Create a Displacement Map



1. Create a new group and add the displacement plugin to it.
2. Add material to the group container.
3. Open the displacement plugin 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



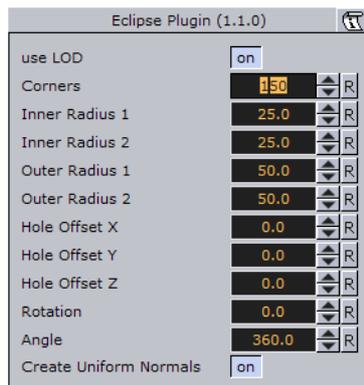
This plugin enables you to create an eclipse shape.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

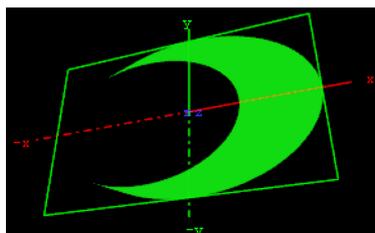
- [Eclipse Properties](#)
- [To Create an Eclipse](#)

Eclipse Properties



- **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 plugin to it.

2. Add material to the group container.
3. Open the eclipse plugin 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



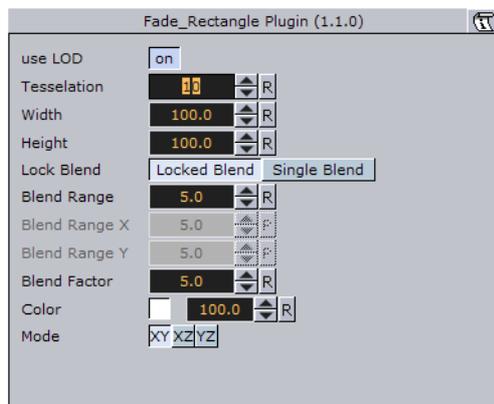
The Fade Rectangle's four sides can be set to fade in a way specified by the user. Both the degree of fading and the area influenced by the fading can be customized.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

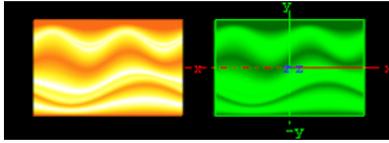
- [Fade Rectangle Properties](#)
- [To Create a Faded Rectangle](#)

Fade Rectangle Properties



- **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:** Shows 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 plugin onto it.
2. Add an image to this group.
3. Open the fade rectangle editor and change the required color through the color parameter and adjust the width and the height.

11.1.18 Filecard



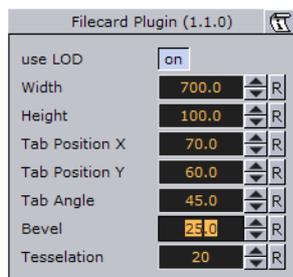
Use this plugin to create filecards with different widths and heights and other attributes.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

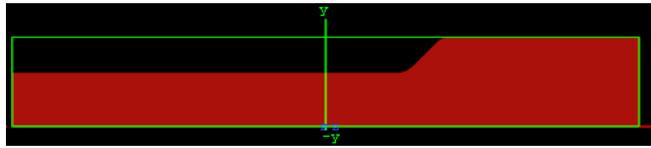
- [Filecard Properties](#)
- [To create a filecard](#)

Filecard Properties



- **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.
- **Tessellation:** 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 plugin 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



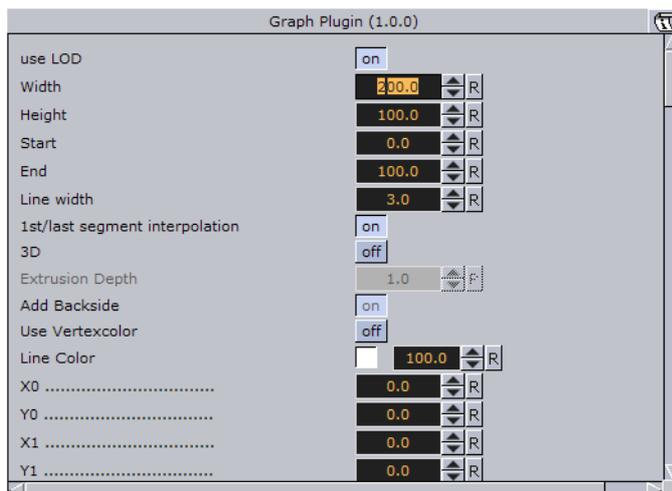
The Graph object allows you to create a 2D or 3D graph with up to 50 values.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

- [Graph Properties](#)
- [To Create a Graph](#)

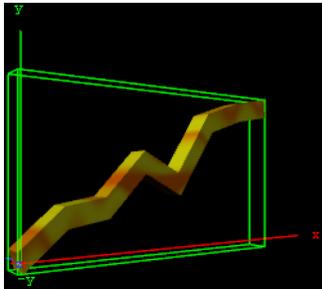
Graph Properties



- **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 plugin 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



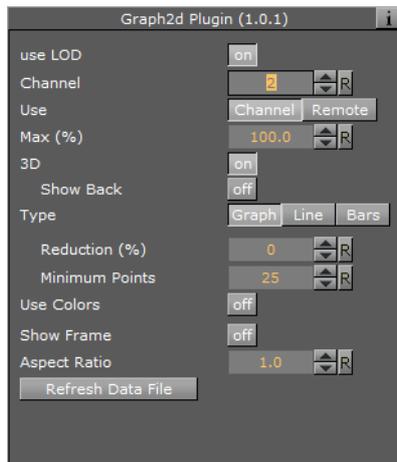
Use this plugin to create 2D and 3D graphs, lines and bars from imported files.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

- [Graph2D Properties](#)
- [To Create a Channel File for a Graph2D](#)
- [To Create a 2D Graph](#)

Graph2D Properties



- **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 Channel File to search for. The file holds the vertex and other data to define the whole graph
- **Use:** Load the file through a Channel File or remote access
- **Max (%):** It sets the rage for rendering. For example, if the range is set to 50% the graph will be cut 50% in width
- **3D:** Enable 3D rendering of the graph, and the **Show Back** option.
 - **Show Back:** Enable 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:** Reload the Channel File, which holds the data value for the x- and y-points

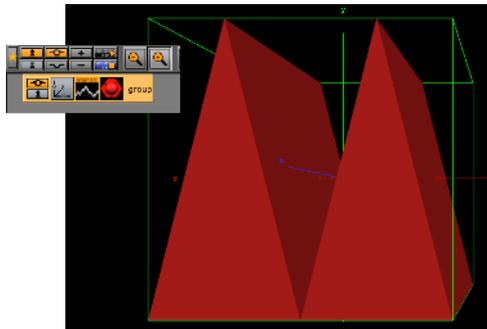
To Create a Channel File for a Graph2D

1. Create a text file (for example, **CHANNEL2**), with no file extension.
2. Add this 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.
3. Create this directory: <viz data folder>\plugin\graph2D. and place the channel file in it.

IMPORTANT! The channel file must always be named **CHANNEL<Channel number>**.

To Create a 2D Graph

1. Create a new Container.
2. Add the **Graph2D** plugin.
3. Add a Material and/or a Texture.
4. Set the **Channel** parameter to the Channel file (for example: **2** for **CHANNEL2**).
5. Click **Refresh data file** if the Channel file contents have been modified.



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.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

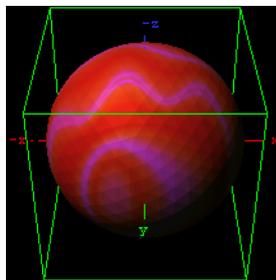
- [Icosahedron Properties](#)
- [To Create an Icosahedron](#)

Icosahedron Properties



- **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 plugin 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 Tessellation 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.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section also contains information on the following topics:

- [Image FX Properties](#)
 - [Image](#)
 - [Geometry](#)
 - [Trigger](#)
 - [Effect](#)
 - [Control](#)
 - [To Add, Load and Remove Images](#)

Image FX Properties



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 plugin container in the scene tree.
- **Load from Scenetree:** Loads images that are added directly as sub-containers to the ImageFX plugin container.
- **Clear all:** Removes all sub-containers to the ImageFX plugin container.

Geometry

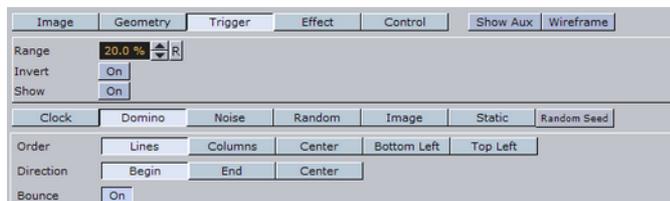


The image is split into a number of stand-alone sections that can be modified independently.

- **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.
- **Tessellation:** 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



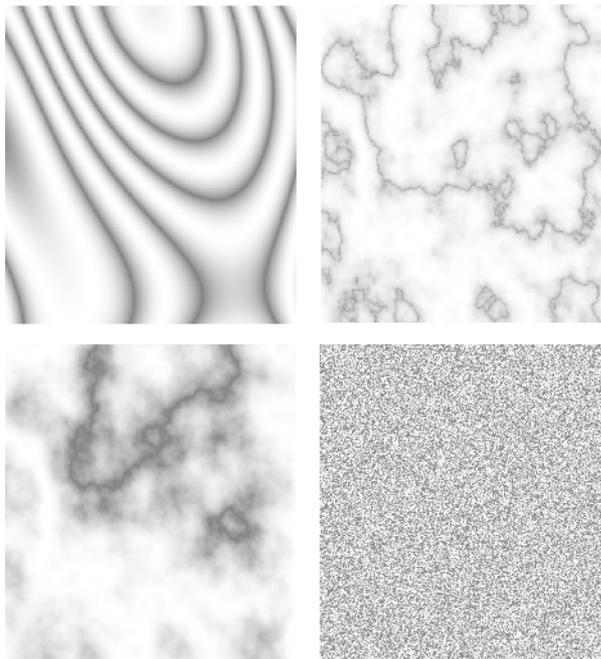
In the trigger section you can select between different progress maps that defines how every section of the image is to be triggered by the global progress. Each section of the image receives its base value, the global progress is a value that changes as you animate by using one of the options in the 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 influenced, 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.

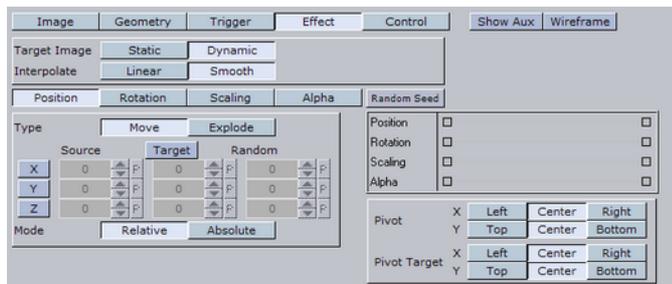
- **Direction:** Sets the direction for the clock hand movement.
- **Domino:** Sets the values for the sections that are computed based on a domino movement. It has the following options:
 - **Order:** Sets the order in which the tiles will be moved. Choose between **Lines** (Line by line movement), **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 required effect. The four images below show how the four noise parameters look like when the show option is enabled:



- **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 shown, 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**.

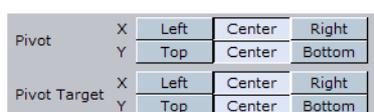
To all options (except alpha) a certain degree of user-controlled randomness can be added.

- **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 Key Frames 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 tessellation 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 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 plugin container.
2. To add images directly as sub-containers to the plugin container you must click the **Load from Scenetree** button to make the plugin 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



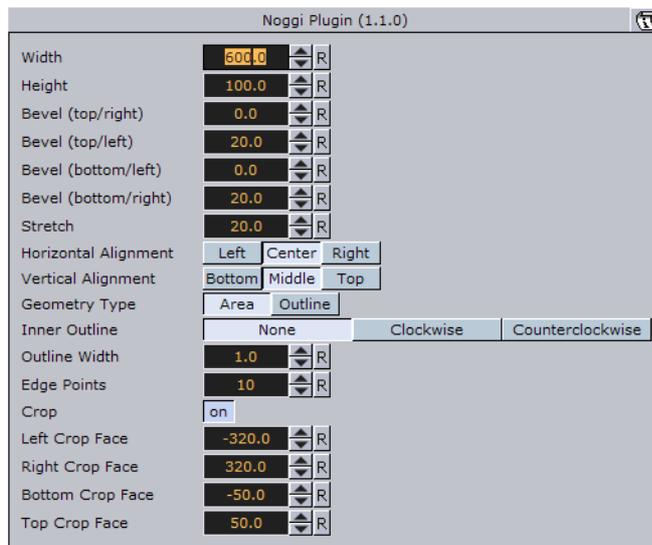
Use this plugin to create geometry objects of beveled rectangles with some attributes.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

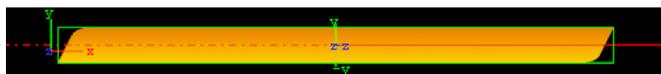
- [Noggi Properties](#)
- [To Create a Noggi](#)

Noggi Properties



- **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 plugin 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 Pointer



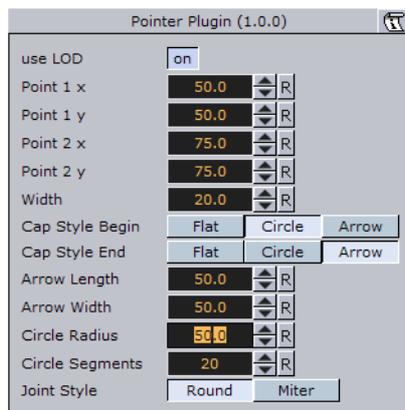
The pointer is similar to the [Arrow](#). The main difference is that the pointer has a joint link, so it looks more like a *bend* arrow. The pointer has two reference points which controls the pointer's position. The first point is in the joint link and the second point is at the arrow's head.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

- [Pointer Properties](#)
- [To create a Pointer](#)

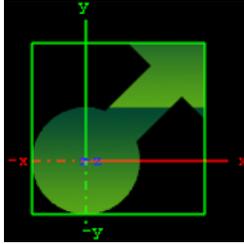
Pointer Properties



- **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 plugin 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.

11.1.25 Polygon



The Polygon Geometry plugin enables the creation of user defined polygon shapes.

The polygons are drawn and edited directly in the [Scene Editor](#). Created Polygons can also be rotated to create a 3D object (see [Revolve: Spline Rotation in 3D](#)).

To create a Polygon, add the Polygon plugin to a container and start drawing by clicking in the Scene Editor to create polygon nodes. A click, hold and drag converts the node into a spline handle for drawing curves.

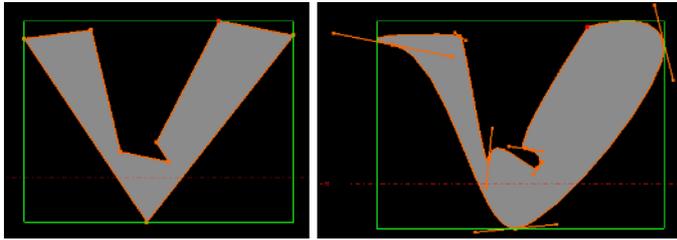
If an existing node is clicked on it removes the node, and if a line or spline is clicked on, between two nodes, it inserts a new node at this location.

There are several other shortcuts for the polygon editor, some are describe in [Create a Polygon](#), see a complete list in [Polygon Plugin Editor Shortcuts](#).

If more than one polygon is created, in the same container, and if they overlap each other, they will act as masks.

Instead of drawing polygons manually they can be imported as Adobe Illustrator files (*.ai) as well.

Note: Only Adobe Illustrator AI 88 format files are supported.



Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

- [Polygon Properties](#)
- [Polygon Scene Editor Context Menu](#)
- [Create a Polygon](#)
- [Scene Editor Grid](#)
- [Revolve: Spline Rotation in 3D](#)

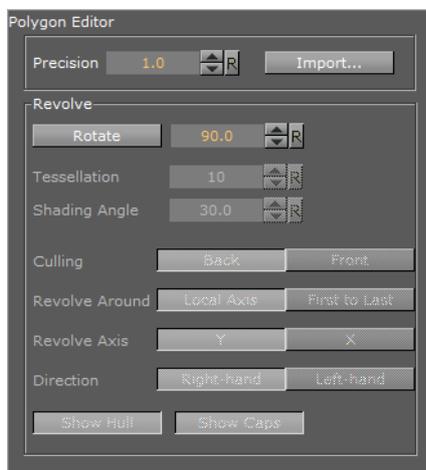
See Also

- [Polygon Plugin Editor Shortcuts](#)

Polygon Properties

To toggle between the Polygon and Container properties:

- Click the Polygon or Transformation icon in the:
 - [Scene Tree](#) Container, or
 - The Properties panel
- In the [Scene Editor](#):
 - Outside the polygon bounding box: *Right click -> properties -> geometry*
 - Inside the polygon bounding box: *Right click -> edit container*

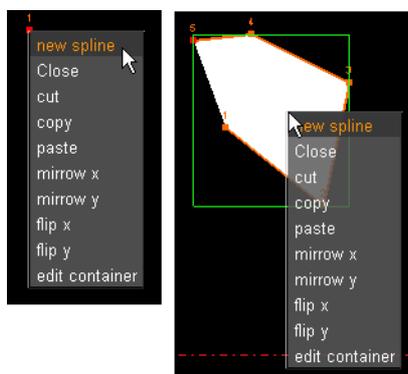


- **Precision:** Set the precision of the spline calculation
- **Import:** Import Adobe illustrator files. Opens a window to select a file and import. Only Adobe Illustrator AI 88 format files are supported.
- **Revolve:** See [Revolve: Spline Rotation in 3D](#)

Polygon Scene Editor Context Menu

Right-click, and hold, in the polygon bounding box, in the Scene Editor, to open the polygon context menu (even if there is just one point created). To select a menu option see [To Select an Option from the Context Menu](#).

Note: The right-click must be inside the created polygon shape box. If a click is outside the bounding box then the standard Scene Editor menu will be shown.



- **New spline:** Create a new polygon in the same container
- **Close / Open:** Open or close a polygon
- **Cut:** Cut the selected polygon or spline nodes
- **Copy:** Copy the selected polygon or spline nodes
- **Paste:** Paste the selected polygon or spline nodes
- **Mirror X / Mirror Y:** Mirror the created shape in the X or Y axis
- **Flip x / Flip y:** Flip the created shape in the X or Y axis
- **Edit container:** Open the Container editor and exit polygon edit mode. To edit the polygon again click on the polygon icon in the properties editor or the Scene Tree

To Select an Option from the Context Menu

1. Right-click and hold.
2. Move the cursor to the required option.
3. Release to select.

Create a Polygon

This section contains the following procedures:

- [To Create a Simple Spline](#)
- [To Draw a New Spline](#)
- [To Move a Spline](#)
- [To Move More Than One Spline](#)

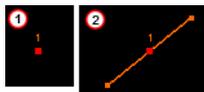
- [To Add or Delete a Spline Point](#)
- [To Move Spline Points](#)
- [To Create or Delete a Spline Point Handle](#)
- [To Resize a Spline Point Left or Right Handle](#)
- [To Resize a Spline Point Handle](#)
- [To Mirror Spline Sections](#)
- [To Flip a Spline](#)

See Also

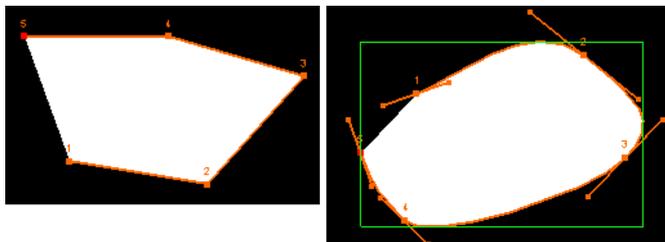
- [Polygon Plugin Editor Shortcuts](#)

To Create a Simple Spline

1. Create a Container.
2. Add the Polygon plugin.
3. Click on the Polygon icon (in the Container or the Properties editor).
4. The polygon splines are created by each click in the Scene Editor:
 - Click in the Scene Editor to create a spline point (1)
 - Click and drag in the Scene Editor to create a spline node, complete with handles (2) (continue the drag to increase the length of the handles)



5. While the polygon plugin is active each click in the Scene Editor will create more polygon points, and create the polygon shape.



Tip: At any time press and release <Delete> to remove the last created point. Press and hold <Delete> to remove more than one spline, or remove the whole polygon.

Once the spline has been created it can be modified, as required, with various shortcuts (see [Polygon Plugin Editor Shortcuts](#)). List below are some common examples:

To Draw a New Spline

1. Right-click in an existing spline.
2. Click **New Spline**.
3. Add intersections as detailed in [To Create a Simple Spline](#).

To Move a Spline

1. Select the spline to be moved:
 - Press <S> and click on a spline.
2. Press <Alt> and drag.

Tip: If there is only one spline in the Container, press <Alt> and drag.

To Move More Than One Spline

1. Select the splines to be moved:
 - Press <S> and drag around the splines.
2. Press <Alt+S> and drag.

To Add or Delete a Spline Point

- Click anywhere on a spline to add another spline point.
- Click on a spline point to delete it

To Move Spline Points

1. Press and hold <Ctrl>.
2. Click on a spline point and drag.
When <Ctrl> is pressed move any spline points as required.

To Create or Delete a Spline Point Handle

1. Press <Shift>:
 - **Create:** Click on a spline point and drag.
 - **Delete:** Click on a spline point with a handle.

To Resize a Spline Point Left or Right Handle

1. Press <Shift>.
2. Click on a spline point handle and drag. Only the selected handle will move.

To Resize a Spline Point Handle

1. Press <Ctrl>.
2. Click on a spline point handle and drag. The left and right handles will move as one handle.

Note: The handle clicked on can be lengthened as required. The other handle will stay the same length

Note: If either the left or right handle has been moved individually, this will set the two handles together again

To Mirror Spline Sections

1. Right-click on a spline, or select particular nodes of it with <S>.

2. Select either:
 - **Mirror X:** Mirrors whole spline along the X axis.
 - **Mirror y:** Mirrors whole spline along Y axis.

To Flip a Spline

1. Right-click on a spline.
2. Select either:
 - **Flip x:** Flips whole spline over X-axis.
 - **Flip y:** Flips whole spline over Y-axis.

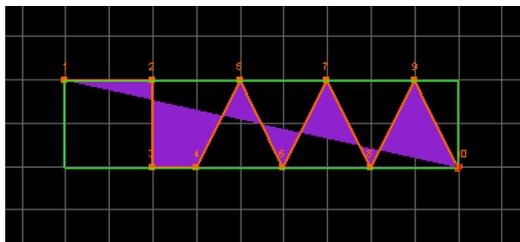
Scene Editor Grid

The Scene Editor [Grid Toolbar](#) can be used as a great tool to help draw polygons. Together with the grid snap option polygon points can be positioned, simply and accurately, in the 3D space.

Make sure that the polygon is aligned with the grid for the grid snap to work.

To Create a Spline in the Scene Editor Grid Snap

1. In the Scene Editor click  (Grid).
2. Click  (Free Grid).
3. Click  (Snap to Grid).
4. Create a spline. Each spline point will snap to the nearest cross hairs on the grid.



To Move a Spline in the Grid with Snap

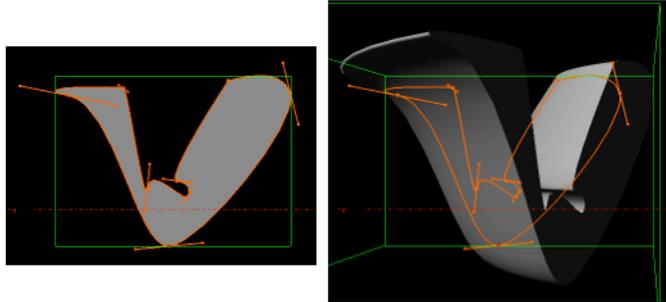
1. Click on  (Move Object to Plane).
2. Press <Alt>.
3. Click on the Spline and drag.

Note: If there is more than one Spline to be moved see [To Move More Than One Spline](#).

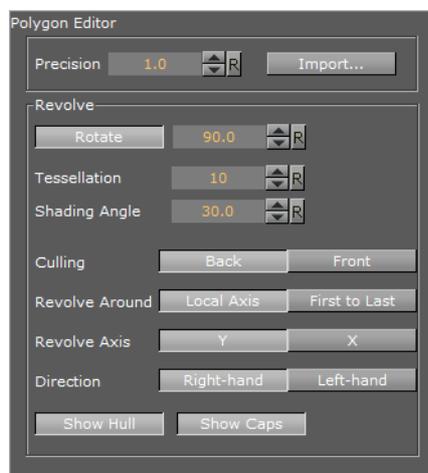
The Spline will snap to the next set of cross hairs, and so on, as it is dragged.

Revolve: Spline Rotation in 3D

The Spline rotation creates a 3D mesh by rotating the created spline.



Polygon Revolve Properties



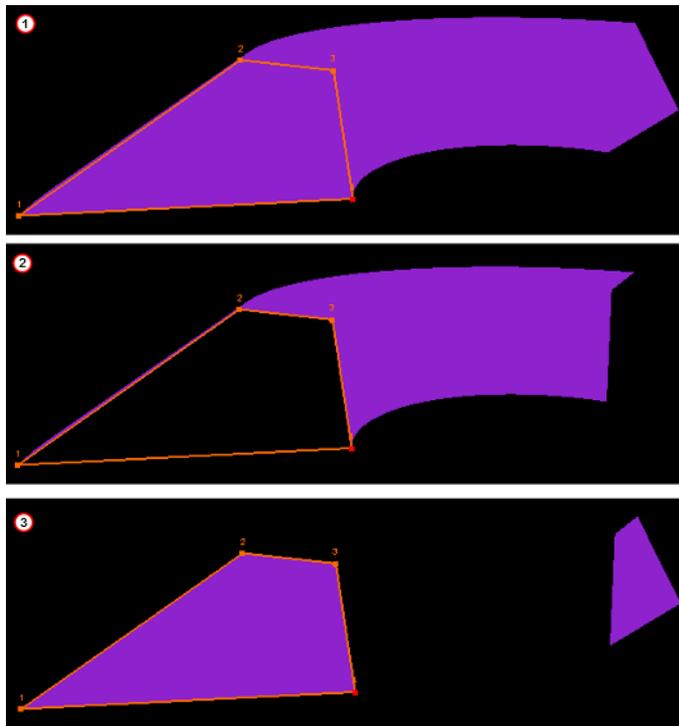
The Polygon Revolve properties are available after the first spline point has been added to the Scene Editor. Click on **Rotate** to activate.

- **Precision:** Set the precision of the spline calculation
- **Import:** Import Adobe illustrator files. Opens a window to select a file and import. Only Adobe Illustrator AI 88 format files are supported.
- **Rotate:** Click to create the revolve mesh and activate all its settings
 - **Rotate in degrees box:** Set the degree of rotation
- **Tessellation:** Sets the degree of detail for the mesh
- **Shading Angle:** Define how the shading normal are aligned on the revolve object surface
- **Culling:** Set which side of the spline to view:
 - **Back:** View the back side of the spline
 - **Front:** View the front side of the spline

Note: To view both sides at the same time, add the [Expert](#) plugin to the Container and active **Back Face**.

- **Revolve Around:** Set the axis to rotate the spline around
 - **Local Axis:** When the Rotation angle is changed the rotation is calculated around the Container axis

- **First to Last:** When the Rotation angle is changed the rotation is around the first and last spline point (**Revolve Axis** is grayed out)
- **Revolve Axis:** Set which axis to rotate around:
 - **Y:** The rotation is around the Y axis
 - **X:** The rotation is around the X axis
- **Direction:** Set which direction to view the rotation:
 - **Right-hand:** View from the right hand side
 - **Left-hand:** View from the left hand side
- **Show Hull:** Show the rotated shape (1 and 2)
- **Show Caps:** Show the original polygon shape at the beginning and end of the rotation (1 and 3)



To Rotate a Spline

1. Create a spline.
2. Click on **Rotate**.
3. Modify the created 3D object with the [Polygon Revolve Properties](#).

11.1.26 Rectangle



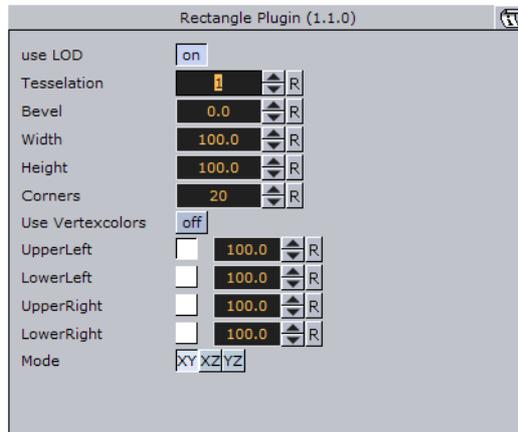
Use this plugin to create a simply 2D rectangle with some attributes.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

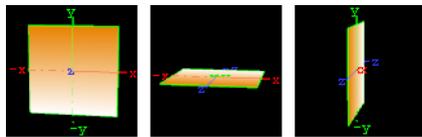
- [Rectangle Properties](#)
- [To Create a Rectangle](#)

Rectangle Properties



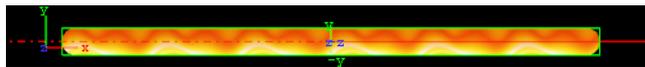
- **Use LOD:** Enables/disables dynamic level of detail.
- **Tessellation:** Sets the degree of detail.
- **Bevel:** Enables and sets the degree of bevel at the rectangle edges. Bevel may have an effect on the height of the rectangle. For example if Height is set to 100.0 and Bevel is set to 51.0, the 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 plugin to it.
2. Add a material and/or a texture to the group, and/or open the Rectangle plugin 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.

11.1.27 Ring



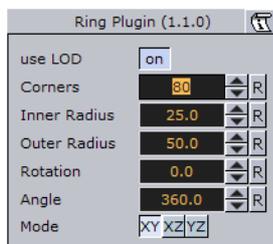
Use this plugin to simply create an open or closed ring with some attributes.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

- [Ring Properties](#)
- [To Create a Ring](#)

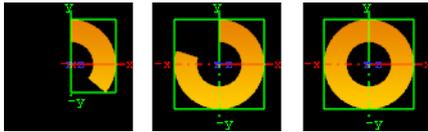
Ring Properties



- **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 required 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 plugin to it.
2. Add a material and/or a texture to the group.
3. Open the Ring plugin editor and **animate** the Angle.
 - Set Angle to 0.0 (60/50 fps).
 - Set Angle to 360.0 (120/100 fps).

11.1.28 Roll



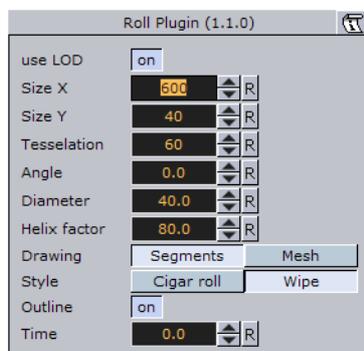
The roll allows you to create a rectangle that rolls up in different ways. This is typically used for different kinds of unveiling. To make the roll look realistic, the expert plugin must be added to the container to enable visualization of backface and two-sided lightening.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

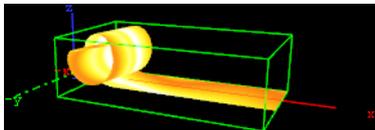
- [Roll Properties](#)
- [To Create a Roll](#)

Roll Properties



- **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 [Create Animations](#) .

To Create a Roll



1. Create a new group and add the Roll plugin to it.
2. Add the [Expert](#) plugin 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.

11.1.29 SoftClip 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 shown synchronized to Viz rendering, or according to the Frames Per Second (FPS) setting, depending on the play mode that is chosen.

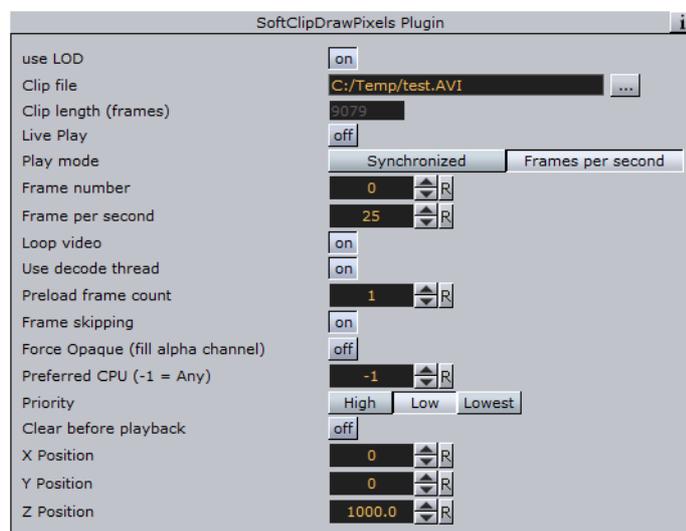
On machines with multiple CPUs turning on the decode thread will help balancing the CPU load over the CPUs. If you use live play, the Preload frame count can help reducing spike-loads and avoid dropped frames.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

- [SoftClip Draw Pixels Properties](#)

SoftClip Draw Pixels Properties



- **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.

11.1.30 Sphere



Use this plugin to create a simple sphere with some attributes.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

- [Sphere Properties](#)

Sphere Properties



- **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.

11.1.31 Spline Path



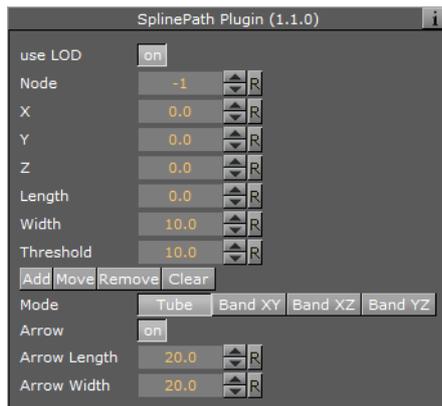
Use this plugin to create complex paths with an arrow look.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

- [Spline Path Properties](#)
- [To Create a Spline Path](#)

Spline Path Properties



- **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:** Set the arrow length (only when **Arrow** is set to On)
- **Arrow Width:** Set the arrow width (only when **Arrow** is set to On)

To Create a Spline Path



1. Create a new group container.
2. Add the Spline Path plugin.
3. Add the [Expert](#) plugin.
4. Open the Expert editor
5. Set **Back Face** to On.
6. Add a Material and/or a Texture to the group.
7. Open the Spline Path editor.
8. Set **Node** to 0 / **X** to 50.0 / **Y** to 50.0
9. Click Add.
10. Set **Node** to 1 / **X** to 150.0 / **Y** to 25.0
11. Click Add.
12. Set **Node** to 2 / **X** to 250.0 / **Y** to 10.0
13. Click Add.
14. Set **Node** to 3, **X** to 350.0, **Y** to 45.0
15. Click Add.

11.1.32 Spline Strip



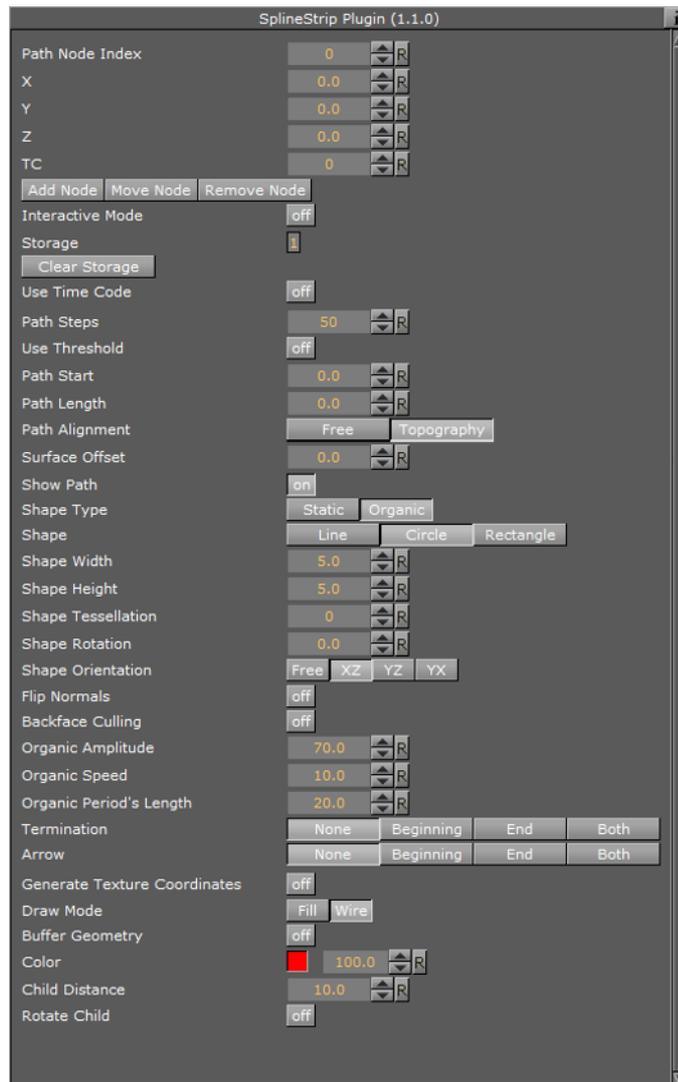
The Spline Strip plugin is used for creating 3D interactive loft extrusions which are able to be animated in their length and surface. You can use modifiable lines, circles or rectangles for the extrusion. Arrows and terrain alignment are also supported.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

- [Spline Strip Properties](#)
- [To Create a Simple Spline Strip](#)

Spline Strip Properties



- **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 required 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 shown 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 plug-ins in Viz Artist/Engine. (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 required 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 required 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

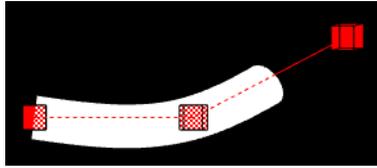
1. Create a new Container.
2. Add the Spline Strip plugin.
3. Open the Spline Strip editor.
4. Use all default values.
5. Click **Add Node**.
6. Change **X** to 100.0.
7. Click **Add Node**.
8. Change **X** to 200.0, **Y** to 50.0.
9. Click **Add Node**.
10. Set te **Shape Width** and **Shape Height** to 50.0.
11. Set **Shape** to **Circle**.
12. Set **Shape Tessellation** to 20.
13. Set **Path Lenght** to 150.



and...

1. Optionally, click **Interactive Mode**.
2. Change **Node Size** to for example 15.0.

3. Active **E** in the Scene Editor
4. Drag and drop the small Cubes as required.



11.1.33 Spring



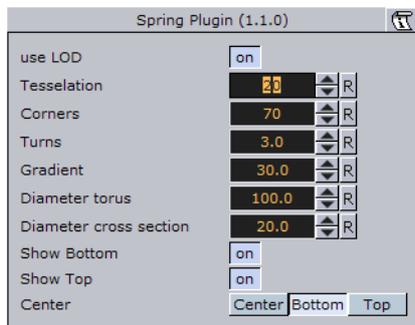
The spring object creates a spiral spring. Be aware this object can easily be very heavy to render if a high level of tessellation and corners are selected.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

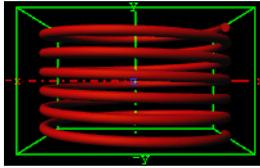
- [Spring Properties](#)
- [To Create a Spring](#)

Spring Properties



- **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 plugin 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.

11.1.34 Star



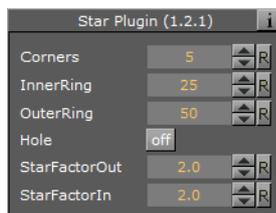
Use this plugin to create a Star with different attributes.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topic and procedure:

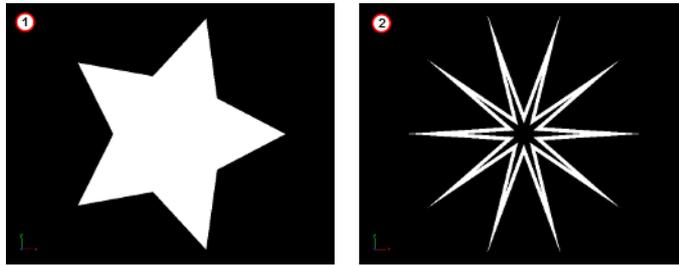
- [Star Properties](#)
- [To Create a Star](#)

Star Properties



- **Corners:** Set the number of corners on the Star.
- **Inner Ring:** Set the inner ring of the Star
- **Outer Ring:** Set the outer ring of the Star
- **Hole:** Create a hole in the Star.
- **Star Factor Out:** Set the outer ring of the hole
- **Star Factor In:** Set the inner ring of the hole

To Create a Star



1. Create a new container.
2. Add the Star plugin to it (1 (default Star)).
3. Open the Star editor.
4. Modify the Star parameters, for example (2).
 - Corners: 10
 - Inner Ring: 30
 - Outer Ring: 60
 - Hole: On
 - Star Factor Out: 6
 - Star Factor In: 6
5. If required, add a material and/or a texture.

11.1.35 Torus



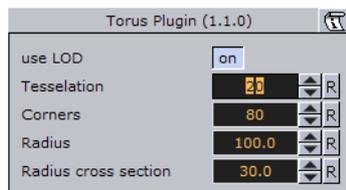
Use this plugin to create a torus with different attributes.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

- [Torus Properties](#)
- [To Create a Torus](#)

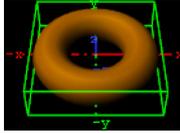
Torus Properties



- **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 plugin 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.

11.1.36 Trio Scroll



The Trio Scroll plugin is essentially a Geometry plugin 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.

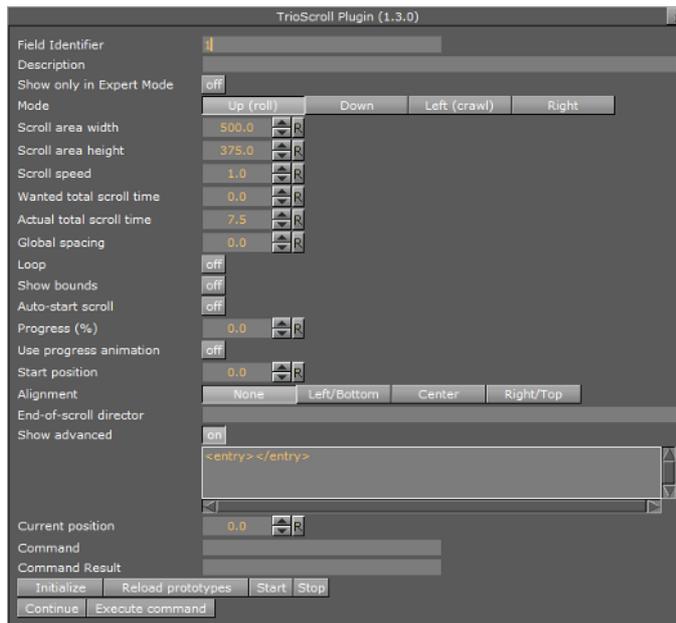
Adding the Trio Scroll plugin to the scene tree will automatically add the [Control Object](#) plugin (if it is not present).

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

- [Trio Scroll Properties](#)

Trio Scroll Properties



- **Field Identifier:** When making templates for Viz Trio, this should be a numeric value, Range: 1-n. For Viz 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 plugins 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 set 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 set On or Off (default) looping. When set to Off looping will not stop the playlist from being played till its end.
- **Show bounds:** Enables and shows 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.

- **Show Advanced:** All parameters under the Advanced section are for debugging, by Vizrt developers or very advanced users only. For more information contact Vizrt support.

The nameless text parameter with the value “<entry></entry>” shows the contents of the scroll elements in the internal VDOM format. It contains the scroll elements that can be dragged into a scroll page in the Trio client with drag and drop

- **Current position:** Scrub the scroller position.
- **Command:** Execute various scroller commands. After setting its value the click on **Execute Command**. If a command has a result it is reflected in **Command result**. The following commands are used from the Trio client:

Example: SET_FOCUS [<element name> [<tab-field name>]] – This is for example used if you double click in Trio on a scroll element to highlight it in the preview.

Example: MOVE_ELEMENT <from element name> <to element name> - This is used when changing the order of scroll elements by drag and drop.

Example: REMOVE_ELEMENT <element name> - This is used to remove a scroll element.

Example: UPDATE_TOTAL_TIME – This commands the plugin to recalculate its total time, typically after adding a new scroll element.

Example: SET_TEMP_EASEPOINT_VALUE – Gets used when editing the easepoint values.

- **Command result:** See **Command**
- **Reload prototypes:** Update the scroll elements due to changes in their templates (scenes) - use a context menu action in Viz Trio to do this (see the [Viz Trio User Guide](#))
- **Start/Stop/Continue:** Control the scroll (used in the local preview and the live controls for playback)
- **Execute command:** See **Command**

See Also

- **Create New Scroll** in the [Viz Trio User's Guide](#).
- [Control Object](#) Plugin
- [Trio Scroll Element](#) Plugin

11.1.37 Wall



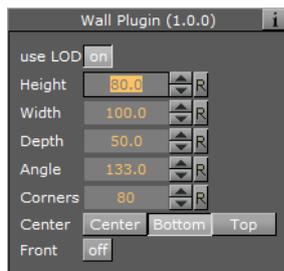
The wall object is a curved wall. The size and curving of the wall can be customized.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

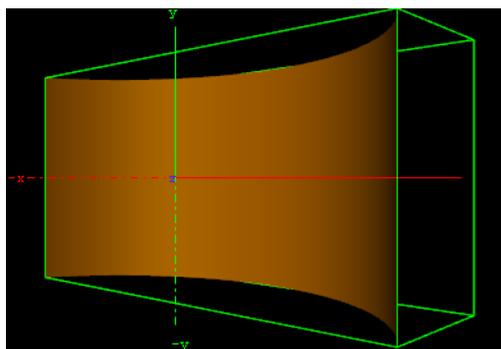
- [Wall Properties](#)
- [To Create a Wall](#)

Wall Properties



- **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 show.
- **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.

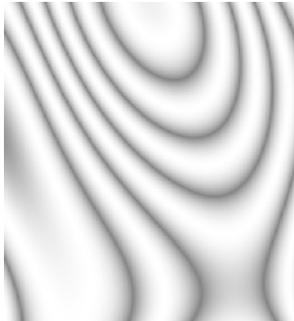
To Create a Wall



1. Create a new group and add the Wall plugin 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.

11.1.38 Wave



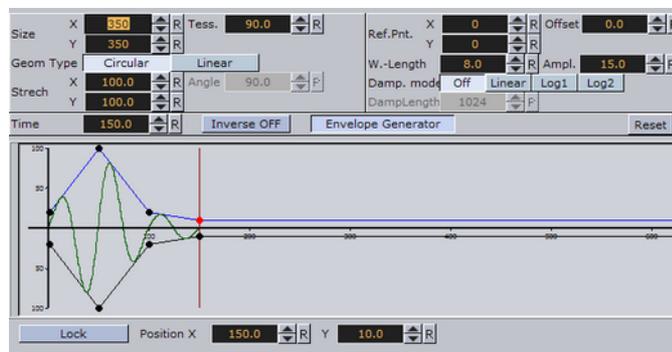
The wave allows you to create a wave sequence on a flat surface. The properties of the sequence are visualized in a graphical display in the wave editor.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Default

This section contains the following topics:

- [Wave Properties](#)
- [To Create a Wave](#)

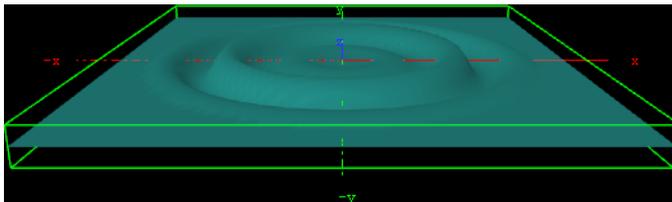
Wave Properties



- **Size X** and **Y**: Sets the size of the rectangle where the waves show.
- **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 required position.

To Create a Wave



1. Create a new group and add the Wave plugin 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 PixelFX Plugins

The Lens Flare plug-ins, which are part of the [PixelFX plugins](#) set, simulate the effect of streaks and spots of light, caused in real cameras when light enters and bounces inside a camera lens. 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. When a Flare Container is created a non-editable pixel shader will be added to the created Flair container.

In addition, the [Expert](#) Container 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.

See Also

- [PixelFXLensFlare](#)



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 plug-ins. 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 following Geom plugins are located in the PixelFX folder:

- [pxLensEnergyBolt](#)
- [Lens Flare Geometries](#)
 - pxLensGlowBall
 - pxLensGlowSpikes
 - pxLensPolyElement
 - pxLensRandomPoly
 - pxLensSpark
 - pxLensStripes
- [pxLensRays](#)

11.2.1 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.

Note: This plugin is located in: Built-ins -> Geom Plugins -> PixelFX

See Also

- [PixelFXLenseFlare](#)

11.2.2 Lens Flare Geometries

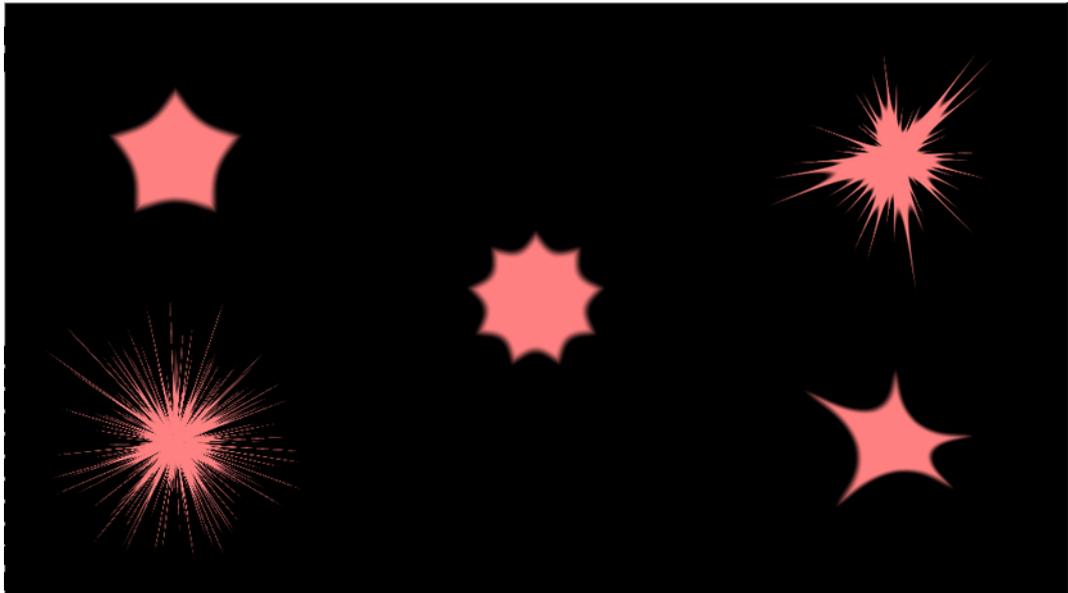
The following lens flare plugins are simply different shapes for the composition of final lens flare effects. Each has its own parameters to define its appearance.

- pxLensGlowBall
- pxLensGlowSpikes
- pxLensPolyElement
- pxLensRandomPoly
- pxLensSpark
- pxLensStripes



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 plug-ins is to

simply play around with the parameter configurations. The following example shows varying manifestations of the pxLensRandomPoly.



Note: This plugin is located in: Built-ins -> Geom Plugins -> PixelFX

See Also

- [PixelFXLenseFlare](#)

11.2.3 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.

Note: This plugin is located in: Built-ins -> Geom Plugins -> PixelFX

See Also

- [PixelFXLenseFlare](#)

11.3 Primitives

The following Geom plug-ins are located in the Primitives folder:

- [N Quad](#)

11.3.1 N Quad



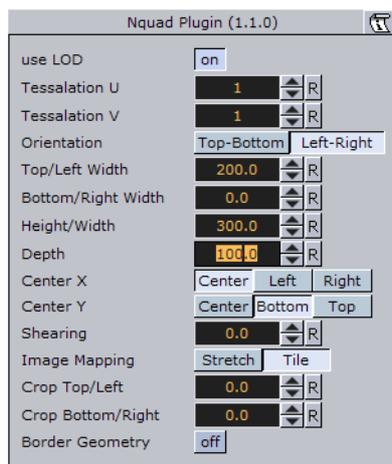
Use this plugin to create a rectangle with different attributes.

Note: This plugin is located in: Built-ins -> Geom Plugins -> Primitives

This section contains the following topics:

- [N Quad Properties](#)
- [To create an N Quad](#)

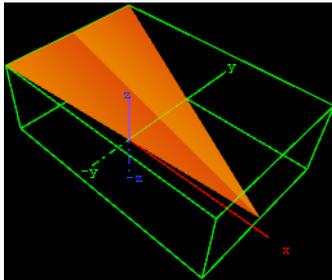
N Quad Properties



- **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 plugin 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.

11.4 RealFX Plugins

The RealFX plugin set enables you to create particle effects in Viz Artist.

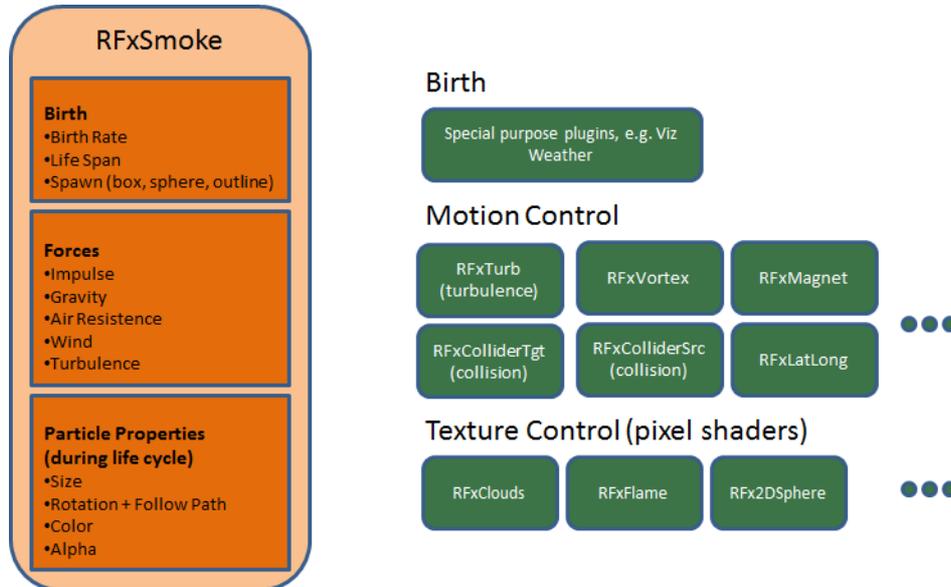
Particle systems are a computer graphics technique to simulate certain physics-based effects, which are otherwise very hard to reproduce with conventional rendering techniques. Examples of such effects which are commonly replicated using particle systems include fire, explosions, smoke, weather effects, sparks, falling leaves, dust, meteor tails, or abstract visual effects like glowing trails, magic spells, etc.

The particle effects in Viz Artist/Engine 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 plug-ins 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 plug-ins 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 plug-ins:** affect the texture mapping and the “look” of each particle by using pixel shader technology



The following Geom plug-ins are located in the RealFX folder:

- [Cloth](#)
- [RFXSmoke](#)

See Also

- [RealFX](#) in [Shader Plugins](#)
- [pxColorWorks](#) in [Container Plugins](#)

11.4.1 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 plugin to a container will also add the [Expert](#) plugin.

Note: This plugin is located in: Built-ins -> Geom Plugins -> RealFX

This section contains the following topics:

- [Cloth Properties](#)
- [To Create a Cloth Effect](#)

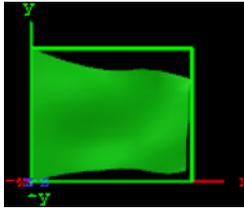
Cloth Properties



- **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** plugin 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](#)

11.4.2 RFXSmoke



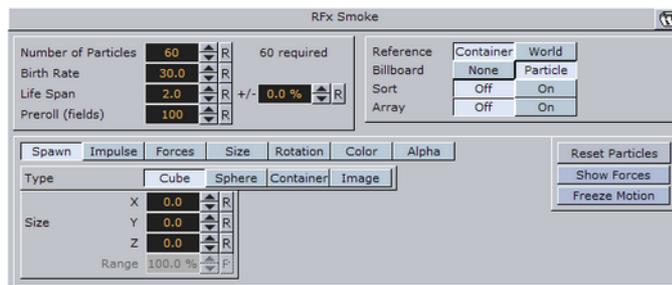
RFXSmoke is the baseline plugin within the RealFX plugin set. It allows you to create realistic visualization of smoke, fire, explosions and much more (the level of creative skill is the only limitation). The basic function of the object is that it emits particles from a point and moves them in some direction until they die or fade away. By using different texture mappings and settings many effects can be achieved.

Note: This plugin is located in: Built-ins -> Geom Plugins -> RealFX

This section contains the following topics:

- [RFXSmoke Properties](#)
- [Spawn](#)
- [Impulse](#)
- [Forces](#)
- [Size](#)
- [Rotation](#)
- [Color](#)
- [Alpha](#)

RFxSmoke Properties



- **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 shows “help lines” that shows the forces settings of some of the parameters.
- **Freeze Motion:** Halts the emitting process.

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 [Create Animations](#) .

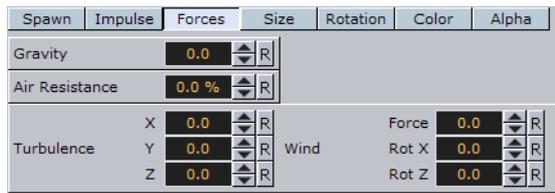
Impulse



Opens a sub section where you can set parameters on how particles are emitted.

- **Speed:** Sets the speed the particles have when they 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 angles 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.

Forces



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

Size



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.

Rotation

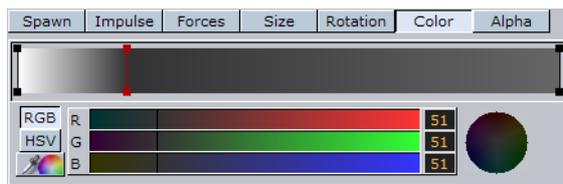


Opens a section where you can define a static rotated position or a spin of the particles. Both the static rotated position and the spin can either be set as an offset

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.

Color

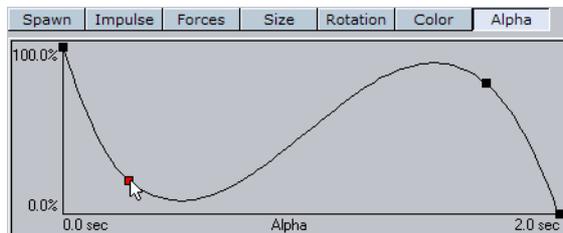


Allows you to set the color of the particles throughout their life cycle. The particles can be set to change their color through stages from their birth to their death by adding more Key Frames (single click the color stage).

The left side of the color stage is at birth time and the right is at death time. By default there are two color Key Frames at each end that you can click on to set their color. The color range between two Key Frames will be an interpolated transformation. More Key Frames can be added to create a more complex color transformation between particles birth and death.

Delete color Key Frames by selecting them and pressing the <Delete> or <Backspace> button on the keyboard.

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 required X and Y position. Delete a point by selecting it and pressing the <Delete> button on the keyboard.

11.5 Ticker

The following Geometry plugins are located in the Ticker folder:

- [Scroller](#)

11.5.1 Scroller



The Scroller plugin is a geometry plugin, mostly used for creating scrolling tickers, that generates a dynamic line of scrolling elements.

When used with Viz Ticker the scene is designed as a transition logic scene where each data element is an instance of a scene template designed as part of the background scene. This is unlike traditional transition logic scenes where each data element would be based on a specific foreground scene controlled by the background scene. Based on the scene template that resides with the background scene the Scroller is able to construct and configure the scene templates on demand for Viz Ticker that manages and reuse them.

To design a scrolling carousel, the Scroller and one or more scene templates must be added to the background scene. The scene templates serve as prototypes for the containers that the Scroller plugin generates. Alignment of the generated containers is done by editing the alignment of the scene templates.

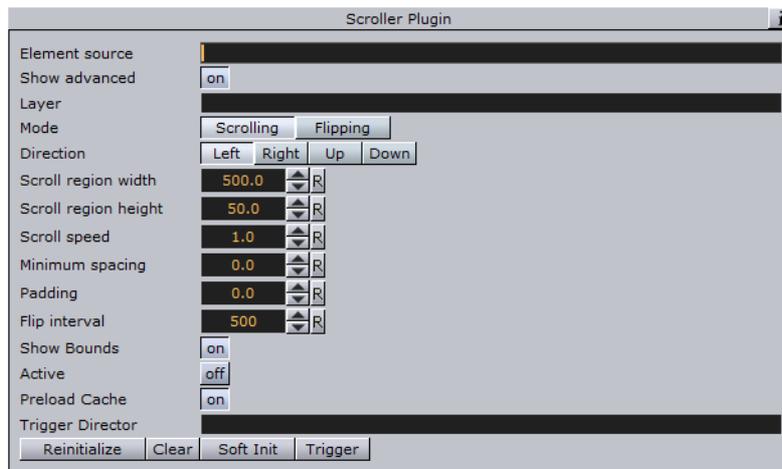
Note: Before saving a Scroller scene, remember to hide the scene template's containers, otherwise the scene template will show on-screen.

Note: This plugin is located in: *Built-ins -> Geom Plugins -> Ticker*

This section also contains information on the following topics:

- [Scroller Properties](#)
- [Global Conventions](#)
- [Scrolling Carousel Conventions](#)
- [Flipping Carousel Conventions](#)
- [Logo Animations as Flipping Carousels](#)

Scroller Properties



- **Element Source:** Element descriptions are added here. Either descriptions are entered in the element source syntax, or they are put in the name of a Shared Memory segment that uses the Shared Memory protocol. When designing and testing a scene, the element source syntax is used to make the system scroll instances of the design containers.
- **Show Advanced:** Shows additional parameters.
- **Layer:** Defines the layer name for the scroller. The layer name must be identical to the name of the director that holds the in and out stop points for the carousel.
- **Mode:** Defines if elements should be shown as scrolling or flipping elements for scrolling and flipping carousels, respectively.
- **Direction:** Defines the direction of the scroller.
- **Scroll Region Width:** Defines the width of the scroller. This option is only relevant for left-to-right/right-to-left scrollers.
- **Scroll Region Height:** Defines the height of the scroller. This option is only relevant for up-down/down-up scrollers.
- **Scroll Speed:** Defines the speed of the scroller. The x position of the scrolling elements will be decremented with this value for each frame that passes.
- **Padding:** Defines an extra padding value that will be added to the width of the bounding box of an element when calculating the spacing between the elements. Negative padding can be used to make elements overlap.
- **Show Bounds:** If enabled, the area in which the scroller tries to maintain the illusion of a continuously scrolling flow of elements is visible on-screen.
- **Active:** Scrolling or flipping is available when enabled.
- **Preload Cache:** Use this on machines without ringbuffer to move all spike-load related to element creation to the initialization phase. Designs named `DESIGN_whatever_Xnnn` will get `nnn` (where `nnn` is an integer) pre-created instances.
- **Reinitialize:** Reinitialize the plugin by deleting all cached objects and reconstructing the initial node structure. This parameter can be used to update the scroller after changing for example the element source.
- **Clear:** Clears all current ticker elements in the rendering window.
- **Trigger:** Triggers the director named in the Trigger Director field provided that there are elements in queue to be scrolled in. If no director is specified then the scroller is activated.

Global Conventions

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

- **Control Plugins:** Elements that should be modified in Viz Ticker Client must have control plug-ins attached. The [Control Object](#) plugin must be added to the template root container, while additional [Control](#) plug-ins should be added to Sub-Containers.

Tip: If text should be edited add the [Control Text](#) plugin to the text container.

- **States:** Carousels must have one out state, and at least one in state. The in state means that the graphics is visible on-screen, while the out state means that the graphics is positioned outside the visual bounds.

The scene can have multiple stop points that define the out state, but they must all be named O, as there is only one out state.

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

A ticker with Arabic text should typically be right centered, as Arabic is a right-to-left language.

Scrolling Carousel Conventions

The following design conventions apply only for scrolling carousel scenes.

Scrolling carousel scenes are characterized by containing one or more Scroller plugins with belonging directors and stop points.

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

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

Flipping Carousel Conventions

The following design conventions apply only for flipping carousel scenes.

Flipping carousel scenes are characterized by containing one or more [Toggle](#) plug-ins with belonging directors and stop points.

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

- The background scene and the scene templates (foreground scenes) must be built as separate scenes.
- All flipping carousel template scenes must have a root container named *object*.
- The container holding the [Toggle](#) plugin must have the same name as the director holding the in and out stop points.
- In the [Control Object](#) editor for each foreground scene (scene template), the Layer Identifier must be identical to the director holding the in and out stop points in the background scene.
- Also in the [Control Object](#) editor in the foreground scene, the *State Identifier* must be identical to an in state stop point of the layer identifier director.

Logo Animations as Flipping Carousels

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

See Also

- [Control](#) plug-ins
- [Toggle](#) plugin
- [Transition Logic](#)

11.6 Visual Data Tools

The following Geometry plugins are located in the Visual 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](#)

11.6.1 Overview

Visual Data Tools (VDT) were designed for Viz Artist to show any kind of statistic data in the best way. All plug-ins use the same transport channel: 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 three types of Shared Memory Maps, see [Data Sharing](#).

For more information about Shared Memory see [Shared Memory \(SHM\)](#).

The VDT plugin suite consists of [Geometry Plugins](#) and [Container Plugins](#). The geometry part is responsible for drawing the chart itself while the container plug-ins are used for chart labeling, data scaling, etc.

To learn more about the use of Visual Data Tools download a tutorial on www.vizrt.com under the Training section.

See Also

- [Container Plugins](#)
- [Geometry Plugins](#)
- [Scripting](#)
- [Scripting Data Types and Procedures](#)

11.6.2 Area Chart



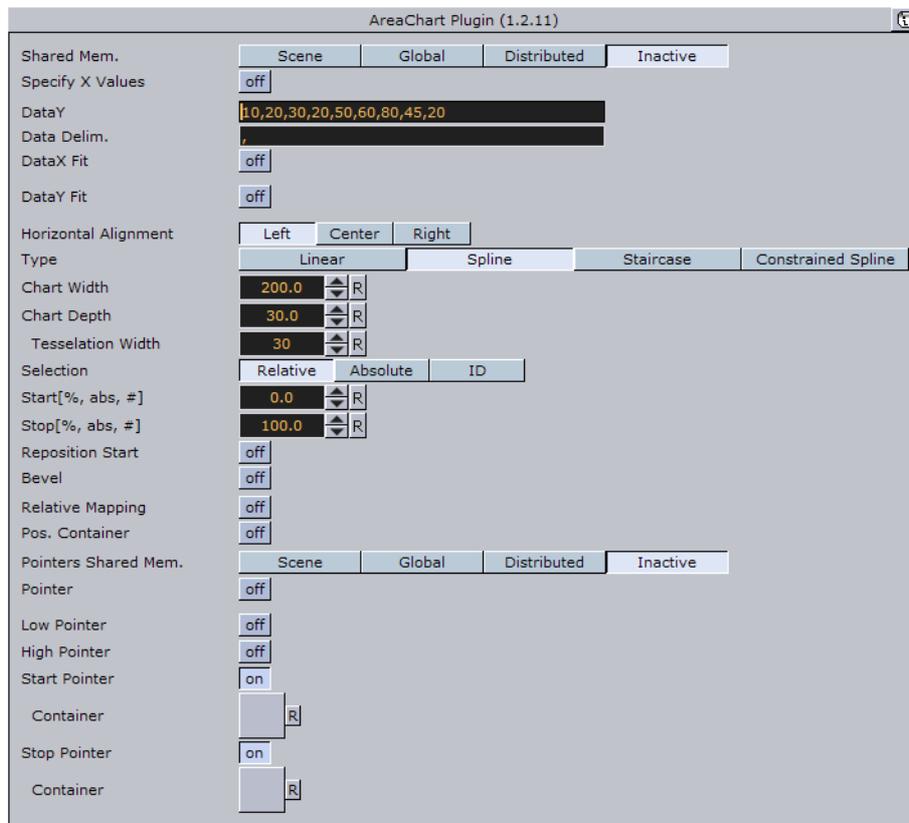
This plugin draws an area chart, filled with data out of a Shared Memory Map. You can use delimited strings or arrays for data transfer via Scene-, Global- or Distributed-Shared Memory.

Note: This plugin is located in: Built-ins -> Geom Plugins -> VisualDataTools

This section contains the following topics:

- [Area Chart Properties](#)
- [To Create a Scene with Area Chart](#)

Area Chart Properties



- **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.

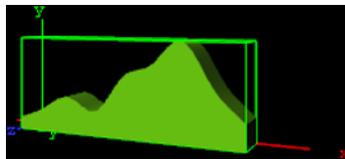
- **DataX, -Y Stop:** Upper Auto Scale edge.
- **Horizontal Alignment:** Sets horizontal orientation to left, center or right.
- **Type:** Creates different graph looks. Available options are Normal, Spline, Staircase and Constrained Spline.
 - **Normal:** Enables direct connections between entered values.
 - **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 with Area Chart



1. Create a new group and add the Area Chart plugin 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 plugin 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 required sign.

See Also

- [Control Chart](#)
- [Data Fit](#)
- [Data Import](#)
- [Data Label](#)
- [Data Storage](#)

11.6.3 Bar Chart



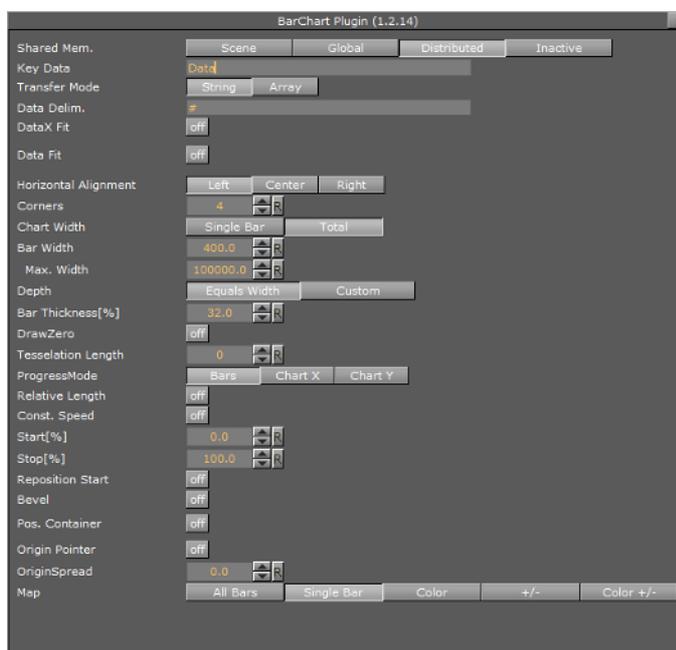
This plugin draws a bar chart, filled with data out of a Shared Memory Map. You can use delimited strings or arrays for data transfer via Scene-, Global- or Distributed-Shared Memory.

Note: This plugin is located in: Built-ins -> Geom Plugins -> VisualDataTools

This section contains the following topics:

- [Bar Chart Properties](#)
- [To create a bar chart](#)

Bar Chart Properties



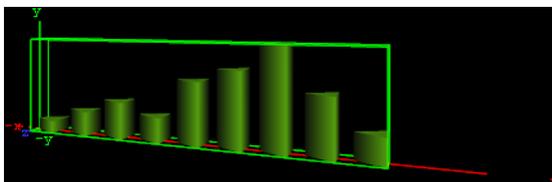
- **Shared Mem.:** Changes between Scene-, Global- and Distributed-Shared Memory. Use Inactive memory to not forward any values via Shared Memory.
- **Data:** Shared Mem. key name.
- **Data:** Input parameter for data.
- **Transfer Mode:** Sets string- or array-based data transfer.
- **Data Delim.:** 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 X:** The chart grows horizontally. Each bar comes in at its final height.
 - **Chart Y:** The chart grows vertically. All bars grow together to 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 they 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 plugin 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](#)

11.6.4 Line Chart



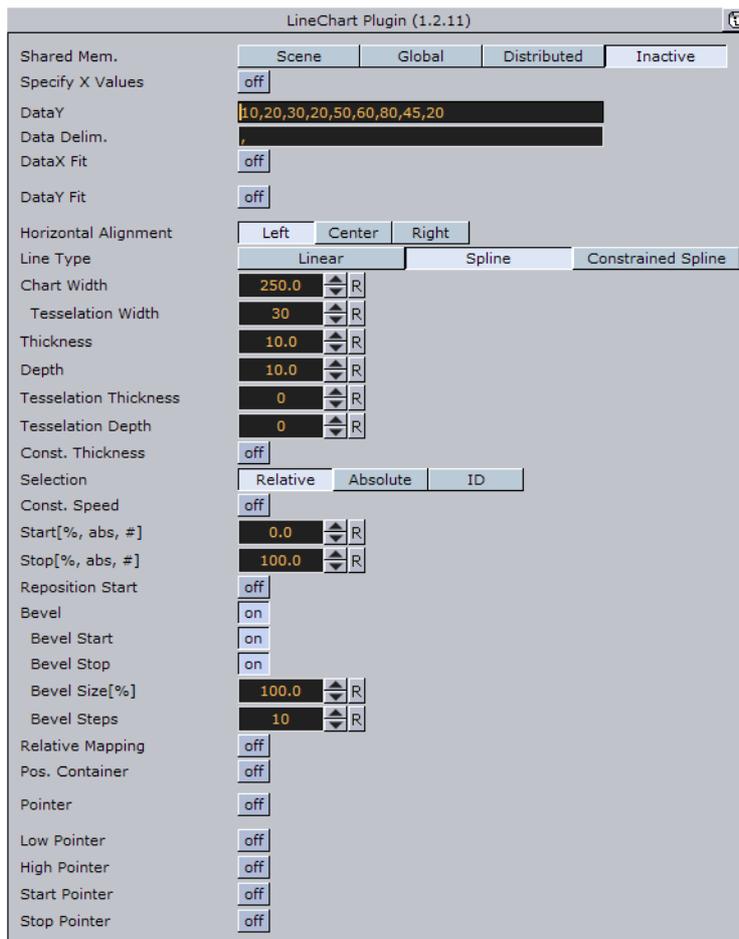
This plugin draws a line chart, filled with data out of a Shared Memory Map. You can use delimited strings or arrays for data transfer via Scene-, Global- or Distributed-Shared Memory.

Note: This plugin is located in: Built-ins -> Geom Plugins -> VisualDataTools

This section contains the following topics:

- [Line Chart Properties](#)
- [To create a line chart](#)

Line Chart Properties



- **Shared Mem.:** Changes between Scene-, Global- and Distributed-Shared Memory. Use Inactive memory to not forward any values via Shared Memory.

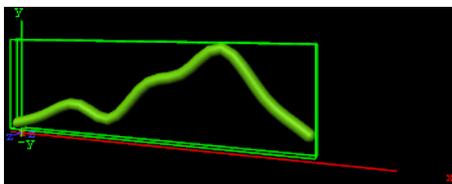
- **Specify X Values:** Enables DataX input.
 - **DataX:** Shared Mem. key name for X values. DataX is the default input parameter for X values.
 - **Key DataX:** Shared Mem. key name for X values.
- **DataY:** Shared Mem. key name for Y values. DataY is the default input parameter for Y values.
- **Key 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 plugin 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 **Shared Mem.** to **Inactive**
- Set the following **DataY** values: 10 , 20 , 30 , 20 , 50 , 60 , 80 , 45 , 20
- Set **Line Type** to **Spline**
- Set **Chart Width** 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](#)

11.6.5 Pie Chart



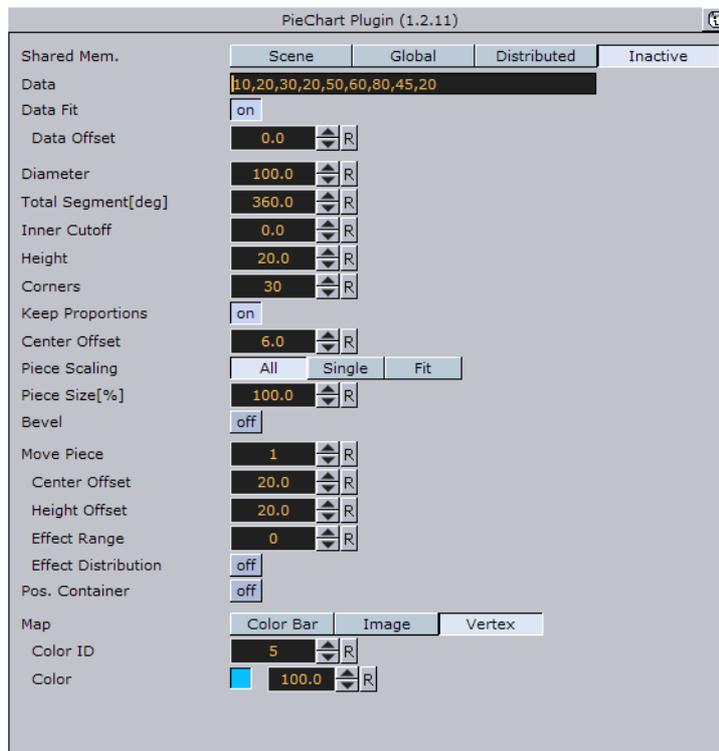
This plugin draws a pie chart, filled with data out of a Shared Memory Map. You can use delimited strings or arrays for data transfer via Scene-, Global- or Distributed-Shared Memory.

Note: This plugin is located in: Built-ins -> Geom Plugins -> VisualDataTools

This section contains the following topics:

- [Pie Chart Properties](#)
- [To Create a Pie Chart](#)

Pie Chart Properties

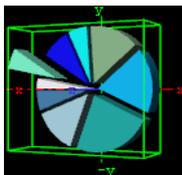


- **Shared Mem.:** Changes between Scene-, Global- and Distributed-Shared Memory. Use Inactive memory to not forward any values via Shared Memory.
- **Data:** Shared Mem. key name. Data is the default input parameter.
- **Transfer Mode:** Sets string- or array-based data transfer.
- **Data Delim.:** 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 plugin 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](#)

11.6.6 Scatter Chart



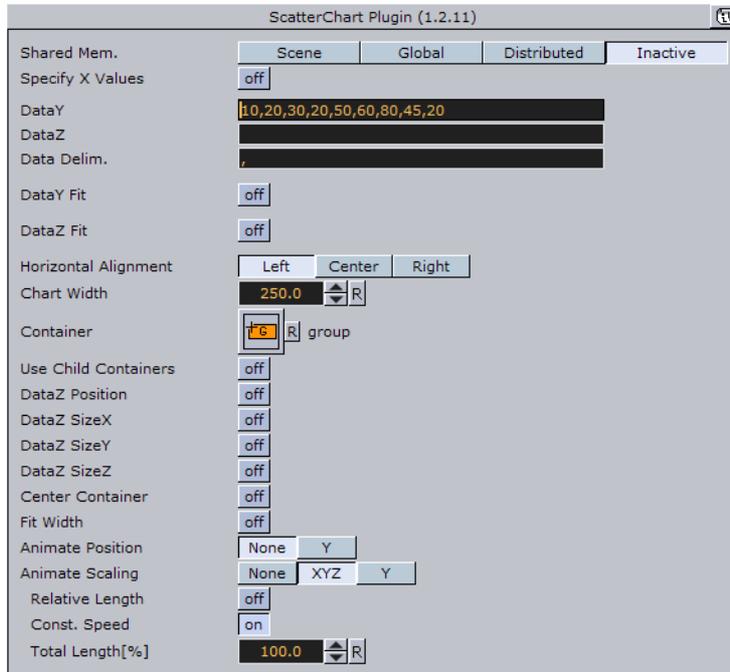
This plugin draws a scatter chart, filled with data out of a Shared Memory Map. You can use delimited strings or arrays for data transfer via Scene-, Global- or Distributed-Shared Memory.

Note: This plugin is located in: Built-ins -> Geom Plugins -> VisualDataTools

This section contains the following topics:

- [Scatter Chart Properties](#)
- [To create a scatter chart](#)

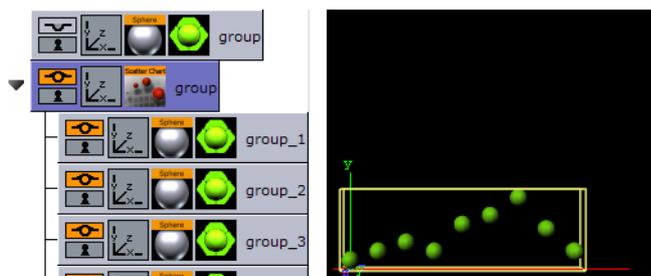
Scatter Chart Properties



- **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](#) plugin 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 plugin 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](#)

11.6.7 Stock Chart



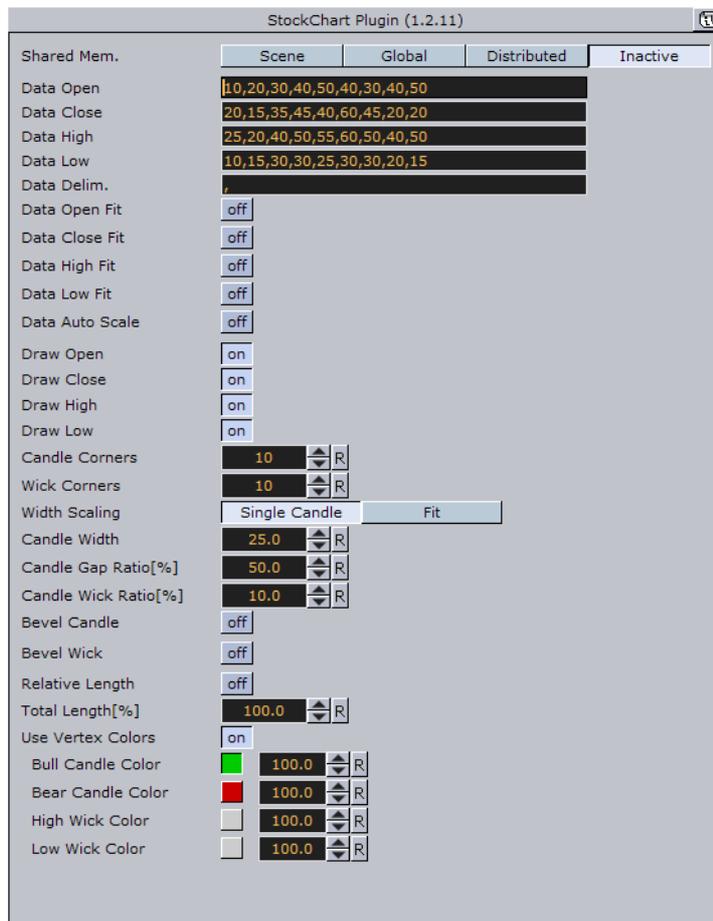
This plugin draws a stock chart, filled with data out of a Shared Memory Map. You can use delimited strings or arrays for data transfer via Scene-, Global- or Distributed-Shared Memory.

Note: This plugin is located in: Built-ins -> Geom Plugins -> VisualDataTools

This section contains the following topics:

- [Stock Chart Properties](#)
- [To Create a Stock Chart](#)

Stock Chart Properties



- **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



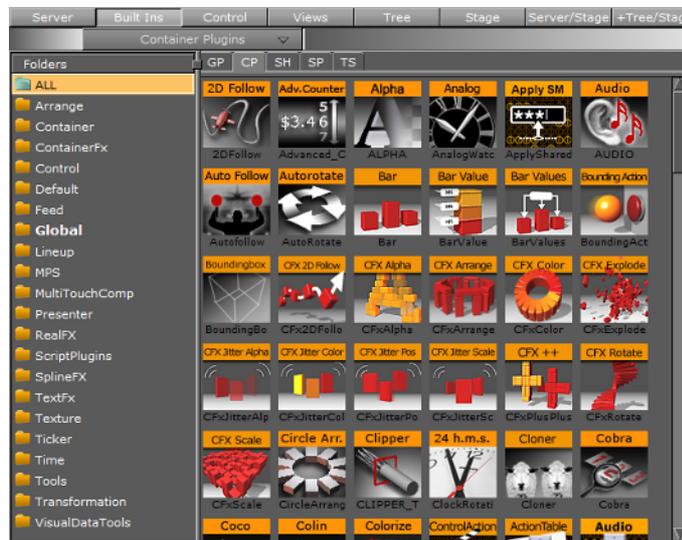
1. Create a new group and add the Stock Chart plugin to it.
2. Open the [Transformation Editor](#) and set Position X and Y to -100.0.

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 Low 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



All Container plugins are located in these folders:

Arrange	pxColorWorks
Container	RealFX
Container FX	Script Plugins
Control	SplineFX
Default	TextFX
Feed	Texture
Global	Ticker
Multi-Touch Plugins	Tools
PixelFX	Transformation
Presenter	Visual Data Tools

Note: The default path for Container plugins is: <viz install folder>\plugin\<plugin name.vip>

12.1 Arrange

These Arrange Container plugins are located in the Arrange folder:

- [Circle Arrange](#)
- [Grid Arrange](#)
- [Time Displacement](#)

12.1.1 Circle Arrange



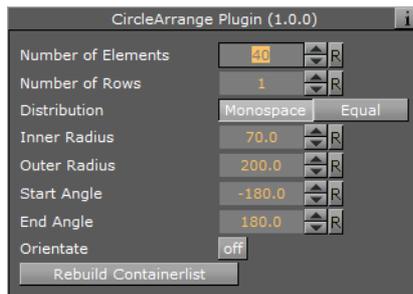
Circle Arrange is a tool for arranging a set of containers into a circular structure.

Note: This plugin is located in: *Built-ins -> Container Plugins -> Arrange*

This section contains information on the following topics:

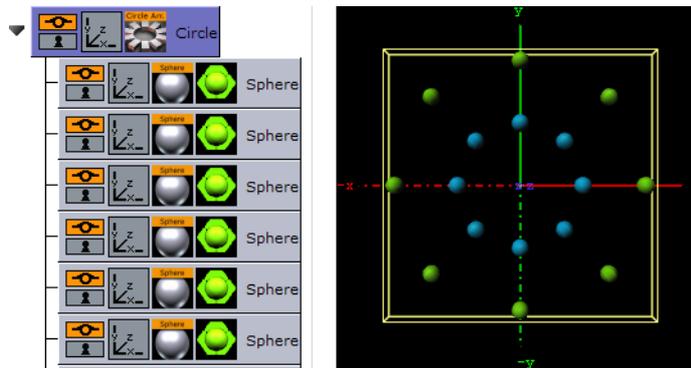
- [Circle Arrange Properties](#)
- [To Arrange Containers in a Circle](#)

Circle Arrange Properties



- **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 plugin 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 plugin 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.

Note: This plugin is located in: *Built-ins -> Container Plugins -> Arrange*

This section contains information on the following topics:

- [Grid Arrange Properties](#)
- [To Arrange Containers in a Grid](#)

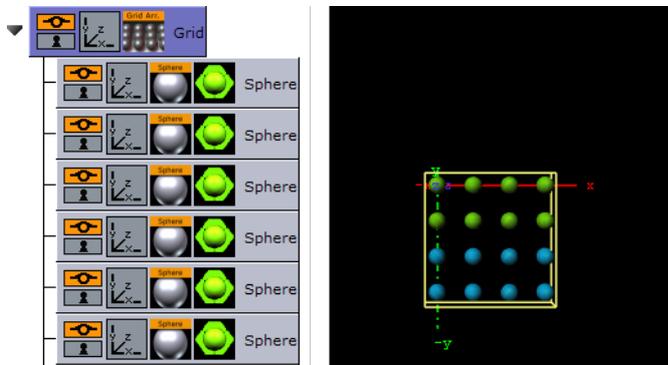
Grid Arrange Properties



- **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.
- **Start In Corner:** Select which corner to start the arrange

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. Add container to the Scene Tree
2. Name it **Grid**.
3. Add a **Sphere** as a Sub-Container to the **Grid** container.
4. Add a material and/or a texture to the Sphere container.
5. Open the [Transformation Editor](#) for the Sphere container.
6. Set **Scaling** (locked) to **0.2**.
7. *Optional:* Animate the Sphere object.
8. Create 15 copies of the Sub-Container (total of 16 Sphere containers). All copies should be placed as Sub-Containers of the Grid container and have the same position as the original.
9. Add the Grid Arrange plugin to the **Grid** container.
10. Open the Grid Arrange editor and set these parameters:

- **Number of Rows:** 4
- **Number of Columns:** 4
- **Row Offset:** 40.0
- **Column Offset:** 40.0

The plugin will position all Sub-Containers 4in 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 grayscale 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.

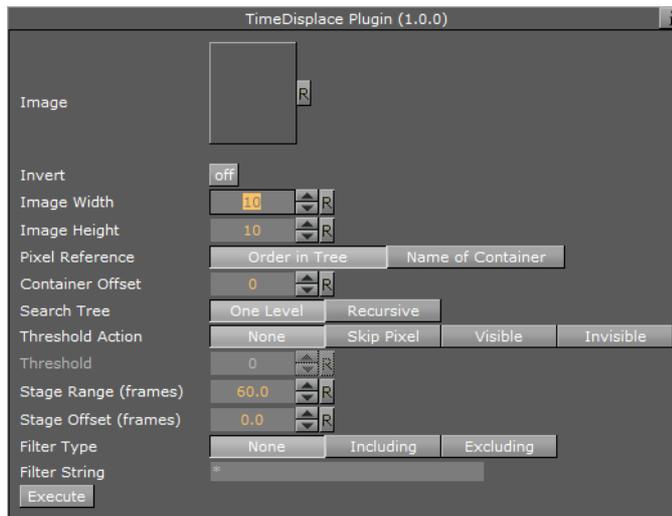
What the function does is to create links between the animated containers and the pixels of the grayscale image and then use the alpha level of each pixel to set the offset of the containers. The range from 100% alpha to 0% alpha on the pixels sets the offset level. By default 100% alpha sets no offset and 0% sets maximum offset.

Note: This plugin is located in: Built-ins -> Container Plugins -> Arrange

This section contains information on the following topics:

- [Time Displacement Properties](#)
- [To Set Animation Time Offsets with a Grayscaled Alpha Image](#)

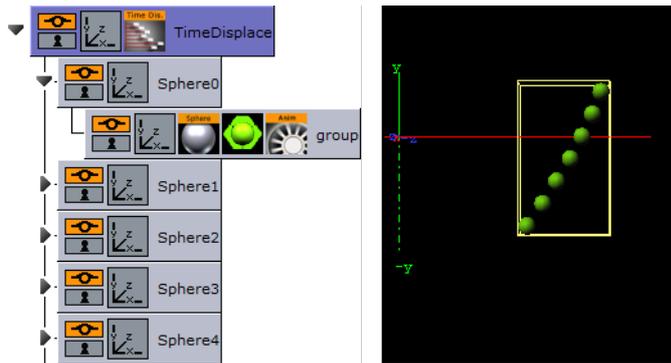
Time Displacement Properties



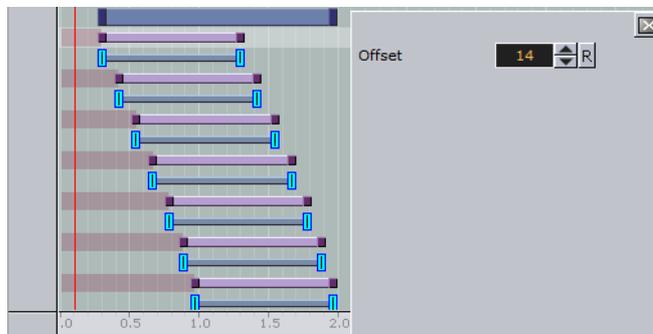
- **Image:** Drag the alpha image you want to use for creating the containers offsets onto the drop zone here.
- **Invert:** Inverts the link between the image pixels and offsets. 0% gives no offset and 100% gives maximum.
- **Image Width:** Sets the height of the alpha image.
- **Image Height:** Sets 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 with a Grayscaled Alpha Image



1. Add a group to the scene tree and name it TimeDisplace, and the TimeDisplace plugin 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 plugin 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](#) plugin 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 plugin.
10. Add the [Corena](#) plugin 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 plugin.
12. Open the TimeDisplace editor and add an alpha image to the Image drop zone.
13. Click Execute.



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

These Container plugins are located in the Container folder:

- [BoundingBox](#)
- [Cobra](#)
- [Coco](#)
- [Colin](#)
- [Cora](#)
- [Corena](#)
- [Toggle](#)

12.2.1 BoundingBox



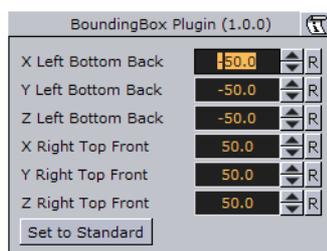
Use this plugin to redefine the standard Bounding Box which is bounding all geometry objects.

Note: This plugin is located in: Built-ins -> Container Plugins -> Container

This section contains information on the following topics:

- [BoundingBox Properties](#)

BoundingBox Properties



- **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



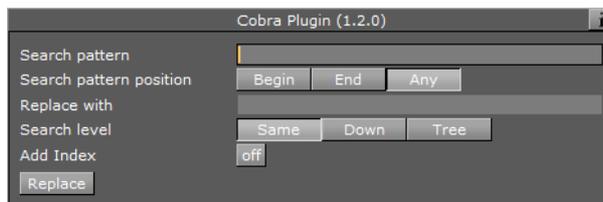
The Cobra function is an easy to use function replace (parts) of container names. Enter a search pattern and a string the search should be replaced with and execute.

Note: This plugin is located in: Built-ins -> Container Plugins -> Container

This section contains information on the following topics:

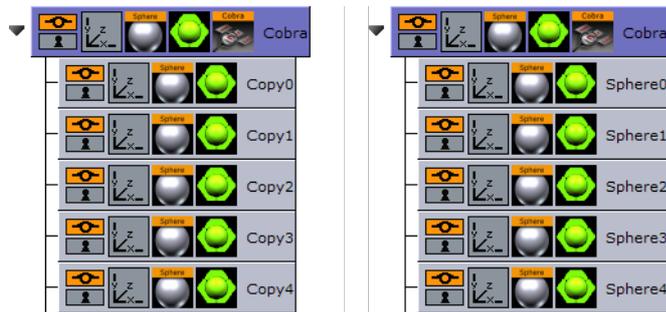
- [Cobra Properties](#)
- [To Rename Container\(s\)](#)

Cobra Properties



- **Search pattern:** Set the text string to be replaced.
- **Search pattern position:** Define the position within the string. It can be set to Begin, End and Any.
- **Replace with:** Sets the text string to be used instead of the search pattern.
- **Search level:**
 - **Same:** Only containers on the same hierarchy level will be affected.
 - **Down:** Only Sub-Containers of the current container will be affected.
 - **Tree:** Search the whole Scene Tree.
- **Add Index:** Set to **On** to add a numerical index to the new name
 - **Starting Index:** Set the number to start the index
 - **Reverse Index:** Set to **On** to use a reverse index list
- **Replace:** Starts the replace process.

To Rename Container(s)



1. Add the Cobra plugin to a container that, for example, has Sub-Containers that need be renamed.
2. Open the Cobra editor.
3. Enter a **Search pattern** text string (e.g. Copy)
4. Set the **Search pattern position** to **Begin**.
5. Enter the **Replace with** text string (e.g. Sphere).
6. Set the **Search level** to **Down**.
7. Click **Replace**.

See Also

- [Coco](#)
- [Corena](#)

12.2.3 Coco



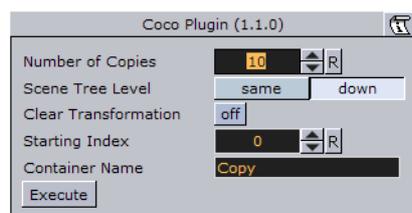
The Coco plugin allows you to create copies of a container with great ease. This function is typically used together with the [Arrange](#) plugins.

Note: This plugin is located in: Built-ins -> Container Plugins -> Container

This section contains information on the following topics:

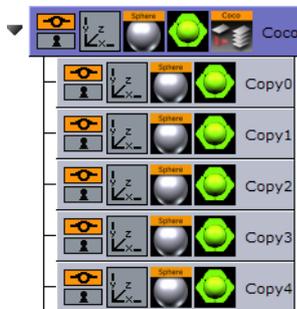
- [Coco Properties](#)
- [To Create Copies of Containers](#)

Coco Properties



- **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 plugin 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



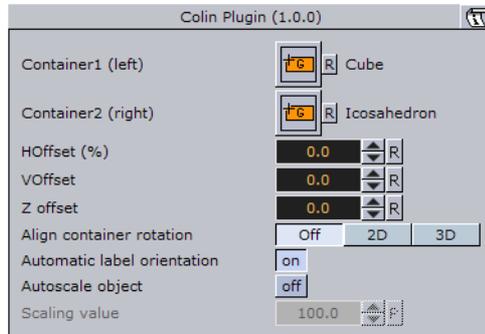
The Colin plugin allows you to align a container in the scene using two other containers as reference. This makes it easy to maintain a good and controlled symmetry in the scene.

Note: This plugin is located in: Built-ins -> Container Plugins -> Container

This section contains information on the following topics:

- [Colin Properties](#)
- [To Align Containers](#)

Colin Properties



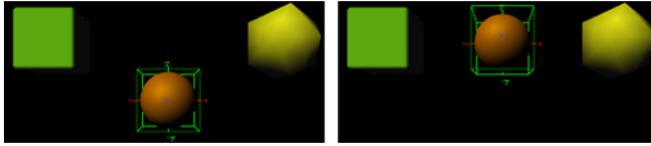
- **Container 1:** Is the drop zone for the first referential container.
- **Container 2:** Is the drop zone for the second referential container.
- **HOffset %:** Sets the horizontal offset between the referential containers in percent. Horizontal means here an invisible line between the referential 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 plugin 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 plugin 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.

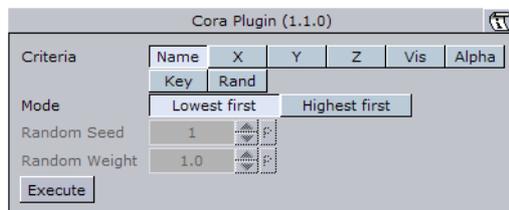
The plugin is often used together with the [Time Displacement](#) plugin which enables many creative possibilities.

Note: This plugin is located in: Built-ins -> Container Plugins -> Container

This section contains information on the following topics:

- [Cora Properties](#)
- [To organize the scene tree](#)

Cora Properties



- **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 plugin 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



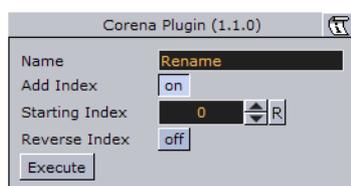
The Corena plugin renames all Sub-Containers of the container you apply it to. The plugin is often used together with the [Time Displacement](#) plugin which enables many creative possibilities.

Note: This plugin is located in: Built-ins -> Container Plugins -> Container

This section contains information on the following topics:

- [Corena Properties](#)
- [To Rename Container\(s\)](#)

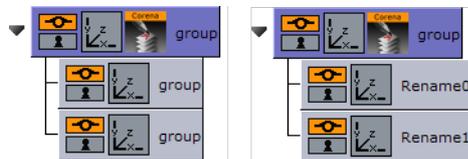
Corena Properties



- **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 plugin 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 plugin is most commonly used with [Transition Logic](#) scenes.

Note: This plugin is located in: Built-ins -> Container Plugins -> Container

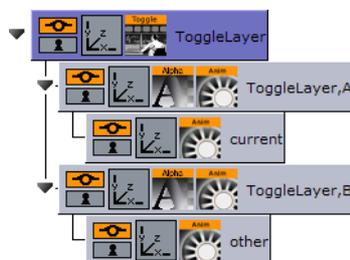
This section contains information on the following topics:

- [Description and Methodological Considerations](#)
- [Toggle Properties](#)

Description and Methodological Considerations

Toggle plugin is one of the main building blocks of [Transition Logic](#), and it is in general a relatively simple plugin. When the Toggle plugin 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.

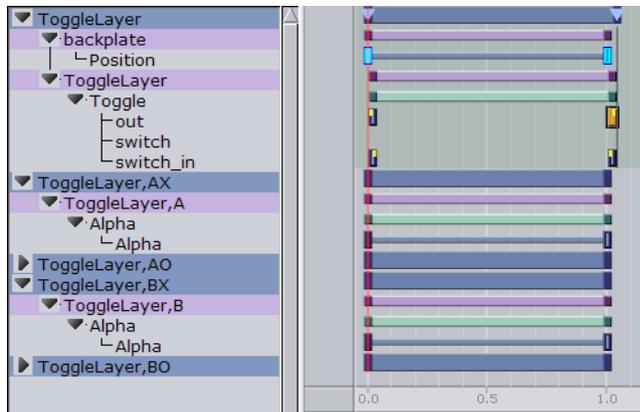
Toggle placeholder containers



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 plugin to keep track of this.

Toggle directors and Key Frames



The Toggle plugin 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.

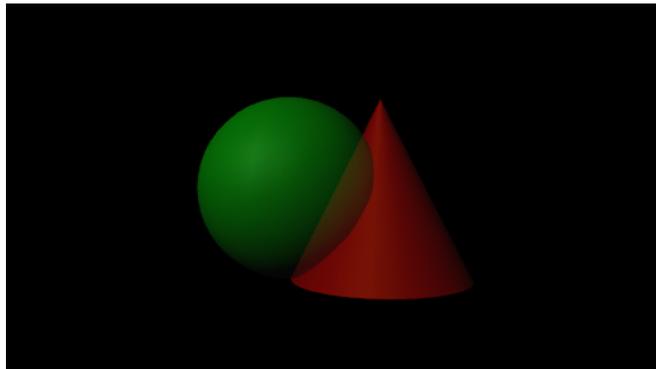
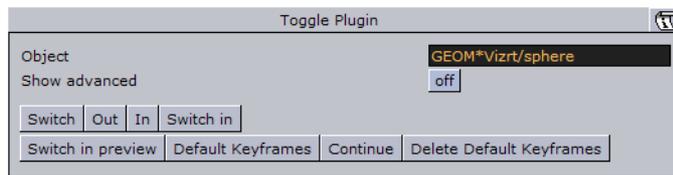
In addition to the cross fade directors the Toggle plugin will also create placeholder directors for the *object* animations. Merged objects that are loaded into these placeholder containers during playback may contain animations embedded in the geometry object. The object animations are automatically 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 required next state. In this scenario the utility directors will be triggered according to the toggle Key Frames in the stage. This means that it is the position of these Key Frames 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.



The plugin 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 plugin 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 plugin. Preloading of the objects needed by the Toggle plugin 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 plugin 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 plugin 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 plugin is in the correct position in the scene tree.
2. The container the Toggle plugin 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.

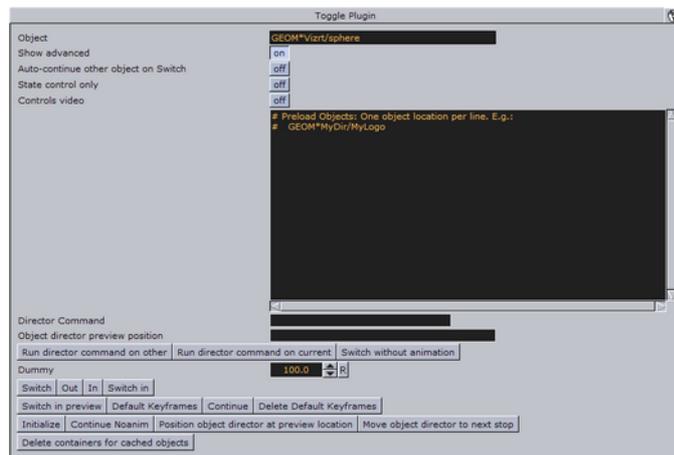
IMPORTANT! One of the [Transition Logic](#) requirements is to have a stop point named "O". Best practice is to always name the first stop point "O" (frame 0).

After the Toggle plugin has been added to the container, the plugin icon can be clicked to see its properties. Clicking the Default Key Frames button will create the most commonly used Key Frames for triggering the toggling between the current and the

other objects. Adjusting the timing can easily be done by moving the Key Frames in the stage.

Toggle Properties

Toggle editor



- **Object:** Shows the name of the currently loaded object (front scene GEOM). This field changes depending on the object loaded in the layer the toggle plugin resides.
- **Show advanced:** Shows 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 means 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 plugin.
 - **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: 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 Key Frames. This will in some cases solve problems with toggle Key Frames not being triggered correctly.

- **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 Key Frames:** Inserts default Toggle Key Frames to invoke the toggles utility directors.
- **Continue:** Sends a continue action to the current object animation director.
- **Delete default Key Frames:** Removes the default Key Frames that were generated by the Default Key Frames 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

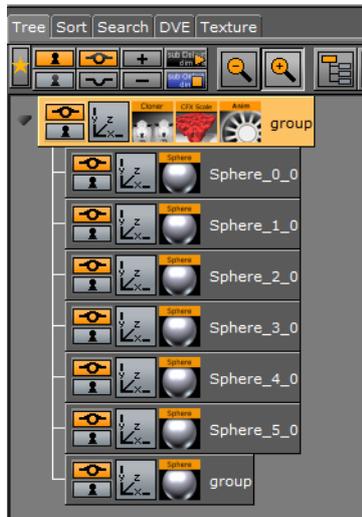
- [Transition Logic](#)

12.3 Container FX

Container FX is much like [TextFX](#), 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.

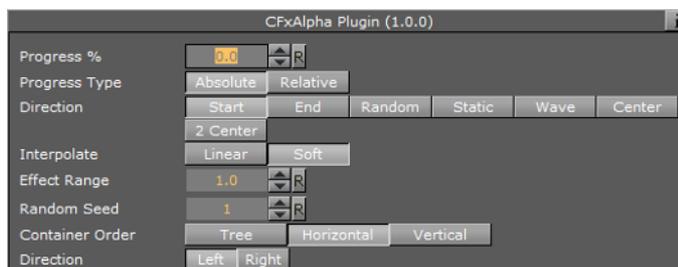


In the above example the containers under the group container with the Sphere names is the container-set, and will be animated here.

These Container FX Container plugins are located in the Container FX folder:

Common Container FX Properties	CFX Jitter Color
CFX 2D Follow	CFX Jitter Position
CFx Alpha	CFX Jitter Scale
CFX Arrange	CFX Plus Plus
CFX Color	CFX Rotate
CFX Explode	CFX Scale
CFX Jitter Alpha	

12.3.1 Common Container FX Properties



The following properties 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, the animation 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 CFX 2D Follow



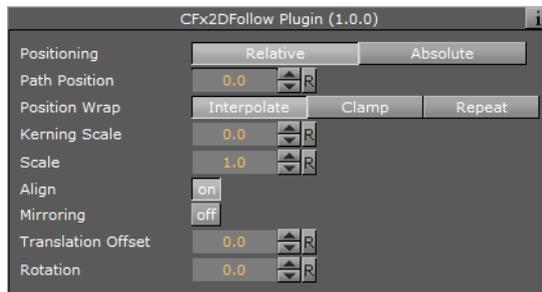
The Container FX 2D Follow plugin allows you to create an animation of the Containers following a path (usually 2DRibbon).

Note: This plugin is located in: Built-ins -> Container Plugins -> ContainerFx

This section contains information on the following topics:

- [CFX 2D Follow Properties](#)

CFX 2D Follow Properties



- **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 CFX Alpha



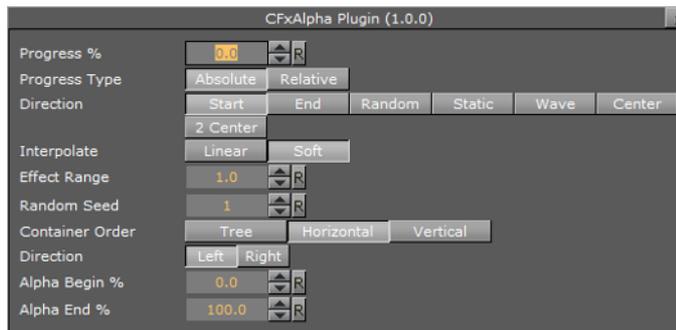
The Container FX Alpha plugin is a container-set effect that creates a fade in effect for the containers in the set. The effect sequence can be set to go many different ways. A fade in and out effect can also be achieved.

Note: This plugin is located in: Built-ins -> Container Plugins -> ContainerFx

This section contains information on the following topics:

- [CFX Alpha Properties](#)

CFx Alpha Properties



- **See:** [Common Container FX Properties](#)
- **Alpha Begin %:** Sets the alpha level of the containers at 0% effect.
- **Alpha End %:** Sets the alpha level of the containers at 100% effect.

12.3.4 CFX Arrange



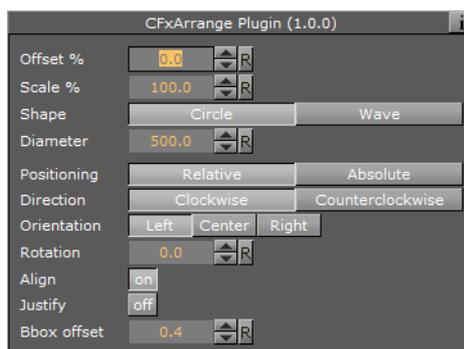
The CFX Arrange plugin arranges containers in either a circular or a wave shape. The containers can be animated on the selected shape by animating the offset value.

Note: This plugin is located in: Built-ins -> Container Plugins -> ContainerFx

This section contains information on the following topics:

- [CFX Arrange Properties](#)

CFX Arrange Properties



- **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 bounding 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.
- **Rotation:** Rotates the container-set on the X-axis.
- **Align:** Aligns the container along the curve.

12.3.5 CFX Color



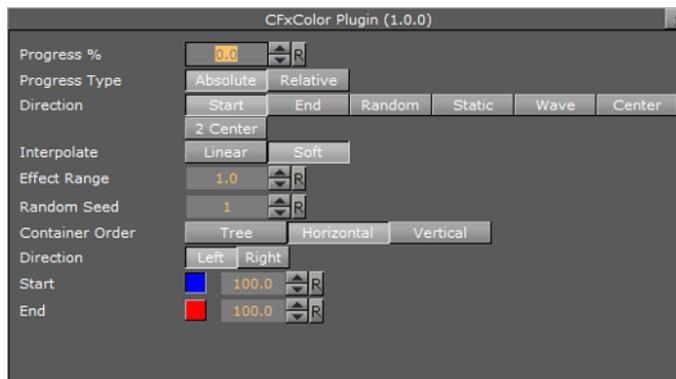
Adds a color effect to the containers-set.

Note: This plugin is located in: Built-ins -> Container Plugins -> ContainerFx

This section contains information on the following topics:

- [CFX Color Properties](#)

CFX Color Properties



- **See:** [Common Container FX Properties](#)
- **Color Start:** Sets the initial color before applying the effect.
- **Color End:** Sets the color the effect should apply.

12.3.6 CFX Explode



The CFX Explode plugin 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.

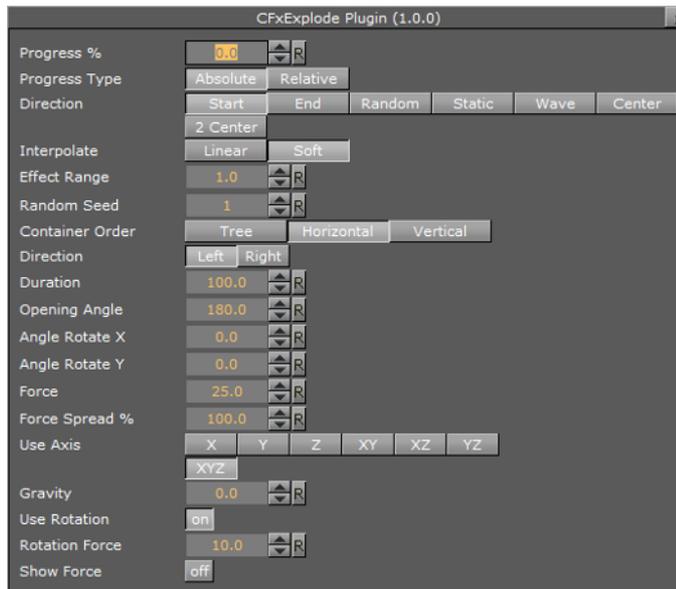
Note: Works only if the containers-set is set to texture

Note: This plugin is located in: Built-ins -> Container Plugins -> ContainerFx

This section contains information on the following topics:

- [CFX Explode Properties](#)

CFX Explode Properties



- **See:** [Common Container FX Properties](#)
- **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 Scene Editor.

12.3.7 CFX Jitter Alpha



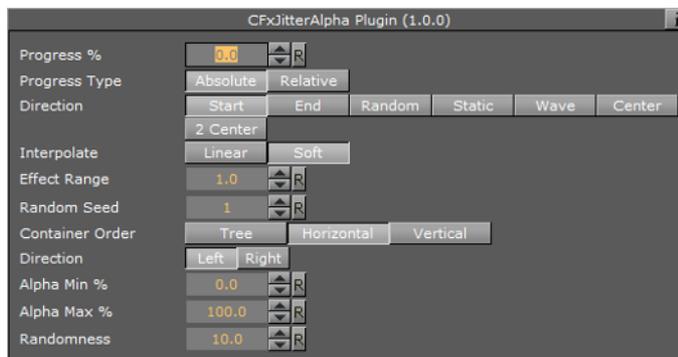
The CFX Jitter Alpha plugin creates a jittering motion of the containers by randomly changing the alpha value of each container. The degree of change and the start sequence of the jittering can be altered. To use the plugin, add it onto a container with a font. To create an animation, animate the progress. Other values can of course be animated as well.

Note: This plugin is located in: Built-ins -> Container Plugins -> ContainerFx

This section contains information on the following topics:

- [CFX Jitter Alpha Properties](#)

CFX Jitter Alpha Properties



- **See:** [Common Container FX Properties](#)
- **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.

12.3.8 CFX Jitter Color



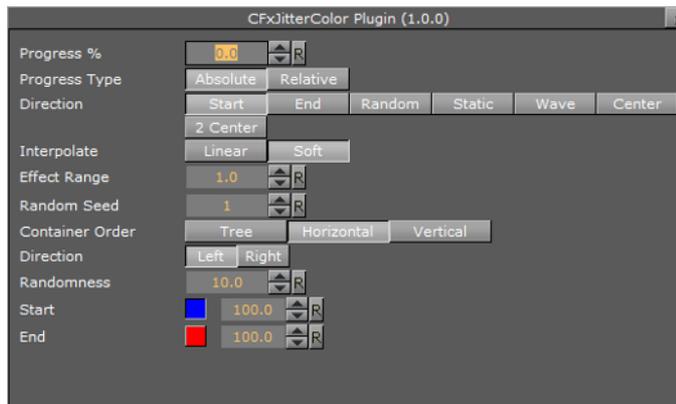
The CFX Jitter Color animates a jittering effect on the color of the container.

Note: This plugin is located in: Built-ins -> Container Plugins -> ContainerFx

This section contains information on the following topics:

- [CFX Jitter Color Properties](#)

CFX Jitter Color Properties



- **See:** [Common Container FX Properties](#)
- **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.

12.3.9 CFX Jitter Position



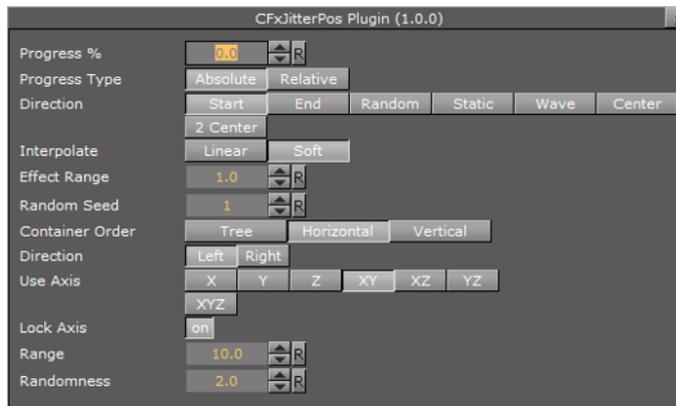
The CFX Jitter Position plugin creates a jittering motion of the containers by randomly changing the position of each container. The degree of position change and the starting sequence of the jittering can be altered. To create an animation, animate the progress. Other values can of course be animated as well.

Note: This plugin is located in: Built-ins -> Container Plugins -> ContainerFx

This section contains information on the following topics:

- [CFX Jitter Position Properties](#)

CFX Jitter Position Properties



- **See:** [Common Container FX Properties](#)
- **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 shown and you can set them individually.
- **Range:** Sets the range of the jittering movement.
- **Randomness:** Sets the intensity of the jittering movement.

12.3.10 CFX Jitter Scale



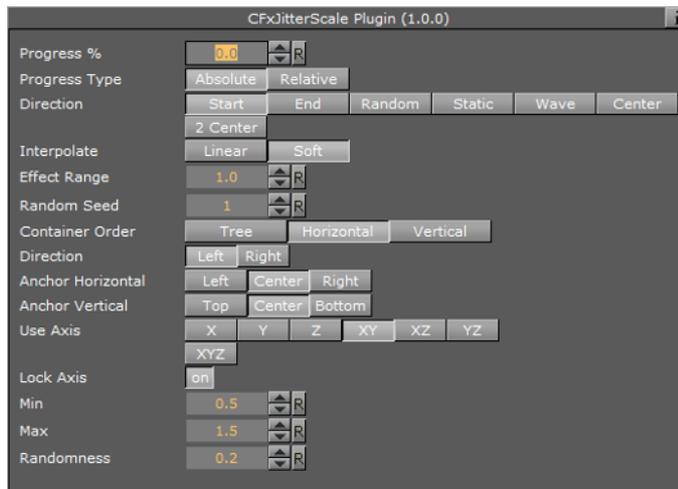
The CFX Jitter Scale animates a jittering effect on the scale of the container.

Note: This plugin is located in: Built-ins -> Container Plugins -> ContainerFx

This section contains information on the following topics:

- [CFX Jitter Scale Properties](#)

CFX Jitter Scale Properties



- **See:** [Common Container FX Properties](#)
- **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.

12.3.11 CFX Plus Plus



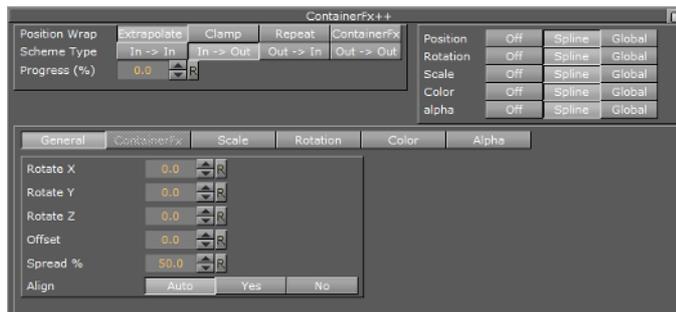
The Container FX Plus Plus plugin allows you to set a number of effects on a container's set.

Note: This plugin is located in: Built-ins -> Container Plugins -> ContainerFx

This section contains information on the following topics:

- [CFX Plus Plus Properties](#)

CFX Plus Plus Properties



- **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).

12.3.12 CFX Rotate



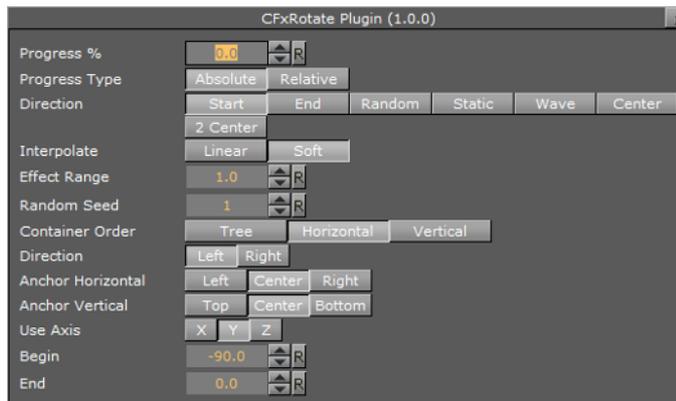
The CFX Rotate plugin allows you to create an effect where the containers rotate on the X-, Y- or Z-axis.

Note: This plugin is located in: Built-ins -> Container Plugins -> ContainerFx

This section contains information on the following topics:

- [CFX Rotate Properties](#)

CFX Rotate Properties



- **See:** [Common Container FX Properties](#)
- **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

- [Expert](#)

12.3.13 CFX Scale



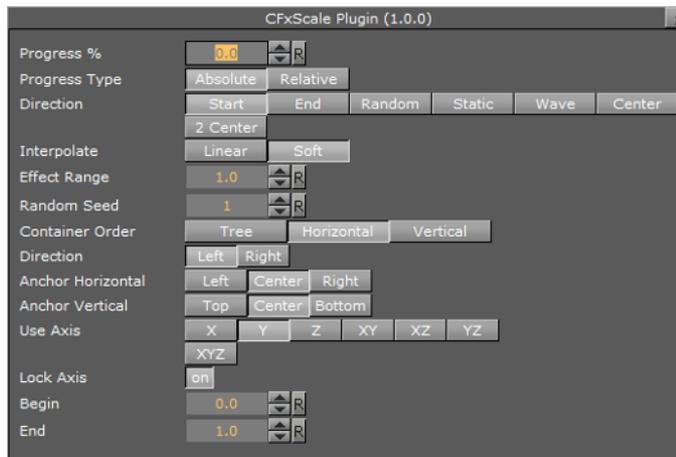
The CFX Scale plugin allows you to create a scaling animation of the containers.

Note: This plugin is located in: Built-ins -> Container Plugins -> ContainerFx

This section contains information on the following topics:

- [CFX Scale Properties](#)

CFX Scale Properties



- **See:** [Common Container FX Properties](#)
- **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.

12.4 Control

The following Container plugins are located in the Control folder:

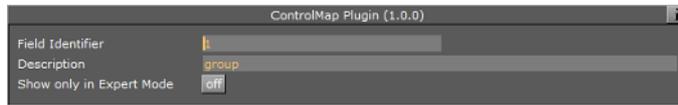
Common Control Plugin Properties	Control Key Frame
Apply Shared Memory	Control List
Control Action	Control Map
Control Action Table	Control Material
Control Audio	Control Num
Control Bars	Control Object
Control Chart	Control Omo
Control Clip	Control Parameter
Control Clock	Control Payload
Control Condition	Control Pie
Control Container	Control Scaling
Control Data Action	Control Sign Container
Control Datapool	Control SoftClip
Control DP Object	Control Stoppoint
Control FeedView	Control Targa Alpha
Control Geom	Control Targa Clip
Control Hide in Range	Control Text

[Control Hide on Empty](#)
[Control Image](#)

[Control Video](#)
[Placeholder](#)

12.4.1 Common Control Plugin Properties

Most control plugins have a set of common properties:



- **Field Identifier:** When making templates for Viz Trio, this should be a numeric value, Range: 1-n. For Viz 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 plugins 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 Apply Shared Memory



The Apply Shared Memory plugin allows values to be transferred into Viz Engine using [Data Sharing](#) mechanisms provided by shared memory maps that can be applied to control plugins such as [Control Text](#). Just as [Control Object](#) acts as a distribution hub for content provided by control applications, the Apply Shared Memory plugin acts as a distribution hub for content provided by external systems feeding real-time data through the shared memory maps.

The control plugins that the Apply Shared Memory plugin can apply content to resides in the same folder.

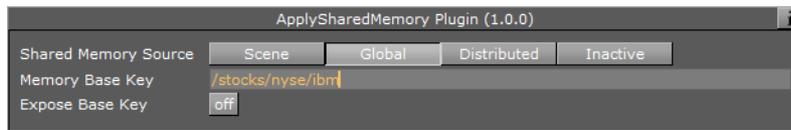
Just like the [Control Object](#) plugin, the Apply Shared Memory plugin only applies values to control plugins residing in the subtree of the scene tree where the Apply Shared Memory plugin resides. Also like the Control Object plugin, it matches the keys of incoming content with the field identifiers of individual control plugins to determine where to apply which incoming content value.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Apply Shared Memory Properties](#)
- [Visualizing composite data structures](#)
- [How shared memory keys are determined](#)
- [To visualize stock data from shared memory](#)
- [To test stock visualization](#)

Apply Shared Memory Properties



- **Shared Memory Source:** Specifies which shared memory map to apply values from. The **Scene**, **Global** and **Distributed** choices select the shared memory maps that are referred to as *Scene.Map*, *System.Map* and *VizCommunication.Map* respectively in the [Data Sharing](#) section.
- **Memory Base Key:** The key prefix path to combine with the field identifiers of control plugins to determine the key to look up in the chosen shared memory map when applying values to a particular field exposed by a control plugin.
- **Expose Base Key:** Expose the **Memory Base Key** property as a controllable field. Once this option is turned on the properties **Field Identifier** and **Description** as described in the [Common Control Plugin Properties](#) section can be specified.

The **Expose Base Key** property is typically used to allow a control application to control the base key which determines where in the shared memory map to pull values from. However, since the exposed property is an exposed control field like any other, a parent Apply Shared Memory plugin can be used to make the shared **Memory Base Key** property controllable by external systems through a shared memory map.

Choosing whether to allow the exposed field to be controlled by control applications or external systems feeding the shared memory map is done by placing either a [Control Object](#) or another Apply Shared Memory plugin on a parent container, respectively.

Visualizing composite data structures

The **Memory Base Key** property combined with the field identifiers exposed by control plugins can be seen as a way to visualize sets of related key-value pairs stored in shared memory maps. While the **Memory Base Key** property specifies where in the shared memory map to locate the set of related key-value pairs, the exposed field identifiers specify which of the related key-value pairs from the selected set to visualize in a particular way in the scene subtree.

For example, the **Memory Base Key** property can specify a shared memory path representing a particular stock symbol, while the field identifiers of control plugins picks out particular values such as stock *name*, *value* and *change* from that location in the shared memory map.

Since the Apply Shared Memory plugin only affects the scene subtree it is placed on, visualizing multiple composite data structure instances sharing the same structure and presentation can easily be achieved by copying subtrees and only change the Memory Base Key property.

For example, design a stock presentation design once, and copy it many times to create a scene showing several distinct stocks using the same visual style. Turning on the **Expose Base Key** property gives users of control applications control of which stocks to show in each presentation copy, while the data content for each stock is still supplied in real-time through the shared memory map.

How shared memory keys are determined

The shared memory keys used to look up values in a shared memory map is constructed by the combination of an absolute path specified in **Memory Base Key**

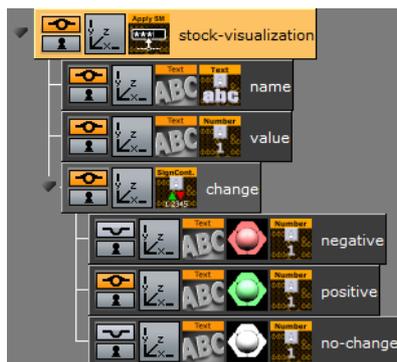
property and an either relative or absolute path consisting of the field identifier of an exposed field. Any path starting with a slash (/) is considered an absolute path, while all other paths are considered relative.

If the field identifier is an absolute path, then that absolute path alone is the result of the resulting shared memory key used. For example, the field identifier `"/weather/norway/bergen/temperature"` will result in the shared memory key `"/weather/norway/bergen/temperature"` regardless of what the **Memory Base Key** property is.

If the field identifier is a relative path, the path is interpreted relative to the absolute base path provided in the **Memory Base Key** property, and the absolute path resulting from that interpretation is used as the shared memory key. For example, a **Memory Base Key** property of `"/weather/sweden/stockholm"` and a field identifier of `"temperature"` results in the shared memory key `"/weather/sweden/stockholm/temperature"`.

Like file system paths, a path segment consisting of a single dot (.) is considered referring to the current directory, while two dots (..) is considered referring to the parent directory. This notation is resolved to a regular absolute path before using the result as a shared memory key. For example, a **Memory Base Key** property of `"/weather/./norway/trondheim"` and a field identifier of `"../oslo/windspeed"` will resolve to the shared memory key `"/weather/norway/oslo/windspeed"`.

To visualize stock data from shared memory

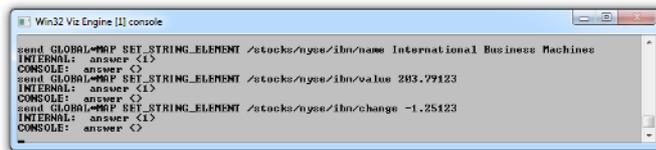


This procedure will visualize composite content fed through a shared memory map to the control plugins [Control Text](#), [Control Num](#) and [Control Sign Container](#).

1. Create a new container
2. Drag and drop a Apply Shared Memory plugin into this container
3. Set the **Memory Base Key** property of the Apply Shared Memory plugin to `"/stocks/nyse/ibm"`
4. Create two text geometries as children of this container
5. Drag and drop a Control Text plugin into the first text geometry container
6. Set the **Field Identifier** property of the Control Text plugin to `"name"`
7. Turn off **Use formatted text** property of the Control Text plugin
8. Drag and drop a Control Num plugin into the second text geometry container
9. Set the **Field Identifier** property of the of the Control Num plugin to `"value"`
10. Create a third container as a child of the Apply Shared Memory plugin container
11. Drag and drop a Control Sign Container plugin into this container.

12. Set the Field Identifier property of the of the Control Sign Container plugin to "value"
13. Create three text geometries as children of the Control Sign Container plugin container
14. Drag and drop a Control Num plugin into each of these three text geometry containers
15. For each of these three Control Num plugins set the **Field Identifier** property to "change"
16. Arrange the layout of the text geometries, and apply materials for styling

To test stock visualization

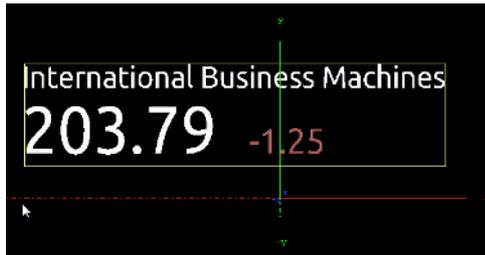


```

Win32 Viz Engine [1] console
send GLOBAL*MAP SET_STRING_ELEMENT /stocks/nyse/ihm/name International Business Machines
INTERNAL: answer <I>
CONSOLE: answer <I>
send GLOBAL*MAP SET_STRING_ELEMENT /stocks/nyse/ihm/value 203.79123
INTERNAL: answer <I>
CONSOLE: answer <I>
send GLOBAL*MAP SET_STRING_ELEMENT /stocks/nyse/ihm/change -1.25123
INTERNAL: answer <I>
CONSOLE: answer <I>

```

To test that stock visualization commands can be sent to populate the global shared memory map, see the descriptions in the [Data Sharing](#) section.



While these commands are suitable for testing the designs, a real external system that feeds the Viz Engine with data would use a more efficient data transfer protocols to populate the shared memory map. See the [External Data Input](#) section for information about sending data efficiently using UDP.

When changing a value in the shared memory map that is visualized by a control plugin the scene updates immediately to reflect the new value. The shared memory map can be populated with stock values for many stock symbols, including stocks that are not visualized at the moment. As soon as the **Memory Base Key** property of the Apply Shared Memory plugin is changed to visualize the values for a different stock symbol, and the scene updates immediately to reflect the value previously populated into the shared memory map for that stock symbol.

12.4.3 Control Action



The Control Action plugin 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

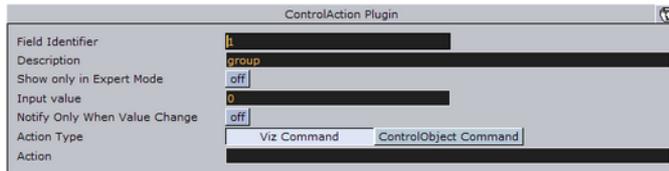
actions will only be triggered if the input value differs from the one already stored on the "input" field from previous invocations.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control Action Properties](#)

Control Action Properties



- **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 Plugin Properties](#)
- [Control Object](#)

12.4.4 Control Action Table



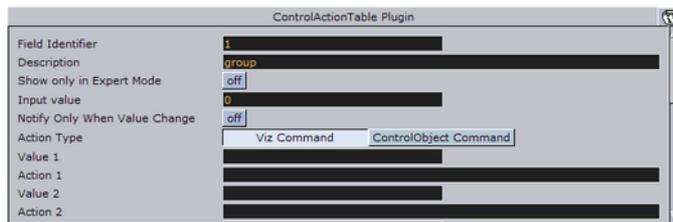
The Control Action Table plugin allows defining a table of actions similar to Control Action. The actions are associated with a given value. When Control Action Table receives input, it compares the received data to each of the values. If the data matches one of the values the plugin starts the corresponding action.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control Action Table Properties](#)

Control Action Table Properties



- **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 Plugin Properties](#)
- [Control Object](#)

12.4.5 Control Audio



The Control Audio plugin enables the operator of a control application to choose a sound clip.

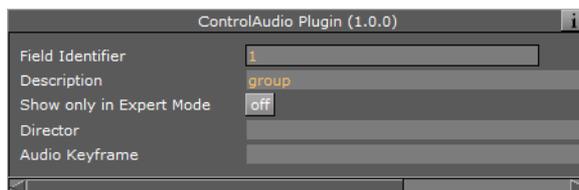
To expose the different properties, and change the audio file in one of the control applications, simply add the Control Audio plugin to the same container as the [Audio](#) plugin.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control Audio Properties](#)

Control Audio Properties



- **Director:** Sets the name of the stage director the audio channel is located below.
- **Audio Key Frame:** Sets the ID/name of the audio clip Key Frame.

Note: Under most circumstances there is no need to set director and audio Key Frame.

See Also

- [Common Control Plugin Properties](#)
- [Control Object](#)
- [Audio](#)

12.4.6 Control Bars



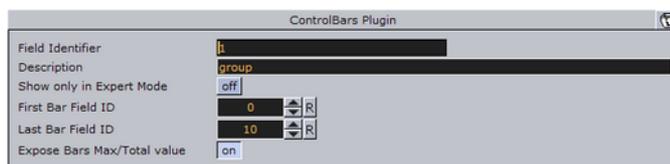
The Control Bars plugin allows binding of tab field values to the Bar Value fields in the Bar values presenter plugin. Each bar value property in the Bar Values editor is presented as a separate tab field in Viz Trio.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control Bars Properties](#)

Control Bars Properties



- **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 Plugin Properties](#)
- [Control Object](#)

12.4.7 Control Chart



The Control Chart plugin binds control of chart data to a Control Object. It currently supports the graph geometry plugin and the bar values, pie values and [Visual Data Tools](#) plugins. 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 plugin 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.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control Chart Properties](#)

Control Chart Properties



- **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 [Visual Data Tools](#) plugins, make sure the Visual Data Tools plugin's Shared Memory is set to Inactive.

See Also

- [Common Control Plugin Properties](#)
- [Control Object](#)
- [Visual Data Tools](#)

12.4.8 Control Clip



The Control Clip binds control of an AVI clip played by MoViz or SoftClip to a Control Object. With the Control Clip plugin the clips may be changed within the control application.

The MoViz plugin enables the user to play media files or media streams inside Viz Engine. MoViz 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.

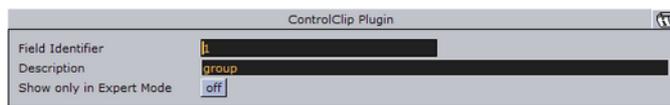
SoftClip is a Viz Artist/Engine plugin that can show AVI clips, either projected on a texture, or rendered directly.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control Clip Properties](#)

Control Clip Properties



This plugin does not have any properties or parameters except those that are common to all plugins.

See Also

- [Common Control Plugin Properties](#)
- [Control Object](#)

12.4.9 Control Clock



The Control Clock plugin 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.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control Clock Properties](#)

Control Clock Properties



- **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 plugin. It is not necessary to set any value here. It is normally only used for debugging purposes.

See Also

- [Common Control Plugin Properties](#)
- [Control Object](#)

12.4.10 Control Condition



With the Control Condition plugin 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 commands like

```
0 RENDERER*STAGE START
```

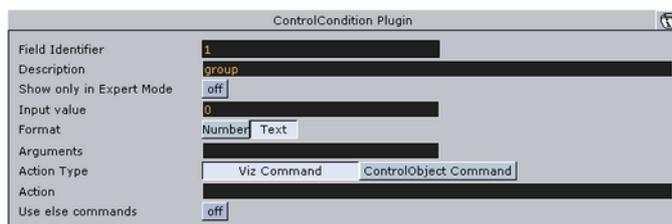
Viz Engine commands must be prefixed with a zero followed by a space. This is to enable the plugin to recognize that the command should be interpreted as a Viz Engine command.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control Condition Properties](#)

Control Condition Properties



- **Input Value:** Shows the current input value to the plugin.
- **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 Plugin Properties](#)
- [Control Object](#)

12.4.11 Control Container



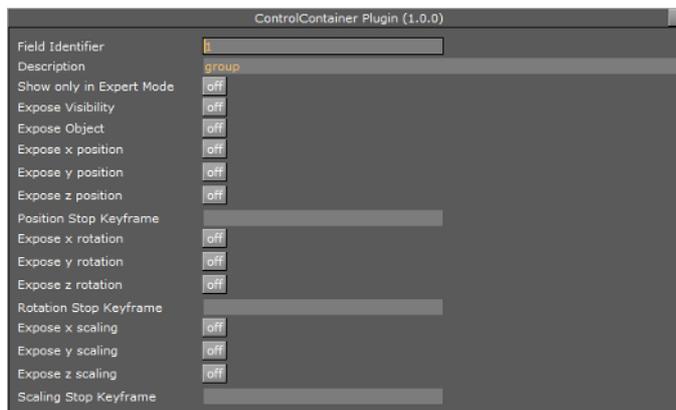
The Control Container plugin exposes a range of different transformation properties for a container.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control Container Properties](#)

Control Container Properties



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 Key Frame can be specified. This enables the user to control a Key Frame in an animation, for instance the end Key Frame in a position animation.

See Also

- [Common Control Plugin Properties](#)
- [Control Object](#)

12.4.12 Control Data Action



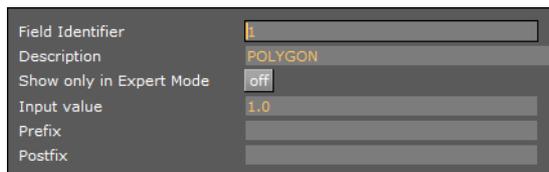
Binds control of a control parameter to a Data Action.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control Data Action Properties](#)

Control Data Action Properties



- **Input value:** Shows the current input value.
- **Prefix:** Add a prefix, if required
- **Postfix:** Add a postfix, if required

See Also

- [Common Control Plugin Properties](#)
- [Control Object](#)

12.4.13 Control Datapool



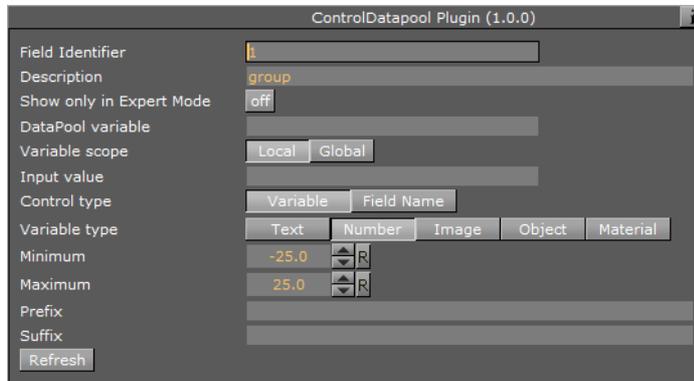
Control data pool can bind a tab field value to a data pool plugin variable. There exist many kinds of datapool plugins. All data pool plugins, which are used in a scene, must have its variable listed in the scene plugin named "data pool". It is these variables which the control data pool plugin links a tab-field to. For more information about the datapool plugins see the data pool documentation.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control Datapool Properties](#)

Control Datapool Properties



- **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 Plugin Properties](#)
- [Control Object](#)

12.4.14 Control DP Object



Allows copying or linking of Data Pool objects.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control DP Object Properties](#)

Control DP Object Properties

Field Identifier	1
Description	POLYGON
Show only in Expert Mode	off
Input value is:	Object to COPY Object to LINK Direct VALUES
Input value	
Direct VALUES	
Value is from table	on
Table Name	

- **Input value is:**
- **Input value:** Shows the current input value.

See Also

- [Common Control Plugin Properties](#)
- [Control Object](#)

12.4.15 Control FeedView



Set the Feed View 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.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control FeedView Properties](#)

Control FeedView Properties

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 Plugin Properties](#)
- [Control Object](#)

12.4.16 Control Geom



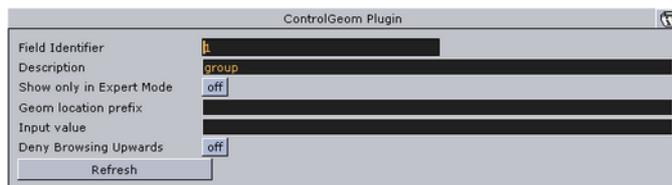
The Control Geom plugin exposes the control of geometry objects to the user.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control Geom Properties](#)

Control Geom Properties



- **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 Plugin Properties](#)
- [Control Object](#)

12.4.17 Control Hide in Range



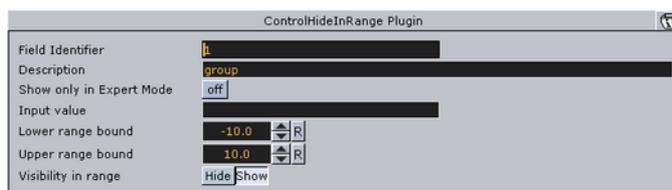
The Control Hide in Range plugin hides or shows a container if the input value given by the user is within or outside a specified value range. The input value must be from a font container that holds a numeric value, typically with a Control Num plugin.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control Hide in Range Properties](#)

Control Hide in Range Properties



- **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 Plugin Properties](#)
- [Control Object](#)

12.4.18 Control Hide on Empty



The Control Hide on Empty plugin performs a very simple function: It hides the container it sits on if the input value is blank (text) or zero (numbers). Use this plugin to hide the graphics elements if the user does not enter any value in the connected tab-field. Together with the [Autofollow](#) plugin, the plugin 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 plugin to hide the bug and the breaking news only if the page in Viz Trio have no value in those tab-fields.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control Hide on Empty Properties](#)

Control Hide on Empty Properties



- **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 Plugin Properties](#)
- [Control Object](#)

12.4.19 Control Image



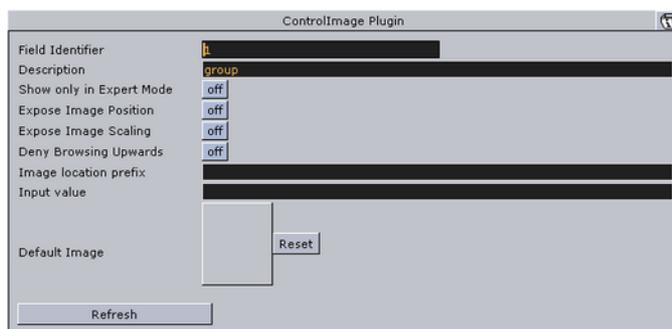
The Control Image plugin creates an image control in the control clients. The plugin must be placed on the container that holds the image/texture. If an image/texture is invalid or Viz Engine for some reason cannot load it, then the current image/texture remains visible.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control Image Properties](#)

Control Image Properties



- **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/textures 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/texture 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/texture on the same tab-field by just entering a text string. This is relevant if a user has a folder containing flag images/textures, and the control client receives "US" from the external system it will load the image/texture 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 plugin. It is not necessary to set any value here. It is used for debugging purposes.

See Also

- [Common Control Plugin Properties](#)
- [Control Object](#)

12.4.20 Control Key Frame



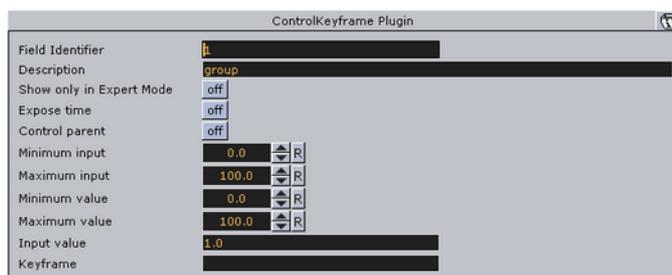
The Control Key Frame plugin can be used to give the user control over the value of a "single value Key Frame" unlike position values for instance where all three axes must be specified. Create an animation, for instance an alpha animation, and give the Key Frame to control a name. Put the Control Key Frame plugin on the object's container and set the parameters.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control Key Frame Properties](#)

Control Key Frame Properties



- **Expose Time:** Enables the user to adjust the Key Frame along the timeline.
- **Control Parent:** When enabled, this option allows for two Control Key Frame plugins to be within one container.

Note: In this case the control Key Frame plugins 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.
- **Key Frame Time:** Sets the time value of the Key Frame.
- **Key Frame:** Sets the name of the Key Frame that is to receive a value by the control client user.

Example for the input values: If the Key Frame is an alpha Key Frame 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 Plugin Properties](#)
- [Control Object](#)

12.4.21 Control List



The Control List plugin allows creation of table controls. By following some design conventions when designing the table in Viz Artist, Viz Trio will show 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.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control List Properties](#)
- [Tables and Preview Points](#)
- [To Design a Table Scene](#)

Control List Properties



- **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 show 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 to show the next 5 rows and so on.

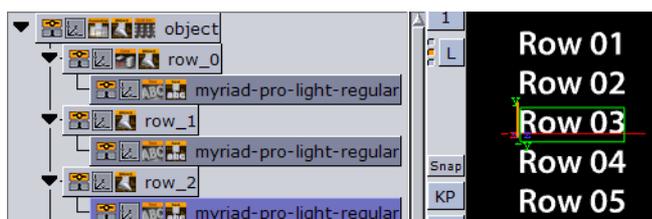
However, in such a situation, while editing table data in Viz Trio, you cannot use a basic jump to Key Frame event for preview, as this will only show the Key Frame that relates to the table's Field identifier (e.g. 3), and not the rows or columns of the table.

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) 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 Key Frame:** When adding Key Frame events (stop points, tags,...) and giving them the name of a control plugin identifier, Viz Trio jumps the timeline to that Key Frame event when selecting the tab field with that identifier.
- **Extended jump to Key Frame:** If the Key Frame event name is e.g. 3_005 and there is a control list with identifier 3, then Viz Trio jumps the timeline to that Key Frame 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 correctly interpreted by the Control List plugin 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 plugin 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 plugin 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 required number of columns.
 - Each of the text containers must get a Control Text plugin 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 plugins will be picked up by the Control List plugin 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* plugin [Coco](#) can be used to automate this task.
 - Add the Coco plugin 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* plugin [Grid Arrange](#) can be used to automate this task.
 - Add the Grid Arrange plugin 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 shown 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 Plugin Properties](#)
- [Control Object](#)

12.4.22 Control Map



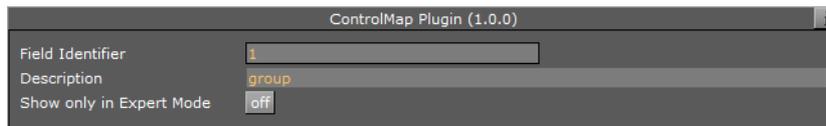
The Map plugin binds the control of the Viz World Client plugin XML property to a ControlObject which allows an operator to use the Viz World Client Editor to select maps.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control Map Properties](#)

Control Map Properties



This plugin does not have any properties or parameters except those that are common to all plugins.

See Also

- [Common Control Plugin Properties](#)
- [Control Object](#)

12.4.23 Control Material



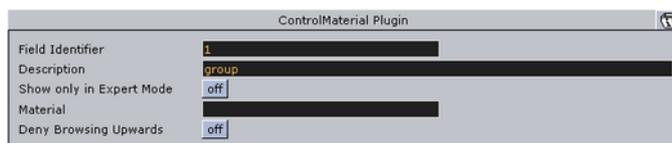
The Control Material plugin exposes the material control to the Viz Trio user.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control Material Properties](#)

Control Material Properties



- **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 Plugin Properties](#)
- [Control Object](#)

12.4.24 Control Num



The Control Number plugin (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

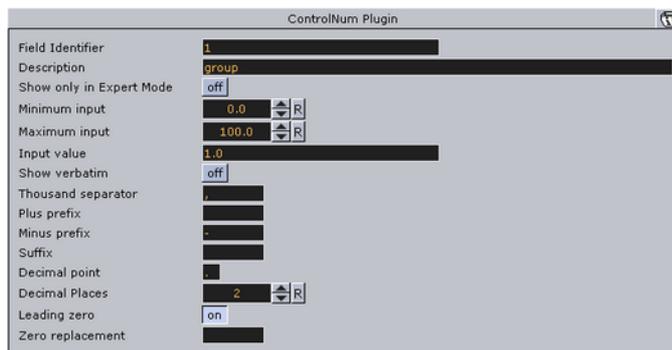
user or by any external source. It should be used instead of Control Text when numbers are to be the input value. To make it work, add the plugin to the container that holds the text object.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control Num Properties](#)

Control Num Properties



- **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 required 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 show instead of zero values.

See Also

- [Common Control Plugin Properties](#)
- [Control Object](#)

12.4.25 Control Object



For each Scene that is to become a Viz Trio page template, a Viz Pilot template or a Viz Ticker template it must have one instance of the Control Object plugin.

Control object provides control channels for subtrees of Viz Scenes. The control channel information is stored in the subtree itself, so the Scene subtree can be merged and stored as objects in the object pool.

When a Control plugin is added to a Scene a control object plugin is automatically added to the root container of the scene subtree to be controlled. The plugin stores and gathers field definitions which describe how values are processed and applied to various properties of the subtree.

The easiest way define fields is to place one or more Control plugins on containers within the subtree. These Control plugins will find the Control Object and register properties related to the container they are placed on.

Note: For Viz Ticker it is possible to have more than one Control Object plugin. It should be on the root container for the tab objects in the scene.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control Object Without Transition Logic](#)
- [Control Object With Transition Logic Enabled](#)
- [Control Object Properties](#)
- [Control Object Advanced Properties](#)

Control Object Without Transition Logic

In the simple scene based mode, the plugin 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.

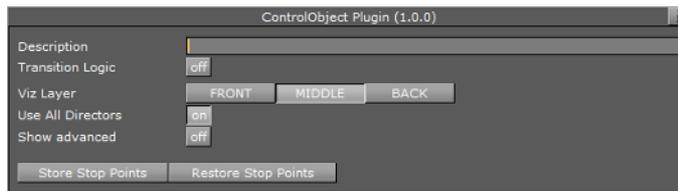
Note: Objects in the Back layer will be rendered behind objects in the Middle and Front layer and so on, so the scenes must then be designed with that in mind.

Control Object 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.

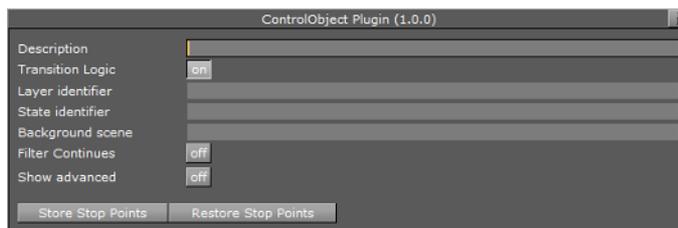
Control Object Properties

Transition Logic Off



- **Description:** A description of the object the scene subtree represents. This description is typically presented to users in GUI applications that allow the user to fill in data that should be presented by the controlled object.
- **Transition Logic:** Specifies whether the controlled object uses transition logic. Transition logic loads objects onto layers in a background scene, and runs animations to change the background scene layer into a required state.
- **Viz Layer:** The Viz layer to load the scene which contains the scene tree which contains this Control Object. This parameter is only used when transition logic is set to Off
- **Use All Directors:**

Transition Logic On

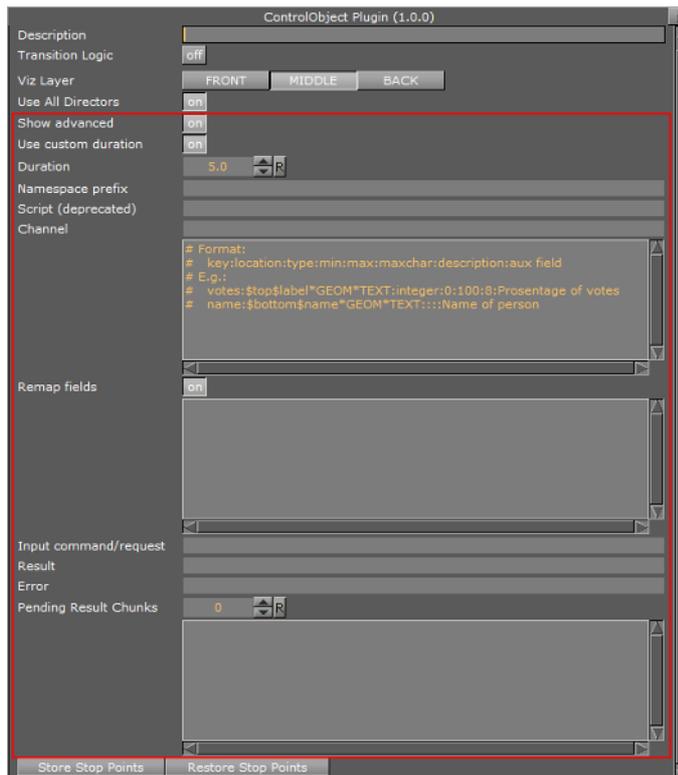


- **Layer identifier:** Select the layer the controlled object should be loaded onto in the background scene. This parameter will only be used when transition logic is set to On. For the transition logic to work, the identifier given here must be identical to the names of the following components of the background scene:
 - **A container with a Toggle plugin:** The transition logic will use the toggle plugin on this layer-container to load the object
 - **A director:** The transition logic will animate this layer-director into the state requested by the object
- **State identifier:** The state which the layer-director of the background scene should be animated to, when the object is loaded into its target layer. The identifier given here must be identical to one or more stop-points defined on the timeline of the layer-director of the background scene. This parameter will only be used when transition logic is set to On.
- **Background scene:** The scene which has the layer-container and layer-director needed by the transition logic to load and animate in the object. This parameter will only be used when transition logic is set to On.
- **Show advanced:** See [Control Object Advanced Properties](#)
- **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 plugin, it will operate on the object's animation director

Control Object Advanced Properties

When the Show Advanced button is set to On, a channel property is shown as well as a text box for manual field definitions. Most of these parameters are used for system integration purposes and are rarely relevant in a normal scene design for Viz Trio, Viz Pilot and Viz Ticker.



- **Use custom duration:** Click to set the duration parameter:
 - **Duration:** The amount of time the graphics should be shown when triggered as an interstitial element, such as sponsor elements in a ticker.
- **Namespace prefix:** Setting this parameter will allow the fields defined by control plugins below, to propagate to control objects further up the Scene Tree. When definitions are propagated upwards, the field identifier will be prefixed with the name prefix
- **Script (deprecitated):** Property used by a Viz Trio client to identify the name of a script that contains functions that will be executed when various operations are completed in Viz Trio.
- **Channel:** Sets the name of the default playback 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:

Field description	Example
-------------------	---------

Field identifier/key	votes
Property location	\$top\$label*GEOM*TEXT
Type	There are many type identifiers and are used by the control applications to determine which content editor control to show for a given field. e.g.: integer bool float integer text richtext duplet triplet
Minimum value	0
Maximum value	100
Maximum number of characters	8
Field description	Percentage of votes
Auxiliary field	Conveys various type specific type constraints and auxiliary information needed by the control application to provide a convenient editing experience. E.g. limiting text to a single text line, allowing only upper case characters, or limiting images to only images found in a certain image pool folder.

- **Remap fields:** For use in scenes with nested control object plugins. This feature is mainly for advanced integration use. When ON, Control Object will give the control clients the list of entries in map, rather than the list of entries registered by control plugins 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.
- **Input command/request:** Field which receives commands and request actions. When this parameter is changed, the command described by the new value of the parameter is executed. Return values for requests are stored in the **Result** parameter.
- **Result:** Stores the result of requests given through the **Input command/request** parameter. Applications which complete the requests typically read the result text from parameter immediately after setting the request text in the **Input command/request** parameter.
- **Error:**
- **Pending Result Chunks:**

See Also

- [Common Control Plugin Properties](#)
- [Toggle](#)

- [Transition Logic](#)

12.4.26 Control Omo



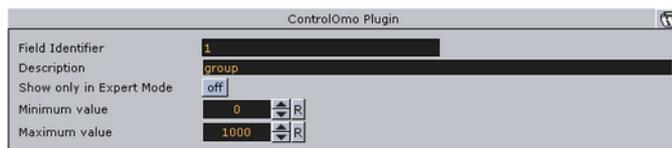
The Control Object moving (Omo) plugin in Viz Artist gives the possibility to add a group of containers and reveal one at the time. This is done by adding an [Omo](#) plugin on the root container and in its editor specify which of the Sub-Containers to show. The Omo value can be made accessible for the control client by adding the Control Omo plugin on the same container as the Omo plugin.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control Omo Properties](#)

Control Omo Properties



- **Minimum Value:** Sets the minimum input value.
- **Maximum Value:** Sets the maximum input value.

See Also

- [Common Control Plugin Properties](#)
- [Control Object](#)
- [Omo](#)

12.4.27 Control Parameter



The Control Parameter plugin works with function plugins, and is a very flexible plugin that can be used to control properties that are not exposed in a specific plugin, 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 plugin. To expose the alpha parameter for a container write ALPHA * ALPHA in the Parameter section in the plugin 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>
```

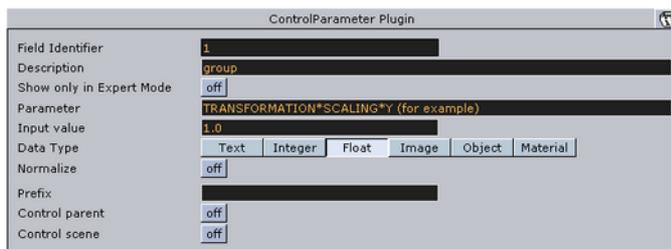
In this case the FUNCTION*PieValues*DATA is the parameter that will expose the object property Pie Max such that it can be controlled through a template.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control Parameter Properties](#)

Control Parameter Properties



- **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 Plugin Properties](#)
- [Control Object](#)

12.4.28 Control Payload



Control Payload distributes a whole payload to children Control Plugins.

Example: Social TV (Feed Streamer) uses this plugin to send payload fields from a Data Hub to Viz Engine.

If the payload value is empty, the Container is hidden.

This section contains the following topics:

- [Control Payload Properties](#)

Control Payload Properties



- **Text Entry Box:** Data from external application (for example Viz Trio, Social TV (Feed Streamer)) to send the payload. Example:

```
<payload xmlns="http://www.vizrt.com/types">
  <field name="topic">
    <value>It\qs Valentine\qs Day, how you doin\q? </value>
  </field>
  <field name="option">
    <value>Hate it! Worst holiday ever|Love it! Wish it were every
    day|Thanks for the reminder, need to buy roses</value>
  </field>
  <field name="count">
    <value>238|189|46</value>
  </field>
</payload>
```

See Also

- [Common Control Plugin Properties](#)

12.4.29 Control Pie



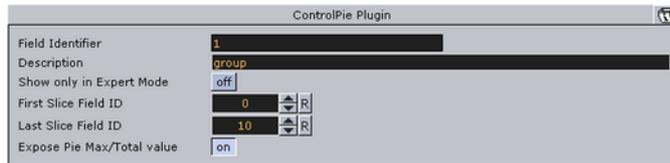
The Control Pie plugin is able to bind a tab-field to values from PieSlice plugins.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control Pie Properties](#)
- [To Build a Pie Chart](#)

Control Pie Properties

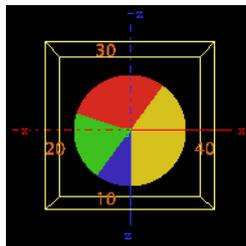


- **First slice ID:** Sets the tab-field number for the first slice (the first container below the group that holds the ControlPie Plugin).
- **Last Slice ID:** Sets the tab-field number for the last slice (the difference between Last and First should correspond to the total number of slices in the pie 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.

See Also

- [Common Control Plugin Properties](#)

To Build a Pie Chart



1. Add a group container and name it, for instance "pie". Add the presenter plugin [Pie Values](#) and the control plugin Control Pie to the group.
2. Add a Cylinder object as a Sub-Container to the group. Add material and the presenter plugin [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, which shows 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

- [Control Object](#)
- [Pie Slice](#)
- [Pie Values](#)

12.4.30 Control Scaling



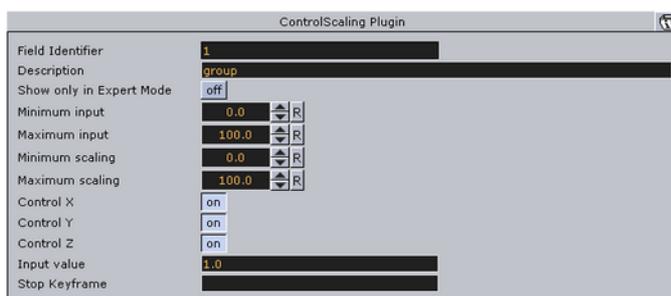
The Control Scaling plugin enables the Viz Trio user to edit transformation scaling values for objects. Normalization between input and output values can be set by parameters.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control Scaling Properties](#)

Control Scaling Properties



- **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 Key Frame:** If the scaled object is animated, the scaling values entered must be put into the end Key Frame 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 Key Frame in the Viz Artist stage and enter that name in the “stop Key Frame” field in the control scaling plugin.

See Also

- [Common Control Plugin Properties](#)
- [Control Object](#)

12.4.31 Control Sign Container



The Control Sign Container value can show one of three containers based on the value of a numerical input. The container that holds the plugin must have three Sub-Containers:

- A negative value will trigger show the first container.
- A positive value will trigger show the second container.
- A zero value will trigger show the third container.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control Sign Container Properties](#)

Control Sign Container Properties



- **Input Value:** Shows the current input value. Enter a value here to test that the correct Sub-Container is shown.

See Also

- [Common Control Plugin Properties](#)
- [Control Object](#)

12.4.32 Control SoftClip



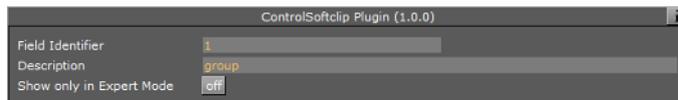
The Control SoftClip plugin allows the user to control and browse for video clips which are played with the Soft Clip plugin in Viz Artist. To make it work, simply add the control plugin on the same container as the SoftClip plugin. Further, specify a Field Identifier and a Description.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control SoftClip Properties](#)

Control SoftClip Properties



This plugin does not have any properties or parameters except those that are common to all plugins.

See Also

- [Common Control Plugin Properties](#)
- [Control Object](#)

12.4.33 Control Stoppoint



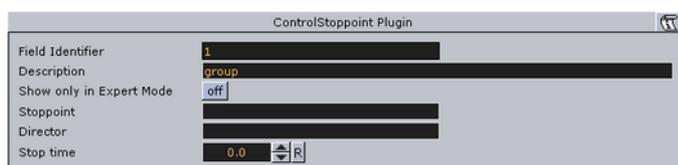
Control Stop point enables control of the time value on a stop point. This can typically be used for setting the length of an animation.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control Stoppoint Properties](#)

Control Stoppoint Properties



- **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 Plugin does not work with transition logic based scenes.

See Also

- [Common Control Plugin Properties](#)
- [Control Object](#)

12.4.34 Control Targa Alpha



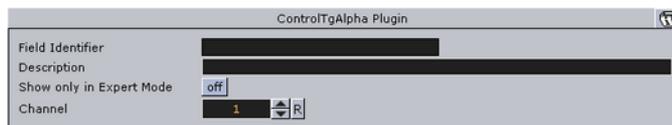
The Control Targa Alpha plugin binds control of Targa video alpha properties to a Control Object.

IMPORTANT! The Control Targa Alpha plugin is deprecated. It is installed because of backward compatibility, and is set to inactive by default. It is not a supported plugin.

This section contains information on the following topics:

- [Control Targa Alpha Properties](#)

Control Targa Alpha Properties



- **Channel:** Sets which channel to control.

See Also

- [Common Control Plugin Properties](#)
- [Control Object](#)

12.4.35 Control Targa Clip



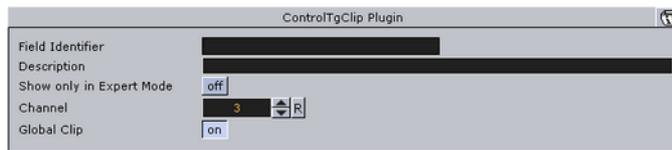
The Control Targa Clip plugin exposes control over a Targa clip channel. The Viz Trio user will get a Windows file browser, where video clips to play can be browsed for.

IMPORTANT! The Control Targa Alpha plugin is deprecated. It is installed because of backward compatibility, and is set to inactive by default. It is not a supported plugin.

This section contains information on the following topics:

- [Control Targa Clip Properties](#)

Control Targa Clip Properties



- **Global Clip:** When set to **On**, the clip specified will not be added to the Stage, but will play globally in the Scene. If set to **Off**, two more fields show, Director and Device key frame. The clip will then be loaded at the named key frame in the specified Director and will run when the Director is triggered.

See Also

- [Common Control Plugin Properties](#)
- [Control Object](#)

12.4.36 Control Text



The Control text plugin binds the control of text to a ControlObject.

This plugin can both be used to control regular text geometries and also text revealed by TFXWrite plugins. The plugin will automatically detect and use the appropriate technique to set the text on its container.

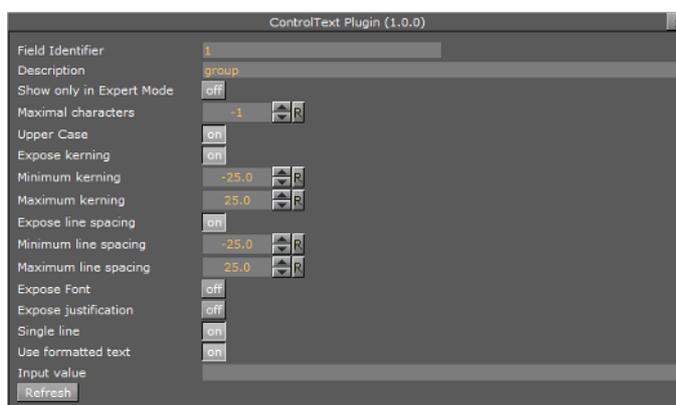
The plugin will also auto detect the [Convert Case](#) plugin and route text through it if present.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control Text Properties](#)

Control Text Properties



- **Maximal characters:** Set the maximum number of characters which can be typed into the text field.
- **Upper Case:** When set to On, lower case input is converted to upper case.
- **Expose kerning:** When set to On, the minimum and maximum kerning parameters can be set.
- **Expose line spacing:** When set to On, the minimum and maximum parameters for line spacing can be set.
- **Expose Font:** When set to On, a different Font can be selected
- **Expose justification:** When set to On, the text justification can be changed
- **Single Line:** Disables text wrap. Text is shown on one single line.
- **Use Formatted Text:** Enables the use of character specific formatting, i.e. kerning, position, rotation, scaling, alpha, and color.

Note: Viz Trio: The user can press an assigned shortcut key to enter a required formatting mode. To change the format of one or more characters, hold down <Alt> and toggle the right and left arrow keys to shift mode. See the [Viz Trio User's Guide](#) for more information on *Character Formatting*.

Note: Viz Multichannel: To use the characters '<' and '>' correctly, make sure that **Use Formatted Text** is set to OFF.

Note: Viz Pilot is not supported.

Note: Formatted text cannot be used together with the TfxWrite text effects plugin.

- **Input Value:** Shows the current value for the plugin. It is not necessary to set any value here. It is used for debugging purposes.

See Also

- [Common Control Plugin Properties](#)
- [Control Object](#)

12.4.37 Control Video



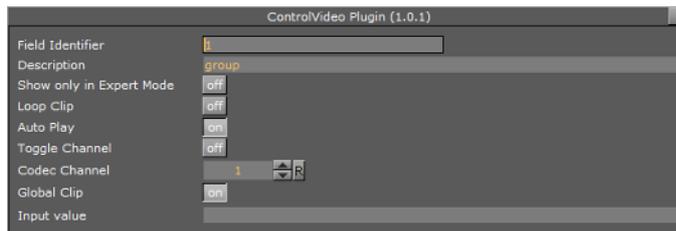
The Control Video plugin exposes control over a video codec channel. The user will get an editor (e.g. Media Search (as part of Viz Trio, Viz Pilot) where video clip files from Viz One can be searched for.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control Video Properties](#)

Control Video Properties



- **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 Key Frame:** Sets the name of the Key Frame.

Disabling Global clip will enable the Director and Clip Key Frame fields. Setting Director and Clip Key Frame will load the clip at the named Key Frame 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 Plugin Properties](#)
- [Control Object](#)

12.4.38 Control World



The ControlWoC plugin allows you to expose a set of properties to the operator. The control is a replacement for the [Control Map](#) plugin with much more options and on-the-fly feedback from Viz Artist/Engine.

Note: The two controls are not compatible. The new control will not work with Viz Pilot, and is only a part of Viz Artist/Engine 3.3 and above.

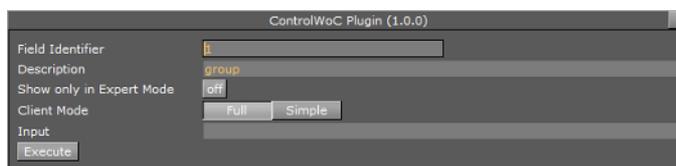
The control can expose different fields based on the container it resides on. When tabbing to the control (in a navigator scene) the camera will jump to map location and all feedback (exact camera position) will be immediate.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Control World Properties](#)

Control World Properties



- **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. The field is based on the input from the NavFinder plugin; however, you may also manually change this field.

See Also

- [Common Control Plugin Properties](#)
- [Control Object](#)

12.4.39 Placeholder



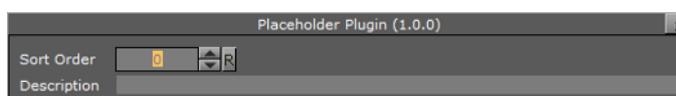
The Placeholder plugin is used to mark placeholder containers when designing library objects for Viz Trio compositing. A placeholder container will be available as a parent to other compositing objects in the scene tree.

Note: This plugin is located in: Built-ins -> Container Plugins -> Control

This section contains information on the following topics:

- [Placeholder Properties](#)

Placeholder Properties



- **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 Plugin Properties](#)
- [Control Object](#)

12.5 Default

The following container plugins are located in the Default folder:

- [Trio Scroll Element](#)

12.5.1 Trio Scroll Element



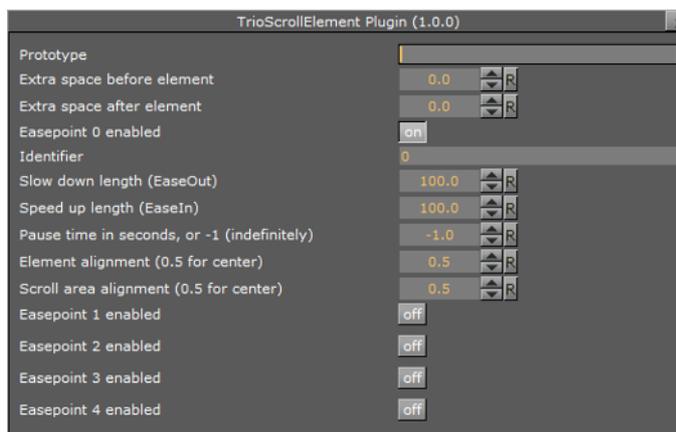
The Trio Scroll Element plugin is used for data storage by the [Trio Scroll](#) Geometry plugin. It is automatically added to each base element of a Viz Trio scroll scene.

Note: This plugin is located in: Built-ins -> Container Plugins -> Default

This section contains information on the following topics:

- [Trio Scroll Properties](#)

Trio Scroll Properties



- **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: This will affect the total time the scroll is scheduled to use.

- **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](#) Plugin

12.6 Feed

The purpose of the Feed plugins is to receive real-time data and apply it to the state of a Viz Artist Scene in various ways. They are mostly used to create Scenes for use in financial data and stock feed visualizations.

Note: They are now mostly replaced by a combination of the shared memory map, Viz scripting, [Control](#) Plugins and the [Apply Shared Memory](#) plugin as the preferred way to bring real-time data into Viz Artist Scenes.

The method of use for the plugins is as follows:

1. Identify text geometry, or container visibility in a scene that an external real-time data feed source is to control.
2. Apply the appropriate Feed plugin to the container of the identified property.
3. In the plugin properties, specify a data feed key to look for in the incoming data feed.
4. Send UDP messages to carry the content of the data feed in a YAML format which carries key-value pairs to a specific port on the Viz Engine host, which the plugin infrastructure listens to.

The following container plugins are located in the Feed folder:

- [Feed Activate](#)
- [Feed View](#)
- [Hide in Range](#)

12.6.1 Feed Activate



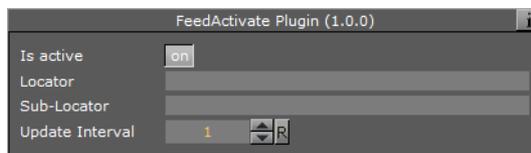
The Feed Activate plugin activates and deactivates plugins (i.e. hides and un hides them), based on a given value in the hierarchy from a FeedRelay.

Note: This plugin is located in: Built-ins -> Container Plugins -> Feed

This section contains information on the following topics:

- [Feed Activate Properties](#)

Feed Activate Properties



- **Is active:** The plugin will update visibility of sub-containers based on container name and value retrieved location
- **Locator:** Get data for text from this location
- **Sub-Locator:** Append this to locator path when retrieving text
- **Update Interval:** How often (in number of fields) an update occurs (increase to gain performance)

12.6.2 Feed View



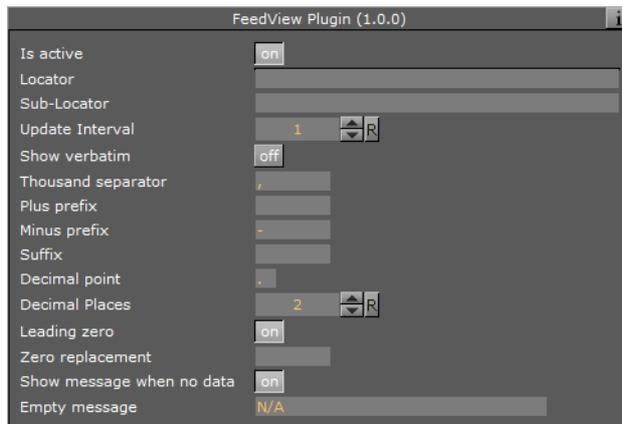
Applies value from a real-time feed onto text geometry with various forms of optional formatting.

Note: This plugin is located in: Built-ins -> Container Plugins -> Feed

This section contains information on the following topics:

- [Feed View Properties](#)

Feed View Properties



- **Is active:** Plugin will update text geometry of container.
- **Locator:** Get data for text from this location.
- **Sub-Locator:** Append this to locator path when retrieving text.
- **Update Interval:** How often (in number of fields) an update occurs.
- **Show verbatim:** Show as given, do not interpret and reformat.
- **Plus prefix:** Use this prefix for positive values.
- **Minus prefix:** Use this prefix for negative values.
- **Suffix:** Append this suffix.
- **Decimal point:** Use these characters as a decimal point.
- **Leading zero:** Use leading zero for values less than one.
- **Zero replacement:** Use this string when value is zero.
- **Show message when no data:** If checked, the contents of the 'Empty Message' field will be set as text if the locator is not found in the cache. If this is unchecked, the text geometry will be left unmodified.
- **Empty Message:** Text to set if the locator is unregistered.

12.6.3 Hide in Range



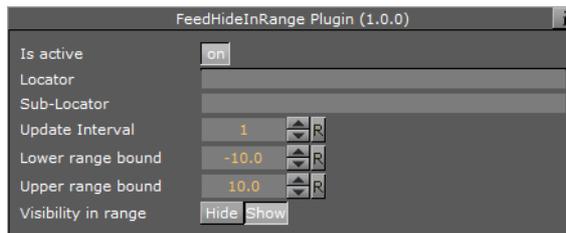
Makes container either visible or invisible depending on whether value from real-time feed is within a specified range of numerical values.

Note: This plugin is located in: Built-ins -> Container Plugins -> Feed

This section contains information on the following topics:

- [Hide in Range Properties](#)

Hide in Range Properties



- **Is active:** The plugin will update text geometry of container.
- **Locator:** Get data for text from this location.
- **Sub-Locator:** Append this to locator path when retrieving text.
- **Update Interval:** How often (in number of fields) an update occurs.

12.7 Global

The following container plugins are located in the Global folder:

Alpha	Mask Source and Mask Target
Audio	Lighting
Clipper	Z-Sort
Expert	Projector Source and Projector Target
Extrude	Script
Glow	Shadow Caster and Shadow Receiver
HDR	Synchronized Properties
Key	Video Clip
Look-At	Window Mask

12.7.1 Alpha



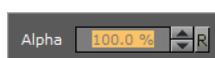
The Alpha plugin adds an alpha channel to the container. The alpha channel defines the degree of transparency for the container and its Sub-Containers. This feature can be used to easily fade in or out a complete part of the scene-tree.

Note: This plugin is located in: Built-ins -> Container Plugins -> Global

This section contains information on the following topics:

- [Alpha Properties](#)

Alpha Properties



- **Alpha:** Sets the alpha value for the container.

12.7.2 Audio



The Audio plugin allows a designer to configure audio channels. The Audio plugin can be applied to any Container.

The Audio plugin has three different audio clip mixing modes (each mode panel has a test function included (see [To Test Audio Channel Setup](#))):

- [Default Mode](#)
- [FX Mode](#)
- [Manual Mode](#)

Note: This plugin is located in: *Built-ins -> Container Plugins -> Global*

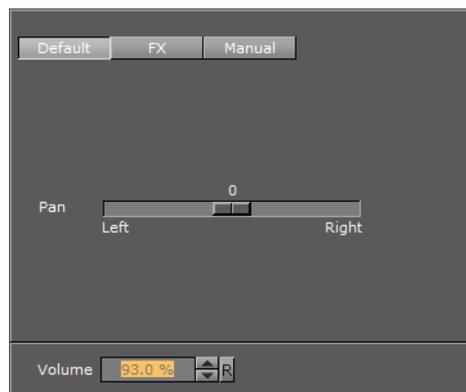
This section contains information on the following topics and procedures:

- [Default Mode](#)
- [FX Mode](#)
- [Manual Mode](#)
- [To Configure Audio Channel Aliases](#)

Default Mode

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.



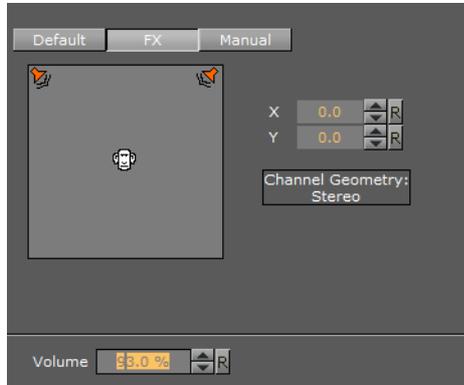
The **Pan** slider uses audio aliases when adjusting the volume of the left and right speakers. This is valid for multichannel configurations as well. If the slider is moved to the right in a 7.1 configuration LeftBack, LeftMiddle and LeftFront will be muted.

The **Volume** field controls the overall audio volume of all clip channels.

FX Mode

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.

The **Volume** field controls the overall audio volume of all clip channels.



Manual Mode

In Manual mixing mode it is possible to enter the Out Channel alias for every clip channel. The mask can hold more than one channel. This allows the mix of every clip channel to an arbitrary amount of out channels.

The **Volume** field controls the overall audio volume of all clip channels.

The syntax for the string is as follows:

CHANNEL_ALIAS%VOLUME+CHANNEL_ALIAS%VOLUME...

or

[CHANNEL_ALIAS%VOLOUME]+

Example: [FRONT_LEFT%0.5]+



To Configure Audio Channel Aliases

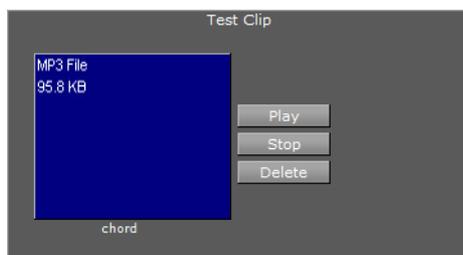
1. Add an Audio clip to a Scene. See:
 - [To Add an Audio Clip to a Scene \(Context Menu\)](#)
 - [To Add an Audio Clip to a Scene](#)
2. Click the **Audio** plugin icon (opens the Audio plugin editor).
3. Click **Manual**.
4. Enter these channel configurations (as entered in **Audio Settings** in [Viz Configuration](#) (see the [Viz Engine Administrator Guide](#)):
 - Out Channel 0: FRONT_LEFT
 - Out Channel 1: FRONT_RIGHT
 - Out Channel 2: FRONT_CENTER
 - Out Channel 3: LOW_FREQUENCY
 - Out Channel 4: BACK_LEFT
 - Out Channel 5: BACK_RIGHT
 - Out Channel 6: SIDE_LEFT
 - Out Channel 7: SIDE_RIGHT

Note: You can use aliases as configured in the Channels tab (see **Audio Settings** in [Viz Configuration](#) (see the [Viz Engine Administrator Guide](#))).

5. 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. Add a group container to the scene tree
2. Add the Audio plugin to the group container
3. Open the Audio plugin 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.

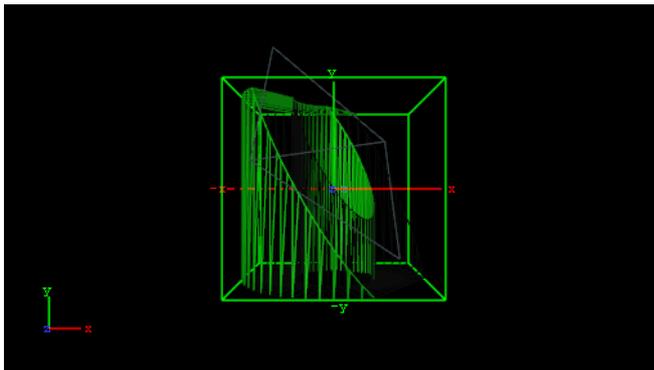
See Also

- Audio Settings (see [Viz Configuration](#) in the [Viz Engine Administrator Guide](#))
- Audio in Viz (see the [Viz Engine Administrator Guide](#))

12.7.3 Clipper



The Clipper plugin is an alternative to the mask function. Objects with the clipper plugin 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 Properties Panel (see [Modify Container Properties](#)).



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 plugin 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.

Note: This plugin is located in: Built-ins -> Container Plugins -> Global

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

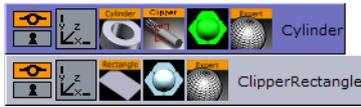
- [Clipper Properties](#)
- [To Create a Clipping Plane Effect](#)

Clipper Properties

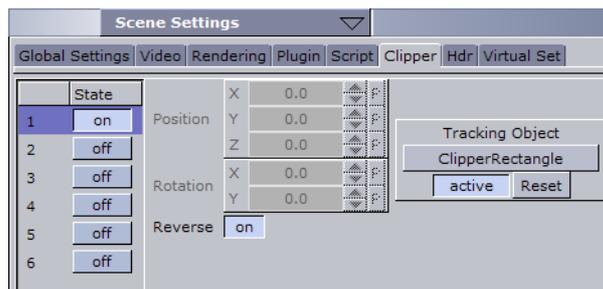


- **Planes:** Enables the planes. By default 1 is enabled.

To Create a Clipping Plane Effect

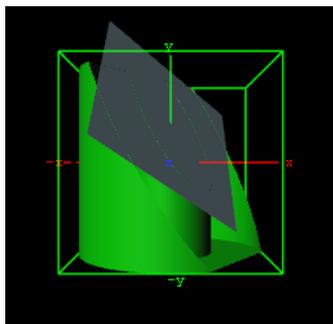


1. Add two group containers to the scene tree (same level) and name them **Cylinder** and **ClipperRectangle**.
2. Add the following to the **Cylinder** container:
 - Cylinder geometry plugin
 - Clipper container plugin
 - Material
 - Optional: Expert container plugin
3. Add the following to the **ClipperRectangle** container:
 - Rectangle geometry plugin
 - Material
 - Optional: Expert container plugin
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.

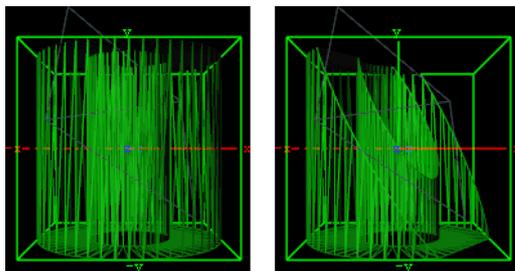


9. Open Scene Settings, and select the [Clipper](#) tab.

10. Enable **State 1**, and **Reverse**.



11. Drag and drop the ClipperRectangle container onto the Tracking Object drop zone.



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](#)

12.7.4 Expert



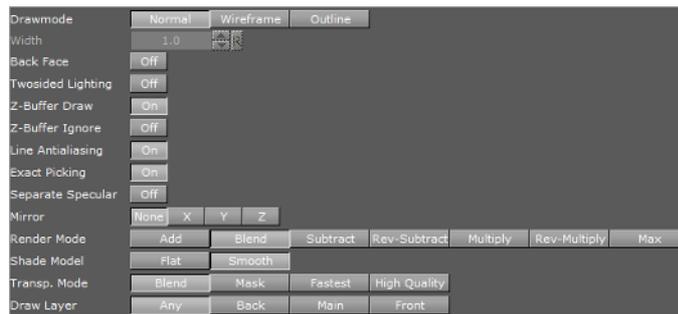
The Expert plugin sets some special properties and adds some advanced functions to a Container.

Note: This plugin is located in: Built-ins -> Container Plugins -> Global

This section also contains information on the following topic:

- [Expert Properties](#)

Expert Properties



- **Draw Mode:** Select to have the container drawn either:
 - **Normal:** Shows the rendered object with material and/or textures.
 - **Wireframe:** Shows the polygons that are drawn 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:** Select **Off** 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.
- **Z-Buffer Ignore:** Set to Off to ignore the OpenGL z-buffer draw option, and will draw the object no matter if it is in the back or not.
- **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.
- **Render Mode:**
 - **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 are 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.
- **Max:** Higher color values will overwrite lower color values (for example, white draws over black - independently from the Z-Sort).
- **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.
- **Draw Layer:** Defines if the objects are drawn in all layers or only if it shows in a certain layer:
 - **Any:** The object is always drawn.
 - **Main:** The object is drawn only if it is in a scene, located in the Main Layer.
 - **Back:** The object is drawn only if it sits in the Back Layer.
 - **Front:** The object is drawn only if it is in the Front Layer.

Note: The Draw Layer option can be used, for example, for a menu in a Scene which should be only visible if it is loaded in the front layer.

12.7.5 Extrude



An Extrude plugin 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. As a result a 3D element with front, back and sides is created from the 2D element.

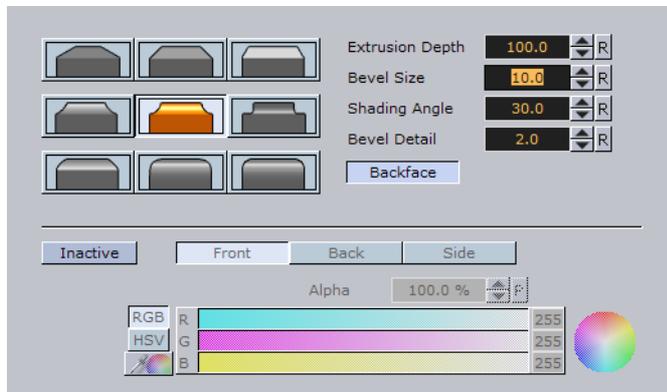
Note: This plugin is located in: Built-ins -> Container Plugins -> Global

This section also contains information on the following topics:

- [Extrude Properties](#)

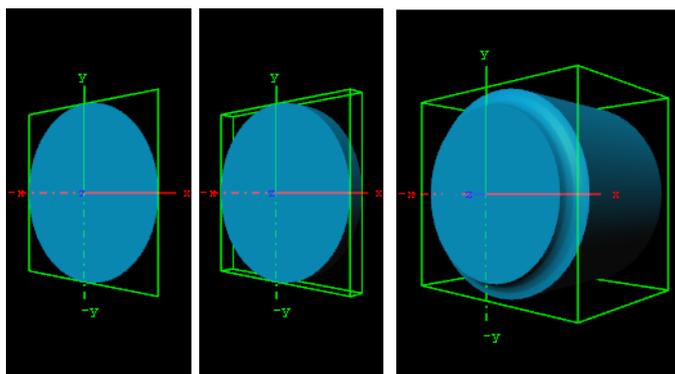
- [To extrude a 2D object](#)

Extrude Properties



- **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 lower 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 plugin and material to it.
2. Add the Extrude plugin 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.

12.7.6 Glow



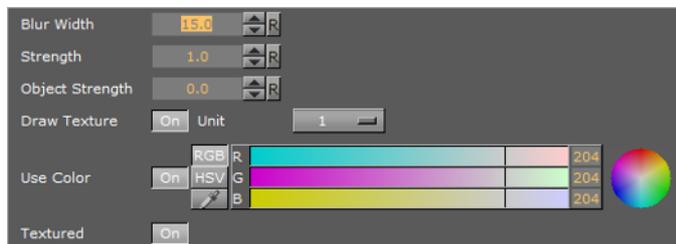
Makes the lit faces of objects glow.

Note: This plugin is located in: Built-ins -> Container Plugins -> Global

This section also contains information on the following topics:

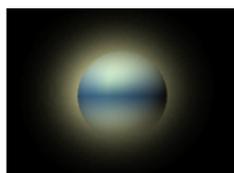
- [Glow Properties](#)
- [To Add Glow](#)

Glow Properties



- **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](#) plugin to the Scene Tree.
2. Add material and/or texture to the Sphere container.
3. Add the Glow plugin.
4. Open the Glow editor and set these parameters:
 - **Blur Width:** 20.0.
 - **Strength:** 6.0.
 - **Object Strength:** 0.0.
 - Enable **Use Color** and set the color parameter.

12.7.7 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.

Note: This plugin is located in: Built-ins -> Container Plugins -> Global

This section also contains information on the following topic:

- [HDR Properties](#)
- [To Add HDR](#)

HDR Properties

This plugin 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. Specify a diffuse and a Cubic Reflection image, as required.

12.7.8 Key



Key adds a Key signal to a container.

Note: This plugin is located in: Built-ins -> Container Plugins -> Global

This section also contains information on the following topic:

- [Key Properties](#)
- [To Add Key](#)

Key Properties



- **Draw RGB:** Enable/disable the graphical object
- **Draw Key:** Enable/disable the key signal
- **Automatic Alpha:** Set to **On** to set the alpha 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:** Set the mode in which the alpha values, of the key element to be rendered, should be mixed with the alpha values already existing in the frame buffer

Note: 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 done 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)

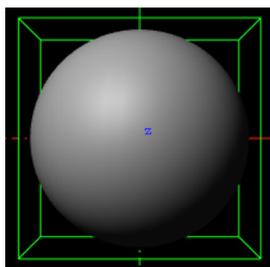
If A is the current source alpha value being rendered now and FA is the alpha of the frame buffer contents (target), the different modes can be explained as following:

- **Add:** $(A + FA)$ 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:** $(A + (1-A)*FA)$ 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:** $(FA - A)$ 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:** $(A - FA)$ 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
- **Max:** Uses the maximum of all key signals

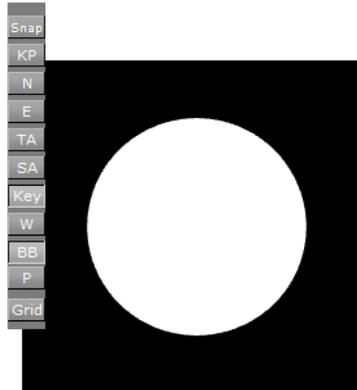
To Add Key



1. Open a Scene.
2. Add an object.
3. Add Key plugin.



4. Click the **Key** button on the left side of the [Scene Editor](#).



See Also

- [Global Settings](#)

12.7.9 Look-At



The Look-At plugin is used to rotate a Container so that it faces a defined direction. This is especially valuable in animations.

Note: This plugin is located in: Built-ins -> Container Plugins -> Global

This section also contains information on the following topic:

- [Look-At Properties](#)

Look-At Properties



- **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
- **Keep Size:**
- **Billboard Axis:** Only with **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

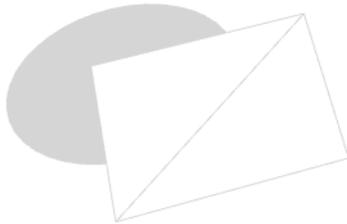
See Also

- [Create Animations](#)

12.7.10 Mask Source and Mask Target



With the Mask (Source and Target) plugin one Container can act as a mask for other Containers. The Mask Container is called the source and the Containers which are affected by the source mask are called targets. A Container affected by a mask becomes transparent where the mask covers it.



Up to eight different target layers and source layers are available for one object. This means that targets with the layer 1 set will be affected by all masks, which have the corresponding 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.

Shadow layers, [Global Settings](#)



Layers 7 and 8 may not be available because global shadow settings are enabled by default. Shadows use two layers of the mask plugin, but can be set to **off** in [Global Settings](#), to use all eight layers.

In [Global Settings](#) 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 a layer is selected to be drawn inverted the opposite situation is the case. Only where the mask covers the mask target will the target be visible.

Note: This plugin is located in: Built-ins -> Container Plugins -> Global

This section also contains information on the following topics:

- [Mask Source and Mask Target Properties](#)
- [To Add a Mask](#)

Mask Source and Mask Target Properties



- **Layer:** Sets the layer the mask should have an effect on. This allows the creation of more than one mask that will have an effect on the target
- **Inherited:** (Mask Source Only) When enabled (On) all Sub-Containers of the mask source will mask the target. If disabled (Off) all Sub-Containers of the mask source will not mask the target as it no longer inherits the mask properties
- **Force Visible:** (Mask Source Only)

To Add a Mask



1. Add three Containers to the Scene Tree.
2. Name them **source_1**, **source_2** and **target**.
3. Add a Font and the Mask Source plugin to the **source_1** Container.
4. Add a Rectangle and the Mask Source plugin to the **source_2** group.
5. Add a Rectangle, material (e.g. red) and the Mask Target plugin to the **target** Container.
6. Open the **source_1** [Transformation Editor](#):
 1. Set **Axis Center X** and **Y** to C
 2. Reset **Position X** and **Y** to 0.0.
7. Open the **source_1** [Text Editor](#):
 1. Enter some text (e.g. Vizrt).
8. Open the **source_1** Mask Source editor:
 1. Set **Layer** to 1.
9. Open the **source_2** [Transformation Editor](#):
 1. Set **Position X** to -120.0 and **Y** to -70.0.
10. Open the **source_2** Rectangle editor:
 1. Set **Width** and **Height** to 50.0.
11. Open the **source_2** Mask Source editor:
 1. Set **Layer** to 2.
12. Open the **target** Rectangle editor:
 1. Set **Width** to 300.0 and **Height** to 200.0.
13. Open the **target** Mask Target editor

1. Enable **Layer 1** and 2.



14. Add a Container as a Sub-Container of **source_2**.
15. Name it **source_inherited**.
16. Add a [Circle](#) to the **source_inherited** Container.
17. Open the **source_inherited** [Transformation Editor](#):
 1. Set **Position X** to 240.0.
18. Open the **source_inherited** Circle editor:
 1. Set **Radius** to 25.0.



19. Open the **source_2** Mask Source editor
 1. Disable **Inherited** (Off).
 - This should make the Circle disappear as it no longer inherits the Mask Source properties.

12.7.11 Lighting



Adds individual lighting per object.

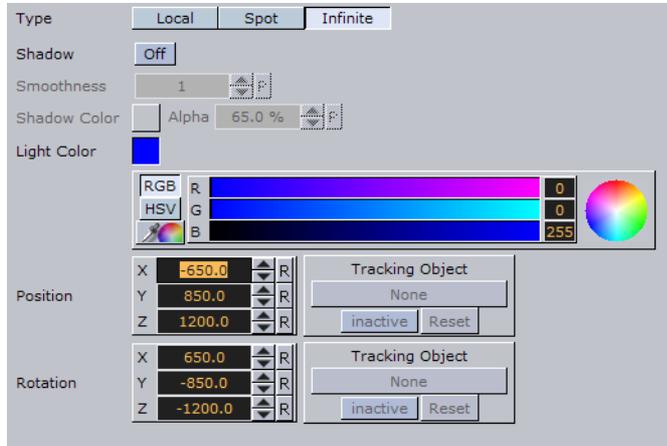
Note: This plugin is located in: Built-ins -> Container Plugins -> Global

This section also contains information on the following topics:

- [Lighting Properties](#)

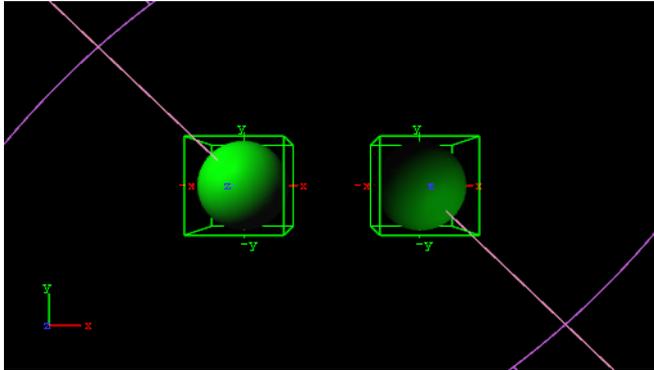
- [Mask Source and Mask Target Properties](#)
- [To Add a Mask](#)

Lighting Properties



- **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 Light



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 plugin 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.

12.7.12 Z-Sort



When this plugin is applied to a group, it disables back to front automatic sorting for the immediate children of that group. This enables designers to set the rendering order manually.

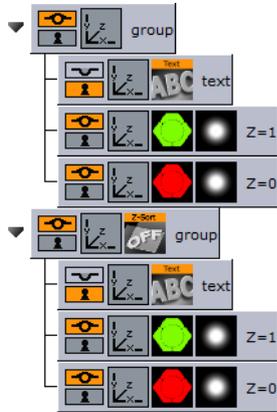
Note: This plugin is located in: Built-ins -> Container Plugins -> Global

This section also contains information on the following topics:

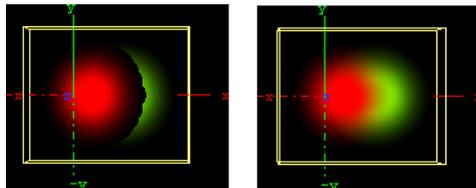
- [Z-Sort Description](#)
- [Transparency, Rendering Order Logic and Priority](#)

Z-Sort Description

To sort objects of a scene on the Z-axis is essential for any 3D program 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 plugin, 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 plugin 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.

Transparency, Rendering Order Logic and Priority

Questions:

- Why does a scene look different when a parent container is set to 99% transparency?
- Why are do some Containers disappear?
- When should the Z-Sort Off Plug-In be used?

Explanation:

The explanation to why changing an alpha plugin to 99% on a parent container, which affects the way objects are rendered, is the Rendering Order. When rendering polygonal Scenes with transparencies (like those created in Viz Artist) the rendering order is important with the current architecture of graphics cards.

The common practice for 3D applications is to render semi-transparent objects from back to front (or rather from far to near).

Example: Think of a scene that shows a city skyline through of a semi-transparent window. Lets assume that the window is rendered **before** the skyline. Some background color is applied to the screen, then the window is rendered. The graphics card will blend the color of the window with the background color (say, blue). In addition to color the graphics card will also mark the distance of the window's pixels from the camera for each pixel, so that farther away pixels will not obstruct the window. This technique for achieving obstruction between objects on per-pixel basis is called Depth-Buffer or Z-Buffer. Now render the skyline. The color of the skyline should have been blended with the color of the window, but it will not. The graphics card will reject every skyline pixel because its distance from the camera is farther away than the distance of the window pixels. This is a case where the positive role of the Depth-Buffer produces undesirable results.

To avoid this undesirable result, Viz Artist splits the rendering of the containers in the Scene into three parts:

1. The mask objects are rendered first to 'mark' the areas of masking on the screen.
2. The opaque objects are rendered front to back.
3. The transparent objects are rendered back to front.

Viz Artist/Engine decides whether an object is transparent based on some criteria, for example:

- If the texture of the object has an alpha channel
- If the material of the object has an alpha value less than 100%
- If the object has an inherited ALPHA function with an alpha value less than 100
- and more...

With all this said, rendering back to front is not perfect because the rendering order is not done per pixel; rendering is not even done per polygon. Viz Artist/Engine renders a scene per complete container; container order is based on the center of the container.

Example: Take a tilted flat map with icons. The map container is large, and its center is in front of some of the icons, but in the back of other icons. Assuming the icons have transparency the map will actually be rendered in between the icons! This behavior is visually incorrect as the map should be rendered first, then all the icons.

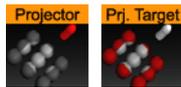
Solution

This behavior can be fine-tuned using the Z-Sort Off Plug-In. When this plugin is applied to a group, it will disable back to front automatic sorting for the immediate children of that group. This enables designers to set the rendering order manually.

Notes/Exceptions

- If an alpha plugin at 99% is applied on the top container it can force the whole collection of containers to be sorted back to front, and can solve the problem, but this is not a best practice. There is much more overhead rendering a scene like this. A more permanent solution would be to optimize the scene with the Z-Sort Off Plug-In.
- Opaque objects are better rendered from front to back because there is some performance gain for each obstructed pixel (it is not being processed after being rejected). With that being said opaque objects can be rendered at any order.

12.7.13 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 plugin does not work on text.

Note: This plugin is located in: Built-ins -> Container Plugins -> Global

This section also contains information on the following topics:

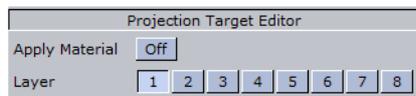
- [Projector Source and Projector Target Properties](#)
- [To project a texture onto an object](#)

Projector Source and Projector Target Properties



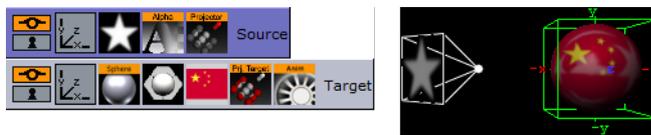
- **fovy:** Sets the field of view for the projector container. The higher the fovy is set, the larger the projected texture on the receiver will be. You can picture it as the zoom function of a slide projector.
- **Show:** Allows you to hide/show the projector object.
- **Layer:** Lets you select by 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.



- **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 plugin 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 plugin, material, image/texture and the Projector Target plugin to the Target container.
7. *Optional:* Animate the sphere.
8. Open the Projector editor and **disable** the **Show** property.

12.7.14 Script



The script functionality is a plugin. Scripts can be added to a Container or the [Scene Settings](#), and stored with the Scene. It is possible to archive and import scripts as part of a Scene Archive as with any other plugin.

Note: This plugin is located in: Built-ins -> Container Plugins -> Global

A script can also be saved as its own script plugin, and used in future Scenes (see [Create Script-based Plugins](#)). To save a script as a script plugin, drag a compiled script into the [Script Plugins](#) folder. The Script Plugins folder is a special folder that lets users create and name plugins.

Note: Script plugins are saved to <viz data folder>\ScriptPlugins.

A script plugin is not stored in the database, and is 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 plugin, and a script that is. As long as it is not converted the content of the script code is data within the 'Script' plugin, just like the width property is data within for example a geometry plugin. Once the script is converted to a plugin it is a plugin itself, like any other plugin.

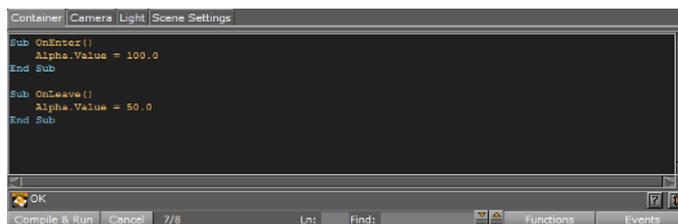
This section also contains information on the following topics:

- [Script Editor Properties](#)
- [Script Editor Context Menu](#)

See Also

- [Script Editor](#)
- [Script Plugins](#)

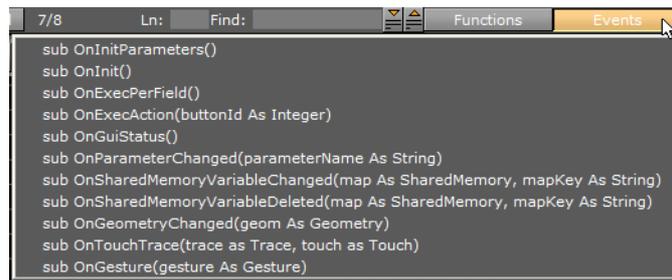
Script Editor Properties



- **OK** : Shows that all parts of the script entered are correct. If there is an error, it will show where the error is.
 - **Compile & Run** : Checks the code entered and translates it into byte code, so the script can be run. If there is an error, a message is shown in the text above.
 - **Cancel** : Ignores the last source code changes and compiles previous version.
 - **Ln:** : Go to the line number entered in the field.
 - **Find:** : Enter free text search. The up icon (▲) searches above the cursor location. The down icon (▼) searches below the cursor location.
- Functions** : Shows a list of used functions in the script (both built-in and self defined ones). Select a function in the list and go there in the script editor.



- **Events** : Shows a list of built-in events. Select one to add it to the script:

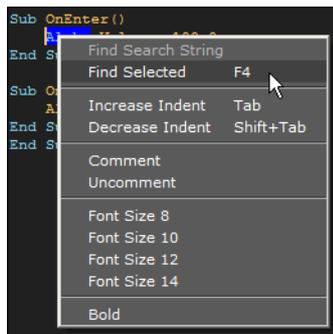


-  : Open the script documentation.
-  : Click the arrow buttons to open or close a script editor floating window.

Tip: The floating window can be scaled to any size.

Script Editor Context Menu

Right-click in the Script Editor to open the Script Editor context menu:



- **Find Search String:** Search for text entered in the **Find** input box
- **Find Selected (F4):** Searches for highlighted text
- **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:** Uncomment the selected text
- **Font Size:** Sets the font size
- **Bold:** Sets the font to bold

12.7.15 Shadow Caster and Shadow Receiver



The Shadow Caster and Receiver plugins 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 plugin 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.

Note: This plugin is located in: Built-ins -> Container Plugins -> Global

This section also contains information on the following topics:

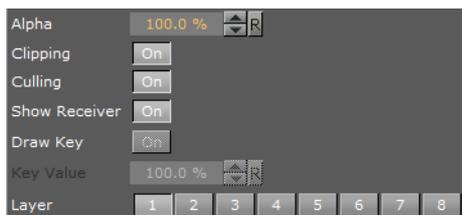
- [Shadow Caster Properties](#)
- [Shadow Receiver Properties](#)
- [To Cast a Shadow](#)

Shadow Caster Properties



- **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.

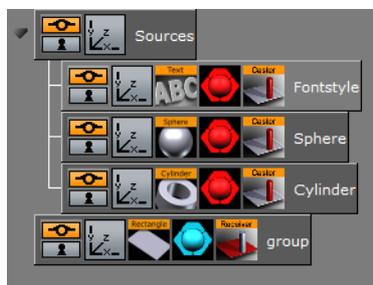
Shadow Receiver Properties



- **Alpha:** Sets the alpha value.

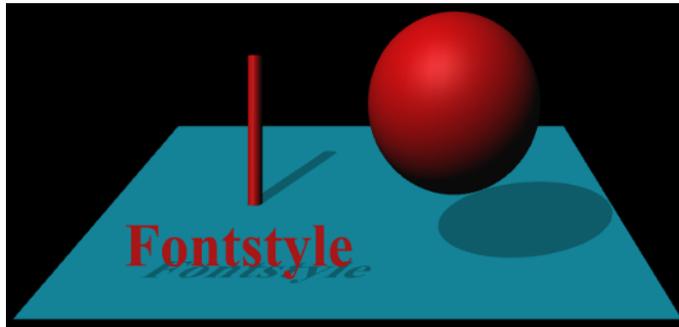
- **Clipping:** With clipping enabled the shadow will be cut on the border of the object. With clipping disabled, the shadow will be rendered onto the background outside the object, which normally looks wrong. Rendering with the clipping options set takes more resources than without. If you for instance 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 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
2. Name it **Sources**.
3. Add a Fontstyle and the Sphere and Cylinder Geometries as Sub-Containers of Sources.
4. Add a Material and the Shadow Caster plugin to each Sub-Container.
5. Add a Group Container at the same level as the **Sources** Container

6. Add a Rectangle Geometry, a Material and the Receiver plugin to it.



7. 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.

12.7.16 Synchronized Properties



The Synchronized Properties plugin 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 Artist/Engine 3.3 versions and works for all container properties and plugin parameters.

Note: This plugin is located in: Built-ins -> Container Plugins -> Global

This section also contains information on the following topics:

- [To Synchronize Scene Property Changes on Multiple Viz Engines](#)

To Synchronize Scene Property Changes on Multiple Viz Engines



- Add the Synch. Properties plugin on all containers that should be synchronized (no scripting is needed).

Note: The Synchronized Property plugin does not inherit to child containers to keep network traffic low.

See Also

- **To synchronize multiple Viz Engines** (see the [Viz Engine Administrator Guide](#))

12.7.17 Video Clip



The Video Clip plugin works in combination with the [Control Video](#) plugin. It allows you to use clip Key Frames in the stage for clip playback. When used with [Transition Logic](#) the Key Frames can be merged with the object.

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](#) plugin 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 is selected.

The plugin is commonly used in a video workflow with Viz One. It automatically adds a Clip Channel plugin. The clip channel has to be activated using the **Video Input** in the http://documentation.vizrt.com/viz-engine-guide/3.7/configuring_viz.html [Viz Configuration](#) (see the Viz Engine Administrator Guide).

Note: This plugin is located in: Built-ins -> Container Plugins -> Global

This section also contains information on the following topics:

- [Video Clip Properties](#)
- [To Add a Video Clip](#)

Video Clip Properties



- **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 Key Frames to the stage. Setting target to Texture adds the [Live Video Media Asset](#) texture plugin and default Key Frames 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 Viz Configuration to see if clip channels are activated.
2. Add the Video Clip plugin to 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 Key Frames can be added to play different clips.

See Also

- [Video Playback](#)

- [Animation Stage](#)
- [Toggle](#)
- [Control Video](#)
- [Transition Logic](#)

12.7.18 Window Mask



Window Mask limits the area into which a Sub-Container of the Window Mask plugin 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.

Note: This plugin is located in: Built-ins -> Container Plugins -> Global

This section also contains information on the following topics:

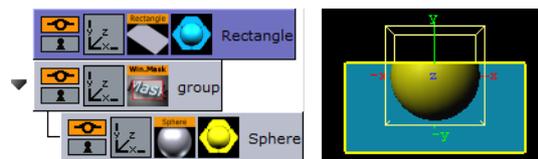
- [Window Mask Properties](#)
- [To Use the Window Mask](#)

Window Mask Properties



- **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 plugin 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.

12.8 Multi-Touch Plugins

The Multi-Touch plugins are used to take incoming touch events, process them and then trigger events, perform transformations on objects, telestrate, or trigger script events. The idea behind these plugins is to bundle often used or needed functionality and eliminate the need to write your own scripts or plugins for that.

They are great tools to bring Multi-Touch interactively fast and easily into a Scene.

The following Container plugins are located in the MultiTouchComp folder:

- [MtTelestrator Plugin](#)
- [MtButton Plugin](#)
- [Mt2D Control Plugin](#)
- [Mt3D Control Plugin](#)
- [MtNavigator Plugin](#)

The Container plugins require the [MtSensor Plugin](#) to be available to work. If a Multi-Touch plugin is added to a Scene it will automatically add the [MtSensor Plugin](#) to the Scene if it is not there.

The Multi-Touch plugins can communicate with Viz scripts, either through event handlers, Shared Memory variables, or Data Pool variables. The Plugins can also be controlled by triggering the 'push buttons' and/or setting the 'parameter' variables exposed by the Plugin. Specifics on what features are available for each Plugin are documented in the Plugin specific sections below.

In the Viz Artist editor interface the 'E' button toggles between enabling and disabling emulated sensor input using the mouse as well as direct input from a sensor.

The Plugins will work with multiple layers. Each Scene layer with multi-touch objects installed in it should also have an instance of the [MtSensor Plugin](#) automatically installed.

This section contains information on the following topics:

- [Plugin Event and Notification System](#)
- [MtTelestrator Plugin](#)
- [MtButton Plugin](#)
- [Mt2D Control Plugin](#)
- [Mt3D Control Plugin](#)
- [MtNavigator Plugin](#)

12.8.1 Plugin Event and Notification System

All the Multi-touch (Mt) Plugins share a common event notification module and associated configuration interface.

For all Plugins that generate events, events are dispatched up the Container hierarchy from the Plugin's Container to the first script found with a handler present for the event being generated.

For Plugins that optional update values in Shared Memory, choose which one of the three maps to have updated:

- System
- Scene
- Distributed

Each shared memory area updated by a Plugin is composed of one 'root' variable specified in the 'Shared Memory Prefix' field in the Plugins user interface. This 'root' variable is updated if any of its additional sub-fields are updated. The root variable contains an integer or 'update id'. Sub fields are identified by adding 'dot delimited' suffix on the root name. For example, if 'Shared Memory Prefix' is set to 'Name' the sub field 'field' would be identified as 'Name.field' in a Viz script.

For all Plugins that can set a Data Pool variable, each has a 'structure' defined in a '.dp' file. There is one '.dp' for each Plugin. Both the file and the name of the structure defined by it are the same as the name of the Plugin. For example, in the case of an MtButton, the file is 'MtButton.dp' and the name of the structure for sub element access is 'MtButton'.

12.8.2 MtTelestrator Plugin



The MtTelestrator plugin is used to do telestration on a Container level (used with the [GraffitiTex](#) plugin (added automatically to the same Container)).

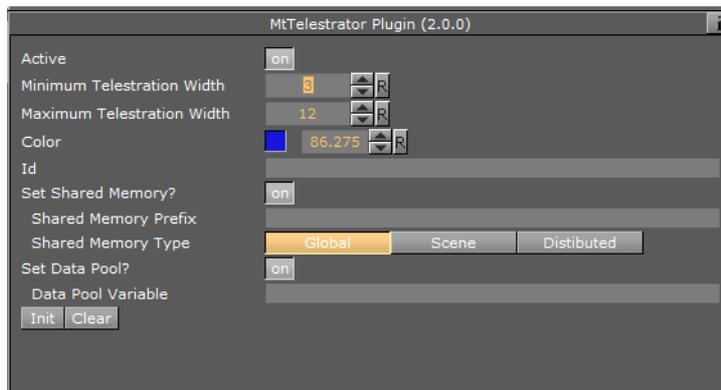
For scene-wide telestration the [MtSensor Plugin](#) has a telestration feature.

Note: This plugin is located in: Built-ins -> Container Plugins -> MultiTouchComp

This section contains information on the following topics:

- [MtTelestrator Configuration](#)
- [Events and Notifications](#)
- [To Create a Simple Scene with Telestration](#)

MtTelestrator Configuration



- **Active:** Select On (enable) or Off (disable) telestration input from the user
- **Minimum Telestration Width:** Minimum brush width to assign based on pressure applied to sensor
- **Maximum Telestration Width:** Maximum brush width to assign based on pressure applied to sensor
- **Color:** The color to assign for subsequent telestration drawing
- **Id:** To give additional context in the handler script, specify a string that will identify any notifications dispatched by this Plugin. This is often included as an argument for the event so a common script may handle events from a number of Plugins.
- **Set Shared Memory:** If ON enables shared memory to be updated for the Plugin notifications.
 - **Shared Memory Prefix:** This sets a 'prefix name' to be prepended to the shared memory variables maintained by the Plugins notifications. For Plugins that maintain multiple fields each field name has the prefix prepended to it followed by a dot '.' so as to mimic member access to an object, i.e: if the prefix is 'Obj' the fields 'field1' and 'field2' would be identified with the strings 'Obj.field1' and 'Obj.field2'. The shared memory field 'Obj' is also maintained and is simply an integer that is modified every time any of its 'subfields' is updated.
 - **Shared Memory Type:** Click either Global, Scene, or Distributed. This selects the shared memory area to update.
- **Set Data Pool:** Shows 'you wish' Plugin notifications to set a Data Pool variable
 - **Data Pool Variable:** Shows the name of the Data Pool variable to have set.
- **Init:** Clear current telestration and make Telestration active
- **Clear:** Clear the contents of the attached telestration

See Also

- [Events and Notifications](#)

Events and Notifications

The events dispatched by MtTelestrator are:

- OnTelestrationStateChange (sent when telestration is activated or deactivated):
 - PluginContainer as String
 - Id as String
 - TelestrationOn as Boolean

- OnTelestrationColorChange (sent when telestration is activated or deactivated):
 - PluginContainer as String
 - Id as String
 - PenColor as Color
- OnTelestrationClear (sent when telestration is cleared):
 - PluginContainer as String
 - Id as String
 - TelestrationOn as Boolean

The shared memory field names are:

- **PluginContainer:** the name of the Plugins container or "" if attached to a scene
- **Id:** the user ID string entered in the notification interface
- **PenColor:** the current pen color
- **Active:** 1 if telestration is active, 0 if not

The Data Pool structure updated is:

- MtTelestrator:
 - string PluginContainer
 - string Id
 - string PenColor
 - string Active

To Create a Simple Scene with Telestration

Note: If a touch screen is not available then go to [Viz Configuration](#) and set **Multi Touch input** to **Mouse**

1. Create a new Scene.
2. Add a new group Container.
3. Add a [Rectangle](#) Geometry to the Container.
4. Add the MtTelestrator plugin to the Container (*Container plugins > MultiTouchComp*).

Note: The GraffitiTex plugin is added to the Container automatically.

5. Set the **Active** parameter for the MtTelestrator and [GraffitiTex](#) to **ON** (in the Properties Panel).
6. Open the Scene in **On Air**. Use the touch input device to create freehand drawing.
7. Use the MtTelestrator and [GraffitiTex](#) plugin editors to modify the telestration affect, as required.

See Also

- [MtTelestrator Configuration](#)
- [GraffitiTex](#) Plugin

12.8.3 MtButton Plugin



The MtButton Plugin triggers an action when tapped or pressed.

The MtButton Plugin will make a container or its immediate children act as buttons. Several methods may be used to trigger the buttons. They are selected in the Viz Artist UI with the 'Trigger Type' radio button.

The MtButton can be enabled or disabled. When the button is disabled, any tap or stroke input it receives will not trigger the associated events. The button can be enabled or disabled through the Plugin parameters, or through a command. A button which is disabled after a stroke has pressed on a button will NOT prevent generation of events for a stroke that is currently active.

Note: This plugin is located in: Built-ins -> Container Plugins -> MultiTouchComp

This section contains information on the following topics:

- [MtButton Configuration](#)
- [Events and Notifications](#)
- [To Create a Simple Scene with the MtButton Plugin](#)

MtButton Configuration



- **Id:** To give additional context in the handler script, specify a string that will identify any notifications dispatched by this Plugin. This is often included as an argument for the event so a common script may handle events from a number of Plugins.
- **Set Shared Memory:** If ON enables shared memory to be updated for the Plugin notifications.
 - **Shared Memory Prefix:** This sets a 'prefix name' to be prepended to the shared memory variables maintained by the Plugins notifications. For Plugins that maintain multiple fields each field name has the prefix prepended to it followed by a dot '.' so as to mimic member access to an object, i.e: if the prefix is 'Obj' the fields 'field1' and 'field2' would be identified with the strings 'Obj.field1' and 'Obj.field2'. The shared memory field 'Obj' is also maintained and is simply an integer that is modified every time any of its 'subfields' is updated.
 - **Shared Memory Type:** Click either Global, Scene, or Distributed. This selects the shared memory area to update.
- **Set Data Pool:** Shows 'you wish' Plugin notifications to set a Data Pool variable
 - **Data Pool Variable:** Shows the name of the Data Pool variable to have set.

- **Active during screen wide telestration?:** If the button is to stay active when the screen-wide telestrator is on, set to On. The default is off, which means the button cannot be triggered when telestration is on.
- **Trigger type:**
 - **On Touch:** Sets the buttons to generate an 'onTrigger' event immediately when they are touched. No other events are generated in this mode.
 - **On Release:** Sets the buttons to generate an 'onTrigger' event when the last stroke present over a button is released. onStrokeEnter() and onStrokeLeave() events are generated as strokes enter and leave the button (see protocol below).
 - **On Tap:** Sets the buttons to generate an 'onTrigger' event when strokes recognized as 'taps' occur after a button is quickly pressed and released without movement or a finger is held down on a key on an object. onStrokeEnter() and onStrokeLeave() events are also generated in this mode.
 - **Taps Only:** Sets the buttons to emit an 'onTap' event when strokes recognized as a 'taps' occur. No other events are generated in this mode.
- **Hit Coordinate Type:** Click either Local, World or Screen. This selects the Coordinate system to use.

See Also

- [Events and Notifications](#)

Events and Notifications

The MtButton generates an **OnTap** event when it is tapped in 'Taps Only' mode:

```
sub OnTap(HitContainer as String, Id as StringHit, Point as VertexTaps as Integer, PluginContainer as String, FirstChildContainer as String)
```

```
end sub
```

- **HitContainer as String:** the name of the container that was 'hit'
- **Id as String:** the user ID string entered in the notification interface
- **HitPoint as Vertex:** the Point where the object was tapped in local coordinates
- **Taps as Integer:** the number of taps ie: 2 == double tap
- **PluginContainer as String:** name of the container the Plugin is attached to
- **FirstChildContainer as String:** first child container below the Plugin the tap occurred on or below

The MtButton generates an **OnTrigger** event when it is triggered in any other mode than 'Taps Only':

```
sub OnTrigger(HitContainer as String, Id as String, HitPoint as Vertex, Taps as Integer, PluginContainer as String, FirstChildContainer as String, StrokeId as Integer)
```

```
end sub
```

- **HitContainer as String:** the name of the leaf container that was 'hit'
- **Id as String:** the user ID string entered in the notification interface
- **HitPoint as Vertex:** the Point where the object was tapped in local coordinates
- **Taps as Integer:** the number of taps if tapped, ie: 2 == double tap
- **PluginContainer as String:** name of the container the Plugin is attached to

- **FirstChildContainer as String:** first child container below the Plugin the trigger occurred on or below
- **Strokeld as Integer:** Unique ID of stroke causing this event.

The MtButton generates an **OnStrokeEnter** event when it is triggered in any other mode than 'Taps Only':

```
sub OnStrokeEnter(Id as String, PluginContainer as String, FirstChildContainer as String, Strokeld as Integer, StrokeCount as Integer, Active as Integer)
```

```
end sub
```

- **Id as String:** the user ID string entered in the notification interface
- **PluginContainer as String:** name of the container the Plugin is attached to
- **FirstChildContainer as String:** first child container below the Plugin the stroke entered.
- **Strokeld as Integer:** Unique ID of stroke causing this event. Only valid from initial enter to final leave.
- **StrokeCount as Integer:** count of stroke on the button after entry. Will be '1' when the first stroke enters
- **Active as Integer:** Will be 0 if stroke went down on button. '1' if it was already down when entering.

The MtButton generates an **OnStrokeLeave** event when it is triggered in any other mode than 'Taps Only':

```
sub OnLeave(Id as String, PluginContainer as String, FirstChildContainer as String, Strokeld as Integer, StrokeCount as Integer, Active as Integer)
```

```
end sub
```

- **Id as String:** the user ID string entered in the notification interface
- **PluginContainer as String:** name of the container the Plugin is attached to
- **FirstChildContainer as String:** first child container below the Plugin the stroke exited.
- **Strokeld as Integer:** Unique ID of stroke causing this event. Only valid from initial enter to final leave.
- **StrokeCount as Integer:** count of stroke on the button after exit. Will be '0' when the last stroke exits
- **Active as Integer:** Will be 0 if stroke went up to cause leave. '1' if still down and 'rolled off'

The shared memory field names are:

- **HitContainer:** the name of the container that was 'hit'
- **Id:** the user ID string entered in the notification interface
- **HitPoint:** the Point where the object was tapped in the selected coordinate type.
- **Taps:** the number of taps, ie: 2 = double tap
- **PluginContainer:** name of the container the Plugin is attached to
- **FirstChildContainer:** first child container below the Plugin that tap occurred on or below
- **Strokeld:** unique ID of last trigger or tap.

The Data Pool structure updated is:

- MtButton:
 - string Id
 - string HitPoint
 - string HitContainer
 - string Taps
 - string PluginContainer
 - string FirstChildContainer
 - string Strokeld

To Create a Simple Scene with the MtButton Plugin

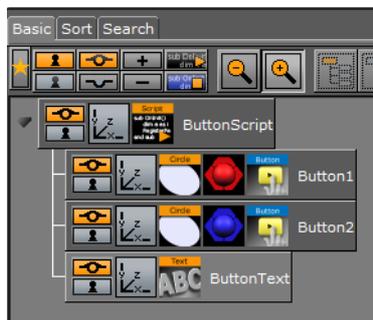
Note: If a touch screen is not available then go to **Communication** in [Viz Configuration](#) (see the [Viz Engine Administrator Guide](#)) and set **Multi Touch** input to **Mouse**

1. Create a new Scene.
2. Add a new group Container.
3. Name the group Container ButtonScript.
4. Add the [Script](#) plugin to the ButtonScript Container (Container plugins > Global).
5. Open the Script Editor and type this script:

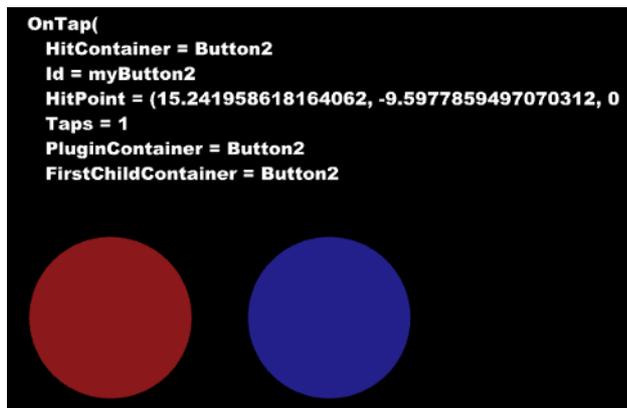
```
sub OnTap(HitContainer as String, Id as String, HitPoint as
Vertex, Taps as Integer, PluginContainer as String,
FirstChildContainer as String )
    dim bt as Container
    dim text as String
    bt = FindSubContainer("ButtonText")
    text = "OnTap(\n HitContainer = " & HitContainer
text &= "\n Id = " & Id
text &= "\n HitPoint = " & HitPoint
text &= "\n Taps = " & Taps
text &= "\n PluginContainer = " & PluginContainer
text &= "\n FirstChildContainer = " & FirstChildContainer &
"\n)"
    bt.Geometry.Text = text
end sub
```

6. Click **Compile and Run**.
7. Add three Sub-containers. Name them, in order:
 - Button1
 - Button2
 - ButtonText
8. Add a [Circle](#) Geometry to the Button1 and Button2 Containers.
9. Use the [Circle](#) Geometry Transformation editors to move the two circles so that both can be seen.
10. Right-click the Button1 and Button2 Containers, in turn, and select **Add Material ...** from the context-menu. Open the [Material Editor](#) and define a color for both Containers.

11. Add the MtButton plugin to the Button1 and Button2 Containers (*Container Plugin > MultiTouchComp*).



1. In the MtButton plugin editor:
 - Type 'myButton1' in the Button1 **Id** field
 - Type 'myButton2' in the Button2 **Id** field
 - In **Trigger Type** field click **On Tap** for both containers
2. Add a Font to the ButtonText Container.
3. Open the Scene in **On Air**.
4. Click on each button, in turn, and see the text change with each selection.
5. Use the MtButton plugin editor to modify the button's actions, as required.



See Also

- [MtButton Configuration](#)

12.8.4 Mt2D Control Plugin



The Mt2D Control Plugin moves, scales, and rotates an object in 2D.

The Mt2D Control plug allows multi-finger 2D manipulation of an object in the scene. When the object is touched, it will translate, rotate, and scale to keep the object underneath the user's fingers as closely as possible. It will move within the screen relative 2D plane through the object's center position defined by the objects orientation axes.

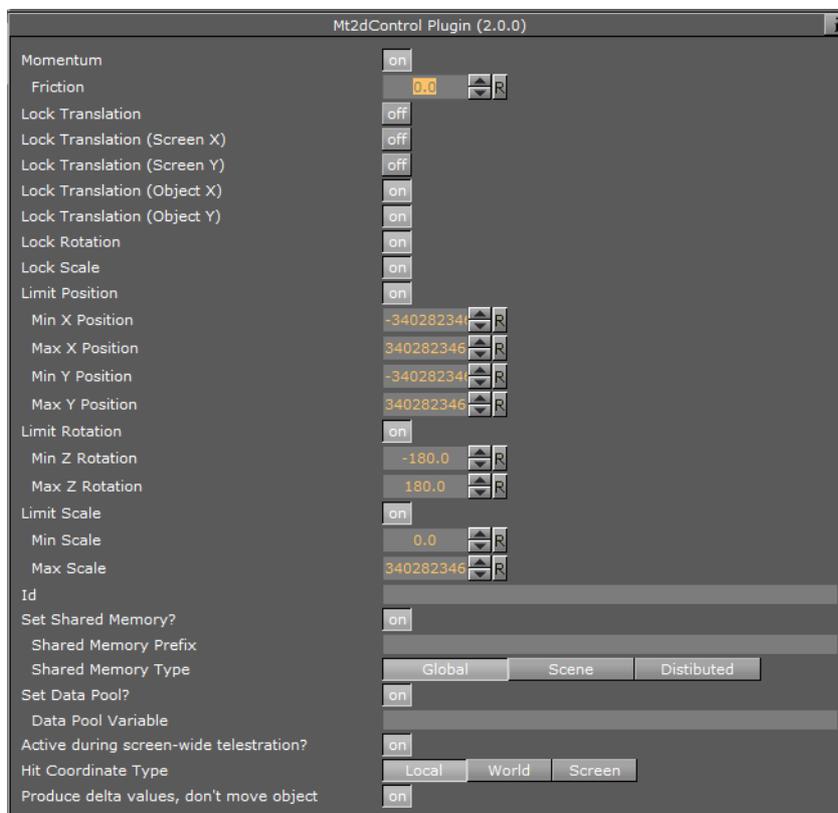
The container being manipulated must not have any local non-uniform scale. The Plugin will force y and z scale so they are equal to x scale when the object is touched. If non uniform scaling is required on object in the container, put the object's container in a group and scale the parent. Or put the Plugin on an identity group and insert the object to be moved as a child container. All transformations set on the manipulated container's parents will be preserved.

Note: This plugin is located in: Built-ins -> Container Plugins -> MultiTouchComp

This section contains information on the following topics:

- [Mt2D Control Configuration](#)
- [Events and Notifications](#)
- [To Create a Simple Scene with the Mt2D Control Plugin](#)

Mt2D Control Configuration



- **Momentum:** Enable movement or rotation with momentum after the object is released.
 - **Friction:** Set the amount of friction used when momentum is enabled

Note: Using this feature will conflict with any externally scripted or other animations applied to the container.

- **Lock Translation:** Allow only rotation and scaling of the object
- **Lock Translation (Screen X):** Prevent movement left-to-right on the screen

- **Lock Translation (Screen Y):** Prevent movement up-and-down on the screen
- **Lock Translation (Object X):** Prevent movement along the object's local X axis
- **Lock Translation (Object Y):** Prevent movement along the object's local Y axis
- **Lock Rotation:** Prevent rotation of the object
- **Lock Scale:** Prevent resizing of the object
- **Limit Position:** Limit the maximum and minimum position of the X and Y axis
- **Limit Rotation:** Limit the maximum and minimum rotation of the X and Y axis
- **Limit Scale:** Limit the maximum and minimum scale
- **Id:** To give additional context in the handler script, specify a string that will identify any notifications dispatched by this Plugin. This is often included as an argument for the event so a common script may handle events from a number of Plugins.
- **Set Shared Memory:** If ON enables shared memory to be updated for the Plugin notifications.
 - **Shared Memory Prefix:** This sets a 'prefix name' to be prepended to the shared memory variables maintained by the Plugins notifications. For Plugins that maintain multiple fields each field name has the prefix prepended to it followed by a dot '.' so as to mimic member access to an object, i.e: if the prefix is 'Obj' the fields 'field1' and 'field2' would be identified with the strings 'Obj.field1' and 'Obj.field2'. The shared memory field 'Obj' is also maintained and is simply an integer that is modified every time any of its 'subfields' is updated.
 - **Shared Memory Type:** Click either Global, Scene, or Distributed. This selects the shared memory area to update.
- **Set Data Pool:** Shows 'you wish' Plugin notifications to set a Data Pool variable
 - **Data Pool Variable:** Shows the name of the Data Pool variable to have set.
- **Active during screen wide telestration?:** If the button to is to stay active when the screen-wide telestrator is on, set to On. The default is off, which means the button cannot be triggered when telestration is on.
- **Hit Coordinate Type:** Click either Local, World or Screen. This selects the Coordinate system to use.
- **Produce delta values, don't move object:** Do not alter transform of object with attached controller, but instead provide 'deltas' in the event notifications.

Note: Rotation scale or position constraints do not operate in this mode.

See Also

- [Events and Notifications](#)

Events and Notifications

The events dispatched by Mt2D Control are:

OnGrab (when object is 'grabbed'):

```
sub ONGrab (HitContainer as String, Id as String, HitPoint as Vertex,
PluginContainer as String, FirstChildContainer as String)
```

```
end sub
```

- **HitContainer as String**
- **Id as String**

- **HitPoint as Vertex**
- **PluginContainer as String**
- **FirstChildContainer as String**

OnGrabUpdate (when object is moved):

sub **OnGrabUpdate (HitContainer as String, Id as String, Position as Vertex, Rotation as Vertex, Scale as Vertex, Pressure as Double)**

end sub

- **HitContainer as String**
- **Id as String**
- **Position as Vertex**
- **Rotation as Vertex**
- **Scale as Vertex**
- **Pressure as Double**

OnAnimation (dispatched when post grab animation (momentum) is started, followed by OnGrabUpdates):

sub **OnAnimation (HitContainer as String, Id as String)**

end sub

- **HitContainer as String**
- **Id as String**

OnGrabRelease (when both touch grab and momentum animation are complete):

sub **OnAnimation (HitContainer as String, Id as String)**

end sub

- **HitContainer as String**
- **Id as String**

The Shared Memory Field names updated are:

- **HitContainer** (the name of the container that was 'hit')
- **Id** (the user ID string entered in the notification interface)
- **HitPoint**
- **PluginContainer**
- **FirstChildContainer**
- **Position**
- **Rotation**
- **Scale**
- **Pressure**
- **State**

The Data Pool structure updated is:

- **Mt2D Control:**

- string HitContainer
- string Id
- string HitPoint
- string PluginContainer
- string FirstChildContainer
- string Position
- string Rotation
- string Scale
- string Pressure
- string State

To Create a Simple Scene with the Mt2D Control Plugin

Note: If a touch screen is not available then go to **Communication** in [Viz Configuration](#) (see the [Viz Engine Administrator Guide](#)) and set **Multi Touch** input to **Mouse**

1. Create a new Scene.
2. Add a new group Container.
3. Add a [Cube](#) Geometry to the Container.
4. Add the Mt2DControl plugin to the Container (*Container plugins > MultiTouchComp*).
5. Open the Scene in **On Air**. Move, rotate or scale the Rectangle with the touch input device.
6. Use the Mt2DControl plugin editor to modify the plugin behavior as required.

See Also

- [MtButton Configuration](#)

12.8.5 Mt3D Control Plugin



The Mt3D Control Plugin moves, scales, and rotates an object in 3D.

The Mt3D Control plugin provides an arc ball-like controller for the object it is attached to. When used with one finger, the object can be rotated. When multiple fingers are used, the object can be moved, rotated and scaled in a 2D plane (the same as the [Mt2D Control Plugin](#)).

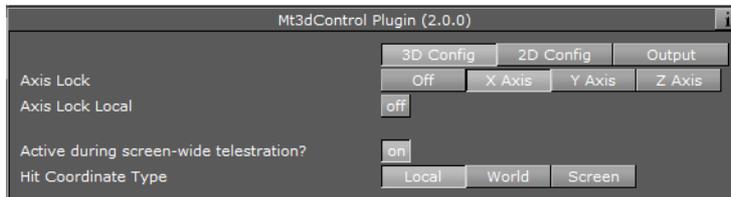
Note: This plugin is located in: Built-ins -> Container Plugins -> MultiTouchComp

This section contains information on the following topics:

- [Mt3D Configuration](#)
- [2D Configuration](#)

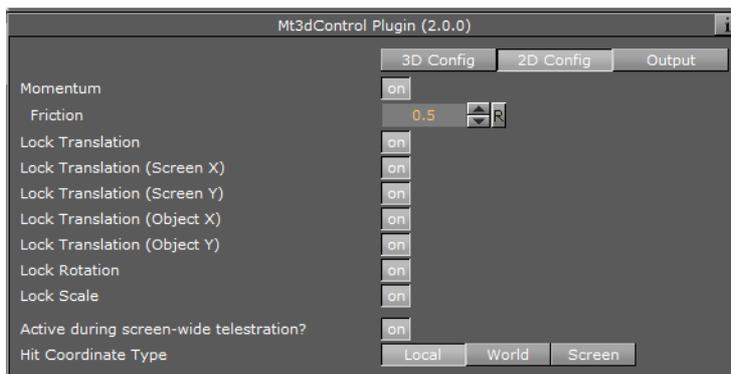
- [Output Configuration](#)
- [Events and Notifications](#)
- [To Create a Simple Scene with the Mt3D Control Plugin](#)

Mt3D Configuration



- **Config:** Select 3D Config
- **Axis Lock:** Constrain the rotation to a specific axis, or allow free rotation. Options are:
 - Off
 - X Axis
 - Y Axis
 - Z Axis
- **Axis Lock Local:** If axis lock is on, use the local axis instead of the world axis.
- **Active during screen wide telestration?:** If the button to is to stay active when the screen-wide telestrator is on, set to On. The default is off, which means the button cannot be triggered when telestration is on.
- **Hit Coordinate Type:** Click either Local, World or Screen. This selects the Coordinate system to use.

2D Configuration



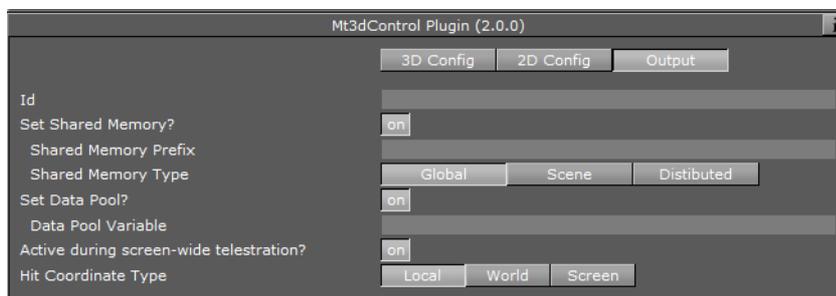
- **Config:** Select 2D Config
- **Momentum:** Enable movement or rotation with momentum after the object is released.
 - **Friction:** Set the amount of friction used when momentum is enabled

Note: Using this feature will conflict with any externally scripted or other animations applied to the container.

- **Lock Translation:** Allow only rotation and scaling of the object

- **Lock Translation (Screen X):** Prevent movement left-to-right on the screen
- **Lock Translation (Screen Y):** Prevent movement up-and-down on the screen
- **Lock Translation (Object X):** Prevent movement along the object's local X axis
- **Lock Translation (Object Y):** Prevent movement along the object's local Y axis
- **Lock Rotation:** Prevent rotation of the object
- **Lock Scale:** Prevent resizing of the object
- **Active during screen wide telestration?:** If the button to is to stay active when the screen-wide telestrator is on, set to On. The default is off, which means the button cannot be triggered when telestration is on.
- **Hit Coordinate Type:** Click either Local, World or Screen. This selects the Coordinate system to use.

Output Configuration



- **Config:** Select Output Config
- **Id:** To give additional context in the handler script, specify a string that will identify any notifications dispatched by this Plugin. This is often included as an argument for the event so a common script may handle events from a number of Plugins.
- **Set Shared Memory:** If ON enables shared memory to be updated for the Plugin notifications.
 - **Shared Memory Prefix:** This sets a 'prefix name' to be prepended to the shared memory variables maintained by the Plugins notifications. For Plugins that maintain multiple fields each field name has the prefix prepended to it followed by a dot '.' so as to mimic member access to an object, i.e: if the prefix is 'Obj' the fields 'field1' and 'field2' would be identified with the strings 'Obj.field1' and 'Obj.field2'. The shared memory field 'Obj' is also maintained and is simply an integer that is modified every time any of its 'subfields' is updated.
 - **Shared Memory Type:** Click either Global, Scene, or Distributed. This selects the shared memory area to update.
- **Set Data Pool:** Shows 'you wish' Plugin notifications to set a Data Pool variable
 - **Data Pool Variable:** Shows the name of the Data Pool variable to have set.
- **Active during screen wide telestration?:** If the button to is to stay active when the screen-wide telestrator is on, set to On. The default is off, which means the button cannot be triggered when telestration is on.
- **Hit Coordinate Type:** Click either Local, World or Screen. This selects the Coordinate system to use.

Events and Notifications

The events dispatched by Mt3dControl are:

OnGrab (when object is 'grabbed'):

sub OnGrab (**HitContainer as String, Id as String, HitPoint as Vertex, PluginContainer as String, FirstChildContainer as String**)

end sub

- **HitContainer as String**
- **Id as String**
- **HitPoint as Vertex**
- **PluginContainer as String**
- **FirstChildContainer as String**

OnGrabUpdate (when object is moved):

sub **OnGrabUpdate** (**HitContainer as String, Id as String, Position as Vertex, Rotation as Vertex, Scale as Vertex, Pressure as Double**)

end sub

- **HitContainer as String**
- **Id as String**
- **Position as Vertex**
- **Rotation as Vertex**
- **Scale as Vertex**
- **Pressure as Double**

OnGrabRelease (when object is "released"):

sub OnGrabRelease (HitContainer as String, Id as String)

end sub

- **HitContainer as String**
- **Id as String**

The Shared Memory Field names updated are:

- HitContainer (the name of the container that was "hit")
- Id (the user ID string entered in the notification interface)
- HitPoint
- PluginContainer
- FirstChildContainer
- Position
- Rotation
- Scale
- Pressure
- State

The Data Pool structure updated is:

- Mt3dControl:

- string HitContainer
- string Id
- string HitPoint
- string PluginContainer
- string FirstChildContainer
- string Position
- string Rotation
- string Scale
- string Pressure
- string State

To Create a Simple Scene with the Mt3D Control Plugin

Note: If a touch screen is not available then go to **Communication** in [Viz Configuration](#) (see the [Viz Engine Administrator Guide](#)) and set **Multi Touch** input to **Mouse**

1. Create a new Scene.
2. Add a new group Container.
3. Add a [Cube](#) Geometry to the Container.
4. Add the Mt3D Control plugin to the Container (*Container Plugins > MultiTouchComp*).
5. Open the Scene in **On Air**. Use the touch input device to transform the object.
6. Use the Mt3D Control plugin editor to modify the plugin behavior, as required.

See Also

- [Mt3D Configuration](#)
- [2D Configuration](#)
- [Output Configuration](#)

12.8.6 MtNavigator Plugin



The MtNavigator Plugin controls the Viz World Navigator Plugin.

The MtNavigator Plugin allows multi-touch free navigation over a map generated by the Curious World Maps Plugins. Currently it only supports the 'Flat' map model, and not the globe or other 3D models. It will send commands to the Navigator Plugin to change the position and distance of the map based on multi-touch input.

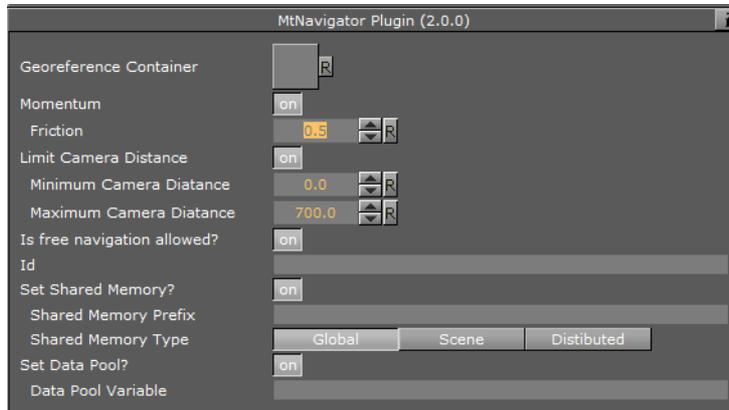
Note: This plugin is located in: Built-ins -> Container Plugins -> MultiTouchComp

This section contains information on the following topics:

- [MtNavigator Configuration](#)

- [Events and Notifications](#)

MtNavigator Configuration



- **Georeference Container:** The container which owns the Navigator plugin is specified here.
- **Momentum:** Enable movement or rotation with momentum after the object is released.
 - **Friction:** Set the amount of friction used when momentum is enabled
- **Limit Camera Distance:** Limit the maximum and minimum camera distance
- **Is free navigation allowed: Activates or deactivates navigation**
- **Id:** To give additional context in the handler script, specify a string that will identify any notifications dispatched by this Plugin. This is often included as an argument for the event so a common script may handle events from a number of Plugins.
- **Set Shared Memory:** If ON enables shared memory to be updated for the Plugin notifications.
 - **Shared Memory Prefix:** This sets a 'prefix name' to be prepended to the shared memory variables maintained by the Plugins notifications. For Plugins that maintain multiple fields each field name has the prefix prepended to it followed by a dot '.' so as to mimic member access to an object, i.e: if the prefix is 'Obj' the fields 'field1' and 'field2' would be identified with the strings 'Obj.field1' and 'Obj.field2'. The shared memory field 'Obj' is also maintained and is simply an integer that is modified every time any of its 'subfields' is updated.
 - **Shared Memory Type:** Click either Global, Scene, or Distributed. This selects the shared memory area to update.
- **Set Data Pool:** Shows 'you wish' Plugin notifications to set a Data Pool variable
 - **Data Pool Variable:** Shows the name of the Data Pool variable to have set.

See Also

- [Events and Notifications](#)

Events and Notifications

The events dispatched by MtNavigator are:

OnNavGrab (when navigable object is 'grabbed'):

sub **OnNavGrab** (HitContainer as String, Id as String, Offset as Vertex, Distance as Double, PluginContainer as String)

end sub

- **HitContainer as String**
- **Id as String**
- **Offset as Vertex**
- **Distance as Double**
- **PluginContainer as String**

OnNavGrabUpdate (when navigable object is moved):

sub **OnNavGrabUpdate** (HitContainer as String, Id as String, Offset as Vertex, Distance as Double, PluginContainer as String)

end sub

- **HitContainer as String**
- **Id as String**
- **Offset as Vertex**
- **Distance as Double**
- **PluginContainer as String**
- **OnNavGrabRelease** (when navigable object is released):
 - HitContainer as String
 - Id as String

The Shared Memory Field names updated are:

- **HitContainer:** the name of the container that was 'hit'
- **Id:** the user ID string entered in the notification interface
- **Offset:** the current lon/lat position in the Navigator
- **Distance:** the current distance in the Navigator
- **PluginContainer:** the name of the container this Plugin is attached to
- **Active:** 1 if 'grabbed' 0 if 'released'

The Data Pool structure updated is:

- MtNavigator:
 - string Id
 - string HitContainer
 - string Offset
 - string Scale
 - string PluginContainer
 - string Active

See Also

- [MtNavigator Configuration](#)

12.9 PixelFX

The PixelFX plugins are part of the [PixelFX plugins](#) set.

The following Container plugins are located in the PixelFX folder:

- [pxLensMulti](#)

12.9.1 pxLensMulti



pxLensMulti serves as a parent container for all the other [PixelFX Plugins](#) Geometry 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 will make sure that all the shapes it handles are constantly facing the camera.

Note: pxLensMulti creates a notional line along which all shapes are scattered.

Note: This plugin is located in: Built-ins -> Container Plugins -> PixelFX

This section contains information on the following topics:

- [pxLensMulti Properties](#)

pxLensMulti Properties

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 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 [Animate 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.

See Also

- [pxColorWorks](#)
- [PixelFX Plugins](#) in Geom Plugins
- [PixelFXLenseFlare](#) in Shader Plugins

12.10 Presenter

The following container plugins are located in the Presenter folder:

- [Bar](#)
- [Bar Value](#)
- [Bar Values](#)
- [Pie Slice](#)
- [Pie Values](#)

12.10.1 Bar



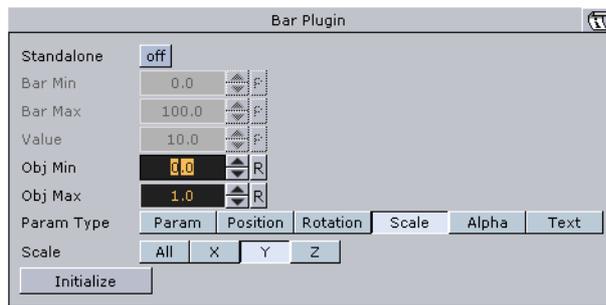
The Bar plugin tells the [Bar Values](#) plugin that the container is part of the bar chart and it also tells which elements of the geometry on the container to control. Each container that is to be a bar, must hold a geometry and a Bar plugin. Bar can be used as a standalone plugin.

.....
Note: This plugin is located in: Built-ins -> Container Plugins -> Presenter

This section contains information on the following topics:

- [Bar Properties](#)

Bar Properties



- **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 required 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 plugin. 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 required 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 plugin to initialize and gain control of the parameters.

See Also

- [Bar Value](#)
- [Bar Values](#)
- [Visual Data Tools](#)

12.10.2 Bar Value



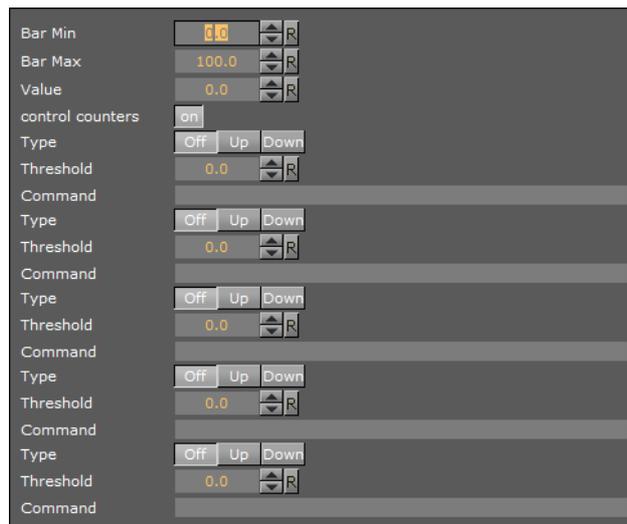
Bar Value controls the value of all Bar plugins and Counters of its child nodes. In addition a set of commands can be specified that are executed when the value passes certain thresholds.

Note: This plugin is located in: Built-ins -> Container Plugins -> Presenter

This section contains information on the following topics:

- [Bar Value Properties](#)
- [To Create a Bar that Shows its Value](#)

Bar Value Properties



- **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 shown in all Sub-Containers with Counter plugins.
- **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 Artist/Engine command that will be executed when the threshold is passed.

To Create a Bar that Shows 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 (Bottom).
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](#)
- [Visual Data Tools](#)

12.10.3 Bar Values



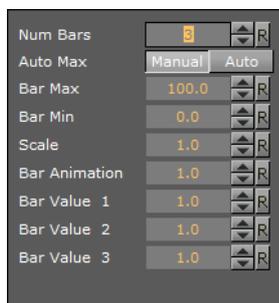
The Bar Values plugin communicates with all Sub-Containers that hold a Bar plugin and lets you set values for each of them.

Note: This plugin is located in: Built-ins -> Container Plugins -> Presenter

This section contains information on the following topics:

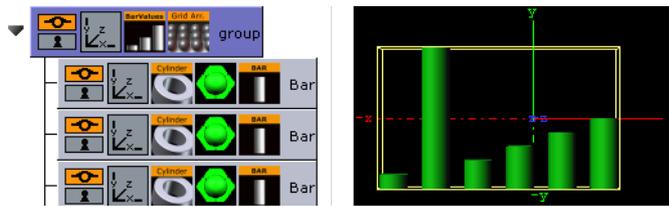
- [Bar Values Properties](#)
- [To Create a Bar Chart with the Bar Values Plugin](#)

Bar Values Properties



- **Num Bars:** Sets the number of bars to be controlled by the Bar Values plugin. 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 with the Bar Values Plugin



1. Add a group container to the scene tree, and add the Bar Values and [Grid Arrange](#) plugin 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](#) plugin 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

- [Visual Data Tools](#)
- [Control Bars](#)
- [Bar](#)
- [Bar Value](#)
- [Pie Slice](#)
- [Pie Values](#)

12.10.4 Pie Slice



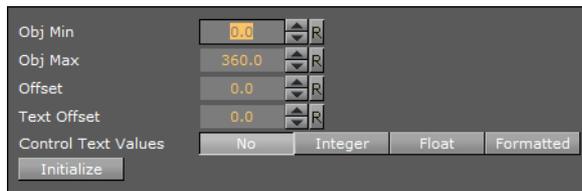
Use the Pie Slice plugin with the [Pie Values](#) plugin to create and animate a pie chart of up to 20 slices.

Note: This plugin is located in: Built-ins -> Container Plugins -> Presenter

This section contains information on the following topics:

- [Pie Slice Properties](#)

Pie Slice Properties



- **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 required 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

- [Visual Data Tools](#)
- [Control Bars](#)
- [Bar](#)
- [Bar Value](#)
- [Bar Values](#)
- [Pie Values](#)

12.10.5 Pie Values



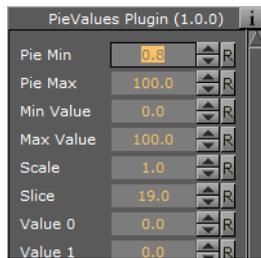
Use the Pie Values plugin with the [Pie Slice](#) plugin to create and animate a pie chart of up to 20 slices.

Note: This plugin is located in: Built-ins -> Container Plugins -> Presenter

This section contains information on the following topics:

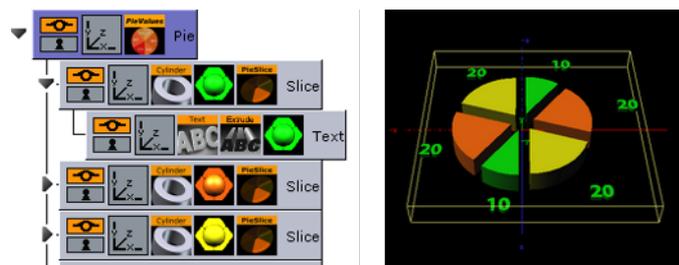
- [Pie Values Properties](#)
- [To Create a Pie Chart with the Pie Values Plugin](#)

Pie Values Properties



- **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 shown 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 with the Pie Values Plugin



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 plugin 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 plugin, material and PieSlice plugin 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](#) plugin 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

- [Visual Data Tools](#)

- [Control Pie](#)
- [Bar](#)
- [Bar Value](#)
- [Bar Values](#)
- [Pie Slice](#)

12.11 pxColorWorks

The color correction plugins, which are part of the [PixelFX plugins](#) set, are generally fast and efficient, however there is a certain performance penalty when animating them in a stacked scenario.

Note: The PixelFX plugins are a separate package from the standard Viz Artist installation, but function within the Viz Artist environment.

The following Container plugins are located in the pxColorWorks folder:

- [pxAddSubtract](#)
- [pxBlackAndWhite](#)
- [pxBrightContrast](#)
- [pxColorMatch](#)
- [pxGamma](#)
- [pxHueRotate](#)
- [pxMask](#)
- [pxSaturation](#)
- [pxStack](#)
- [pxTint](#)

See Also

- [PixelFX](#) in [Shader Plugins](#)
- [PixelFX Plugins](#) in [Geometry Plugins](#)

12.11.1 PixelFX plugins

The PixelFX plugins affect pixels and geometry, and produce various special effects. They do 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 show certain properties of the plugin in question.



- At the bottom left, there is a performance bar, with three blocks (like a cellphone battery indicator). One block means the plugin will not have a drastic performance effect, and three means the plugin is taxing and should therefore be used with caution.
- At the bottom right the orange stack shows if the plugin is stackable, meaning 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 plugins 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 a Scene, therefore perform a color correction to the whole scene. Another example is [pxNoise](#) - apply it to the root level container of a scene or a whole tree branch to add a kind of film-grain effect.

One of the parameters that feature in a number of the PixelFX 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.



12.11.2 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.

Note: This plugin is located in: Built-ins -> Container Plugins -> pxColorWorks

12.11.3 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.

Note: This plugin is located in: Built-ins -> Container Plugins -> pxColorWorks

12.11.4 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.

Note: This plugin is located in: Built-ins -> Container Plugins -> pxColorWorks

12.11.5 pxColorMatch

pxColorMatch lets you replace one color with another. It works on images and tree branches holding geometry alike.

Note: This plugin is located in: Built-ins -> Container Plugins -> pxColorWorks

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.

12.11.6 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.

Note: This plugin is located in: Built-ins -> Container Plugins -> pxColorWorks

12.11.7 pxHueRotate



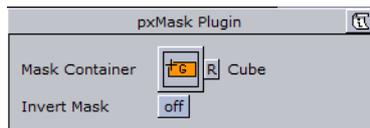
pxHueRotate allows you to adjust the hue rotation on an image. It can be applied to Everything, Shadows, Midtones and Highlights.

Note: This plugin is located in: Built-ins -> Container Plugins -> pxColorWorks

12.11.8 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.



Note: This plugin is located in: Built-ins -> Container Plugins -> pxColorWorks

12.11.9 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.

Note: This plugin is located in: Built-ins -> Container Plugins -> pxColorWorks

12.11.10 pxStack



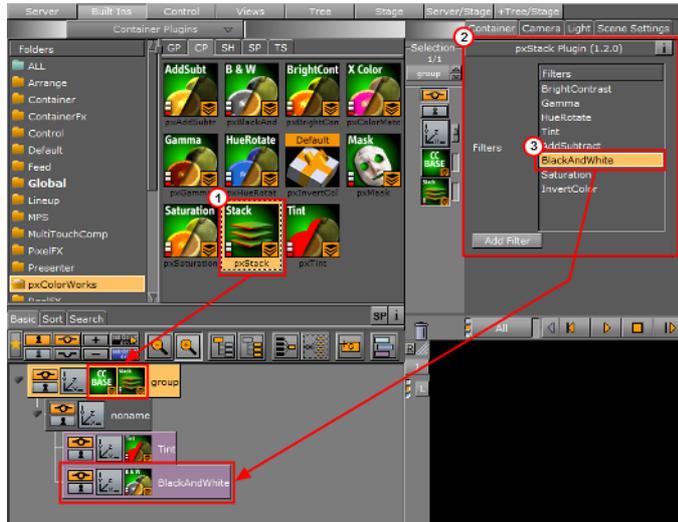
Note: This plugin is located in: Built-ins -> Container Plugins -> pxColorWorks

This section contains information on the following topics:

- [To apply to the pxStack plugin](#)

To apply to the pxStack plugin

1. Add the pxStack plugin to the required Container in the Scene Tree (1).



Note: When the pxStack plugin is added to a Container, it automatically adds the [pxCCBase](#) plugin to the same Container. Also, it automatically generates a child Container to host the color correction nodes you will define later.

2. Click the pxStack icon in the Container. The plug-in editor (2) shows a list of color correction filters.
3. Click on a color correction filter.
4. Click **Add Filter**. The color correction filter is added as a new Container under the 'noname' Container in the Scene Tree (3). This termed as a 'Color Correction Node'.
5. Repeat to add more color correction filters. Each color correction filter applied is added (stacked) as a new Color Correction Node under the 'noname' Container in the Scene Tree.
6. Click on a Color Correction Node to open its editor. Modify the effect properties as required.
7. 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.

Note: To disable or enable a Color Correction Node click the Locked/Unlock icon in the top left of the Container (see [Container Icons](#)).

Note: To remove a color correction function, delete the plugin.

12.11.11 pxTint



pxTint allows you to add a color tint and apply it to an image in the required amount. It can be applied to Everything, Shadows, Midtones and Highlights. You also have the option to preserve highlights, or not.

Note: This plugin is located in: Built-ins -> Container Plugins -> pxColorWorks

12.12 RealFX

The RealFX plugin set enables you to create particle effects in Viz Artist.

See Also

- [RealFX](#) in [Shader Plugins](#)
- [RealFX Plugins](#) in [Geometry Plugins](#)

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.

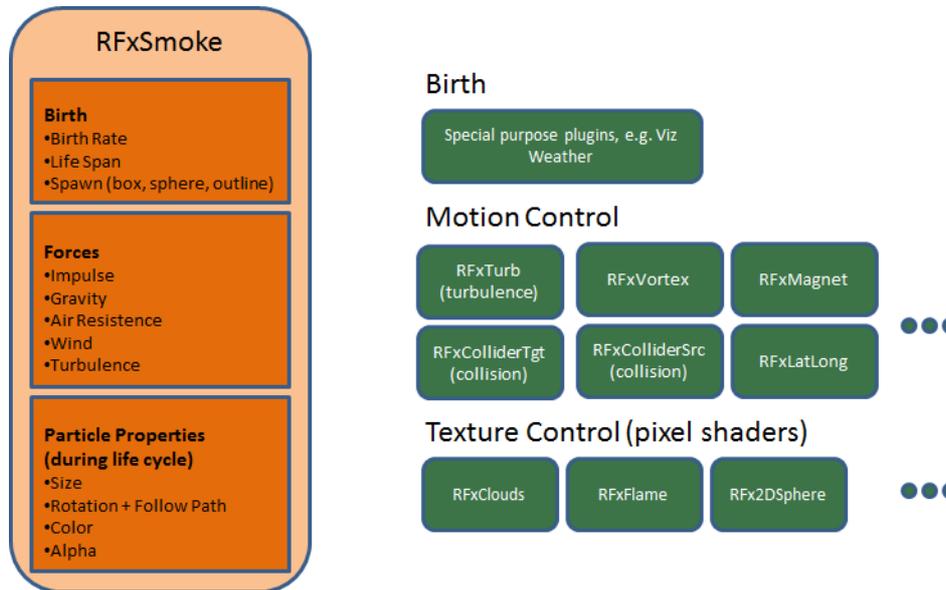
The particle effects in Viz Artist/Engine run in real-time, meaning that there are a few inherent constraints that must be taken into account when considering best practices for employing this plugin set. For example, there is a trade-off between the number of particles and performance optimization; 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

RealFX Architecture



The following Container plugins are located in the RealFX folder:

- [RFXColliderSrc](#)
- [RFXColliderTgt](#)
- [RFXLatLong](#)
- [RFXMagnet](#)
- [RFXTurb](#)
- [RFXVortex](#)

12.12.1 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.

Note: This plugin is located in: Built-ins -> Container Plugins -> RealFX

Adjust the following parameters as required to achieve the required 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](#)

12.12.2 RFxColliderTgt



Located under the Container Plugins tab, RFxColliderTgt is used in conjunction with [RFxTurb](#) 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: This plugin is located in: Built-ins -> Container Plugins -> RealFX

Adjust the following parameters as required to achieve the required effect:

- Flag 1-8 (on/off)

See Also

- [RFxSmoke](#)
- [RFxTurb](#)

12.12.3 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.

Note: This plugin is located in: Built-ins -> Container Plugins -> RealFX

Adjust the following parameters as required to achieve the required 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

- [pxColorWorks](#)

12.12.4 RFXMagnet



Located under the Container Plugins tab, RFXMagnet creates motion patterns that resemble a magnetic field.

Note: This plugin is located in: Built-ins -> Container Plugins -> RealFX

Adjust the following parameters as required to achieve the required effect:

- Amount
- Center X
- Center Y
- Center Z
- Power Law
- Min Distance Clamp

See Also

- [RFXSmoke](#)

12.12.5 RFXTurb



Located under the Container Plugins tab, RFXTurb applies turbulence-like forces to particles. It can be used to create the effect of random changes in wind force and direction.

Note: This plugin is located in: Built-ins -> Container Plugins -> RealFX

Adjust the following parameters as required to achieve the required effect:

- Amount
- Inertia
- Wavelength
- Update Rate

See Also

- [RFxVortex](#)

12.12.6 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.

Note: This plugin is located in: Built-ins -> Container Plugins -> RealFX

Adjust the following parameters as required to achieve the required effect:

- Amount
- Inertia
- Center X
- Center Y
- Center Z
- Rot X
- Rot Y

See Also

- [RFxSmoke](#)

12.13 Script Plugins

The Script Plugins is an empty folder to save created [Script](#) plugins.

A script can be saved as its own script plugin, and used in future Scenes (see [Create Script-based Plugins](#)). To save a Script Plugin, drag a compiled script into the Script Plugins folder.

Note: Script plugins are saved to <viz data folder> \ScriptPlugins.

See Also

- [Script](#) plugin
- [Script Editor](#)

12.14 SplineFX

The following container plugins are located in the SplineFX folder:

- [2D Follow](#)

12.14.1 2D Follow



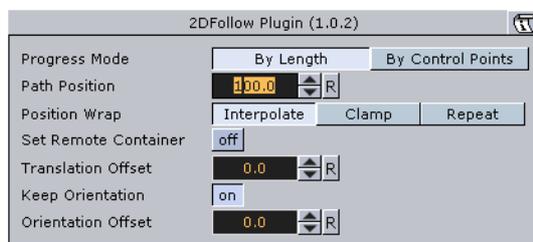
The 2D Follow function works together with the [2D Ribbon](#). What the plugin basically does is to automatically create an animation path that follows the form of a 2D Ribbon. The 2DRibbon must be created first.

Note: This plugin is located in: Built-ins -> Container Plugins -> SplineFX

This section contains information on the following topics:

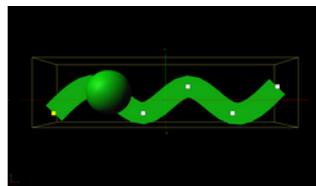
- [2D Follow Properties](#)
- [To Set up the 2D Follow Plugin](#)

2D Follow Properties



- **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 plugin to the container.

- Open the 2D Follow editor and animate its Path Position value from 0.0 to 100.0.

See Also

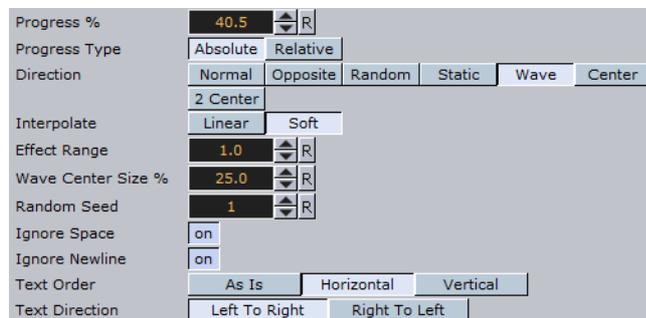
- [2D Ribbon](#)
- [Create Animations](#)

12.15 TextFX

The following Container plugins are located in the TextFX folder:

Common Text FX Properties	Text FX Jitter Position
Convert Case	Text FX Jitter Scale
Mark Text	Text FX Plus Plus
Text FX Alpha	Text FX Rotate
Text FX Arrange	Text FX Scale
Text FX Color	Text FX Size
Text FX Color Per Vertex	Text FX Slide
Text FX Emoticons	Text FX Vertex Explode
Text FX Explode	Text FX Write
Text FX Jitter Alpha	

12.15.1 Common Text FX Properties



The following properties are common to most of the Text 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 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.

12.15.2 Convert Case



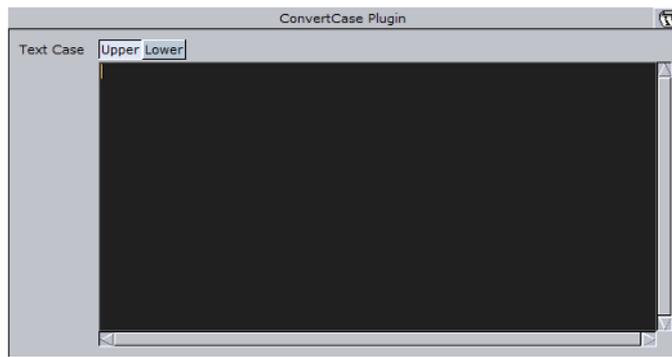
Converts ASCII strings to upper or lower case. Current container must hold a string.

Note: This plugin is located in: Built-ins -> Container Plugins -> TextFx

This section contains information on the following topics:

- [Convert Case Properties](#)
- [To Convert a String to Upper or Lower Case](#)

Convert Case Properties



- **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 plugin 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.

12.15.3 Mark Text



The Mark Text plugin allows you to underline words in a text string with a.

Note: This plugin is located in: Built-ins -> Container Plugins -> TextFX

This section contains information on the following topics:

- [Mark Text Properties](#)
- [To Mark Text](#)

Mark Text Properties

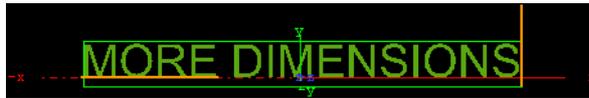


- **Text to Mark:** Sets the text that should be marked.

Note: 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 plugin 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 plugin.

12.15.4 Text FX Alpha



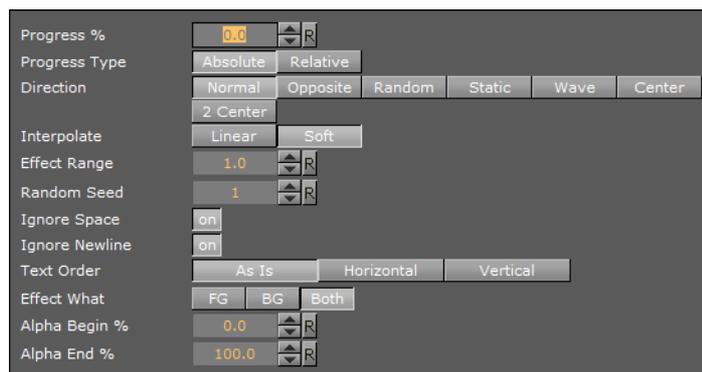
The Text FX Alpha plugin is a text effect that creates a fade in effect for the text characters. The effect sequence can be set to go many different ways. A fade in and out effect can also be achieved.

Note: This plugin is located in: Built-ins -> Container Plugins -> TextFX

This section contains information on the following topics:

- [Text FX Alpha Properties](#)

Text FX Alpha Properties



- **Alpha Begin %:** Sets the alpha level of the character at 0% effect.
- **Alpha End %:** Sets the alpha level of the character at 100% effect.

See also [Common Text FX Properties](#).

12.15.5 Text FX Arrange



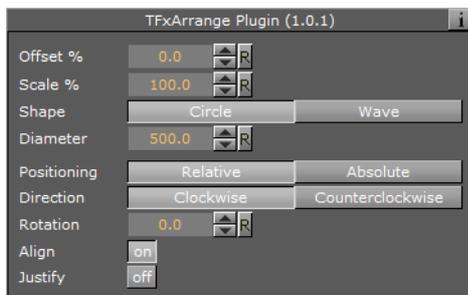
The Text FX Arrange plugin allows you to arrange characters in either a circular- or a wave shape. The characters can be animated on the selected shape by animating the offset value.

Note: This plugin is located in: Built-ins -> Container Plugins -> TextFX

This section contains information on the following topics:

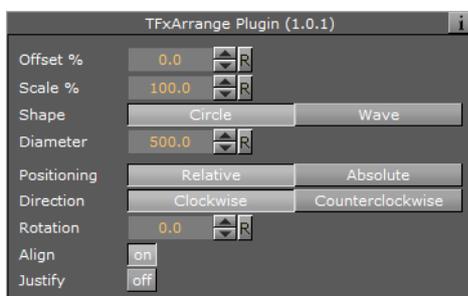
- [Text FX Arrange Properties](#)
 - [Circle Parameters](#)
 - [Wave Parameters](#)

Text FX Arrange Properties



- **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. Options are
 - **Circle**, or
 - **Wave**

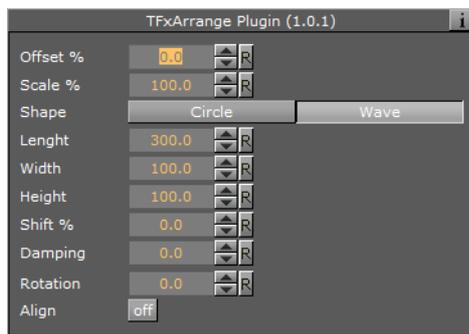
Circle Parameters



- **Diameter:** Set the diameter of the circular shape
- **Positioning:** Define the position of the text on the circular shape:

- **Relative:** The text spacing of the text object is maintained
- **Absolute:** The text is evenly distributed out on the circle
- **Direction:** Set the direction of the text on the circle to either:
 - **Clockwise,** or
 - **Counterclockwise.**
- **Rotation:** Rotate the characters on the X-axis
- **Align:** Rotate the characters on the Z-axis to align the X-axis with the tangent of the shape at the characters position

Wave Parameters



- **Length:** Set the wave length. A high value creates many and small waves, a low value creates fewer and bigger waves
- **Width:** Set the width of the wave
- **Height:** Set 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
- **Rotation:** Rotate the characters on the X-axis
- **Align:** Rotate the characters on the Z-axis to align the X-axis with the tangent of the shape at the characters position

12.15.6 Text FX Color



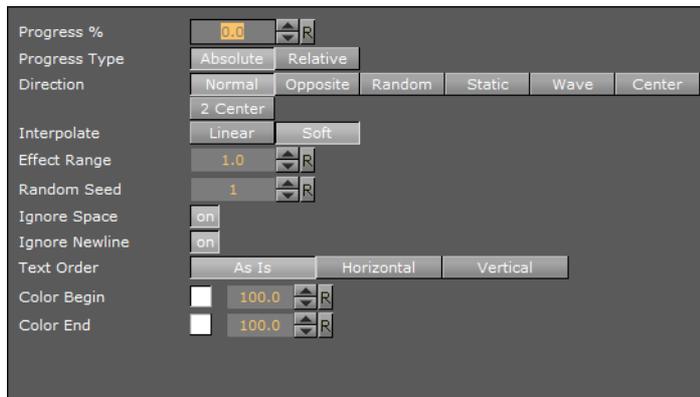
Adds a color effect to the text.

Note: This plugin is located in: Built-ins -> Container Plugins -> TextFX

This section contains information on the following topics:

- [Text FX Color Properties](#)

Text FX Color Properties



- **Color Begin:** Sets the initial color before applying the effect.
- **Color End:** Sets the color the effect should apply.

See also [Common Text FX Properties](#).

12.15.7 Text FX Color Per Vertex



Adds a color per vertex effect to the text.

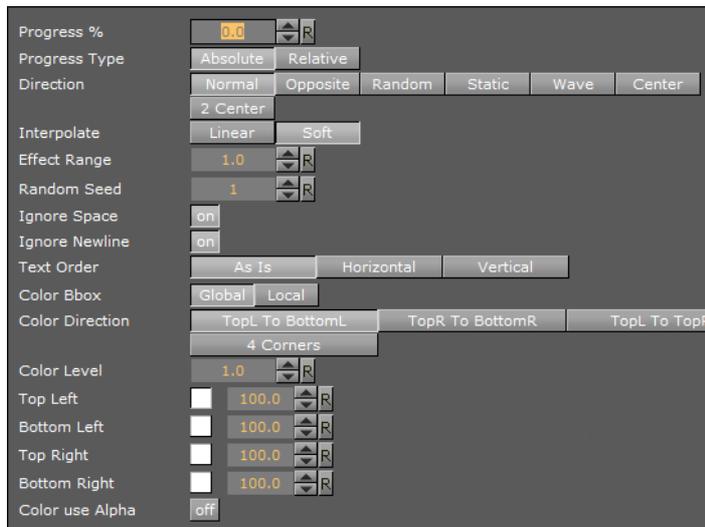
Note: Works only if the text is set to texture (does not work with geometry text).

Note: This plugin is located in: Built-ins -> Container Plugins -> TextFX

This section contains information on the following topics:

- [Text FX Color Per Vertex Properties](#)

Text FX Color Per Vertex Properties



- **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.

See also [Common Text FX Properties](#).

12.15.8 Text FX Emoticons



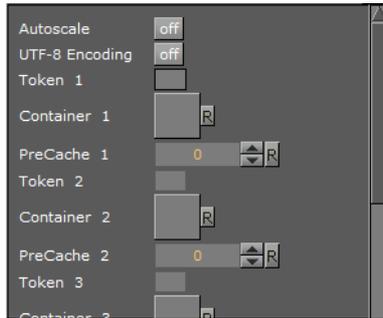
The Text FX Emoticons allows you to create an emoticon container, add it to the Emoticons plugin and replace characters (tokens) in a text string with an assigned container.

IMPORTANT! This plugin is located in: Built-ins -> Container Plugins -> TextFX

This section contains information on the following topics:

- [Text FX Emoticons Properties](#)

Text FX Emoticons Properties



- **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 Artist/Engine to cache the objects.

12.15.9 Text FX Explode



The Text FX Explode plugin 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.

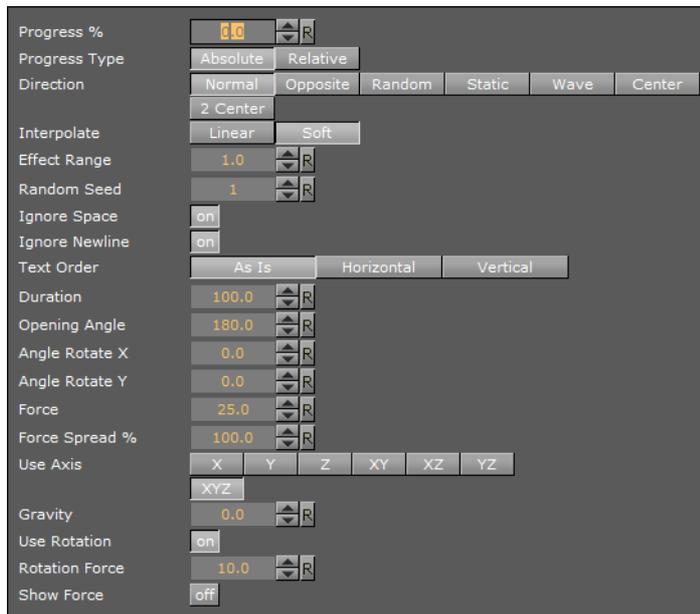
Note: Works only if the text is set to texture (does not work with geometry text).

Note: This plugin is located in: Built-ins -> Container Plugins -> TextFX

This section contains information on the following topics:

- [Text FX Explode Properties](#)

Text FX Explode Properties



- **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 Scene Editor.

See also [Common Text FX Properties](#).

See Also

- [Text FX Vertex Explode](#)

12.15.10 Text FX Jitter Alpha



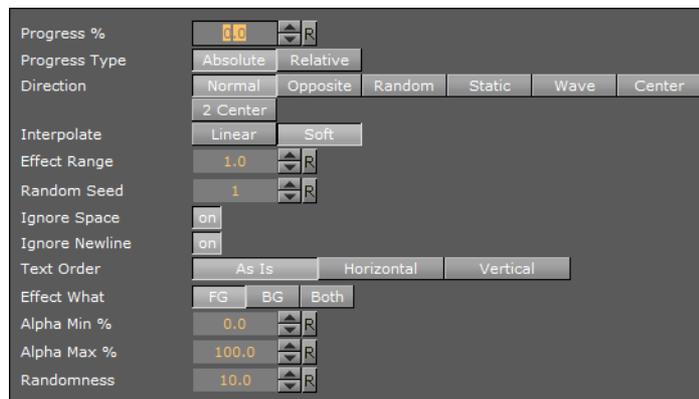
The Text FX Jitter Alpha plugin creates a jittering motion of the characters by randomly changing the alpha value of each character. The degree of change and the start sequence of the jittering can be altered. To use the plugin, add it onto a container with a font. To create an animation, animate the progress. Other values can of course be animated as well.

Note: This plugin is located in: Built-ins -> Container Plugins -> TextFX

This section contains information on the following topics:

- [Text FX Jitter Alpha Properties](#)

Text FX Jitter Alpha Properties



- **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.

12.15.11 Text FX Jitter Position



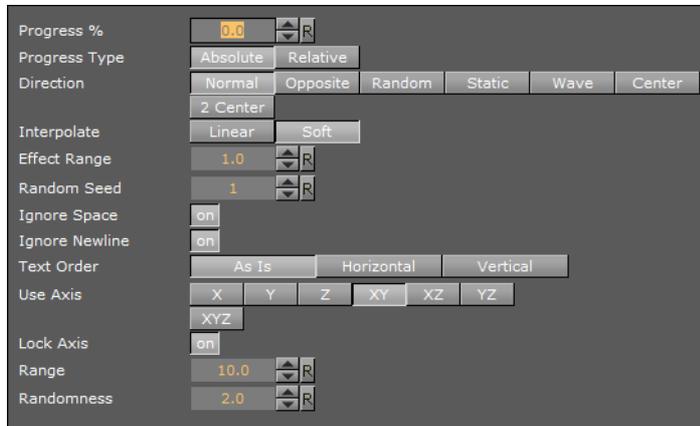
The Text FX Jitter Position plugin creates an jittering motion of the characters by randomly changing the position of each character. The degree of position change and the starting sequence of the jittering can be altered. To use the function add it onto a container which holds a font. To create an animation, animate the progress. Other values can of course be animated as well.

Note: This plugin is located in: Built-ins -> Container Plugins -> TextFX

This section contains information on the following topics:

- [Text FX Jitter Position Properties](#)

Text FX Jitter Position Properties



- **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 shown 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 Text FX Properties](#).

12.15.12 Text FX Jitter Scale



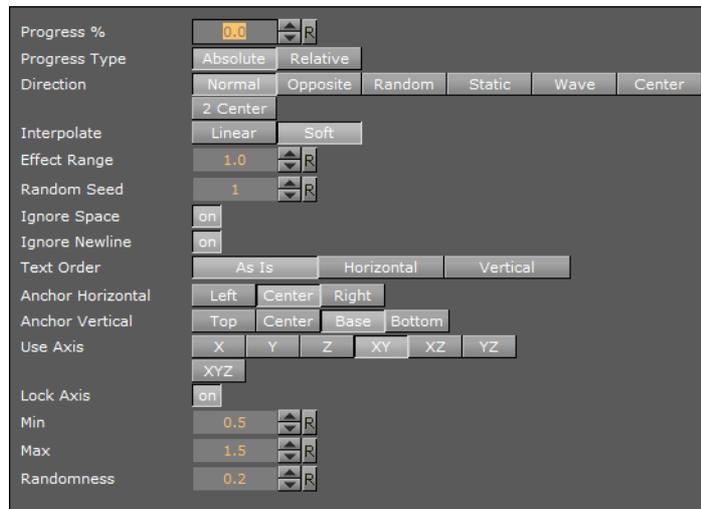
The Text FX Jitter Scale animates a jittering effect on the scale of the character.

Note: This plugin is located in: Built-ins -> Container Plugins -> TextFX

This section contains information on the following topics:

- [Text FX Jitter Scale Properties](#)

Text FX Jitter Scale Properties



- **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.

See also [Common Text FX Properties](#).

12.15.13 Text FX Plus Plus



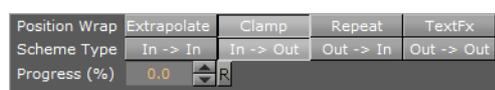
The Text FX Plus Plus plugin allows you to set a number of effects on a text string.

Note: This plugin is located in: Built-ins -> Container Plugins -> TextFX

This section contains information on the following topics:

- [Text FX Plus Plus Properties](#)

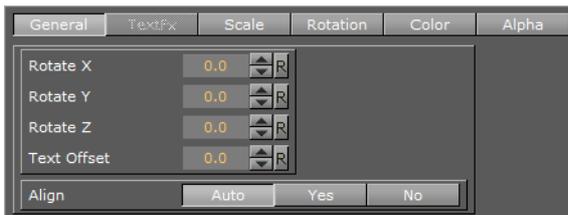
Text FX Plus Plus Properties



- **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	Off	Spline	Global
Rotation	Off	Spline	Global
Scale	Off	Spline	Global
Color	Off	Spline	Global
Alpha	Off	Spline	Global

- **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.



- **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](#).
 - **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.

12.15.14 Text FX Rotate



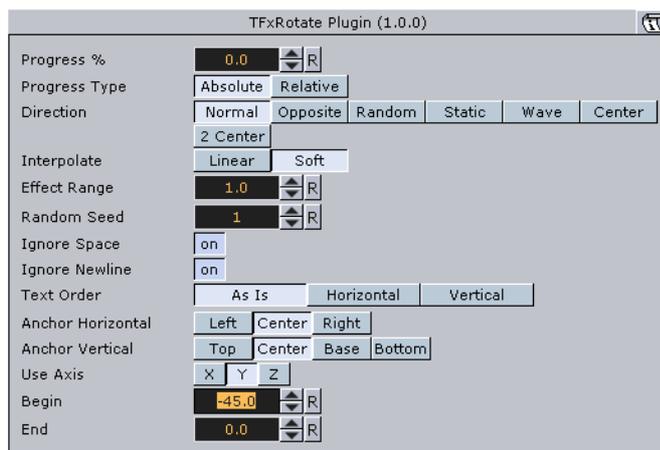
The Text FX Rotate plugin allows you to create an effect where the characters rotate on the X-, Y- or Z-axis.

Note: This plugin is located in: Built-ins -> Container Plugins -> TextFX

This section contains information on the following topics:

- [Text FX Rotate Properties](#)

Text FX Rotate Properties



- **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.

See also [Common Text FX Properties](#).

See Also

- [Expert](#)

12.15.15 Text FX Scale



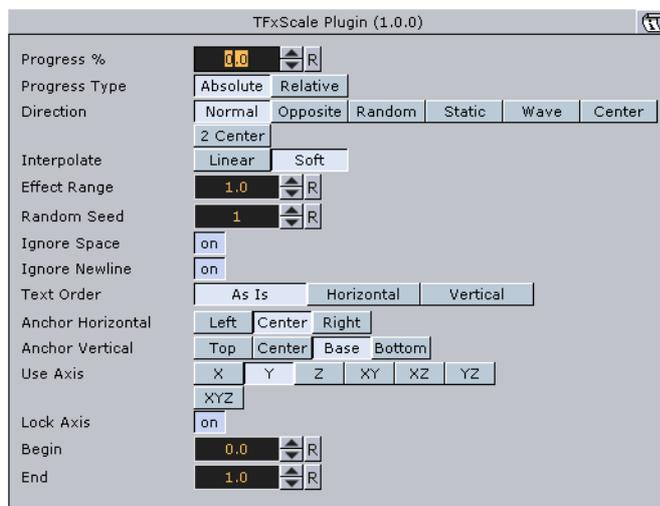
The Text FX Scale plugin allows you to create a scaling animation of the characters.

Note: This plugin is located in: Built-ins -> Container Plugins -> TextFX

This section contains information on the following topics:

- [Text FX Scale Properties](#)

Text FX Scale Properties



- **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.

See also [Common Text FX Properties](#).

See Also

- [Text FX Size](#)

12.15.16 Text FX Size



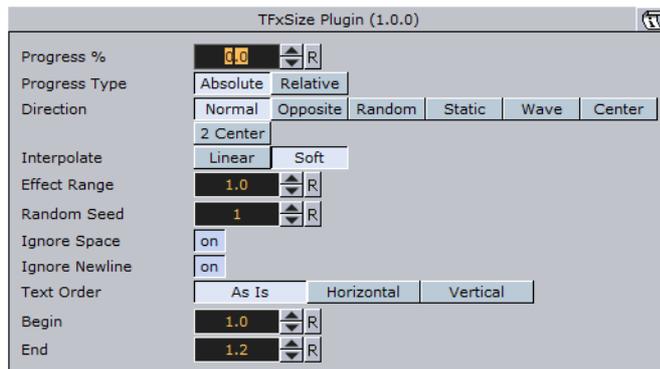
The Text FX Scale plugin allows you to create a sizing animation of the characters.

Note: This plugin is located in: Built-ins -> Container Plugins -> TextFX

This section contains information on the following topics:

- [Text FX Size Properties](#)

Text FX Size Properties



- **Begin:** Sets the initial scale of the text before the effect.
- **End:** Sets the final scale of the text after the effect.

See also [Common Text FX Properties](#).

See Also

- [Text FX Scale](#)

12.15.17 Text FX Slide



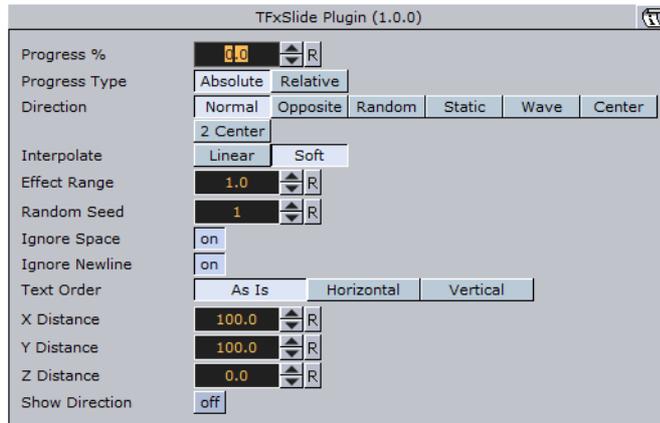
The Text FX Slide plugin allows you to create a sliding animation of the characters on the X, Y and Z-axis.

Note: This plugin is located in: Built-ins -> Container Plugins -> TextFX

This section contains information on the following topics:

- [Text FX Slide Properties](#)

Text FX Slide Properties



- **X, Y and Z Distance:** Defines direction and distance of moved text.
- **Show Direction:** Enables and disables view of 'movement' vector.

Note: The direction will not be shown in On Air mode.

See also [Common Text FX Properties](#).

12.15.18 Text FX Vertex Explode



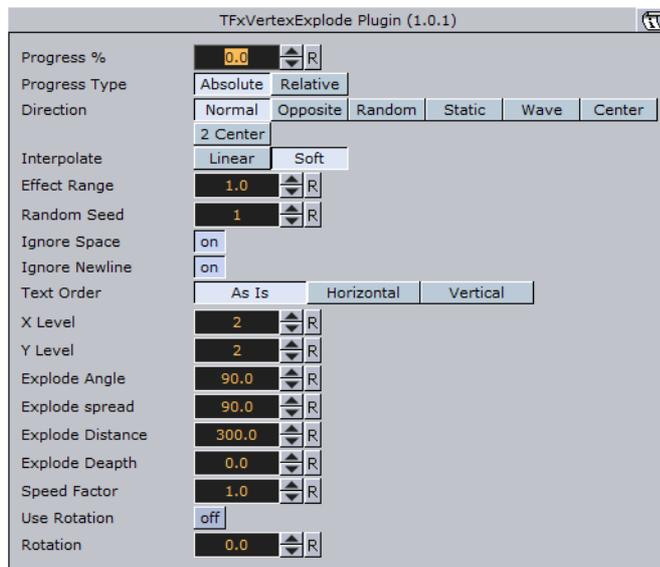
The Text FX Vertex Explode plugin allows you to create an effect where text characters are exploded with a vertex effect.

Note: This plugin is located in: Built-ins -> Container Plugins -> TextFX

This section contains information on the following topics:

- [Text FX Vertex Explode Properties](#)

Text FX Vertex Explode Properties



- **X, Y and Z Distance:** Defines direction and distance of moved text.
- **Show Direction:** Enables and disables view of 'movement' vector.

Note: The direction will not be shown in On Air mode.

See also [Common Text FX Properties](#).

See Also

- [Text FX Explode](#)

12.15.19 Text FX Write



The Text FX Write plugin allows you to create an effect where text characters get shown 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 plugin is entered in the plugin's own editor.

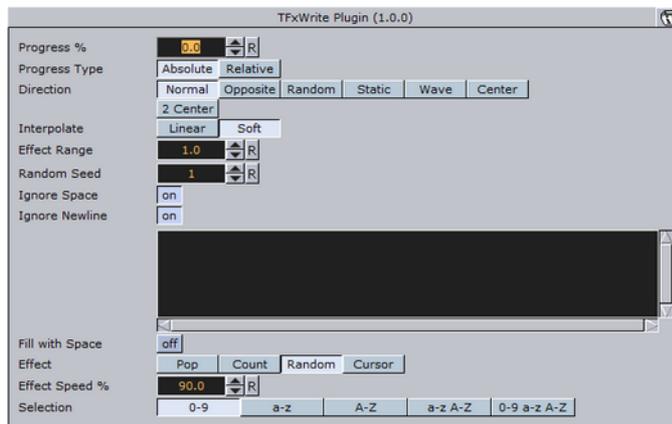
Note: Text FX Write expects UTF-8 encoded text as input if a font encoding other than *Default* is selected (see **Font Options** (see [Viz Configuration](#) in the [Viz Engine Administrator Guide](#))).

Note: This plugin is located in: Built-ins -> Container Plugins -> TextFX

This section contains information on the following topics:

- [Text FX Write Properties](#)

Text FX Write Properties



- **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 shown.
 - **Count:** Creates a counting sequence by showing a random symbol before showing the character to be shown.
 - **Random:** This (in combination with effect range) writes random characters before the final character is shown. Use a non proportional font when you use this function to get the better looking effect.
 - **Cursor:** Shows 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.

See also [Common Text FX Properties](#).

12.16 Texture

The following container plugins are located in the Texture folder:

- [BrowserCEF](#)
- [Grabbit](#)
- [GraffitiTex](#)
- [Image Clip](#)
- [Light Blur](#)
- [MoViz](#)

- [Noise](#)
- [SoftClip](#)

12.16.1 BrowserCEF



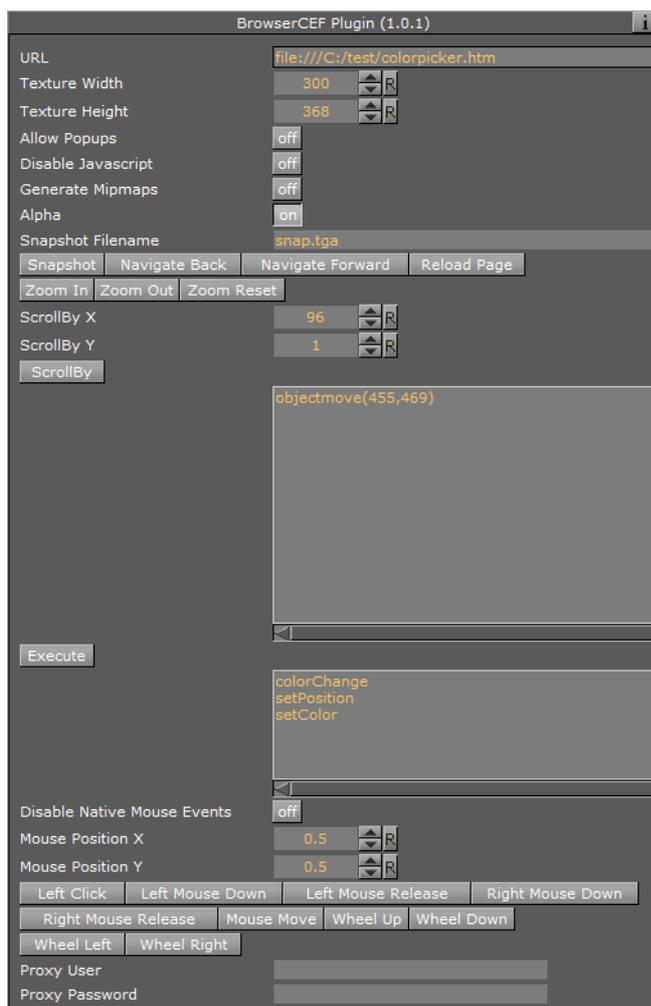
The BrowserCEF plugin gives the ability to stream browser content on a Texture.

Note: This plugin is located in: Built-ins -> Container Plugins -> Texture

This section contains information on the following topics:

- [BrowserCEF Properties](#)
- [To Create a Simple Scene with BrowserCEF](#)
- [Interact with a Scene with JavaScript](#)
- [Example HTML Document](#)

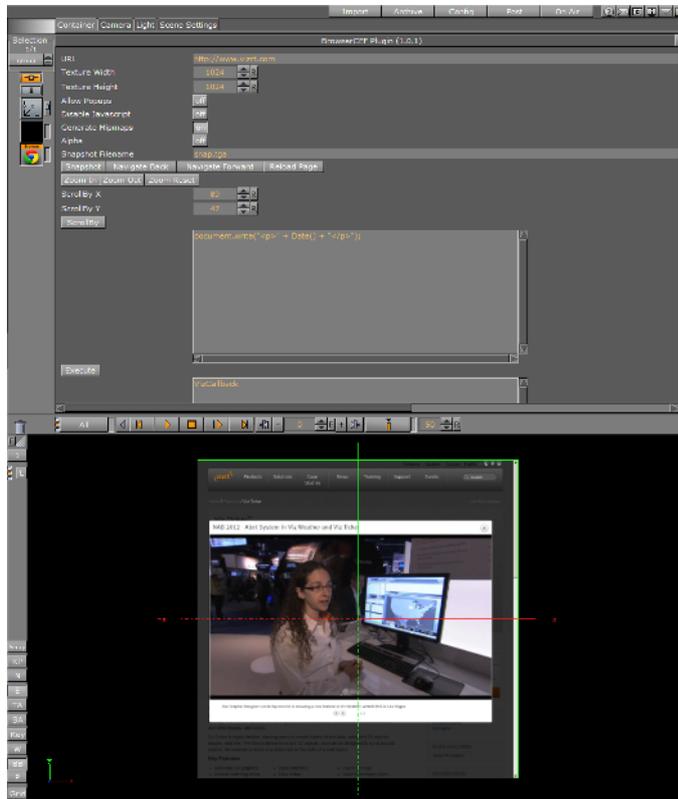
BrowserCEF Properties



- **URL:** Enter the URL address to be opened
- **Texture Width:** Width in pixels of the browser window
- **Texture Height:** Height in pixels of the browser window
- **Allow Popups:** Allow the browser to render popup windows. The popup will be rendered into the current texture. Close the popup by navigating back
- **Disable JavaScript:** Disable JavaScript execution
- **Generate Mipmaps:** Generate all mipmap levels.
 - Set to OFF for more performance
 - Set to ON for more quality when the texture is rendered on a smaller area on the screen than the width/height of the browser itself
- **Alpha:** Enable a transparent browser window
- **Snapshot Filename:** The full path of the snapshot filename
- **Snapshot:** Take a snapshot of the current browser texture. The snapshot is saved as a file as specified in Snapshot Filename
- **Navigate Back:** Navigate back to the previous page
- **Navigate Forward:** Navigate forward to the next page
- **Reload Page:** Reload the current page
- **Zoom In:** Zoom into the page
- **Zoom Out:** Zoom out of the page
- **Zoom Reset:** Reset zoom to default value
- **ScrollBy X:** Set the scroll in the horizontal axis
- **ScrollBy Y:** Set the scroll in the vertical axis
- **ScrollBy:** Execute ScrollBy X and Y values
- **Execute:** Execute the code entered in the code entry field
- **Disable Native Mouse Events:** Disables all mouse events (ie, move, click and wheel events). Useful to control mouse events from outside (e.g. scripting) or if it interferes with other plugins (e.g. like Graffiti)
- **Mouse Position X and Y:** The u/v parameters control the position of the mouse cursor. The coordinates are normalized in [0,1], being (0,0) the lower left corner and (1,1) the upper right corner of the browser window. All mouse click and move events described below relate to this coordinate:
 - **Left Click:** Emulates a click (mouse down and mouse up) of the left mouse button
 - **Left Mouse Down:** Emulates a left mouse button down event
 - **Left Mouse Up:** Emulates a left mouse button up event
 - **Right Mouse Down:** Emulates a right mouse button down event
 - **Right Mouse Up:** Emulates a right mouse button up event
 - **Mouse Move:** Emulates a mouse move event (might be triggered after the u/v coordinate has been changed)
 - **Wheel Up:** Emulates a mouse wheel up event
 - **Wheel Down:** Emulates a mouse wheel down event
 - **Wheel Left:** Emulates a mouse wheel left event
 - **Wheel Right:** Emulates a mouse wheel right event
- **Proxy User:** The user name in case the system's http-proxy requires authentication

- **Proxy Password:** The user name password in case the system's http-proxy requires authentication

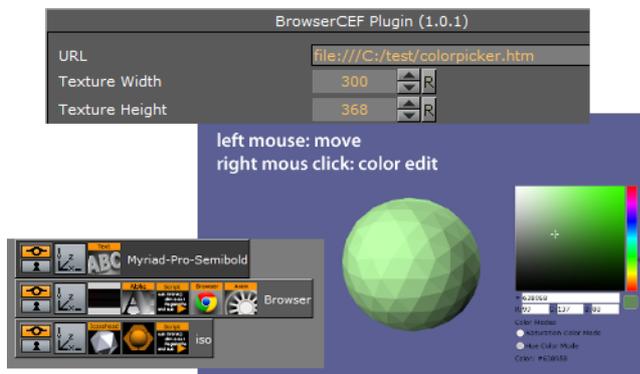
To Create a Simple Scene with BrowserCEF



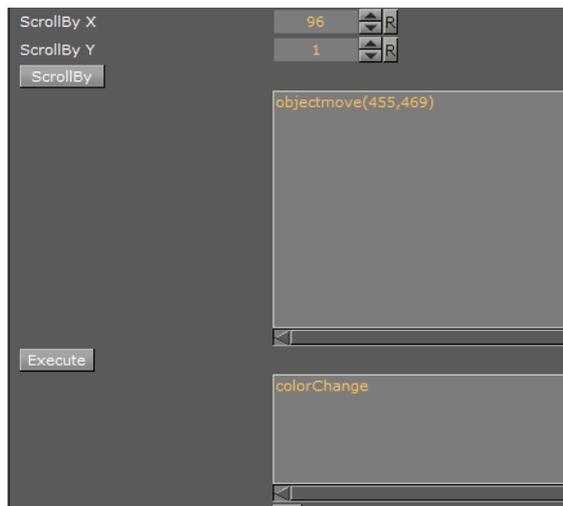
1. Add a new Container to the Scene tree.
2. Add a **rectangle** geometry to the Container
3. Add the BrowserCEF plugin.
4. Open the BrowserCEF properties and enter a valid URL.
5. Click on **E** (enable the handling of interactive script/plugin events (Scene Editor)).
6. Use the mouse or controls to navigate the web site.

Interact with a Scene with JavaScript

It is possible to interact with a Scene from within the JavaScript of a html document, for example, colorpicker.html.



In the JavaScript section of a html document interact with the Scene with the **window.vizrt** object. Append any function name on this object, for example, **window.vizrt.colorChange("255 100 100")**. The function **colorChange** must be defined in the **JavaScript function** section of the **BrowserCEF** plugin.



To add more functions, add them in this GUI section, separated by new lines:



Only one argument is supported for each function, to add more arguments to a function use separators and encode all arguments into one string.

Calls from JavaScript are translated into Viz Shared Memory variable changes, on the **system.map**. To register these changes use the Shared Memory variable composed by the Container ID of the BrowserCEF plugin and the string, for example, "**BROWSER STATUS**".

IMPORTANT! There must be a space between the double quotation mark (") and BROWSER: " BROWSER STATUS".

Script Example: When the script sits on the same Container:

```
sub OnInit()
    System.map.RegisterChangedCallback( this.VizId & " BROWSER STATUS" )
end sub
sub OnSharedMemoryVariableChanged(map As SharedMemory, mapKey As String)
    println( "shared map changed: " & map[mapKey] )
    Dim str As String = map[mapKey]
    Dim strstr As array[String]
    str.split(" ", strstr)
    if strstr[1] = "javascript" then
        if strstr[2] = "colorChange" then
            scene.findContainer("iso").Material.diffuse.red =
cDb1(strstr[3])/255.0
            scene.findContainer("iso").Material.diffuse.green =
cDb1(strstr[4])/255.0
            scene.findContainer("iso").Material.diffuse.blue =
cDb1(strstr[5])/255.0
        end if
    end if
end sub
```

Example string passed to Shared Memory:

```
"0 JavaScript colorChange 0 53 4"
```

- The first number, either 0 or 1, can be ignored
- **JavaScript**, shows that a JavaScript function was invoked
- The function name (**colorChange**) itself followed by the argument(s).

Example call in the html document:

```
        window.vizrt.colorChange(r + " " + g + " "
+ b);
```

Example HTML Document

Entered in **URL**: file:///C:/test/colorpicker.htm (see [BrowserCEF Properties](#)):

```
<script type="text/javascript">
function colorAnim()
{
    window.vizrt.colorAnim()
}
function objectmove(x, y)
{
    PositionX.value = x;
    PositionY.value = y;
}
</script>
```

```

</script>
<!DOCTYPE html>
<html lang="en">
<head>
  <title id='Description'>The jqxColorPicker widget allows you to
  easily pick a color.
  </title>
  <link rel="stylesheet" href="http://www.jqwidgets.com/jquery-widgets-
  demo/jqwidgets/styles/jqx.base.css" type="text/css" />
  <script type="text/javascript" src="http://www.jqwidgets.com/jquery-
  widgets-demo/scripts/jquery-1.11.1.min.js"></script>
  <script type="text/javascript" src="http://www.jqwidgets.com/jquery-
  widgets-demo/scripts/demos.js"></script>
  <script type="text/javascript" src="http://www.jqwidgets.com/jquery-
  widgets-demo/jqwidgets/jqxcore.js"></script>
  <script type="text/javascript" src="http://www.jqwidgets.com/jquery-
  widgets-demo/jqwidgets/jqxcolorpicker.js"></script>
  <script type="text/javascript" src="http://www.jqwidgets.com/jquery-
  widgets-demo/jqwidgets/jqxradiobutton.js"></script>
</head>
<body class='default'>
  <div id='content'>
    <script type="text/javascript">
      $(document).ready(function () {
        // Create jqxColorPicker widgets.
        $("#jqxColorPicker").jqxColorPicker({ width: '250',
height: '250'});
        $("#hueMode").jqxRadioButton({ width: '150', height:
'25'});
        $("#saturationMode").jqxRadioButton({ checked: true,
width: '180', height: '25'});
        $("#hueMode").on('change', function (event) {
          if (event.args.checked) {
            $("#jqxColorPicker").jqxColorPicker({ colorMode:
'hue' });
          }
          else {
            $("#jqxColorPicker").jqxColorPicker({ colorMode:
'saturation' });
          }
        });
        $("#jqxColorPicker").on('colorchange', function (event) {
          //console.log(event.args.color.r);

window.vizrt.colorChange(event.args.color.r + " " + event.args.color.g +
" " + event.args.color.b);
          $("#colorlog").html("<div>Color: #" +
event.args.color.hex + "</div>");
        });
      });
    </script>
    <div id='jqxWidget'>

```

```

<div id="jqxColorPicker">
</div>
<div style="font-size: 13px; font-family: verdana; margin-
top: 10px;">
    <div style="margin-bottom: 5px;">Color Modes</div>
    <div id="saturationMode">Saturation Color Mode</div>
    <div id="hueMode">Hue Color Mode</div>
</div>
<div style="font-family: Verdana; font-size:13px;"
id="colorlog"></div>
</div>
</div>
</body>
</html>

```

12.16.2 Grabbit



The Grabbit plugin gives the ability to use captured video inside Viz Engine. Grabbit uses the Microsoft DirectShow Filter graph framework to capture the video.

Devices which are supported by DirectShow and are self-contained such as capture cards or web cameras, can be used. Self-contained means the capture device does not need any further input.

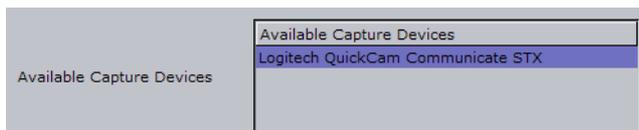
Note: This plugin is located in: Built-ins -> Container Plugins -> Texture

This section contains information on the following topics:

- [Grabbit Properties](#)
- [Required Software](#)
- [Best Practices](#)
- [Known Limitations](#)
- [To Capture a Video Stream with Grabbit](#)
- [To Capture a Video Stream with Grabbit and a Vfw Supported Device](#)

Grabbit Properties

Following is a list of all parameters of the plugin. Depending on the state of the plugin, they may be hidden. Some of them are not visible at all.



- **Available Devices:** Lets you decide which video capture device to use. This list is created during loading of the plugin. 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 plugin. Since the list is static is just shows all capture devices which were available at startup. If

the parameter is changed and the plugin 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:** 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 plugin is already started it will stop the capturing before loading the new capture device.
 - Parameter name: **CaptureDevice** (**string**).

Note: This parameter is hidden from the user interface.

- **Capture Format:** 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.

Note: This parameter is hidden from the user interface.

- Parameter name: **CaptureFormat** (**string**).



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.

Video Filter	Video Capture Pin	Audio Filter	Audio Capture Pin
Crossbar 1	Crossbar 2	TV-Tuner Video	TV-Tuner Audio

Parameters are shown if the filter supports the DirectShow dialog interface. Since the dialogs are handled by Windows and not the Viz Artist GUI 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 Artist GUI.

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.

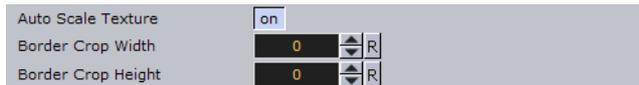
- **Video Capture Pin:** If available choose RGB24 or UYV as format. Other settings may not work. The dialog window may be behind the renderer GUI after activation. Please be careful to close the dialog window. Otherwise the renderer and the GUI may lock up. Changes are effective immediately.
 - Parameter 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 Artist GUI 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 Artist GUI.

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:** 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:** Shows if the capture graph has been built successfully.
 - Parameter Name: **StatusOpened.**
 - Type: toggle button (show only).

- **Status Started:** Shows if the capture graph has been started.
 - Parameter Name: `StatusStarted`.
 - Type: toggle button (show only).
- **Status Stream Width:** Shows the captured video width.
 - Parameter Name: `StatusStream_x`.
 - Type: integer (show only).
- **Status Stream Height:** Shows the captured video height.
 - Parameter Name: `StatusStream_y`.
 - Type: integer (show 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 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 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 with Grabbit

1. Add the Rectangle geometry to the Scene Tree.
2. Add the Grabbit plugin to it.
3. Select the capture device 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 plugin or by another application).

4. If you have a device that supports the capture format enumeration and the ProcAmp interface you will get a screen similar to the following.
5. Set the **Capture Format** by using the dialogs behind the configuration buttons Video Filter, and so on.
6. 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 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.
7. 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.
8. Adjust the effect settings. You might want to experiment with them. Some parameters have discreet values as for example the *Gamma* parameter.

Note: The effect settings are highly dependant on the selected device.

To Capture a Video Stream with Grabbit and 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 Artist GUI.
 - Make sure that each dialog window is closed before any other changes are made in the Viz Artist 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.

12.16.3 GraffitiTex



The Graffiti Texture plugin 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 plugin can also recognize some rendered shapes and replace the hand-drawn element with the recognized shape (circle, ellipse, cross, arrow).

The plugin can either work in combination with Viz Engine's multitexturing technology or can add a standalone texture.

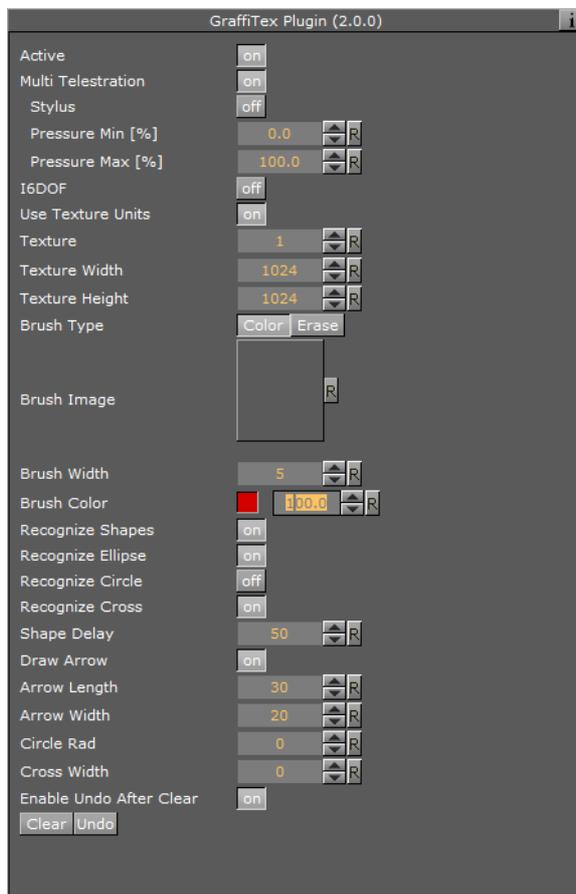
Note: There are two graffiti plugins, one that can be used globally as a scene plugin, and this one that is applied to the container level.

Note: This plugin is located in: Built-ins -> Container Plugins -> Texture

This section contains information on the following topics:

- [GraffitiTex Properties](#)
- [To Create a Container Level Graffiti Effect](#)

GraffitiTex Properties



- **Active:** Enable/disable drawing
- **Multi Telestration:** Set to On or OFF. Allows handling of multiple touch events for multitouch devices. Some devices incorporate a pressure factor, this influences the width of the brush:
 - **Pressure Min [%]:** Set the minimum width of the brush
 - **Pressure Max [%]:** Set the maximum width of the brush
- **I6DOF:** Specifies whether input comes from mouse or 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:** Set the length of an arrow head
- **Arrow Width:** Set the width of an 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
- **Enable Undo After Clear:** Enable undo after the Clear button has been used.
- **Clear (button):** Clears the canvas
- **Undo (button):** Undo

To Create a Container Level Graffiti Effect



- Add the plugin to the container, set the plugin [GraffitiTex Properties](#), set Viz Artist in [On Air Mode](#) and start drawing.

See Also

- [Graffiti](#) (scene plugin)

12.16.4 Image Clip



The Image Clip plugin 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 plugin 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 plugin should be uncomplicated as most video applications has support for saving out TIFF. 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 [Grabbit Properties](#) 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.

Note: This plugin is located in: Built-ins -> Container Plugins -> Texture

This section contains information on the following topics:

- [Image Clip Properties](#)
- [To Add an Image Clip](#)

Image Clip Properties



- **Image:** Sets the image path and the first image you would like to use in the animation. Do not use clip names with numbers (except the counter (e.g. 000, 001, 002, etc.)).
- **Play Mode:** Sets the play mode:
 - Once:
 - Loop:
 - Swing:
 - Animation:
 - Anim-Dissolve:
- **Reverse:** Reverses the animation (not in **Animation** play mode).
- **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**.
- **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.
- Geo Ref:
- **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.
- Base Path:
- Format:
- Weather:
- Blend Images:
 - Dissolve Speed:
- Control Texture Map:
- **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.
- Mapping:
- **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.
2. Add the Image Clip plugin to it.
3. In the Image Clip properties click the  (Browse) button to locate the folder that has the required images.
4. Select the first image in the sequence.
5. Click **OK**.
 - The image sequence now loads into the ImageClip plugin, and is visible in the scene.
6. Click **Play** to test the image sequence.
 - If you change the image sequence and you need to reload it, click the Reinitialize button.
7. Change the Play Mode to Loop to play the image sequence continuously.
 - The image sequence will start playing automatically.

8. Change the Play Mode to Animation to animate the image sequence frame position to control the sequence within the [Stage Editor](#).
 - 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 shown for the image sequence.
9. Animate the Animation Position value.
 - The Animation Position maximum value is restricted to the number of images.
10. 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.
11. 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 Key Frames, so if you choose to slow down an animation then the motion will not be smooth.

See Also

- [Video Playback](#)
- [MoViz](#)
- [SoftClip](#)
- [Background Clip](#)

12.16.5 Light Blur



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 Artist/ Engine core. For future use it is therefore recommended to use the [Dynamic Texture Media Asset](#) plugin.

Note: This plugin is located in: Built-ins -> Container Plugins -> Texture

This section contains information on the following topics:

- [Light Blur Properties](#)

Light Blur Properties



- **Width and Height:** Sets the width and height of the blurred light. Available options are 2 to 2048
- **Format:** Sets the image format. Options are:
 - RGB
 - RGBA
 - Alpha
 - Luminance
 - Luminance Alpha
- **Save:** 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

12.16.6 MoViz



With the MoViz plugin media files or streams can be played inside the Viz Engine.

MoViz is not a QuickTime player or an AVI player. It uses the Microsoft DirectShow Filter Graph framework to play the media files, which means it can play both 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 not all QuickTime codecs can be used, only the ones that are supported within DirectX wrappers and filters.

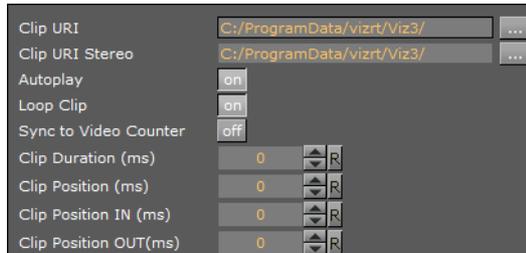
Note: Throughout this plugin documentation, media file and media stream is referred to as media and may be used interchangeably unless stated otherwise.

Note: This plugin is located in: Built-ins -> Container Plugins -> Texture

This section contains information on the following topics:

- [MoViz Properties](#)

MoViz Properties

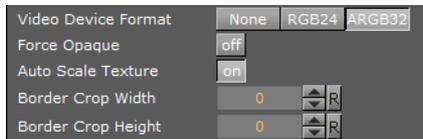


- **Clip URI:** Click to load clips from the file system (or enter the location directly). It takes the form of a valid URI (e.g. http://... mms://...). If the parameter is changed and the plugin is already started it will stop the playback before loading a new clip. Changes are effective immediately.

IMPORTANT! All video clips from the file system should be stored locally on the D: 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 D: drive so that the clip can be seen in preview.

- **Clip URI Stereo:**
- **Autoplay:** Set the automatic start of the clip after it has loaded, to On or Off. Changes are effective immediately (Default is off)
- **Loop Clip:** Set looped playback to On or Off. 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 ((Default is Off)
- **Sync to Video Counter:**
- **Clip Duration (ms):** Shows the duration of the clip in milliseconds (read only). This parameter makes only sense with media that supports this kind of information.
- **Clip Position (ms):** Gives the current position of the clip in 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.
- **Clip Position IN (ms):** Gives the first frame where the clip should start in milliseconds. This parameter makes only sense with media that supports this kind of information. Changes are effective immediately.

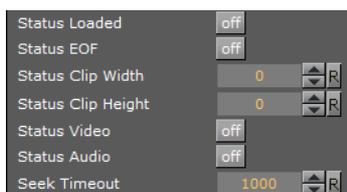
- **Clip Position OUT (ms):** Gives the last frame where the clip should stop in milliseconds. This parameter makes only sense with media that supports this kind of information. Changes are effective immediately.



- **Video Device Format:** Select a video format for rendering. Changes are effective at load time.
 - **None:** The video part is not rendered
 - **RGB24:** The video part is delivered to Viz Artist/Engine with the RGB24 format
 - **ARGB32:** The video part is delivered to Viz Artist/Engine with the ARGB32 format
- **Force Opaque:** (ARGB32 only) If the clip has an alpha channel set the alpha value to 100% opaque, if required. Changes are effective for each frame delivered
- **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
 - **Border Crop Width:** Crop pixels on the horizontal sides of the video. Changes are effective immediately but not for already delivered video
 - **Border Crop Height:** Crop pixels on the vertical sides of the video. Changes are effective immediately but not for already delivered video



- **Audio Device/Format:** Select which audio device/format to use for rendering. This may be useful until audio via Viz Artist/Engine is supported although it is not recommended due to performance and frame drop issues. Changes are effective immediately but not for already delivered audio
 - **None:** The audio part is not rendered
 - **Default Device:** The audio part is rendered using the default DirectShow renderer



- **Status Loaded:** Show if the clip has loaded
- **Status EOF:** Show if the clip is at EndOfFile
- **Status Clip Width:** Shows the video width
- **Status Clip Height:** Shows the video height
- **Status Video:** Show if the clip has a usable Video component
- **Status Audio:** Show if the clip has a usable audio component
- **Seek Timeout:** Specify the timeout for seeking to a position in the movie file. Higher values can prevent jump back to the expected position when clip playback

takes longer to start than expected. However, the playback time may not be as exact



- **Eject:** Unload the clip and destroy the graph. Changes are effective immediately
- **Load:** Load the given clip and builds the graph. Changes are effective immediately
- **Rewind:** Rewind the clip to the start, when supported. Changes are effective immediately
- **Stop:** Put the graph in stop mode. Changes are effective immediately
- **Play:** Put the graph in play mode. i.e. start playback. Changes are effective immediately
- **Pause:** Puts the graph in pause mode. Changes are effective immediately
- **Play From Start:** Like **Play**, but rewinds the clip to the start first. Changes are effective immediately
- **Clear:** Clear the texture to opaque white

See Also

- [Video Playback](#)
- [Advanced Issues with Video Codecs](#)
- [Image Clip](#)
- [SoftClip](#)

12.16.7 Noise



Allows the user to configure and create a noise image.

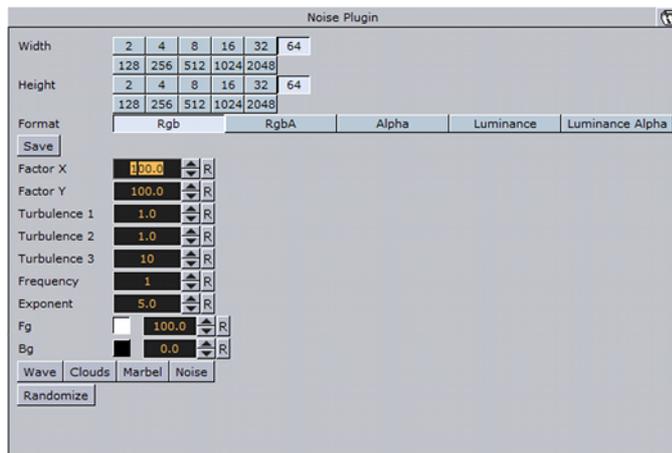
IMPORTANT! The noise functionality is now a built in feature of the Viz Artist/Engine core. For future use it is therefore recommended to use the [Dynamic Texture Media Asset](#) plugin.

Note: This plugin is located in: Built-ins -> Container Plugins -> Texture

This section contains information on the following topics:

- [Noise Properties](#)

Noise Properties



- **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.

12.16.8 SoftClip



The SoftClip is used for playing video clips projected on a texture, and supports AVI files only. This format supports RGB and RGBA (alpha channel).

In its simplest form, the SoftClip plugin can exist on an empty container to play back a video clip in Viz Artist/Engine. Other plugins can be mixed with SoftClip to obtain different results. For example you can apply a SoftClip plugin to a Sphere and add a Material to affect the overall shape, color and shading of the video clip.

With SoftClip you are able to control the video clip within the stage, unlike with the [Live Video Media Asset](#) 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 SoftClip plugin might have an affect on performance and quality, hence, it is recommended to use the [Performance Editor](#) when working with the SoftClip plugin.

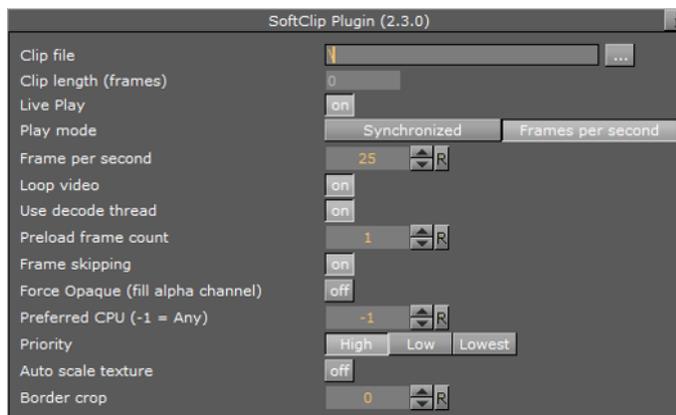
IMPORTANT! Viz Engine keeps all designer/operator/automation loaded scenes – and plugins – in memory. As a consequence, if you load 30 different templates that load 30 different scenes – that each has a single SoftClip plugin on it, then Viz Engine will have thirty SoftClip plugins with 30 threads activated in memory. If every plugin has 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 renders (a function in many Vizrt control client programs) should be used.

Note: This plugin is located in: Built-ins -> Container Plugins -> Texture

This section contains information on the following topics:

- [SoftClip Properties](#)
- [To Use SoftClip](#)

SoftClip Properties



- **Clip file:** Use this function to browse for the clip to be played out.

Note: It is important that all video clips are stored locally on the D: 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 D: drive so that the clip can be seen in preview.

- **Clip length (frames):** Shows the number of frames for the clip
- **Live Play:** Make sure that the clip will be played and synchronized to the frame rate of the Viz Engine. This is set in [Viz Configuration](#) / Output Format options. Disable the Live Play to allow the clip playback to be Key Frame animated.
- **Play mode:**
 - **Synchronized:** The clip will run synchronized with the scene playback. This is set in the [Viz Configuration](#)/Output Format options
 - **Frames per second:** Adjust for different speeds, set in the next parameter

- **Frame number:** Enables the frame numbers to be manually animated. This is an alternative to playing the whole clip (available when **Live Play** is set to Off)
- **Frames per second:** Set the frame rate (available when **Play mode** is **Frames per second**).

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:** Enable to split the processing of the clip evenly between the machine's processors. It is recommended to have this **on** (default). 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.

Tip: If **Priority** is set to **High**, for best results set **Use Decode Thread** to **off**.

- **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 playback. With frame skipping disabled the clip will play out in full, as fast as possible but will not try to maintain real-time playback. (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 specifications of the running machine:
 - If it is a quad core processor the CPU value can be set to 4, but to avoid errors it is recommended to set the value to 2
 - If it is a machine with lower specifications then the recommended value is 1
 - The value -1 enables the machine operating system to select which CPU to use
- **Priority:** Sets the priority of drawing the video to a custom level in Viz Engine giving it more or less processor priority
- **Clear before playback:** Set to ON to clear the image before playback. **Live Play** must be set to **Synchronized** and the **Frame number** to be shown from the clip is smaller than the current frame number +1 (available when **Live Play** is set to Off)
- **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 SoftClip

1. Add a group container to the Scene Tree.
2. Add the **SoftClip** plugin.

3. Open the SoftClip editor.
4. Click the **Browse...** button to select and load a clip.
 - Loading a clip will make it visible in the Scene. By default, the SoftClip plugin has the Live Play and Loop video settings enabled, hence, the clip will instantly start playing (and looping) in the render window.
5. Optional: Enable the **Use decode thread** option.
6. Disable the **Live Play** option to enable the **Frame number** setting.
7. Animate the Frame number setting (e.g. from frame 10 to 30).

Note: Viz Engine will not interpret between Key Frames, so if you choose to slow down an animation then the motion will not be smooth.

See Also

- [Video Playback](#)
- [Advanced Issues with Video Codecs](#)
- [Image Clip](#)
- [MoViz](#)

12.16.9 Tex Component



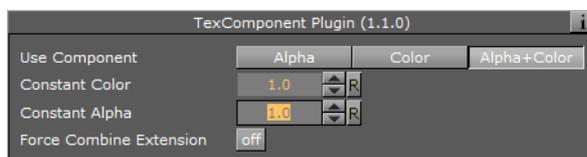
The Tex Component plugin selects which component of a texture is used for the texture mapping process.

Note: This plugin is a legacy plugin and the functionality is now incorporated into the Texture Editor.

This section contains the following topics:

- [Tex Component Properties](#)

Tex Component Properties



- **Use Component:**
 - **Alpha:** Only the alpha of the texture will be used. The RGB color of the texture will be set to a constant color specified by the **Constant Color** parameter
 - **Color:** Only the RGB color of the texture will be used. The alpha will be set to a constant value specified by the **Constant Alpha** parameter
 - **Alpha+Color:** Both the alpha and the RGB color of the texture will be used (default setting)

- **Constant Color:** Set the constant RGB color of the texture when **Alpha** is selected
- **Constant Alpha:** Set the constant alpha value when **Color** is selected

12.17 Ticker

The following container plugins are located in the Ticker folder:

- [Scroller Action](#)

12.17.1 Scroller Action



The Scroller Action plugin makes it possible to design scrolling tickers that trigger actions when scroller elements are played out. Actions can be triggered both before and after reading the next element from the input source, as well as delaying the reading of the next element a certain number of frames.

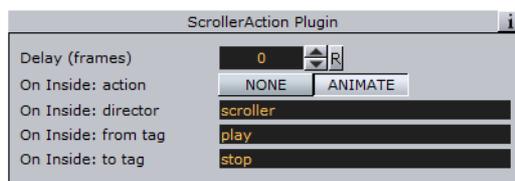
A scrolling carousel scene must have a [SoftClip Draw Pixels](#) plugin attached to a container and be configured to receive elements from an external source. In most cases this external source is Viz Ticker's Ticker Service running on the same machine. During scene design, it is useful to use a debugging syntax to specify a fake source that refers to graphics templates located directly in the scene.

Note: This plugin is located in: *Built-ins -> Container Plugins -> Ticker*

This section contains information on the following topics:

- [Scroller Action Properties](#)
- [Controlling the Progression](#)
- [To Control the Progression](#)

Scroller Action Properties



- **Delay (Frames):** Adds a delay after the current element is fully inside the scroller, so that insertion of the next element is postponed the given number of frames.
- **On Inside:** Defines if the element should trigger anything after it fully enters the scroller.
 - **Action:** If NONE is selected, nothing is triggered. If ANIMATE is selected, it is possible to trigger a director and define the start and stop point of the director animation.
 - **Director:** Defines the director to run when On Inside is triggered.
 - **From Tag:** Defines where to start the director animation.
 - **To Tag:** Defines where to stop the director animation.

- **After Delay:** Defines if the element should trigger anything after the scroller has finished delaying the next element.
 - **Action:** If NONE is selected, nothing is triggered. If ANIMATE is selected, it is possible to trigger a director and define the start and stop point of the director animation.
 - **Director:** Defines the director to run when After Delay is triggered.
 - **From Tag:** Defines where to start the director animation.
 - **To Tag:** Defines where to stop the director animation.

The source syntax (separator) (text) will create a scroller showing an endless stream of elements alternating between a separator and a text graphics design. The scroller will search through the scene tree for containers named separator and text, use the graphics template found there, and then copy them to create the element instances that are being scrolled.

Controlling the Progression

A typical example of when the Scroller Action plugin is necessary, is when text messages (SMS) should scroll into the screen, animate to a halt, wait a few seconds, and then restore the scroll speed whenever new text messages are available.

This paragraph provides guidance on how to create an upwards scroll to show text messages, where the scroller should pause for a while after playing out each message to the screen. The scroller should not immediately follow the text design with the next separator design. Instead, the text design should enter the screen fully, and then the scroller should smoothly slow down to a halt. The scroller should wait a given number of frames before slowly starting the scroller again and pushing the next separator element onto the screen. If an external source is used, the scroller should not start again until there is a new element available from the source.

To Control the Progression

1. Create a director named **scroller**.
 - This director animates the scroller speed.
2. On this director, create a stop point, and call it **stop**.
3. Set the stop scroll speed to **0**.
4. Create another stop point, and call it **play**.
5. Set the play scroll speed to a suitable speed.

The actual names that are used here are not important, but they must match the names specified in step 8.

6. Add the Scroller Action plugin to the text template top node (the container holding the [Control Object](#) plugin).
7. In the Scroller Action plugin editor, define the **Delay** frame count.

This setting adds a delay after the current element, so that the insertion of the next element is postponed the given number of frames.

The delay counter starts to count from the time the previous element has fully entered the screen, which is a few frames after it would normally try to insert a new element. When delay is set to 0, the scroller will perform as normal, and not wait until the previous element has fully entered the scroller before inserting a new element.

The *delay* parameter specifies the minimal amount of time that must pass between pushing out the current element, which is based on the template

graphics with the *Scroller Action* plugin, and the next element. The actual amount of time might be greater if there are no new elements available after the *delay* frame count has expired. Setting this parameter does not affect the speed of the scroller, it only delays the inserting of the next element, in effect increasing the resulting gap between the previous element and the next.

The default value of the *delay* frame count parameter is 0, which means that the scroller will attempt to insert the next element immediately after the previous element.

8. In the Scroller Action plugin editor, click the *On Inside: action* **Animate** button. When *animate* is selected, three additional *on inside* related parameters become available:
 - **On Inside:** director (default value is *scroller*)
 - **On Inside:** from tag (default value is *play*)
 - **On Inside:** to tag (default value is *stop*)

The parameters should remain at the default values.

When the current element has fully entered the screen, the scroller will start to animate the *scroller* director from a *play* tag to a *stop* tag. The rules of the *GOTO_TRIO* director command defines this behavior.

9. In the Scroller Action plugin editor, click the *After Delay: action* **Animate** button. When *animate* is selected, three additional *after delay* related parameters become available:
 - **After Delay: director** (default value is *scroller*)
 - **After Delay: from tag** (default value is *stop*)
 - **After delay: to tag** (default value is *play*)

The parameters should remain at the default values.

When the *delay* frame count expires, the scroller will start to animate the *scroller* director from a *stop* tag to a *play* tag. The rules of the *GOTO_TRIO* director command defines this behavior.

See Also

- [Transition Logic](#)

12.18 Time

The following container plugins are located in the Time folder:

- [Analog Watch](#)

12.18.1 Analog Watch



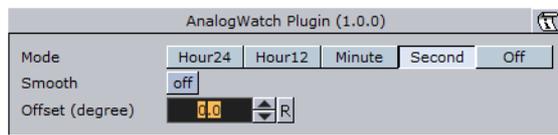
The Analog Watch plugin lets you create a real-time animated clock of objects.

Note: This plugin is located in: *Built-ins -> Container Plugins -> Time*

This section contains information on the following topics:

- [Analog Watch Properties](#)
- [To Create an Analog Watch](#)

Analog Watch Properties



- **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:
 - 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:
 - 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:
 - Style1 to Flat.
 - Width and Arrow Width to 2.0

- Percent to 50.0.
- 7. Add the Analog Watch plugin 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](#)

12.19 Tools

The following container plugins are located in the Tools folder:

Advanced Counter	Max Size
Autofollow	Max Size Lines
Autorotate	Object Zoom
Bounding Actions	Omo
Clock Rotation	Pablo
Cloner	Parliament
Colorize	Rotations Order
Counter	Slide Show
DVE Follow	System Time
Heartbeat	Temo
Image Link	TextBG
Jack	Text Link
Level Of Detail (LOD)	Text Parameters
Magnify	VCF Parameter
Match It	

12.19.1 Advanced Counter



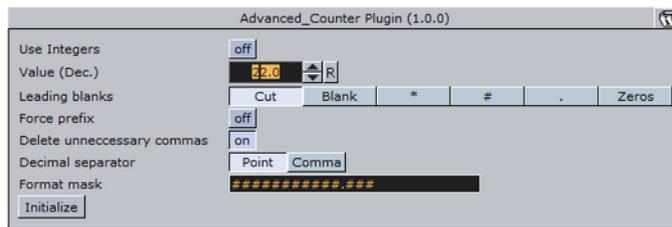
The Advanced Counter plugin allows you to easily create an animated counting sequence. It shows values with decimals and it has the possibility to create a user defined format mask. Furthermore it can have prefixed values.

Note: This plugin is located in: Built-ins -> Container Plugins -> Tools

This section contains information on the following topics:

- [Advanced Counter Properties](#)
- [To create an advanced counter](#)

Advanced Counter Properties



- **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 shown 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 plugin to the group container.
4. Open the Advanced Counter editor and animate the parameters.

12.19.2 Autofollow



The Autofollow plugin is used to connect one object with another. This object is from then on dependent on the connected plugin. 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).



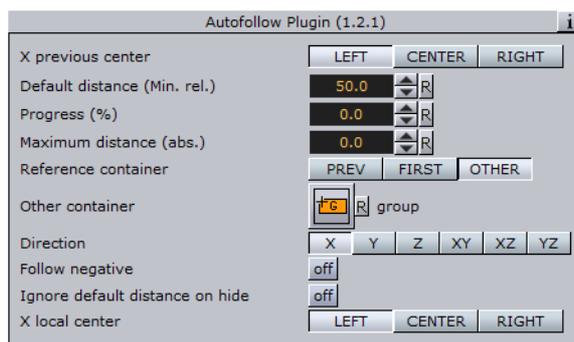
Drag the Autofollow plugin over the object container which should be dependent. It will automatically follow the Previous object in the scene tree. You may also set this to be the First object or select one specific object by adding it to the object placeholder exposed when selecting the Other option.

Note: This plugin is located in: Built-ins -> Container Plugins -> Tools

This section contains information on the following topics:

- [Autofollow Properties](#)

Autofollow Properties



- **X Previous Center:** Defines the alignment of your related object.
- **Default Distance (Min):** Allows you to set a distance from the object to the related one. This is a relative value that is dependent on the size of the related container.
- **Progress (%):** Allows you 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 Key Frames on the progress. At the beginning of the animation you can for example have 100%, and at the end 0%. The object will run from the maximum to the default distance value.
- **Maximum Distance:** Allows you to set the maximum distance from the object to the related one. This is an absolute value, which is independent of 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 shows when you select 'Other'.

- **Direction:** Allows you to choose in which axes there should be a dependency on. Options are: X, Y, Z, XY, XZ or YZ.
- **Follow Negative:** If you select this feature, all numerical values are negated. For example when you change 'Default Distance (Min)' to 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 visible container depending on the reference container setting (only if set to Previous or First).
- **Ignore default distance on hide:** Allows you to hide the reference container if you do not want the default distance to be considered. In this case it will be null. If you make the reference container visible, the default distance will show again.
- **X Local Center:** Allows you to select your alignment for the local center in X-direction.

See Also

- [Merge Containers](#)

12.19.3 Autorotate



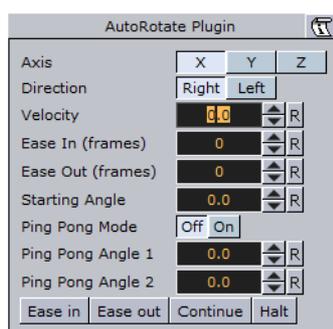
With the Autorotate function, a container can be set to continuously rotate round one of its axes.

Note: This plugin is located in: Built-ins -> Container Plugins -> Tools

This section contains information on the following topics:

- [Autorotate Properties](#)

Autorotate Properties



- **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 2 for 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**.

12.19.4 Bounding Actions



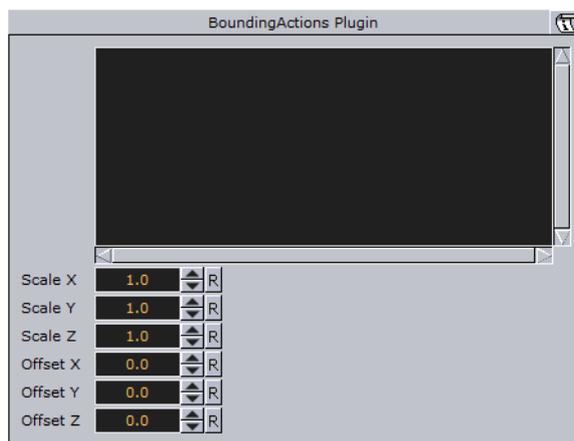
The Bounding Actions plugin enables you to run Viz Artist/Engine actions depending on the size of the bounding box.

Note: This plugin is located in: Built-ins -> Container Plugins -> Tools

This section contains information on the following topics:

- [Bounding Actions Properties](#)

Bounding Actions Properties



- **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 plugin 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 plugin is used to modify it), this command is executed.

12.19.5 Clock Rotation



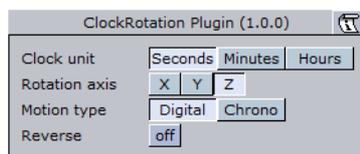
Animates any object as a rotating clock.

Note: This plugin is located in: Built-ins -> Container Plugins -> Tools

This section contains information on the following topics:

- [Clock Rotation Properties](#)

Clock Rotation Properties



- **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](#)

12.19.6 Cloner



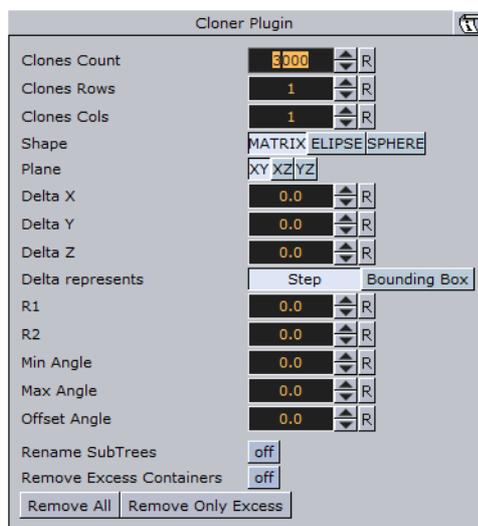
Creates a number of clones of a given container (target).

Note: This plugin is located in: Built-ins -> Container Plugins -> Tools

This section contains information on the following topics:

- [Cloner Properties](#)

Cloner Properties



- **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 plugin 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 to create a layout for the newly generated clones.

12.19.7 Colorize



The colorize plugin 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 plugin 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.

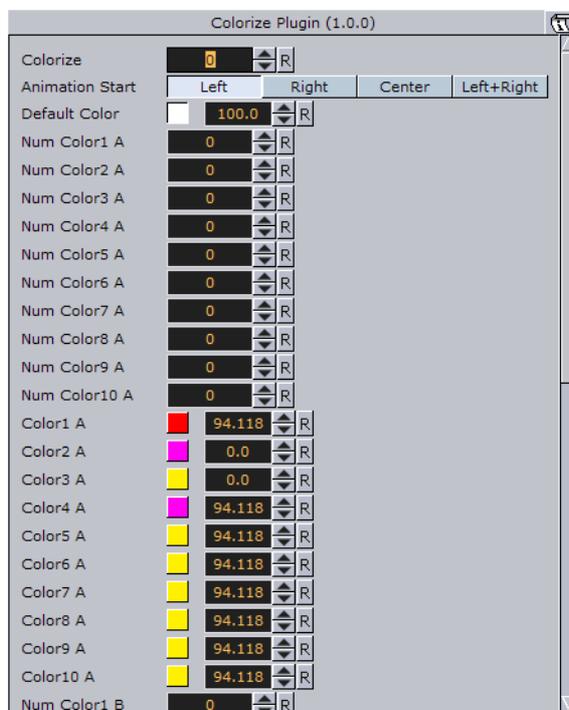
The plugin can define up to 10 colors. Each color is able to influence more than one container. The Num Color1-10 parameters allow you to link multiple containers to one color in the colorize plugin. Each of the Sub-Containers must have its own material for the plugin to work.

Note: This plugin is located in: Built-ins -> Container Plugins -> Tools

This section contains information on the following topics:

- [Colorize Properties](#)

Colorize Properties



- **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.

12.19.8 Counter



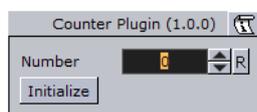
The counter is a simple plugin that can create a count up or count down at any given range and at any given speed. It shows integral numbers.

Note: This plugin is located in: Built-ins -> Container Plugins -> Tools

This section contains information on the following topics:

- [Counter Properties](#)
- [To use the counter](#)

Counter Properties



- **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 plugin to a container with a font, and click Initialize.

12.19.9 DVE Follow



DVE Follow connects a video input channel or a clip channel of the containing scene with the bounding box of a container. When connected the channel, if set to DVE, will follow the bounding box of the container. The plugin does not actively set the channel to DVE.

IMPORTANT! Be aware of the fact that the DVE animation is done on the video board while the animation of the ruling container is done together with all other graphics on the GPU. The output delays of the graphics and the DVE animation are different and this can lead to notable offsets between graphics and DVE (especially notable with fast animations).

DVE Follow Properties



- **Channel Type:** Type of channel to manipulate:
 - Video Channel
 - Clip Channel
- **Video/Clip Channel Number:** The plugin supports a maximum of 8 video channels and 16 clip channels.
- **Layer:**
- **Video Width:** Width of the containers bounding box.
- **Video Height:** Height of the containers bounding box.
- **Alpha:** Alpha value of the DVE channel.
- **Priority:** Priority value of the DVE channel.
- **Crop:** Left, right, top and bottom crop value of the DVE channel.

12.19.10 Heartbeat



The Heartbeat plugin creates a heart beat animation. Includes animation of:

- Size, along different axes, axis-pairs or all axes.

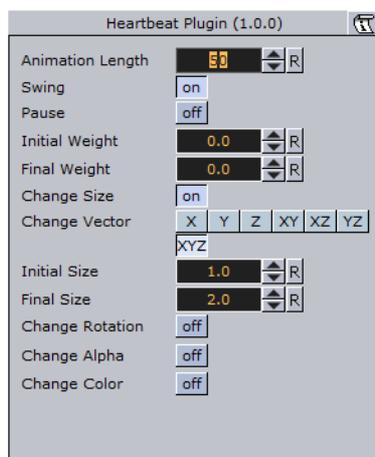
- Rotation, around three main axes.
- Alpha.
- Color.

Note: This plugin is located in: Built-ins -> Container Plugins -> Tools

This section contains information on the following topics:

- [Counter Properties](#)

Heartbeat Properties



- **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. See also 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`.

12.19.11 Image Link



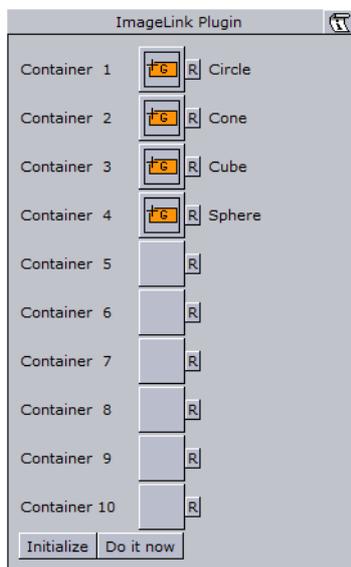
The Image Link plugin copies the image/texture from the source container to up to 10 linked containers. The image you load in the source container will be repeated in all the linked containers.

Note: This plugin is located in: Built-ins -> Container Plugins -> Tools

This section contains information on the following topics:

- [Image Link Properties](#)
- [To link an image to multiple containers](#)

Image Link Properties



- **Container 1: 10:** Drop zones for the containers you want to link.
- **Initialize:** Initializes the plugin.
- **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 plugin 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](#)

12.19.12 Jack



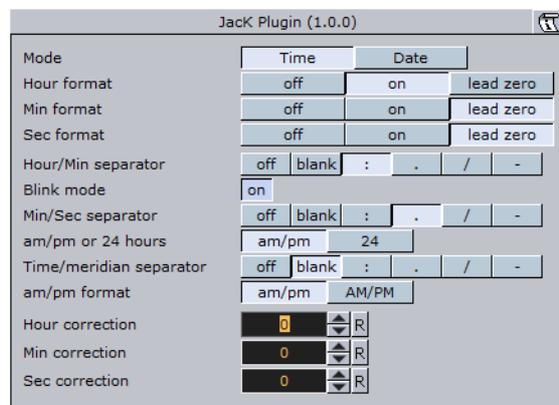
The Jack plugin allows you to show a digital clock and date. The plugin is highly customizable allowing you to set your own time and date formats and correction values.

Note: This plugin is located in: Built-ins -> Container Plugins -> Tools

This section contains information on the following topics:

- [Jack Properties](#)

Jack Properties

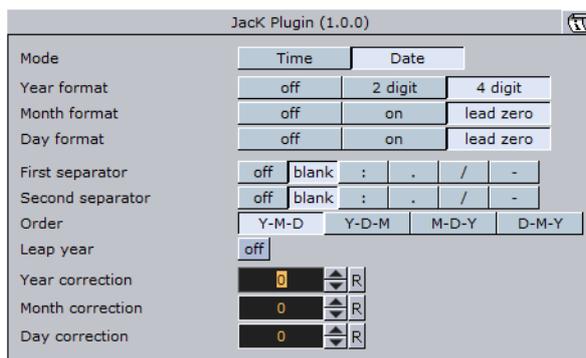


Time mode has the following properties:

- **Hour format:** Sets how to show hours (either off, on or with a leading zero if the value is less than 10).
- **Min format:** Sets how to show minutes (see Hour format).
- **Sec format:** Sets how to show 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 shown.

- **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 to show 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).



Date mode has the following properties:

- **Year format:** Sets how to show 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 shown, either off, on or with a leading zero from January to September.
- **Day format:** Sets how the day is shown (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.

12.19.13 Level Of Detail (LOD)

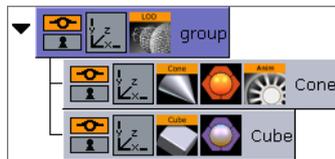


Level Of Detail (LOD) is a mechanism for controlling the level of complexity of a 3D object, depending on camera position and object size. As an object becomes smaller

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 losing 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 plugin can be used. The LOD plugin 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 plugin 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.



To use the LOD plugin 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 plugin 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 plugin.

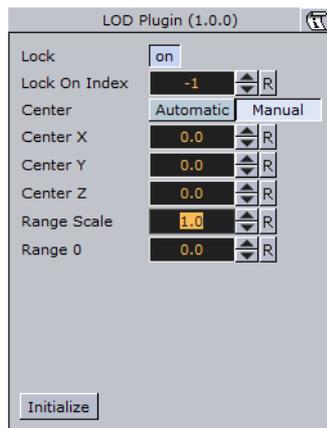
The LOD container plugin works in conjunction with the [Level Of Detail \(LOD\) Manager](#) scene plugin that controls all the LOD plugins in the scene.

Note: This plugin is located in: Built-ins -> Container Plugins -> Tools

This section contains information on the following topics:

- [LOD Properties](#)

LOD Properties



- **Lock:** In certain cases it is desirable to disable the distance switching and lock the LOD plugin to show one of the levels.
- **Lock On Index:** Shows 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 plugin. This is required if the number of children of the LOD group has changed after the LOD plugin was assigned to the group.

See Also

- [Level Of Detail \(LOD\) Manager](#)

12.19.14 Magnify



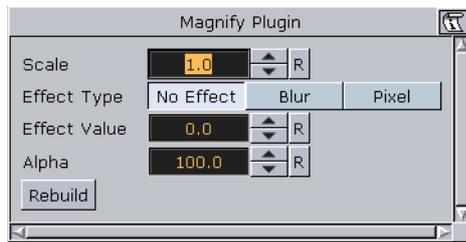
The Magnifying plugin offers the possibility to create a simulation of a looking glass and other similar effects. At an area defined by an overlying object an image can be shown in a magnified way.

Note: This plugin is located in: Built-ins -> Container Plugins -> Tools

This section contains information on the following topics:

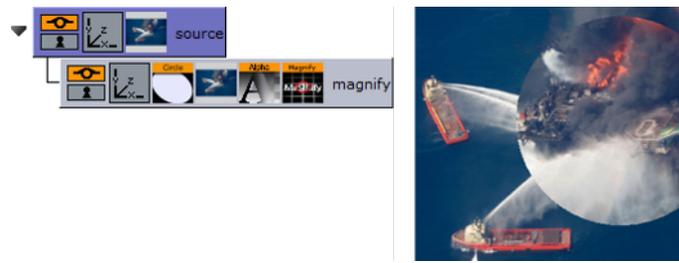
- [Magnify Properties](#)

Magnify Properties



- **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 plugin to the magnify container.
8. Open the Magnify editor and set Scale to 2.0.

12.19.15 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.

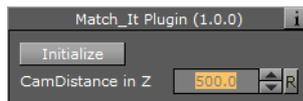
Note: Keep in mind that you have to re-initialize the plugin if the camera is moved.

Note: This plugin is located in: Built-ins -> Container Plugins -> Tools

This section contains information on the following topics:

- [Match It Properties](#)

Match It Properties



- **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.

12.19.16 Max Size



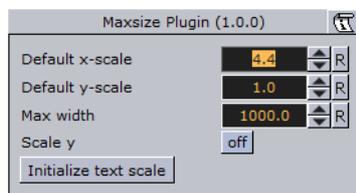
The Max size plugin is used to set max size parameters for text objects. This allows you to have control over the space the text should use such that it does not 'overflow' the designated text area.

Note: This plugin is located in: Built-ins -> Container Plugins -> Tools

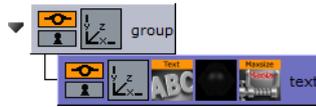
This section contains information on the following topics:

- [Max Size Properties](#)
- [To set max size](#)

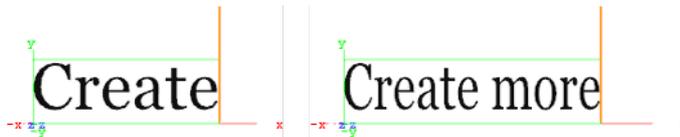
Max Size Properties



- **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 plugin 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](#)

12.19.17 Max Size Lines

The Max Size Lines plugin solves the problem of using [TextFX](#) with multiline text in right-to-left languages.

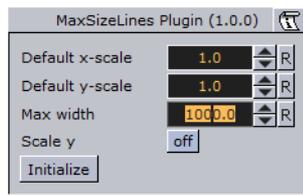
Note: Each line of text must be in separate containers, and organized as a sub group of the container holding the Max Size Lines plugin.

Note: This plugin is located in: Built-ins -> Container Plugins -> Tools

This section contains information on the following topics:

- [Max Size Lines Properties](#)

Max Size Lines Properties



- **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 plugin parameters.

See Also

- [Max Size](#)

12.19.18 Object Zoom



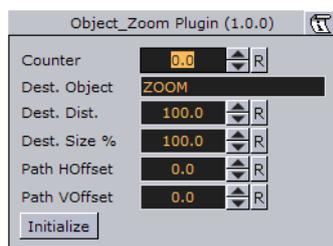
Object Zoom lets an object follow the camera and match its orientation. This creates an effect as if the camera would smoothly zoom to the object.

Note: This plugin is located in: Built-ins -> Container Plugins -> Default

This section contains information on the following topics:

- [Object Zoom Properties](#)
- [To zoom an object towards the camera](#)

Object Zoom Properties



- **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.

12.19.19 Omo



The Object moving (Omo) plugin 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 plugin 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.

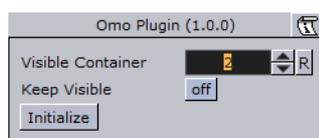
The plugin hides all Sub-Containers and shows them one by one. The one to be shown is controlled by the **Visible Container** parameter. By animating this value an animation sequence of all the containers is made. The plugin can also be used to show one group at a time.

Note: This plugin is located in: Built-ins -> Container Plugins -> Tools

This section contains information on the following topics:

- [Omo Properties](#)

Omo Properties



- **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 plugin. 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 plugin 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](#)

12.19.20 Pablo



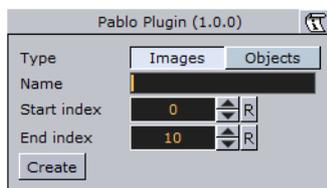
Pablo is an object import plugin that enables you to import multiple indexed images/objects from the pool under the local container Pablo is added to.

Note: This plugin is located in: Built-ins -> Container Plugins -> Tools

This section contains information on the following topics:

- [Pablo Properties](#)

Pablo Properties



- **Type:** Select whether 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.

12.19.21 Parliament



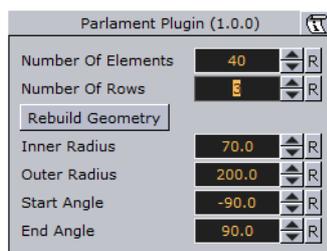
The Parliament plugin is specially developed for creating visualizations of parliament seats in graphics designed for election programs, but it can of course be used in other ways. The plugin creates a “parliament like” seating structure using an object of your selection.

Note: This plugin is located in: Built-ins -> Container Plugins -> Tools

This section contains information on the following topics:

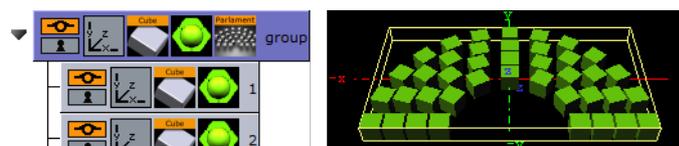
- [Parliament Properties](#)
- [To Create a Parliament Shape](#)

Parliament Properties



- **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 plugin to the container, and click the Initialize button.
 - By default the seating is created with 40 seats in 3 rows. The plugin 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 plugin 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 one 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.

12.19.22 Rotations Order



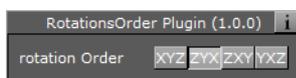
Rotations Order changes the rotation order of a container in Viz Artist to match the Softimage XSI rotation order. This plugin is automatically added to some containers when Softimage XSI scenes are imported into Viz Artist.

Note: This plugin is located in: Built-ins -> Container Plugins -> Tools

This section contains information on the following topics:

- [Rotations Order Properties](#)

Rotations Order Properties



- **Rotation Order:** Sets the order of rotation to either:
 - **XYZ**
 - **ZYX**
 - **ZXY**
 - **YXZ**

12.19.23 Slide Show



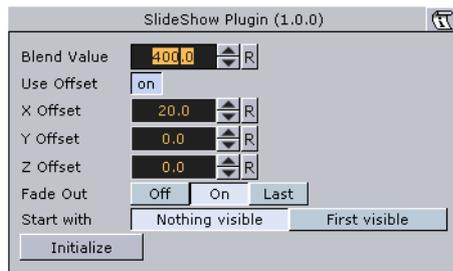
The Slide Show plugin allows you to create a sequence where a group of containers fade in and out one by one in the same manner as a normal slide show.

Note: This plugin is located in: Built-ins -> Container Plugins -> Tools

This section contains information on the following topics:

- [Slide Show Properties](#)

Slide Show Properties



- **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 from 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 plugin 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 plugin.
 - Each of the Sub-Containers automatically get an alpha icon.

12.19.24 System Time



The System Time plugin creates a variety of time and date settings based on the system time.

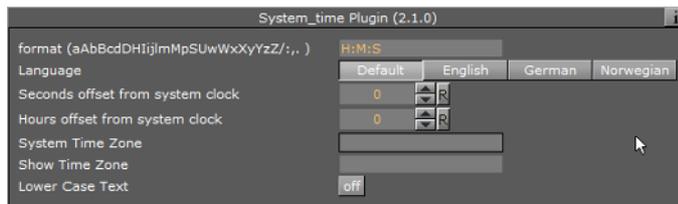
Note: This plugin is located in: Built-ins -> Container Plugins -> Tools

Note: Also see '[Calculate the Days of a Week and Month](#)' for greater control over time and date presentation.

This section contains information on the following topics:

- [System Time Properties](#)

System Time Properties



- **format (aAbBcdDHijlmMpSUwWxXyYzZ/;,.)**: Set the format of the system time.
- **Language**: Set the language format of the system time. Options:
 - Default (system specific)
 - English
 - German
 - Norwegian
- **Seconds offset from system clock**: Set the number of seconds the system clock should be offset (default is 0)
- **Hours offset from system clock**: Set the number of hours the system clock should be offset (default is 0)
- **System time zone**: Set the system time zone.
- **Show Time Zone**: Show the result of the system time zone.
- **Lower Case Text**: Show 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](#)

12.19.25 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 plugin is a single image consisting of many tiled and equally sized squares set up in a matrix.

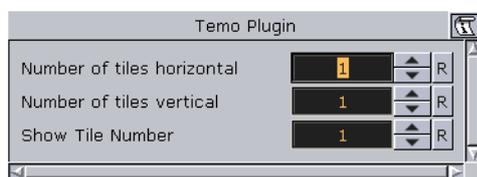
Each square of the image is made up to be a snapshot of an animation sequence, just like each picture frame is on a normal film. The image must be made up in advance with the aid of a image editing tool. In the property editor you tell the Temo plugin how many tiled squares there are in the X- and the Y-axis. The plugin is then able to show the tiles of the image one by one. By animating this, a "film like" sequence is created. The plugin has only a small influence on the rendering performance.

Note: This plugin is located in: Built-ins -> Container Plugins -> Tools

This section contains information on the following topics:

- [Temo Properties](#)

Temo Properties



- **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.

12.19.26 TextBG



The TextBG plugin, when used with the Rectangle, Noggi, Fade Rectangle or Cube Geometry plugins, creates a background which grows as text is typed in the Text Editor. To use other Geometry plugins click on the **Scale** option.

Note: Not all Geometry plugins will work with the TextBG plugin.

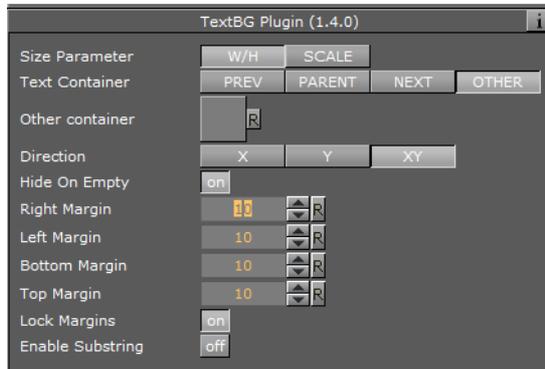
Note: The TextBG plugin can also be used with Geometries, Images and Video clips.

Note: This plugin is located in: Built-ins -> Container Plugins -> Tools

This section contains information on the following topics:

- [TextBG Properties](#)
- [To Add Background to the Previous or Next Containers](#)

TextBG Properties



- **Size Parameter:** Sets the size of the background to adjust to:
 - **W/H (width/height):** The width and height of the background geometry will adjust to the size of the typed text.
 - **Scale:** The background will scale to the size of the text.
- **Text Container:** Sets the TextBG plugin to use the background in selected containers:
 - **PREV (previous):** The background geometry is applied to the Container previous to the TextBG container.
 - **PARENT:** The background geometry is applied on the parent Container
 - **NEXT:** The background geometry is applied to the next container from the TextBG container.
 - **OTHER:** Select a Container (from the same group) to apply the background geometry to.
- **Other container:** Available when **OTHER** is selected. Drag a container (from the same group), other than the previous or next container, to the drop area. Click on **R** to remove the Container.
- **Direction:** Sets the direction in which the background should “grow” 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 is 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 the Previous or Next Containers



1. Add a group Container to the Scene Tree.
2. Add three Sub-Containers to the group container:
 - Name the top one Text1.
 - Name the bottom one Background.
3. Add a Fontstyle to the Text1 Container.
4. Open the Transformation Editor of the Text1 Container.
 1. Set **Position Z** to 1.0.
 2. Type some text (for example, More Control).
5. Copy the Text1 container and place below the Background container.
6. Rename the new Text container to Text2.
7. Add a Geometry (for example, Noggi), a material and the TextBG plugin to the Background container.
8. Open the TextBG editor.
9. Toggle the **Text Container** property from **PREV** to **NEXT**. The background geometry moves from the Text1 container (the previous container) to Text2 (the next container).

To apply a background to another Container, other than the previous or next Containers (in the same group), click on **OTHER** (in the TextBG properties) and drag the selected Container to the drop area.

To apply a background to the parent Container click on **PARENT** (in the TextBG properties). In this case the parent Container needs to hold a text.

Note: If the text in the Text2 container is of a different size the background geometry will size to the text.

Note: Sub string is not supported together with multiple texts.

12.19.27 Text Link



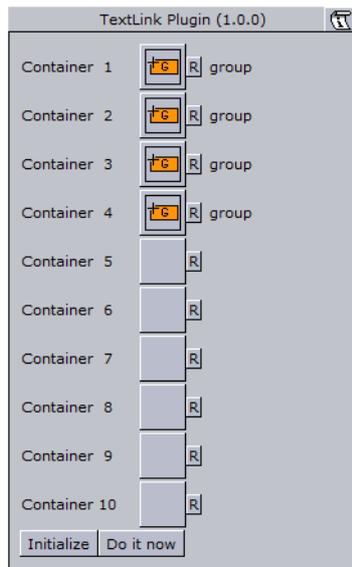
The Text Link plugin copies the text from the source container to up to 10 linked containers. The text you enter in the source container will be repeated in all the linked containers. The linked containers must contain a font.

Note: This plugin is located in: Built-ins -> Container Plugins -> Tools

This section contains information on the following topics:

- [Text Link Properties](#)
- [To link text to multiple containers](#)

Text Link Properties



- **Container 1: 10:** Drop zones for the containers you want to link.
- **Initialize:** Initializes the plugin.
- **Do it now:** Performs the operation immediately.

To link text to multiple containers

1. Add a group container to the scene tree.
2. Add the Text Link plugin 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](#)

12.19.28 Text Parameters



The Text Parameters plugin (TextParam plugin) gives the user the possibility to animate text parameters because standard text parameters in the Text Editor cannot be animated. The parameters for the Text Parameters can only be used for animation.

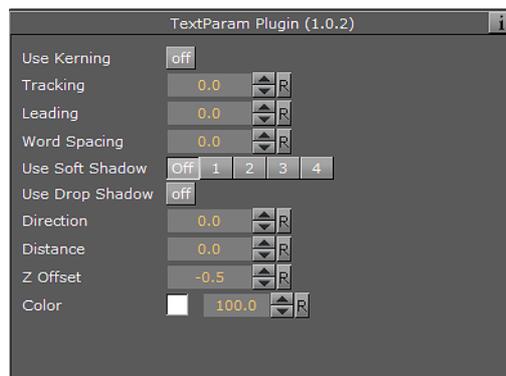
The settings are not visible on the text until an animation update has been made.

Note: This plugin is located in: Built-ins -> Container Plugins -> Tools

This section contains information on the following topics:

- [Text Parameters Properties](#)

Text Parameters Properties



- **Use Kerning:** Enable or disable the use of Kerning.
- **Tracking:** Set the space between each letter.
- **Leading:** Set the space between each line of text.
- **Word Spacing:** Set the space between each word.
- **Use Soft Shadow:** Enable 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:** Enable or disable Drop Shadow.
- **Direction:** Set the direction of the Drop Shadow in degrees.
- **Distance:** Set the distance of the Drop Shadow.
- **Z Offset:** Set the offset of the Drop Shadow in the Z-axis.
- **Color:** Set the color of the shadow.

12.19.29 VCF Parameter



The VCF (Virtual Camera Flight) Parameter container plugin works in conjunction with the [VCF](#) scene plugin. 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 studio expansion components.

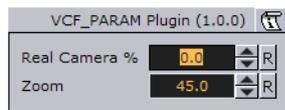
You must have a real camera with data tracking enabled, which is set in remote mode in the camera editor. In case you have several tracked cameras, the virtual camera will interpolate its position to the real camera that is selected on air.

Note: This plugin is located in: Built-ins -> Container Plugins -> Tools

This section contains information on the following topics:

- [VCF Parameter Properties](#)
- [To use the virtual camera flight parameter plugin](#)

VCF Parameter Properties



- **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 plugin

1. Add the [VCF](#) scene plugin to the scene setup plugins 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 plugin (e.g. T_POS and T_ROT).
4. Click **Initialize** to finish.
5. Add the VCF Parameter container plugin 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 Key Frame (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 studio in the scene setup: background.
8. Make sure your FG object/talent are not seen while switching to the real camera. They should show in the frame after the switch.

Note: It is not possible to animate the roll of the camera.

See Also

- [VCF](#)

- [Create Animations](#)

12.20 Transformation

The following container plugins are located in the Transformation folder:

- [Justifier](#)

12.20.1 Justifier



The Justifier plugin 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 Artist/Engine coordinate system.

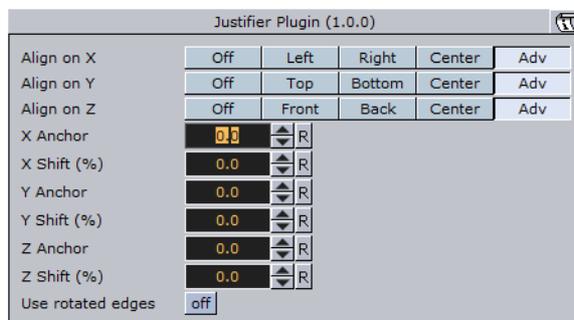
This plugin acts on the local space of the container it is attached to. So a parent container can be used to rotate the container for use of arbitrary (world space) axes.

Note: This plugin is located in: Built-ins -> Container Plugins -> Transformation

This section contains information on the following topics:

- [Justifier Properties](#)

Justifier Properties



- **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.

12.21 Visual Data Tools

The following container plugins are located in the Visual Data Tools folder:

- [Data Fit](#)
- [Data Import](#)
- [Data Label](#)
- [Data Storage](#)

See Also

- Visual Data Tools [Overview](#)
- [Area Chart](#)
- [Bar Chart](#)
- [Line Chart](#)
- [Pie Chart](#)
- [Scatter Chart](#)
- [Stock Chart](#)
- Tutorial on www.vizrt.com under the Training section

12.21.1 Data Fit



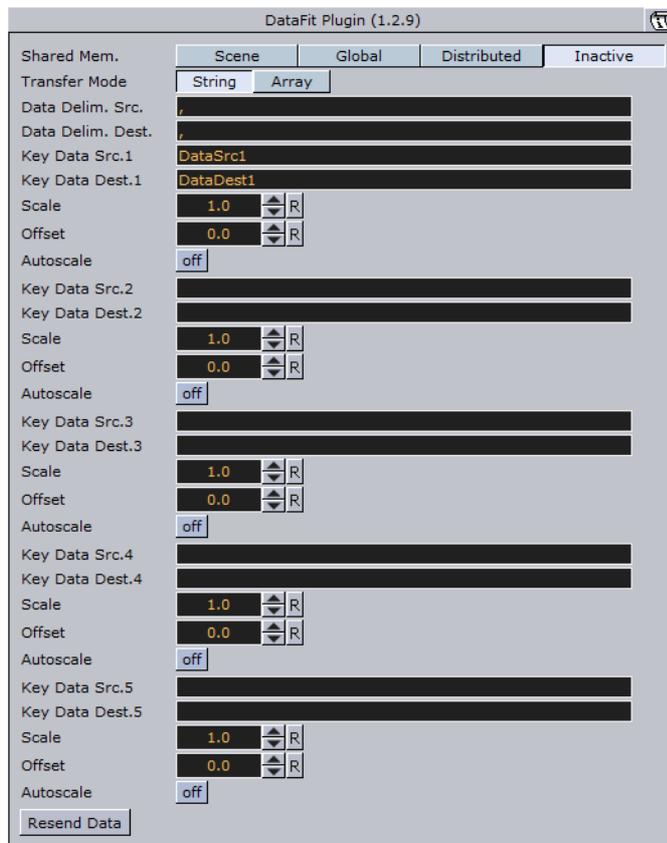
Data Fit listens for incoming data and modifies and redistributes it to another Shared Memory key.

Note: This plugin is located in: Built-ins -> Container Plugins -> VisualDataTools

This section contains information on the following topics:

- [Data Fit Properties](#)
- [To Create a Scene with Data Fit](#)

Data Fit Properties



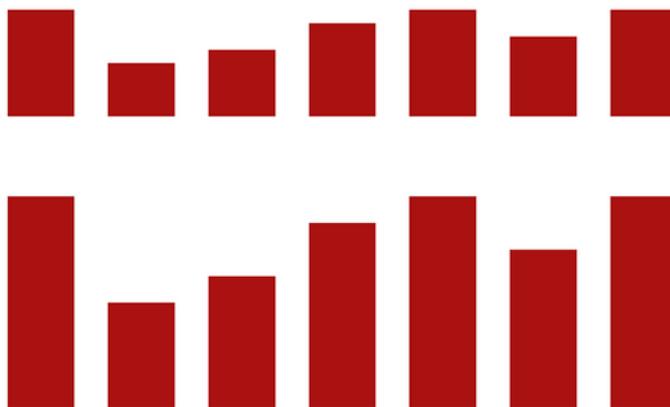
- **Shared Mem:** Changes between Scene-, Global- and Distributed-Shared Memory. Use Inactive memory to not forward any values via Shared Memory.
- **Transfer Mode:** Sets string- or array-based data transfer.
- **Data Delim. Src.:** Defines the value separator sign(s) for all data sources.
- **Data 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 with 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.



12.21.2 Data Import



This plugin enables Microsoft Excel (.xls and .xlsx) and Microsoft Access (.mdb) file import via ADO, and distributes it to a text field or a shared memory map. The plugin supports reading both numbers and text.

IMPORTANT! Viz Artist 64-Bit: If Microsoft Office 32-Bit is installed on a x64 Windows machine, with Viz Artist/Engine 64-Bit installed, uninstall Microsoft Office completely before Microsoft Access Database Engine 2010 Redistributable is installed. If, for any reason, Microsoft Office is required, it must be the 64-Bit version.

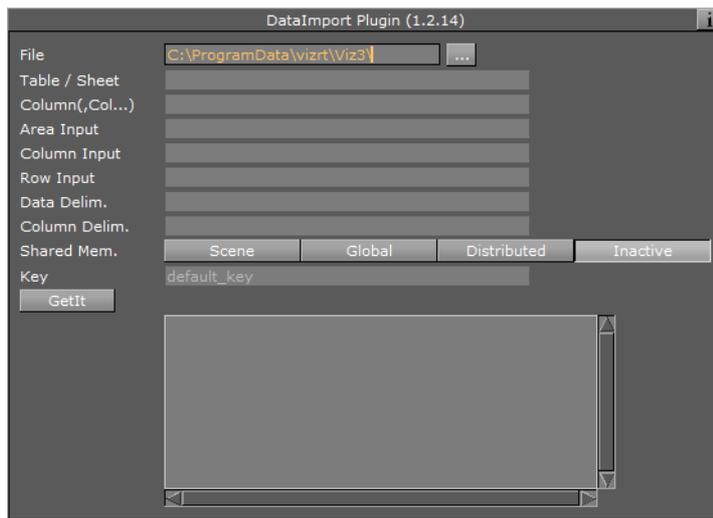
Note: To access Excel or Access data using this plugin install the [Microsoft Access Database Engine 2010 Redistributable](#).

Note: This plugin is located in: *Built-ins -> Container Plugins -> VisualDataTools*

This section contains information on the following topics:

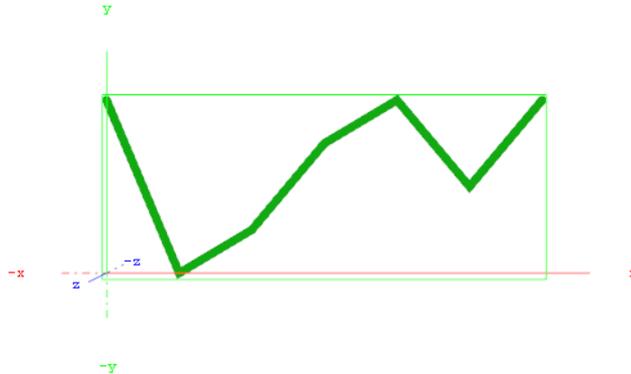
- [Data Import Properties](#)
- [To Create a Scene with Data Import](#)

Data Import Properties



- **File:** Choose the required 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 required data and shows it in the text box

To Create a Scene with Data Import



This procedure will import data from an Microsoft Excel sheet and feed the [Line Chart](#) plugin with data through Shared Memory.

1. Start Microsoft Excel.
2. Enter **ExcelDataY** into cell A1
3. Add some sample 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 for example *ExcelData.xls* at C:\
6. Start Viz Artist and **create a new scene**



7. Add a group container to the scene tree
8. Drag and drop the [Line Chart](#) and **Data Import** plugin to the group container
9. Add a material to the same group container
10. Open the [Transformation Editor](#) and set **Position X** to -200.0 and **Position Y** to -100.0
11. Set **Shared Mem.** to **Scene** with **MyDataY** as Key name for both plugins
12. Set **Data Delim.** to **#** for both plugins
13. For the *Line Chart* plugin set **ChartWidth** to 500.0 .
14. Activate **DataY Fit**, **DataY Auto Scale** and **DataY Detect Limits**
15. Adjust **DataY Stop** to 200.0
16. Activate **Const. Thickness** to give the chart a constant line width
17. For the *Data Import* plugin set the stored Excel file as the **File** parameter
18. Enter **MyTable** for the Table/Sheet parameter and **ExcelDataY** for Column (,Col...)
19. Click the **GetIt** button and the chart shows

12.21.3 Data Label



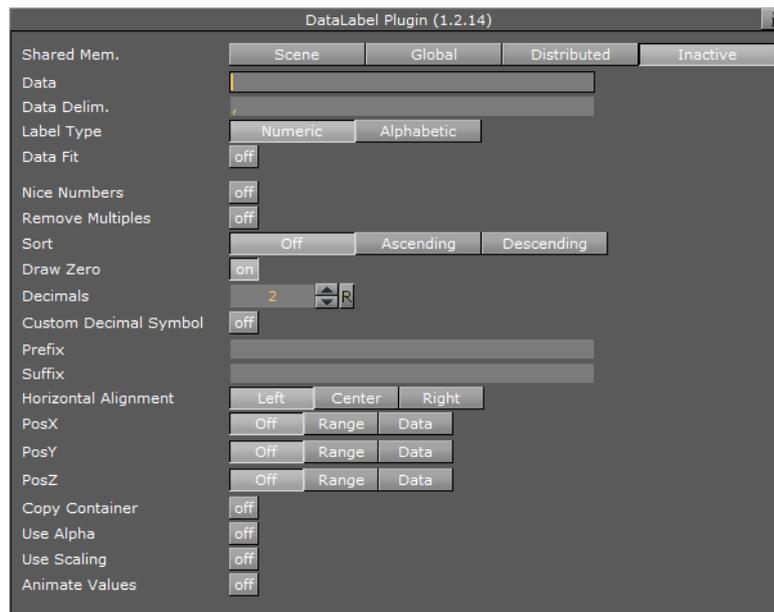
This plugin enables graph labeling for all Financial Plugins. All containers beneath the current container get labeled with the values from your Shared Memory variables.

Note: This plugin is located in: *Built-ins -> Container Plugins -> VisualDataTools*

This section contains information on the following topics:

- [Data Label Properties](#)
- [To Create a Scene with Data Label](#)

Data Label Properties



- **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 (to show 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 data 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 its 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 with 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 Key Frame, set Total Length[%] back to 100.0 and create another Key Frame.
 - 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.

12.21.4 Data Storage



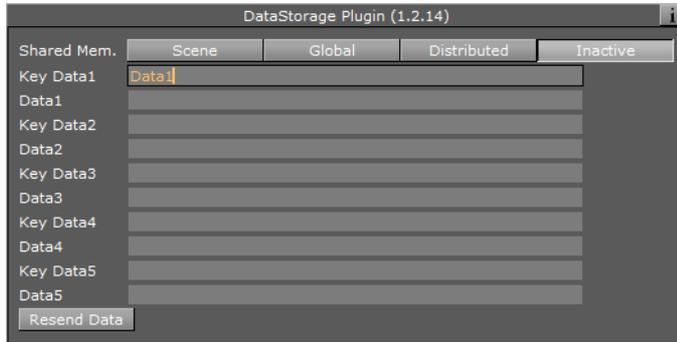
This plugin stores data in the current scene and sends it to the other plugins on startup. This plugin helps artists during scene design if there is no data source available, and to hold data within e.g. the scene (see [Data Sharing](#)).

Note: This plugin is located in: *Built-ins -> Container Plugins -> VisualDataTools*

This section contains information on the following topics:

- [Data Storage Properties](#)
- [To Store and Send Data](#)
- [To Enable Data Input from External Sources](#)

Data Storage Properties



- **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.

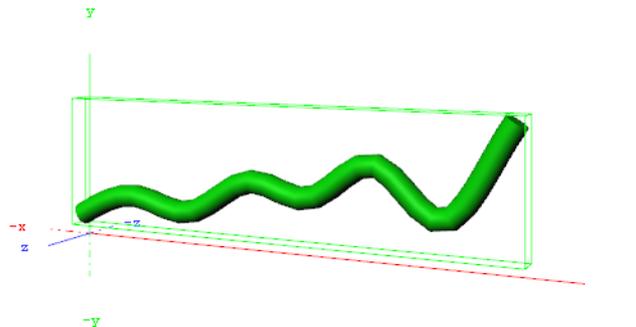
To Store and Send Data

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



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

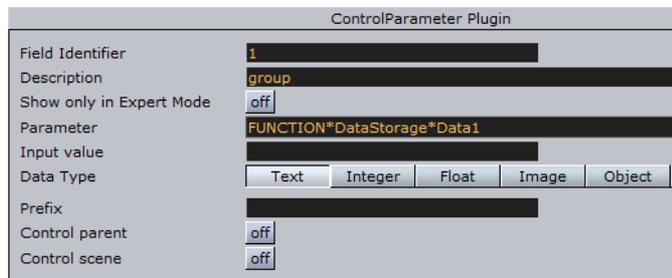
Note: The chart plugins have different property names (e.g. Data and Key Data). Also, some have X, Y or Z appended at the end, defining the axis.



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



The following example builds on the previous example on how [To Store and Send Data](#).

1. Add the [Control Parameter](#) to the scene tree
 - The [Control Object](#) plugin is automatically added
2. Open the Control Parameter editor
3. Enter **FUNCTION*DataStorage*Data1** into the **Parameter** field
4. 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 with Data Fit](#)
- [To Create a Scene with Data Import](#)
- [To Create a Scene with Data Label](#)
- [To create a scene using area chart and data storage](#)
- [Data Sharing](#)

13 Shader Plugins



Shader plugins are located in these folders:

Default	PixelFXLensFlare
Effects	RealFX
Filter	RTT Advanced Materials
Material	Texture
PixelFX	

Note: The default path for the Shader plugins is: <viz install folder>\plugin
 \<plugin name.vip>

13.1 Default

This folder is empty and reserved for custom Shader plugins.

13.2 Effects

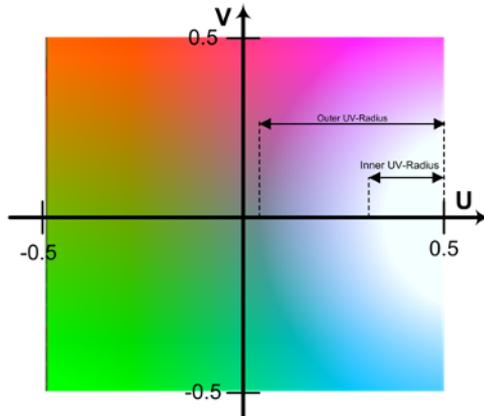
The following Shader plugins are located in the Effects folder:

- [Chroma Keyer](#)
- [Fluid](#)
- [Frame Mask](#)
- [Image Mask](#)
- [Soft Mask](#)
- [Water Shader](#)

13.2.1 Chroma Keyer



This Shader clears regions in a video defined by a color. The colorspace of this Shader is YUV which is defined as following:



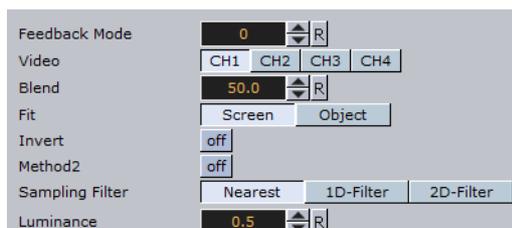
The luminance (Y) component determines the brightness of the color, while the U and V components determine the color itself (the chroma). Luminance (Y) ranges from 0.0 (dark) to 1.0 (bright). U and V ranges from -0.5 to 0.5.

Note: This plugin is located in: Built-ins -> Shader -> Effects

This section contains information on the following topics:

- [Chroma Keyer Properties](#)

Chroma Keyer Properties



- **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 [Keying Mode for Video](#)).
- **Blend:** Sets the alpha for the key signal
- **Fit:** Select Screen or Object
- **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	▲▼	R
OpeningAngle	10.0	▲▼	R
KeyGain	1.0	▲▼	R
SatMin	0.0	▲▼	R
SatMinGrad	0.1	▲▼	R
SatMax	1.0	▲▼	R
SatMaxGrad	0.1	▲▼	R

- **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	▲▼	R
V-Color:	0.3	▲▼	R
UVDiameter	0.05	▲▼	R
UVGradient	0.2	▲▼	R
UVScale	1.0	▲▼	R

- **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	▲▼	R
LumMinGrad	0.1	▲▼	R
LumMax	0.15	▲▼	R
LumMaxGrad	0.1	▲▼	R

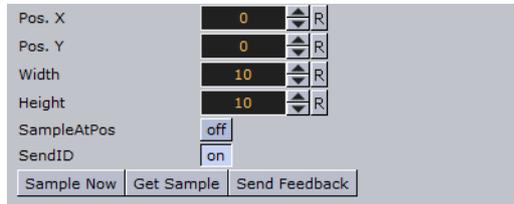
- **LumMin:** Lower luminance limit for keyed colors.
- **LumMinGrad:** Gradient for lower luminance limit (0=sharp, 1=soft).
- **LumMax:** Upper luminance limit for keyed colors.
- **LumMaxGrad:** Gradient for upper luminance limit (0=sharp, 1=soft).

Sampling reference point:

UVRangeCheck	on		
URangeMin	0.0	▲▼	R
URangeMax	1.0	▲▼	R
VRangeMin	0.0	▲▼	R
VRangeMax	1.0	▲▼	R
SampleTrustiness[%]	0.0	▲▼	R

- **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:** 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.

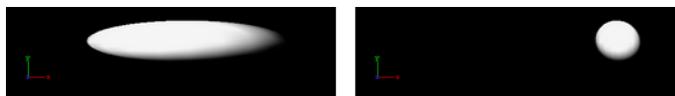
See Also

- [Keying Mode for Video](#)

13.2.2 Fluid



With Fluid you can get a geometry mesh to be elastic during animation.

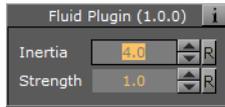


Note: This plugin is located in: Built-ins -> Shader -> Effects

This section contains information on the following topics:

- [Fluid Properties](#)

Fluid Properties



- **Inertia:** Sets the inertia of the motion.
- **Strength:** Sets the strength of the motion.

13.2.3 Frame Mask



This plugin adds a framed mask around an object (e.g. image or geometry). In the properties the aspect ratio can also be set.

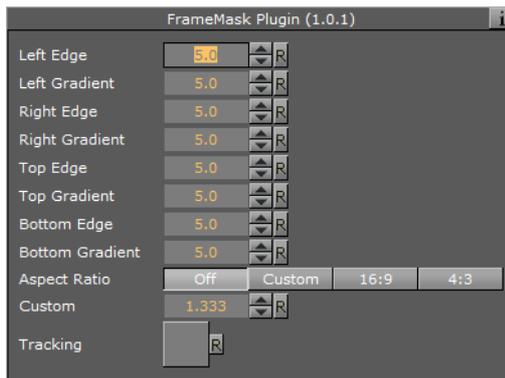


Note: This plugin is located in: Built-ins -> Shader -> Effects

This section contains information on the following topics:

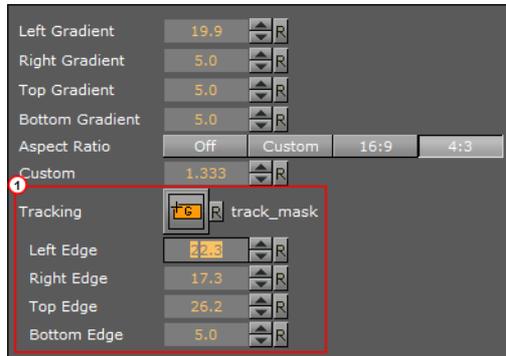
- [Frame Mask Properties](#)

Frame Mask Properties



- **Left, Right, Top and Bottom Edge:** Set the tracked borders (see also **Tracking**)
- **Left, Right, Top and Bottom Gradient:** Set the gradient from the edge of the frame to the edge of the object
- **Aspect Ratio:** Set the aspect ratio (**16:9**, **4:3** or **Custom**), if required (default is **Off**)
- **Custom:** When **Aspect Ratio** is set to **Custom**, use to set a custom aspect ratio (e.g. 1.85:1 or 2.39:1)

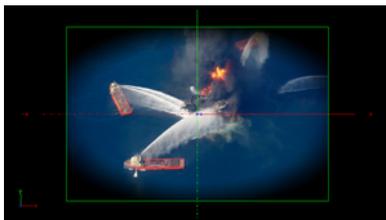
- **Tracking:** Instead of animating all tracked borders manually, use a Container to track the location of the actual frame mask. Drag a Container to the drop zone (1):



13.2.4 Image Mask



The Image Mask plugin applies a mask to an image using other images (e.g. an alpha image).

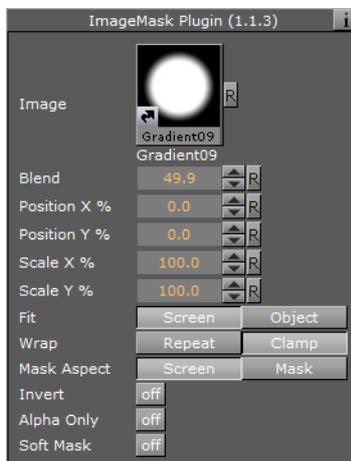


Note: This plugin is located in: Built-ins -> Shader -> Effects

This section contains information on the following topics:

- [Image Mask Properties](#)

Image Mask Properties



- **Image:** Placeholder for image
- **Blend:** Blend the image with the object
- **Position X, Y %:** Set the position of the mask in percentage for the screen or object
- **Scale X, Y %:** Set the size of the mask in percentage for the screen or object
- **Fit:** Select the size of the image mask to the screen or the object:
 - **Screen:** The image mask is sized to the screen. If the object is moved it will move while the image mask stays fixed
 - **Object:** The image mask is sized to the object. If the object is moved the image mask will move with it
- **Wrap:**
 - **Repeat:** The image mask is applied multiple times on the screen or object
 - **Clamp:** The image mask is applied once on the screen or object
- **Mask Aspect:** Set the aspect of the image mask to the screen or the object
- **Invert:** Inverts the image
- **Alpha only:** Enable the alpha channel for the image (for example, an RGBA image)
- **Soft Mask:** Apply a soft mask (available when **Fit** is set to **Screen**)

13.2.5 Soft Mask



The Soft Mask plugin applies a soft mask to an object (e.g. image or geometry).

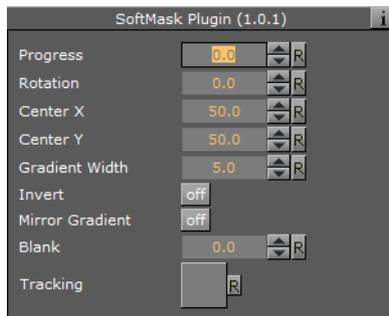


Note: This plugin is located in: Built-ins -> Shader -> Effects

This section contains information on the following topics:

- [Soft Mask Properties](#)

Soft Mask Properties

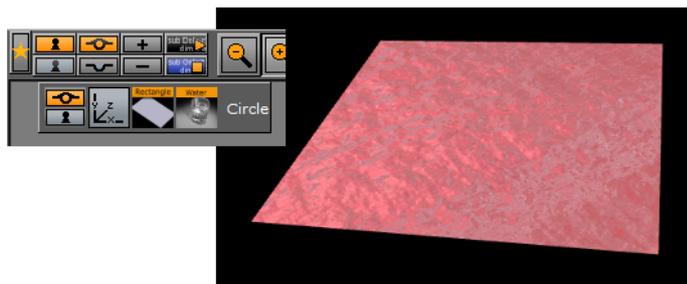


- **Progress:** Move the mask perpendicular to the mask direction. Default is 0.0 and is based on the configured center position.
- **Rotation:** Rotate the mask around the center.
- **Center X and Y:** Define the X and Y position of the center the mask will have.
- **Gradient Width:** Set the size of the gradient.
- **Invert:** Invert the mask.
- **Mirror Gradient:** Mirror the mask on both axis.
- **Blank:** Offset the area that should not be affected by the mask.
- **Tracking:** The progress can be controlled by a tracked Container to build the progress animation easier. Drag a Container to the drop zone

13.2.6 Water Shader



This shader simulates a water surface.

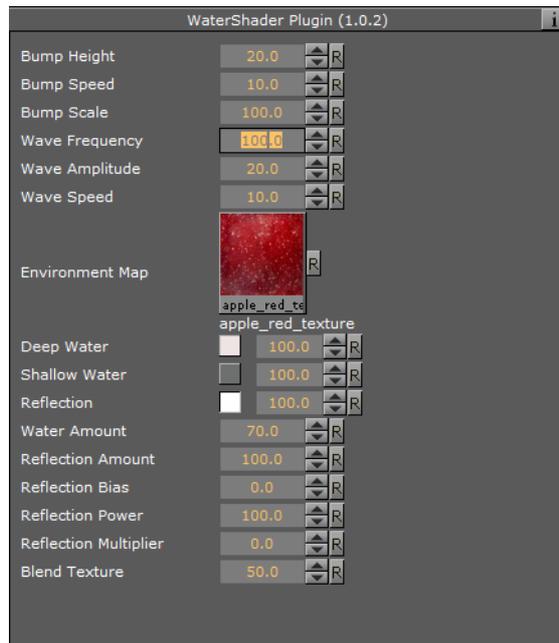


Note: This plugin is located in: Built-ins -> Shader -> Effects

This section contains information on the following topics:

- [Water Shader Properties](#)

Water Shader Properties



- **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

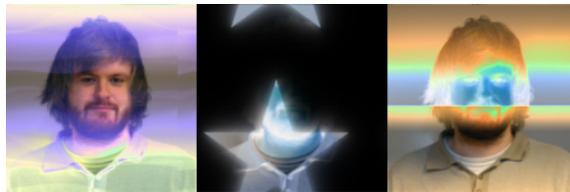
The following shader plugins are located in the Filter folder:

- [Blend Image](#)
- [Blur](#)
- [Color Balance](#)
- [Radial Blur](#)
- [Sepia](#)
- [Sharpen](#)

13.3.1 Blend Image



Use this plugin to blend two images in different ways.

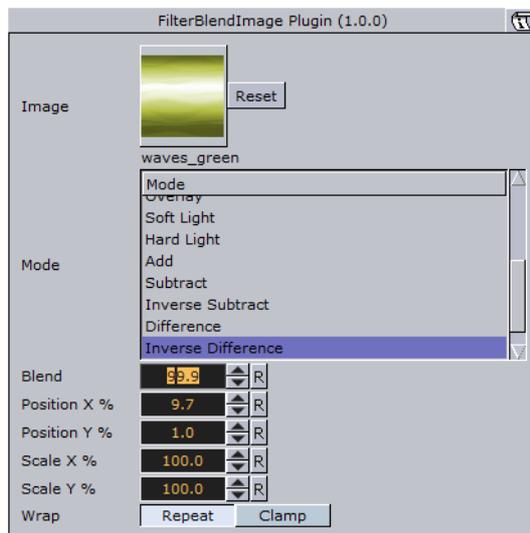


Note: This plugin is located in: Built-ins -> Shader -> Filter

This section contains information on the following topics:

- [Blend Image Properties](#)

Blend Image Properties



- **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 inverts 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 inverts 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 plugin to blur an object with an image texture and/or a material on it. Values can be animated. You also have a blur plugin 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.

Note: This plugin is located in: Built-ins -> Shader -> Filter

This section contains information on the following topics:

- [Blur Properties](#)

Blur Properties



- **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.



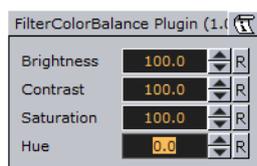
Tip: Most shaders also work on video clips.

Note: This plugin is located in: Built-ins -> Shader -> Filter

This section contains information on the following topics:

- [Color Balance Properties](#)

Color Balance Properties



- **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 plugin 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.

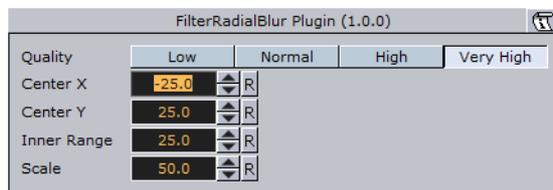
Note: Radial blur does not clamp the object.

Note: This plugin is located in: Built-ins -> Shader -> Filter

This section contains information on the following topics:

- [Radial Blur Properties](#)

Radial Blur Properties



- **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 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 plugin.

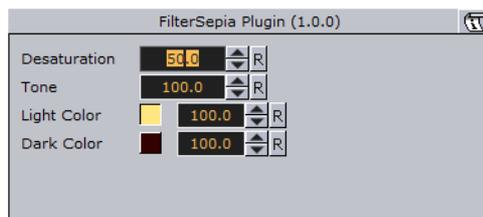
Note: The color range is from 0.0 - 100.0 divided by 255 (0.392, 0.784 etc.).

Note: This plugin is located in: Built-ins -> Shader -> Filter

This section contains information on the following topics:

- [Sepia Properties](#)

Sepia Properties



- **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



Use this plugin for sharpening the given images on objects.



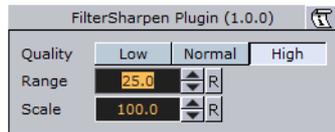
The samples above depict the same image with Quality set to High, and Scale values at 0.0, 100.0, 200.0 and 300.0, respectively. Range is set to 25.0 (default). The first image is the original (reference) with Scale set to 0.0.

Note: This plugin is located in: Built-ins -> Shader -> Filter

This section contains information on the following topics:

- [Sharpen Properties](#)

Sharpen Properties



- **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 located 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.

In the figure below the left image is with anisotropic light and the right image is with a regular material.

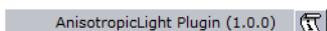


Note: This plugin is located in: Built-ins -> Shader -> Material

This section contains information on the following topics:

- [Anisotropic Light Properties](#)

Anisotropic Light Properties



This plugin does not have any properties or parameters.

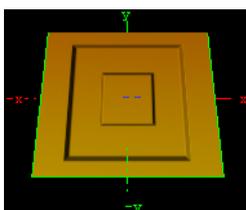
13.4.2 Bump Map



With this plugin 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 plugins 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.

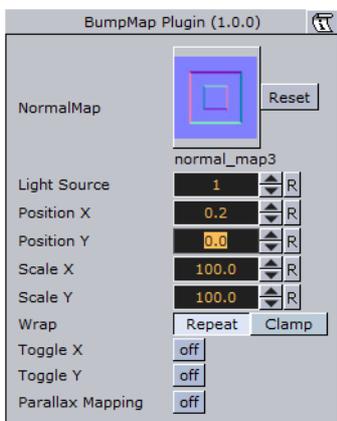


Note: This plugin is located in: Built-ins -> Shader -> Material

This section contains information on the following topics:

- [Bump Map Properties](#)

Bump Map Properties



- **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.

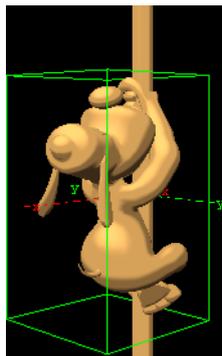
See Also

- [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.

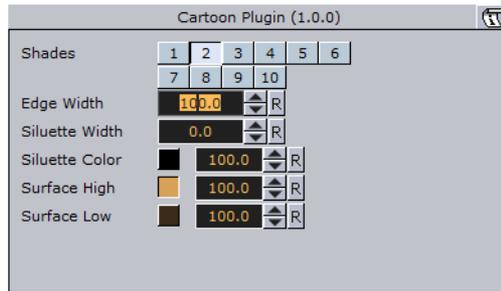


Note: This plugin is located in: Built-ins -> Shader -> Material

This section contains information on the following topics:

- [Cartoon Properties](#)

Cartoon Properties



- **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.

In the figure below the left image is with Gooch and the right image is with a regular material.

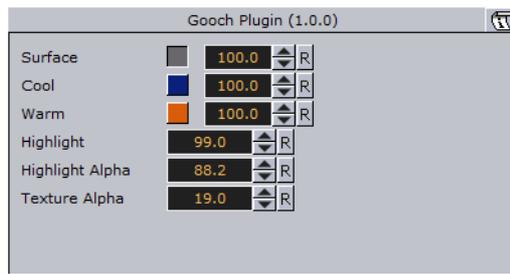


Note: This plugin is located in: Built-ins -> Shader -> Material

This section contains information on the following topics:

- [Gooch Properties](#)

Gooch Properties



- **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.

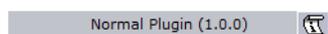
If the Normal is pointing along the Y-axis, this will give the RGB value (100,200,100). A normal along the X-axis give the RGB value (200,100,100). 45° between x and Y-axis will give RGB value (150,150,100).

Note: This plugin is located in: Built-ins -> Shader -> Material

This section contains information on the following topics:

- [Normal Map Properties](#)

Normal Map Properties



This plugin does not have any properties or parameters.

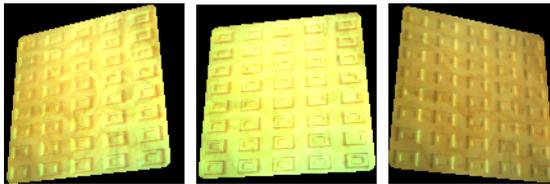
13.4.6 Simple Bump Map



With this plugin 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](#) plugin instead which is using an RGB normal map image.

To generate a Simple Bump Map you can use the plugins 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 grayscale image (white -> highest, black -> lowest bumps).



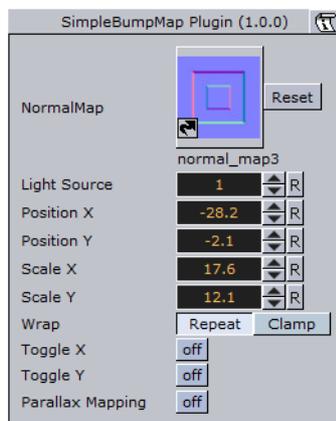
Note: All calculations are done in object space.

Note: This plugin is located in: Built-ins -> Shader -> Material

This section contains information on the following topics:

- [Simple Bump Map Properties](#)

Simple Bump Map Properties



- **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 PixelFX

The Pixel Shader plugins are part of the [PixelFX plugins](#) set.

Viz Artist 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 are not.

The following Shader plugins are located in the *PixelFX* folder:

Common Properties	pxPixelate
pxBCubic	pxPosterize
pxCCBase	pxRecolor
pxEqualize	pxRipple
pxGradient	pxSparkle
pxInvert	pxTurbDissolve and pxTurbWipe
pxLensDistort	pxTurbulence
pxMotionBlur	pxTwirl
pxNoise	pxWaves

13.5.1 Common Properties

Most of the plugin icons there are mini icons that show certain properties of the plugin in question:



- At the bottom left, there is a performance bar, with three blocks (like a cellphone battery indicator). One block means the plugin will not have a drastic performance effect, and three means the plugin is taxing and should therefore be used with caution.
- At the bottom right the orange stack shows if the plugin is stackable, meaning 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.

13.5.2 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.

Note: This plugin is located in: Built-ins -> Shader -> PixelFX

13.5.3 pxCCBase



The pxCCBase (Color Correction) applies a cumulative set of color corrections using the [pxStack](#) plugin.

Note: This plugin is located in: Built-ins -> Shader -> PixelFX

13.5.4 pxEqualize



pxEqualize allows you to adjust the black, mid-range and white color levels in an image.

Note: This plugin is located in: Built-ins -> Shader -> PixelFX

13.5.5 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 make sure 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.

Note: This plugin is located in: Built-ins -> Shader -> PixelFX

13.5.6 pxInvert



Often you create an image with an alpha channel in Photoshop, import it to Viz Artist/Engine and only then realize that your alpha channel is inverted. Or you find an image in the Viz Artist/Engine 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.

Note: This plugin is located in: Built-ins -> Shader -> PixelFX

13.5.7 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.

Note: This plugin is located in: Built-ins -> Shader -> PixelFX

13.5.8 pxMotionBlur



pxMotionBlur applies motion blur to the texture. Supports [pxColorWorks](#) and [pxStack](#) plugins.

Adjust the following parameters as required to achieve the required effect:

- X Center
- Y Center
- Rotate
- Zoom
- X Shift
- Y Shift
- Fix Edges (on/off)
- Samples (enter a value)

Note: This plugin is located in: Built-ins -> Shader -> PixelFX

13.5.9 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 required 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.

Note: This plugin is located in: Built-ins -> Shader -> PixelFX

13.5.10 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 required effect:

- Lock X/Y (on/off)
- X Size
- Y Size
- Smoothness

Supports [pxColorWorks](#) and [pxStack](#) plugins.

Note: This plugin is located in: Built-ins -> Shader -> PixelFX

13.5.11 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.

Note: This plugin is located in: Built-ins -> Shader -> PixelFX

13.5.12 pxRecolor



pxRecolor 'false-colors' an image by using either Red, Green, Blue or Luma as the input to a color ramp.

Note: This plugin is located in: Built-ins -> Shader -> PixelFX

13.5.13 pxRipple



pxRipple creates a pond ripple effect. Supports [pxColorWorks](#) and [pxStack](#) plugins.

Adjust the following parameters as required to achieve the required effect:

- X Center
- Y Center
- Radius
- Amount
- Wavelength
- Progress

Note: This plugin is located in: Built-ins -> Shader -> PixelFX

13.5.14 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 set 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.

Note: This plugin is located in: Built-ins -> Shader -> PixelFX

13.5.15 pxTurbDissolve and pxTurbWipe



Both pxTurbDissolve and pxTurbWipe utilize the native multi-texturing support of Viz Artist.

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 make sure 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 make sure that the Inheritable option is inactive.
5. Apply the Dissolve plugin to the Image A (the parent image).

- In the plugin's editor, first adjust the Transition, followed by the other parameters, as per your liking.

Note: This plugin is located in: Built-ins -> Shader -> PixelFX

See Also

- [pxGradient](#)

13.5.16 pxTurbulence



pxTurbulence creates a distortion effect using a Perlin turbulence function. Supports [pxColorWorks](#) and [pxStack](#) plugins. Adjust the wavelength, amplitude and speed to achieve the required effect.

Note: This plugin is located in: Built-ins -> Shader -> PixelFX

13.5.17 pxTwirl



pxTwirl creates a distortion effect that twists an image around a central point. Supports [pxColorWorks](#) plugins and [pxStack](#) plugins.

Adjust the following parameters as required to achieve the required effect:

- X Center
- Y Center
- Radius
- Amount
- Edge Softness

Note: This plugin is located in: Built-ins -> Shader -> PixelFX

13.5.18 pxWaves



pxWaves creates a distortion effect that makes an image look wavy. Supports [pxColorWorks](#) plugins and [pxStack](#).

Adjust the following parameters as required to achieve the required effect:

- Type: Sine, Noise, Triangle, Square

- Angle
- Amount
- Wavelength
- Progress
- Seed

Note: This plugin is located in: Built-ins -> Shader -> PixelFX

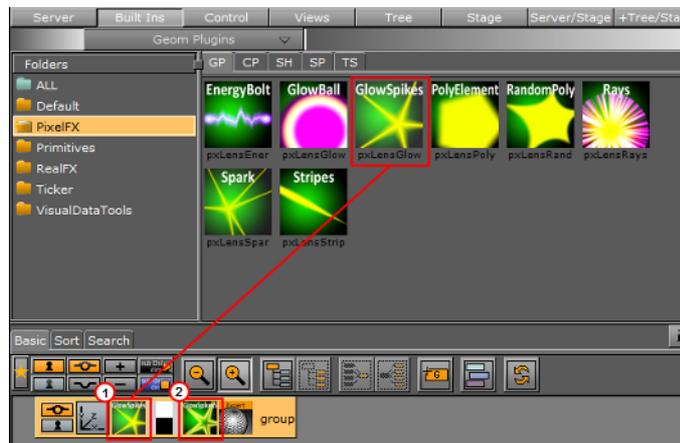
13.6 PixelFXLenseFlare

The Lense Flare plugins, which are part of the [PixelFX plugins](#) set, 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 lense flare or simple flare effects.

The Flare draws itself as a combination of geometry shader and pixel shader.

When a Flare Container is created (1) a non-editable pixel shader (2) is automatically added to the created Flair Container.

The non editable shaders are needed to do more (detailed, on a per pixel basis) graphical effects on the GPU (manipulating colors, textures, lighting, etc.), which cannot be done with Geometries only.



These plugins are located at: Built-ins -> Shader -> PixelFXLenseFlare

See Also

- [PixelFX Plugins](#) in [Geometry Plugins](#)

13.7 RealFX

The RealFX plugin set enables you to create particle effects in Viz Artist.

See Also

- [RealFX Plugins](#) in [Geometry Plugins](#)

- [pxColorWorks](#) in [Container Plugins](#)

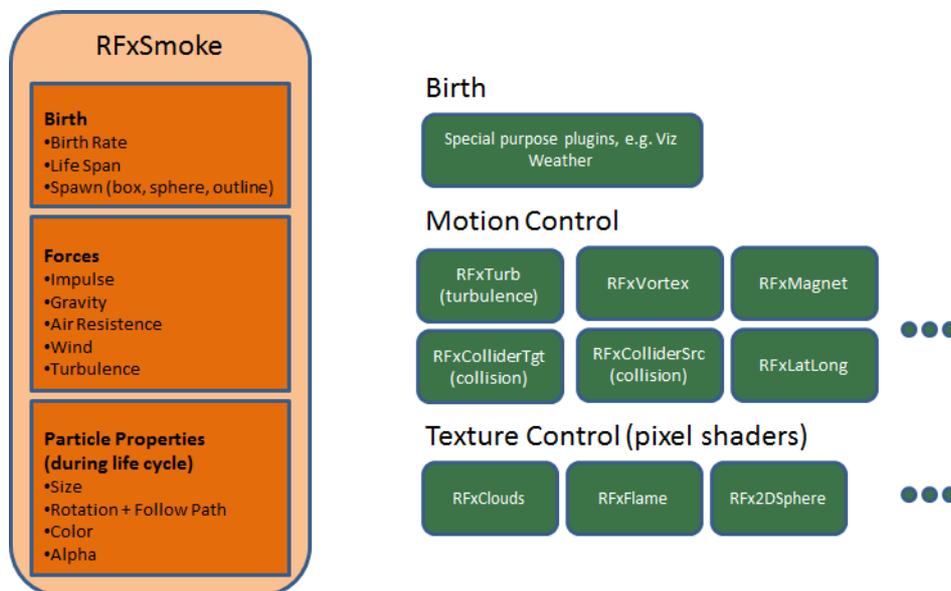
Particle systems are a computer graphics technique to simulate certain physics-based effects, which are otherwise very hard to reproduce with conventional rendering techniques. Examples of such effects which are commonly replicated using particle systems include fire, explosions, smoke, weather effects, sparks, falling leaves, dust, meteor tails, or abstract visual effects like glowing trails, magic spells, etc.

The particle effects in Viz Artist/Engine 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



This section contains information on the following topics:

- [RFX2DSphere](#)
- [RFXClouds](#)
- [RFXFlame](#)

13.7.1 RFX2DSphere



RFX2DSphere creates a 2D sphere-like look on each particle. The sphere is affected by light sources.

Adjust the following parameters as required to achieve the required effect:

- Color Mode (Material, Color, Color-Material)
- Radius
- Blend

Note: This plugin is located in: Built-ins -> Shader -> RealFX

See Also

- [RFXSmoke](#)

13.7.2 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 required effect:

- Brightness
- Turb Scale X
- Turb Scale Y
- Turb Speed

Note: This plugin is located in: Built-ins -> Shader -> RealFX

See Also

- [RFXSmoke](#)

13.7.3 RFXFlame



RFXClouds create a dynamic flame-like texture on each particle, and changes randomly for each particle over time.

Note: This plugin is located in: Built-ins -> Shader -> RealFX

See Also

- [RFxSmoke](#)

13.8 RTT Advanced Materials

The following [RTT](#) shader plugins are located in the RTTAdvancedMaterials folder:

Anisotropic Light Shader	Gooch Shader
Brushed Metal Shader	Lacquered Surfaces Shader
Bump Shader	Metal Reflection Shader
Bump Optimized Shader	Microstructure Shader
Fabric Shader	Monitor Shader
Glass Shader	Velvet Shader

[RTT](#) Advanced Materials plugins allow you to create a number of high-quality materials using state-of-the-art shader technology. Using the plugins is very easy. Simply drag the shader plugin of your choice onto selected geometry, assign a material and set up the additional parameters as shown by the shader in its own editor. Some plugins remain inactive until you have applied all necessary textures, depending on the shader plugin. This may include a basic texture on your container and/or additional textures to be dropped in the plugin interface. You can modify all parameters of the applied 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 plugins (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 container is used. It is also possible to change the alignment of the environment textures to achieve the correct 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 base texture will be used.

All Shaders support up to five light sources. By enabling more than five lights, the last activated light will be disabled. The supported light types are infinite and local lights; also the light color of these lights is supported.

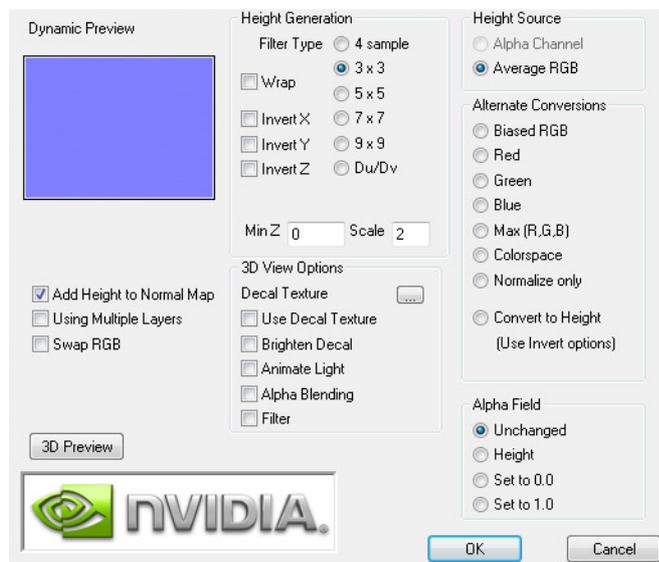
If you import the Viz Artist/Engine archive **RTT_ShaderPresets_V1** you will obtain about 60 sample scenes which are meant to give you an idea how to work with the Shaders. These presets are free to use, but in general, they are thought as a guideline for how to work with textures and the Shaders. Correctly 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 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 plugins, or use this link [NVIDIA Texture Tools for Adobe Photoshop](#).

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's Guide); yet there is not much work involved. Just have a look at the settings in the NVIDIA Photoshop Plugin Image below and start exploring them on your own.



Creating Cubic Environment maps is no secret either. Cubic Environment Maps can be rendered either with a preferred Render Engine (Mental Ray, Vray, etc.), if it should be a 3D generated environment map, or they can be created with HDR Images, which are then transformed with HDR Shop.

Note: HDR Shop has a menu for panoramic transformations, here you can create a cubic environment image by transforming it from latitude/longitude to an cubic environment by just a few mouse clicks (see instructional video).

Note: Creating Cubic Environment Maps (HDR or LDR) is a well known technique for 3D experts. This Viz Artist User Guide does not explain the creation of these Environment Maps in detail, as there are numerous tutorials on the web for this texture creation procedure.

If you do not have a license for the [RTI](#) Advanced Materials plugins, 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 plugins 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" plugin. To achieve the best results with the Shaders we strongly recommend to create your cube maps in HDR format.

Note: 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

HDR (High Dynamic Range) 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 plugins 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 show usually in some shades of blue. They can be created easily from (grayscale) height maps with plugins 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.

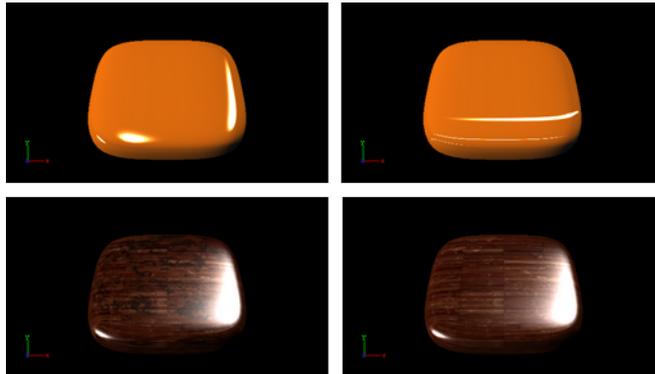
RTT

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.

13.8.1 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.



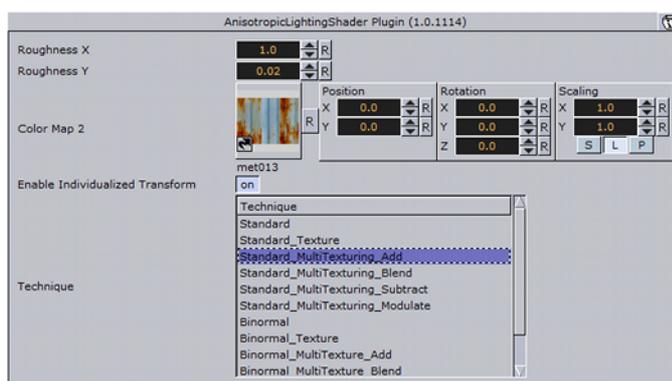
Note: All [RTT](#) shaders can be uninstalled from the Viz Artist program menu.

Note: This plugin is located in: Built-ins -> Shader -> RTTAdvancedMaterials

This section contains information on the following topics:

- [Anisotropic Light Shader Properties](#)
- [Best Practices](#)

Anisotropic Light Shader Properties



- **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:** Shows 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 texture.
 - **Standard_MultiTexture_Blend:** Is a technique to blend Color Map 2 with the texture according to their alpha values.
 - **Standard_MultiTexture_Subtract:** Is a technique to subtractively blend Color Map 2 with the texture.
 - **Standard_MultiTexture_Modulate:** Is a technique to blend Color Map 2 by multiplying it by the 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 texture.
 - **Binormal_MultiTexture_Blend:** Is a technique to blend Color Map 2 with the texture according to their alpha values.
 - **Binormal_MultiTexture_Subtract:** Is a technique to subtractively blend Color Map 2 with the texture.
 - **Binormal_MultiTexture_Modulate:** Is a technique to blend Color Map 2 by multiplying it by the texture.

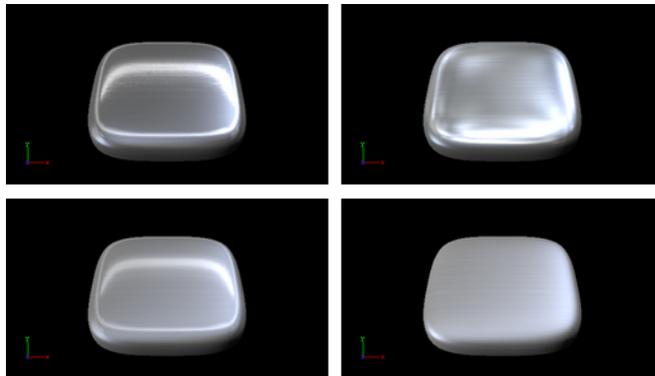
Best Practices

Apart from the parameters in the plugin container, it is necessary to assign a material. When applying a texture technique it is compulsory to assign a 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 material parameters in addition, such as ambient, diffuse, specular color and shininess.

13.8.2 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.



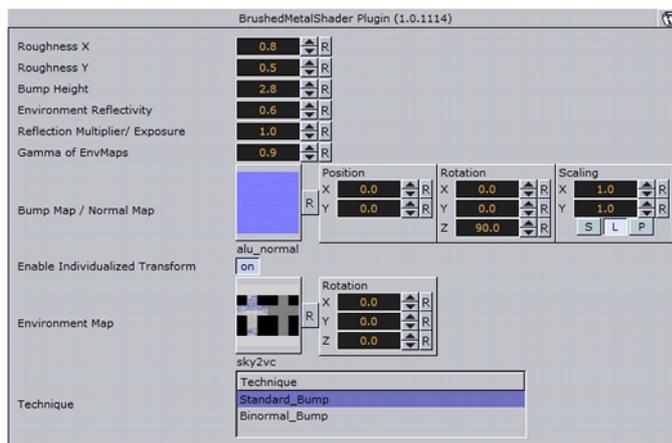
Note: All [RTT](#) shaders can be uninstalled from the Viz Artist/Engine program menu.

Note: This plugin is located in: Built-ins -> Shader -> RTTAdvancedMaterials

This section contains information on the following topics:

- [Brushed Metal Shader Properties](#)
- [Best Practices](#)

Brushed Metal Shader Properties



- **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:** Shows 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

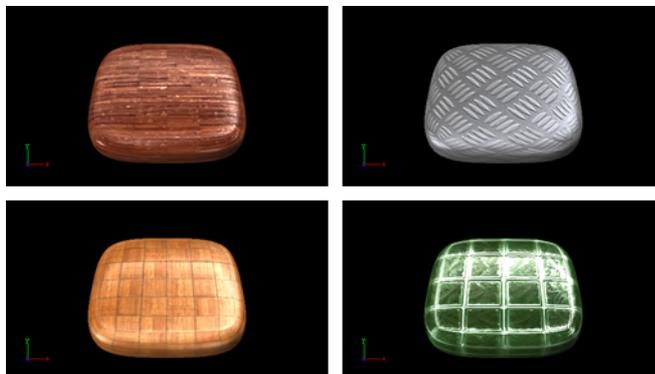
The Brushed Metal Shader should be preferred to [Anisotropic Light Shader](#) whenever you want to create highly detailed metal surfaces. However, the shader is more complex and requires more rendering time. Therefore, it may be unsuitable, depending on the complexity of the other parts of the scene and also on the graphics hardware available.

Apart from the parameters in the plugin container, it is necessary to assign a texture as a color/basic texture and a material. Do not forget to adjust material parameters in addition, such as ambient, diffuse, specular color and shininess, to achieve the required look for your surface.

13.8.3 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 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 shown and where the surface shows only diffuse lighting.



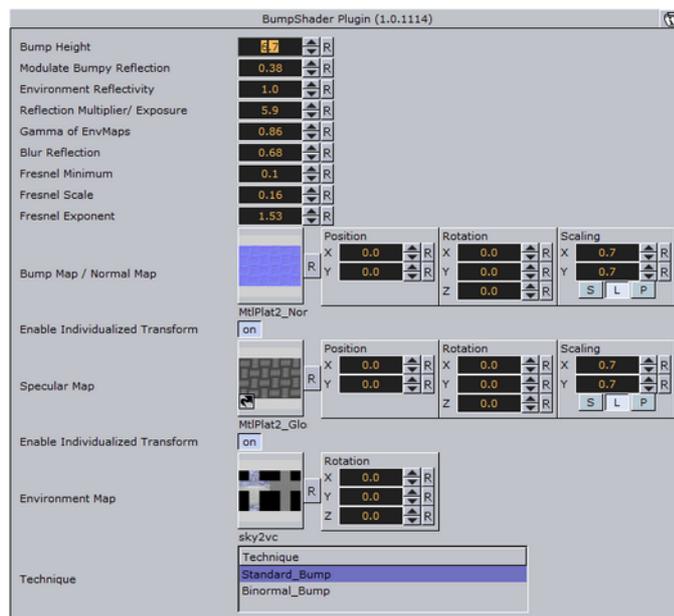
Note: All [RTT](#) shaders can be uninstalled from the Viz Artist/Engine program menu.

Note: This plugin is located in: Built-ins -> Shader -> RTTAdvancedMaterials

This section contains information on the following topics:

- [Bump Shader Properties](#)
- [Best Practices](#)

Bump Shader Properties



- **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:** Shows 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

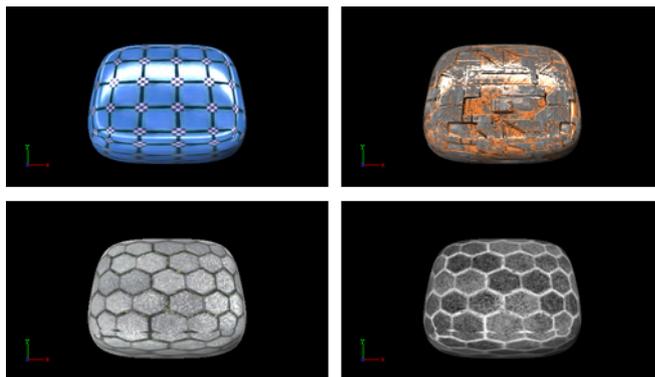
If you use this shader in complex scenes, you should favor the [Bump Optimized Shader](#). The difference between the Bump Shader and the Bump Optimized Shader is that you can modify the texture coordinates inside the Bump Shader, which, however, needs more rendering performance. If you do not have to adjust the texture coordinates individually, or animate them separately, you should use the BumpOptimizedShader to save rendering performance.

Apart from the parameters in the plugin container, it is necessary to assign a texture as a color texture and a material.

13.8.4 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 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 shown and where the surface shows only diffuse lighting.



This plugin offers the same functionality as the [Bump Shader](#) plugin, 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.

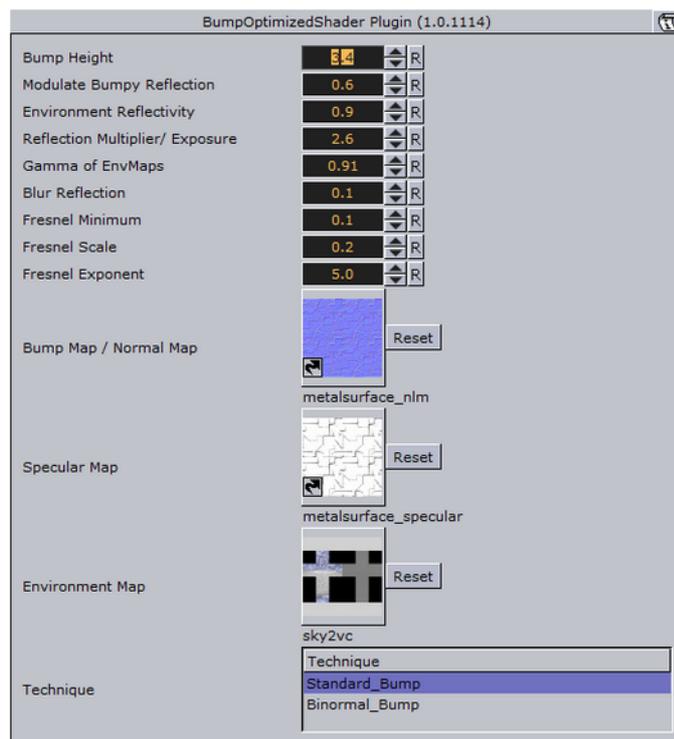
Note: All [RTT](#) shaders can be uninstalled from the Viz Artist/Engine program menu.

Note: This plugin is located in: Built-ins -> Shader -> RTTAdvancedMaterials

This section contains information on the following topics:

- [Bump Optimized Shader Properties](#)

Bump Optimized Shader Properties



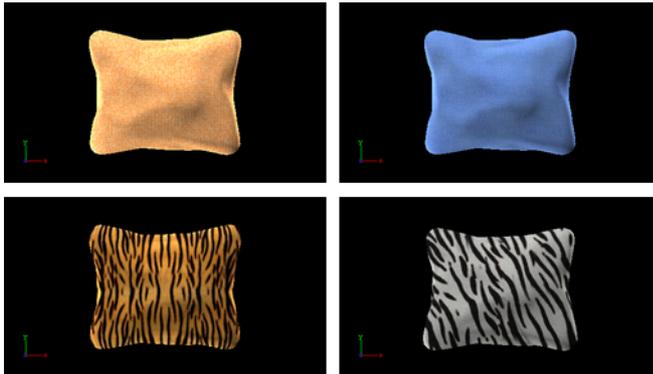
- **Bump Height:** Manipulates the height of the created bump on the surface.
- **Modulate Bumpy Reflection:** Controls whether the environment reflection is smooth or influenced by surface details. To calculate reflections on a smooth clear coat layer, value 0.0 is set by default. On the other hand, 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:** Shows 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.

13.8.5 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.



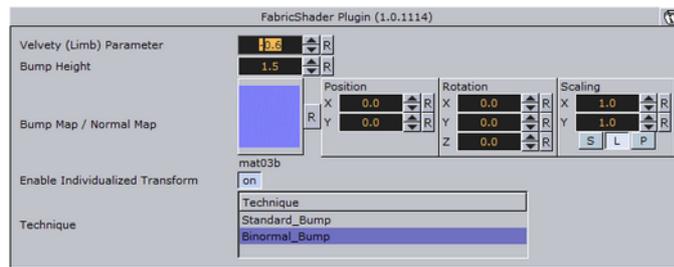
Note: All [RTT](#) shaders can be uninstalled from the Viz Artist/Engine program menu.

Note: This plugin is located in: Built-ins -> Shader -> RTTAdvancedMaterials

This section contains information on the following topics:

- [Fabric Shader Properties](#)

Fabric Shader Properties



- **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:** Shows 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.

13.8.6 Glass Shader



Apart from the parameters in the plugin container, it is necessary to assign a texture as a color/basic texture and a material. Do not forget to adjust material parameters in addition, such as ambient, diffuse, specular color and shininess, to achieve the required 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.



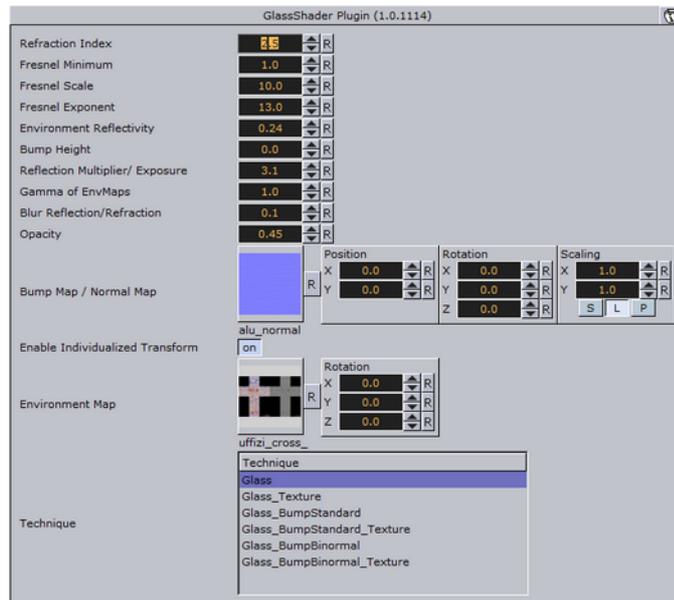
Note: All [RTT](#) shaders can be uninstalled from the Viz Artist/Engine program menu.

Note: This plugin is located in: Built-ins -> Shader -> RTTAdvancedMaterials

This section contains information on the following topics:

- [Glass Shader Properties](#)
- [Glass Shader Best Practices](#)

Glass Shader Properties



- **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:** Shows 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.

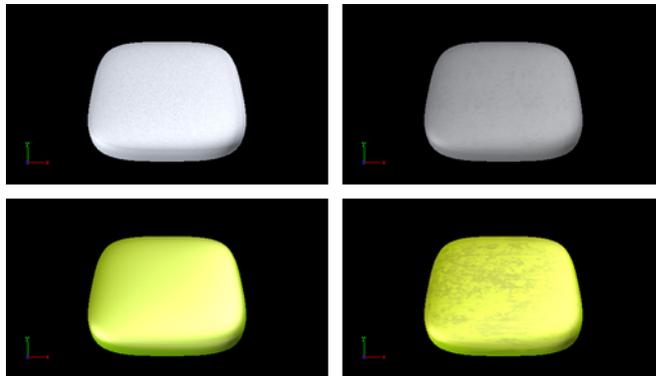
Glass Shader Best Practices

Apart from the parameters in the plugin container, it is necessary to assign a material and a texture as a color/basic texture. Do not forget to adjust material parameters in addition, such as ambient, diffuse, specular color and shininess, to achieve the required look for your surface.

13.8.7 Gooch Shader



Gooch shading is used to calculate the lighting of the surface. Accordingly, a warm color is shown 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.



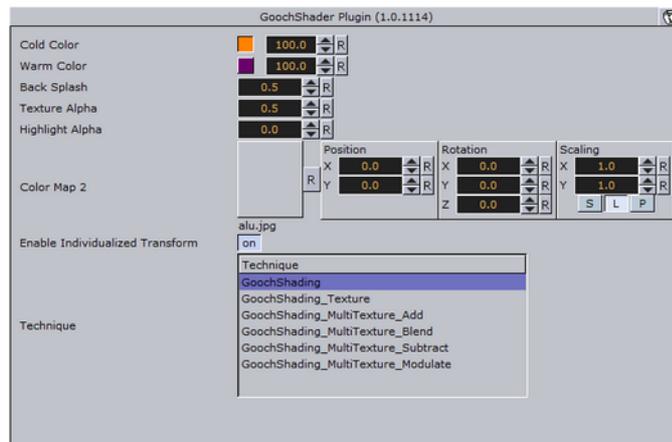
Note: All [RTT](#) shaders can be uninstalled from the Viz Artist/Engine program menu.

Note: This plugin is located in: Built-ins -> Shader -> RTTAdvancedMaterials

This section contains information on the following topics:

- [Gooch Shader Properties](#)
- [Gooch Shader Best Practices](#)

Gooch Shader Properties



- **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:** Shows 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 texture.
 - **GoochShading_MultiTexture_Blend:** Is a technique to blend Color Map 2 with the texture according to their alpha values.
 - **GoochShading_MultiTexture_Subtract:** Is a technique to subtractively blend Color Map 2 with the texture.
 - **GoochShading_MultiTexture_Modulate:** Is a technique to blend Color Map 2 by multiplying it by the texture.

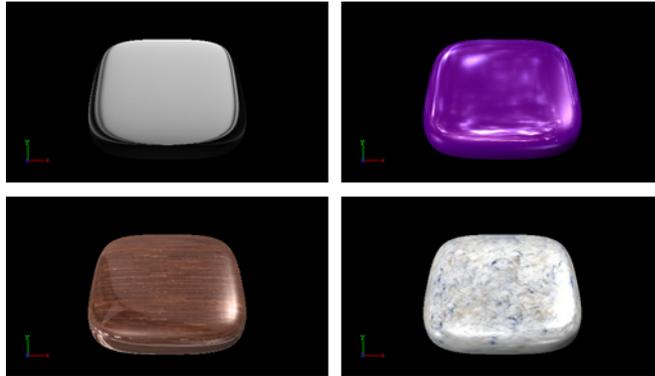
Gooch Shader Best Practices

Apart from the parameters in the plugin container, it is necessary to assign a material. Therefore, do not forget to adjust material parameters in addition, such as ambient, diffuse, specular color and shininess. When applying a texture technique it is compulsory to assign an texture as a color/basic texture. In case of multi texturing an additional, second texture must be assigned in the corresponding Shader rollout.

13.8.8 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.



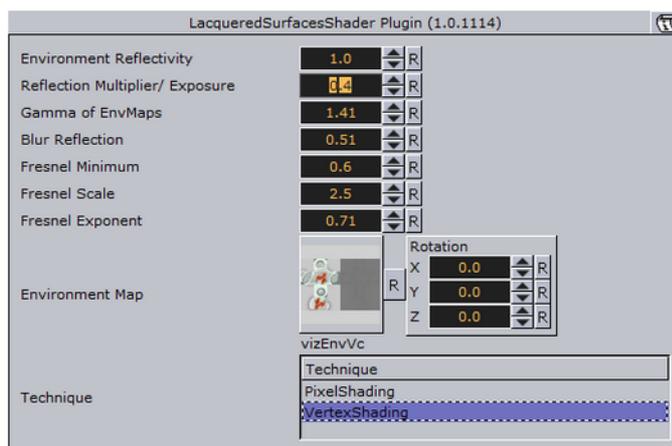
Note: All [RTT](#) shaders can be uninstalled from the Viz Artist/Engine program menu.

Note: This plugin is located in: Built-ins -> Shader -> RTTAdvancedMaterials

This section contains information on the following topics:

- [Lacquered Surfaces Shader Properties](#)
- [Lacquered Surfaces Shader Best Practices](#)

Lacquered Surfaces Shader Properties



- **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:** Shows 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.

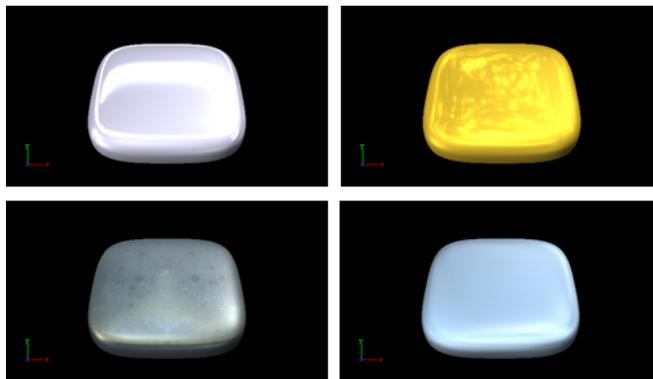
Lacquered Surfaces Shader Best Practices

Apart from the parameters in the plugin container, it is necessary to assign a material and a texture as a color/basic texture. Do not forget to adjust material parameters in addition, such as ambient, diffuse, specular color and shininess, to achieve the required look for your surface.

13.8.9 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.



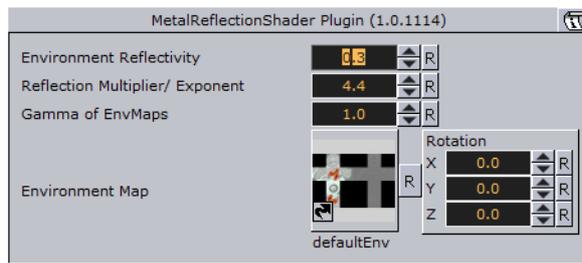
Note: All [RTT](#) shaders can be uninstalled from the Viz Artist/Engine program menu.

Note: This plugin is located in: Built-ins -> Shader -> RTTAdvancedMaterials

This section contains information on the following topics:

- [Metal Reflection Shader Properties](#)
- [Metal Reflection Shader Best Practices](#)

Metal Reflection Shader Properties



- **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.

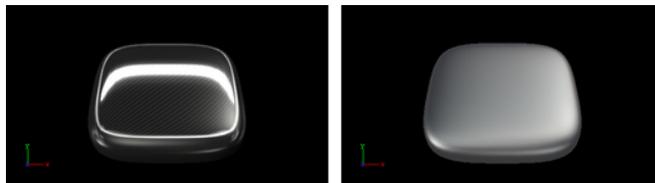
Metal Reflection Shader Best Practices

Apart from the parameters in the plugin container, it is necessary to assign a material and a texture as a color/basic texture. Do not forget to adjust material parameters in addition, such as ambient, diffuse, specular color and shininess, to achieve the required look for your surface. To achieve a photorealistic, metallic look, it is recommended to assign any color but white as a specular color.

13.8.10 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 Artist/Engine program menu.

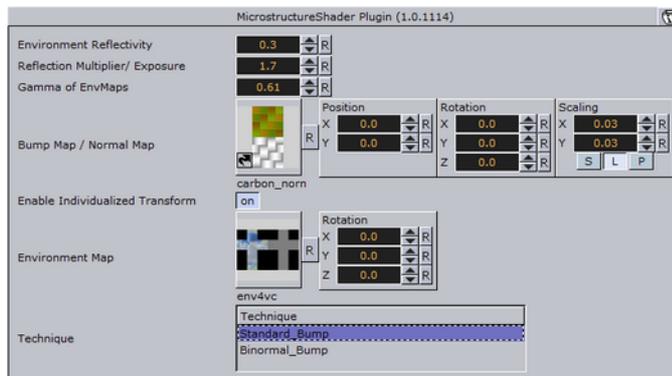
Note: This plugin is located in: Built-ins -> Shader -> RTTAdvancedMaterials

This section contains information on the following topics:

- [Microstructure Shader Properties](#)

- [Microstructure Shader Best Practices](#)

Microstructure Shader Properties



- **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:** Shows 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.

Microstructure Shader Best Practices

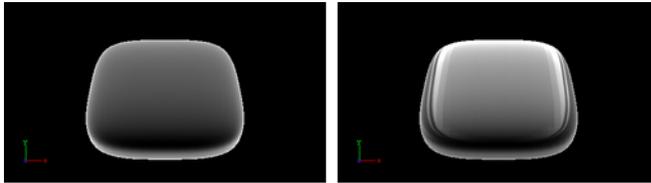
Apart from the parameters in the plugin container, it is necessary to assign a material and a texture as a color/basic texture. Do not forget to adjust also the material parameters like ambient, diffuse, specular color and shininess, to achieve the required look for your surface

13.8.11 Monitor Shader



The RTT Monitor shader simulates the appearance of a flat screen display where saturation decreases at grazing angles of view. Therefore, the shown 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.



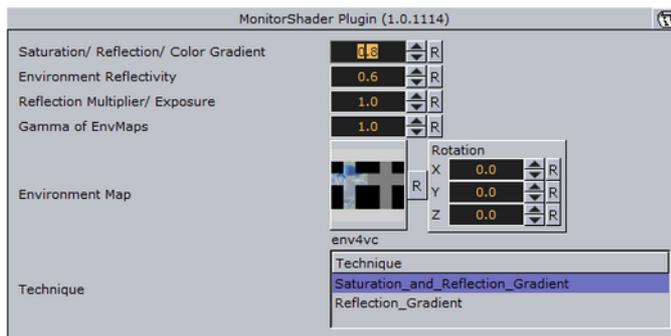
Note: All [RTT](#) shaders can be uninstalled from the Viz Artist/Engine program menu.

Note: This plugin is located in: Built-ins -> Shader -> RTTAdvancedMaterials

This section contains information on the following topics:

- [Monitor Shader Properties](#)
- [Monitor Shader Best Practices](#)

Monitor Shader Properties



- **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:** Shows 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.

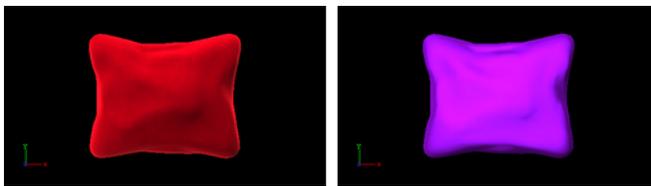
Monitor Shader Best Practices

Apart from the parameters in the plugin container, it is necessary to assign a 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 material parameters in addition, such as ambient, diffuse, specular color and shininess, to achieve the required look for your surface.

13.8.12 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.



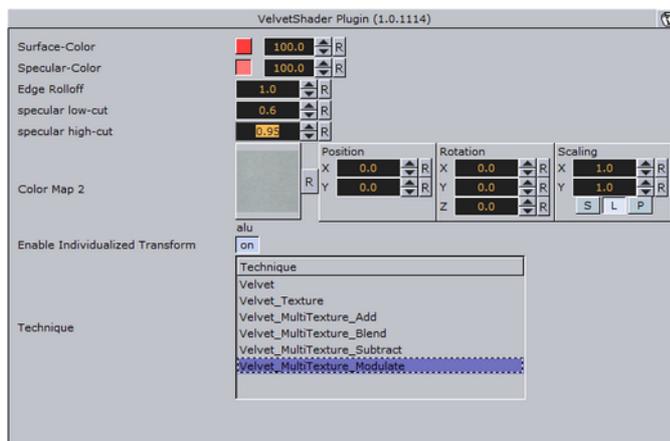
Note: All [RTT](#) shaders can be uninstalled from the Viz Artist/Engine program menu.

Note: This plugin is located in: Built-ins -> Shader -> RTTAdvancedMaterials

This section contains information on the following topics:

- [Velvet Shader Properties](#)
- [Velvet Shader Best Practices](#)

Velvet Shader Properties



- **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:** Shows 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 texture.
 - **Velvet_MultiTexture_Blend:** Is a technique to blend Color Map 2 with the texture according to their alpha values.
 - **Velvet_MultiTexture_Subtract:** Is a technique to subtractively blend Color Map 2 with the texture.
 - **Velvet_MultiTexture_Modulate:** Is a technique to blend Color Map 2 by multiplying it by the texture.

Velvet Shader Best Practices

Apart from the parameters in the plugin container, it is necessary to assign a material. Therefore, do not forget to adjust material parameters in addition, such as ambient, diffuse, specular color and shininess. When applying a texture technique it is compulsory to assign an texture as a color/basic texture. In case of multi texturing an additional, second texture must be assigned in the corresponding Shader rollout.

13.9 Texture

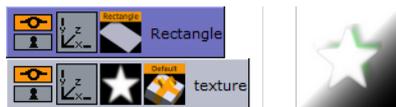
The following shader plugins are located in the Texture folder:

- [Drop Shadow](#)
- [Emboss](#)
- [MultiTexture](#)

13.9.1 Drop Shadow



Drop Shadow is a shader plugin that generates 2D shadow of a texture.

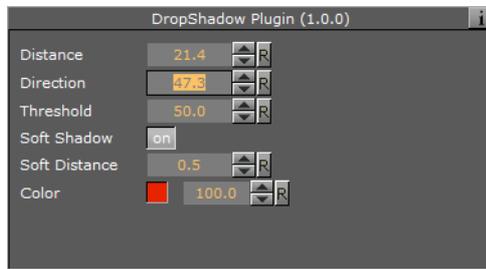


Note: This plugin is located in: Built-ins -> Shader -> Texture

This section contains information on the following topics:

- [Drop Shadow Properties](#)

Drop Shadow Properties



- **Distance:** Set the distance of the shadow.
- **Direction:** Set the direction of the shadow in degrees.
- **Threshold:** Set the shadow threshold.
- **Soft Shadow:** Enable the Soft Distance parameter for applying a soft shadow.
 - **Soft Distance:** Set the distance of the soft shadow.
- **Color:** Set the color of the shadow.

13.9.2 Emboss



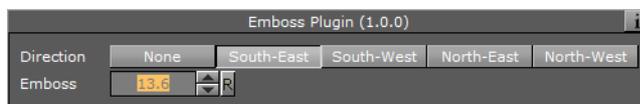
Emboss is a simple shader plugin that allows you to raise highlighted surfaces and lower shadows of textures and images creating an embossed look.

Note: This plugin is located in: Built-ins -> Shader -> Texture

This section contains information on the following topics:

- [Emboss Properties](#)

Emboss Properties



- **Direction:** Set the direction of the emboss. Options are:
 - None
 - South-East
 - South-West
 - North-East
 - North-West
- **Emboss:** Sets the Emboss level, in percentage.

13.9.3 MultiTexture



The MultiTexture shader plugin blends (linear) two textures together.

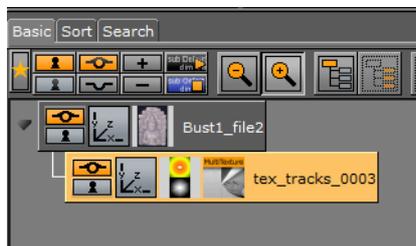
Note: This plugin is located in: Built-ins -> Shader -> Texture

This section contains information on the following topics:

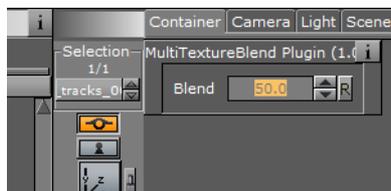
- [To Create a Simple Blended Image Scene](#)
- [Blended Images Example](#)

To Create a Simple Blended Image Scene

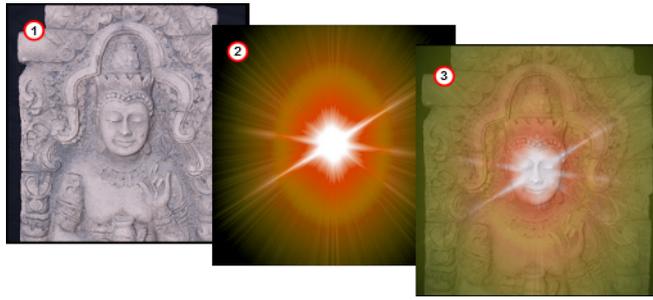
1. Create a new Container.
2. Add an image to the new Container.
3. In the Image Editor set the image to:
 - Unit: Level one
 - Inheritable
4. Create a Sub Container.
5. Add an image to the Sub Container.



6. In the Image Editor set the image to:
 - Unit: Level two
7. Add the MultiTexture plugin to the Sub Container.
8. Click on the MultiTexture plugin.
9. Adjust the image blend as required.



Blended Images Example



14 Scene Plugins

Scene plugins are global functions that have influence on the whole Scene. They are mainly used when external programs are interfaced with Viz Artist.



Scene plugins are located in the following folders:

Default	Multi-Touch Plugins
Image	Script Plugins
Libero	Texture
Lineup	Tools

Note: The default path for the Scene plugins is: <viz install folder>\plugin
 \<plugin name.vip>

See Also

- [Multi-Touch Plugins](#) in [Container Plugins](#)

14.1 Default

The following scene plugins are located in the Default folder:

- [VCF](#)

14.1.1 VCF



The VCF scene plugin works in conjunction with the [VCF Parameter](#) plugin that allows you to create a seamless interpolated transition from a virtual camera flight (VCF)

to a real camera - and conversely. This is only a relevant plugin to set up if you have purchased the virtual studio expansion components.

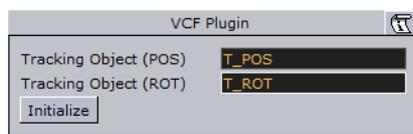
You must have a real camera with data tracking enabled which is set in remote mode in the camera editor. In case you have several tracked cameras, the virtual camera will interpolate its position to the real camera that is selected on air.

Note: This plugin is located in: Built-ins -> Scene Plugins -> Default

This section contains information on the following topics:

- [Properties](#)
- [To animate a virtual camera flight](#)

Properties



- **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 plugin 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 plugin (for example T_POS and T_ROT).
 4. Click Initialize to finish.
 5. Add the [VCF Parameter](#) plugin 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 located in the Image folder:

- [Background Clip](#)

14.2.1 Background Clip



This plugin 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 do not run out of memory on the render engine. Scene plugins 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:

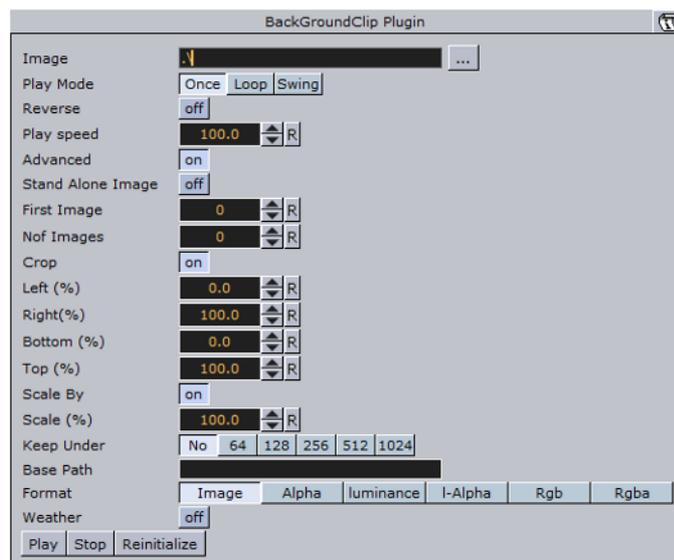
number of images * image width * image height * 3 (or 4 for alpha).

Note: This plugin is located in: Built-ins -> Scene Plugins -> Image

This section contains information on the following topics:

- [Background Clip Properties](#)
- [To animate a virtual camera flight](#)

Background Clip Properties



- **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 Scene Editor.
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

- [Video Playback](#)
- [Image Clip](#)

14.3 Libero

The following Scene Plugin is located in the Libero folder:

- [EVControl Plugin](#)

The plugins in this folder are set to inactive by default. To view the folder and use the plugins, set all plugins to active in **Plugins** (see [Viz Configuration](#) in the [Viz Engine Administrator Guide](#)).

14.3.1 EVControl Plugin



The [EVControl Plugin](#) allows an EVS video server to be controlled from the Viz Engine (like a tape deck over RS422). This gives the ability to load and control playback of EVS video server clips.

Note: This plugin is located in: Built-ins -> Scene Plugins -> Libero

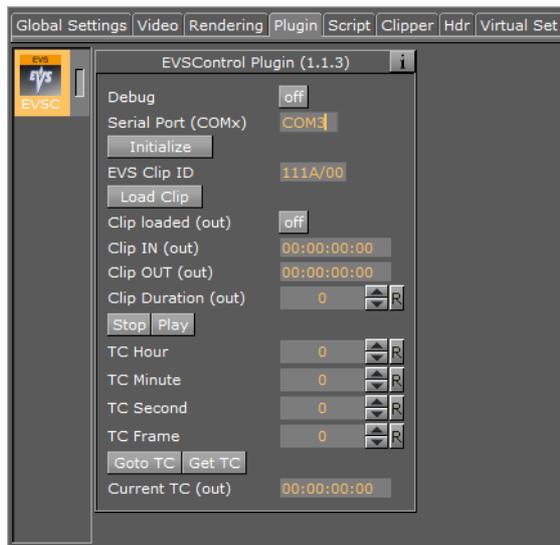
This section contains information on the following topics:

- [Properties](#)
- [Clip loaded successfully](#)
- [Clip loaded successfully](#)
- [Get the Current TimeCode](#)
- [Example Scenes](#)

14.3.2 Properties

Various EVControl plugin settings can be modified in Viz Artist from the [EVControl Plugin Editor](#).

EVSControl Plugin Editor



Note: Do not change OUT parameters manually or using a script.

- **Debug:** Toggle to enable more debug output on the console
- **Serial Port (COMx):** The serial port to be used for controlling the EVS.
- **Initialize:** Click to initialize the Controller. Without initialization, the plugin does not work.
- **EVS Clip ID:** The ID of the EVS clip to be loaded. The value must match the format "PBCK/MM" where:
 - P = Page number (1-9)
 - B = Bank number (1-9)
 - C = Clip number (1-9)
 - K = Camera letter (A-F)
 - MM = EVS Machine number (00, 01, ..), where 00 is the local EVS.
- **Load Clip:** Click to load the clip shown in the "EVS Clip ID" field
- **Clip loaded:** Shows when the Load Clip was clicked to confirm the clip was loaded.
- **Clip IN:** The IN time code of the successfully loaded clip (Format hh:mm:ss:ff)
- **Clip OUT:** The OUT time code of the successfully loaded clip (Format hh:mm:ss:ff)
- **Clip Duration:** The duration of the successfully loaded clip in milliseconds
- **Stop:** Stops play back
- **Play:** Starts play back
- **TC Hour:** The time code hour value for the "Goto TC" action
- **TC Minute:** The time code minute value for the "Goto TC" action
- **TC Second:** The time code second value for the "Goto TC" action
- **TC Field:** The time code frame value for the "Goto TC" action
- **Goto TC:** Click to go to the time code shown by the above fields
- **Get TC:** Get the current timecode
- **Current TC:** The value of the "Get TC" action

Clip loaded successfully

If a clip has been loaded successfully the EVSControl Plugin Editor will show:

- the 'Clip loaded (out)' flag is set.
- the 'Clip IN (out)' shows the Clip IN timecode.
- the 'Clip OUT (out)' shows the Clip OUT timecode.
- the 'Clip Duration (out)' shows the Clip length in milliseconds.
- EVS goes to IN timecode of clip.

Go to a Specific TimeCode

1. Enter the hour, minute, second, and frame in the fields:

- TC Hour
- TC Minute
- TC Second
- TC Field

2. Click on the **Goto TC** button

Get the Current TimeCode

1. Click on the **Get TC** button
2. The current TimeCode is shown in the Current TC field.

14.3.3 Example Scenes

The plugin comes with two example scenes:

- One shows a slider to scrub a clip
- The other shows scanning for clips and creating clip buttons for playback control

14.4 Lineup

The following scene plugins are located in the Lineup folder:

- [Tree Status](#)

14.4.1 Tree Status



The Tree Status plugin is needed by the Lineup template in Viz Pilot to collect information about the scene hierarchy.

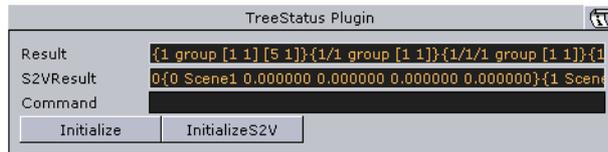
Consider that the Tree Props plugin is required for using the Tree Status plugin. Move the Tree Props plugin onto the group which is holding the required transformation to control. Viz Pilot is then able to build its own internal tree properties list.

Note: This plugin is located in: Built-ins -> Scene Plugins -> Lineup

This section also contains information on the following topics:

- [Properties](#)

Properties



- **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 Studio.
- **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.

14.5 Multi-Touch Plugins

The following Scene plugins are located in the MultiTouchComp folder:

- [MtSensor Plugin](#)

All the [Multi-Touch Plugins](#) Container Plugins require the [MtSensor Plugin](#) to be available. If a [Multi-Touch Plugins](#) Container Plugin is added to a Scene it will automatically add the [MtSensor Plugin](#) to the Scene if it is not there.

The plugins in this folder are set to inactive by default. To view the folder and use the plugins, set all plugins to active in **Plugins** (see [Viz Configuration](#) in the [Viz Engine Administrator Guide](#)).

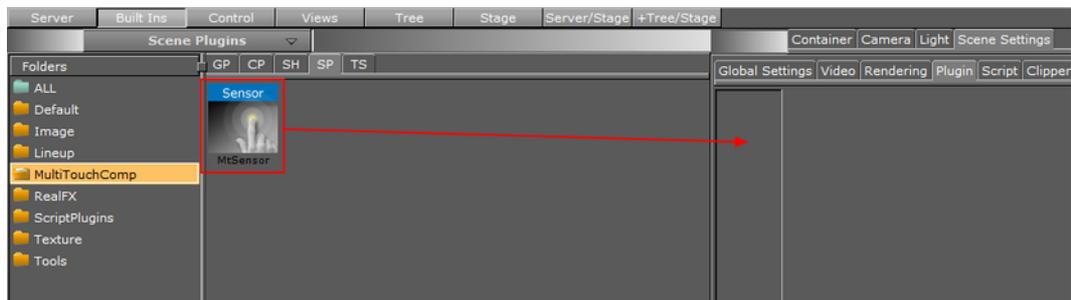
14.5.1 MtSensor Plugin



The MtSensor Plugin manages global telestration and plugin settings.

The MtSensor Plugin is the scene Plugin that must be available for any Multi-Touch feature to work. It is automatically added to the Scene if any of the Container or Geometry Plugins that require it are inserted in the Scene. It can also be dragged from

the Server area under *Built Ins* -> *SP* -> *MultiTouchComp*, to the Plugin tab under Scene Settings.



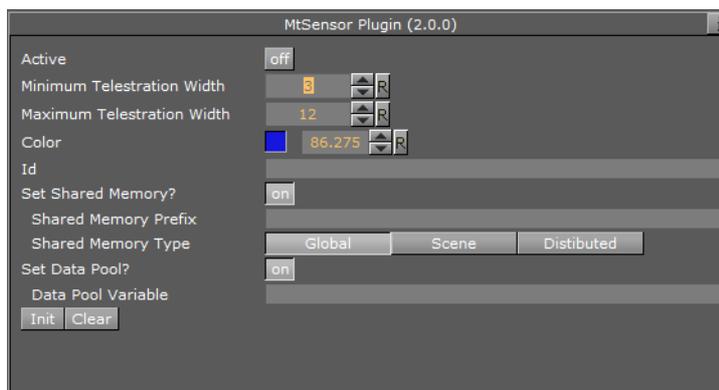
The MtSensor Plugin also provides a screen-wide telestration facility. It can be controller through Plugin parameters and buttons that are available for scripting.

Note: This plugin is located in: Built-ins -> Scene Plugins -> MultiTouchComp

This section contains information on the following topics:

- [MtSensor Configuration](#)
- [Events](#)

MtSensor Configuration



- **Active:** Select On (enable) or Off (disable) telestration input from user
- **Minimum Telestration Width:** Minimum brush width to assign based on pressure applied to sensor
- **Maximum Telestration Width:** Maximum brush width to assign based on pressure applied to sensor
- **Color:** The color to assign for subsequent telestration drawing
- **Id:** To give additional context in the handler script, specify a string that will identify any notifications dispatched by this Plugin. This is often included as an argument for the event so a common script may handle events from a number of Plugins.
- **Set Shared Memory:** If ON enables shared memory to be updated for the Plugin notifications.
- **Shared Memory Prefix:** This sets a 'prefix name' to be prepended to the shared memory variables maintained by the Plugins notifications. For Plugins that maintain multiple fields each field name has the prefix prepended to it followed by a dot

`.` so as to mimic member access to an object, i.e: if the prefix is 'Obj' the fields 'field1' and 'field2' would be identified with the strings 'Obj.field1' and 'Obj.field2'. The shared memory field 'Obj' is also maintained and is simply an integer that is modified every time any of its 'subfields' is updated.

- **Shared Memory Type:** Either Global, Scene, or Distributed. This selects the shared memory area to update.
- **SetDataPool:** Shows 'you wish' Plugin notifications to set a DataPool variable
- **DataPoolVariable:** Shows the name of the DataPool variable one wishes to have set.
- **Init:** Clear current telestration and make Telestration active
- **Clear:** Clear the contents of the attached telestration

See Also

- [Events](#)

Events

The MtSensor Plugin does not generate any events for the scripting engine. The scene-wide telestration feature emits the same events that the [MtTelestrator Plugin](#) emits.

See Also

- [MtSensor Configuration](#)

14.6 Script Plugins

The Script Plugins is an empty folder to save created Script plugins.

A script can be saved as its own script plugin, and used in future Scenes (see [Create Script-based Plugins](#)). To save a Script Plugin, drag a compiled script into the Script Plugins folder.

Note: Script plugins are saved to *<viz data folder> \ScriptPlugins*.

See Also

- [Script Editor](#)

14.7 Texture

The following scene plugins are located in the Texture folder:

- [Graffiti](#)

14.7.1 Graffiti



The Graffiti scene plugin 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 plugin can also recognize some rendered shapes and replace the hand-drawn elements with the recognized shapes (e.g. circle, ellipse, cross, arrow).

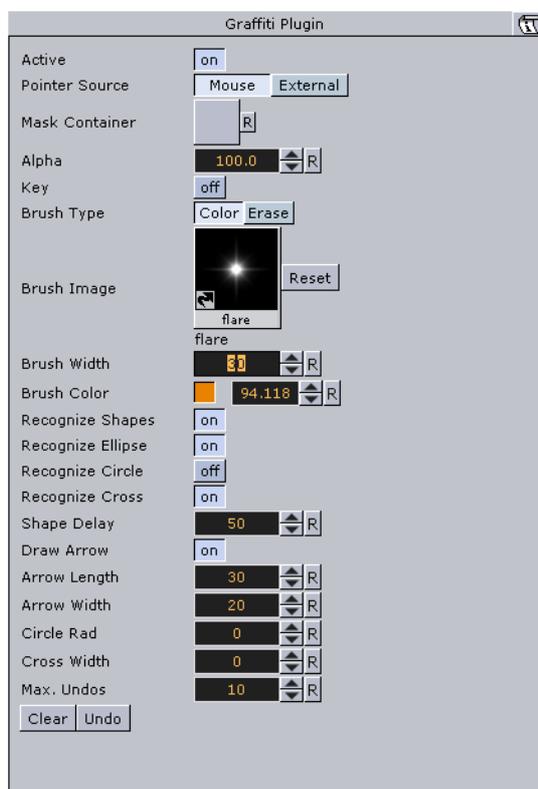
The graffiti scene plugin is not intended for use on video output. To use graffiti textures on video output, use the [GraffitiTex](#) container plugin.

Note: This plugin is located in: Built-ins -> Scene Plugins -> Texture

This section contains information on the following topics:

- [Properties](#)
- [To Create a Scene Level Graffiti Effect](#)

Properties



- **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 plugin to the [Plugin](#) panel under [Scene Settings](#), set the plugin [Properties](#), set Viz Artist in [On Air Mode](#) and start drawing.

See Also

- [Plugin](#)
- [GraffitiTex](#) (container plugin)

14.8 Tools

The following scene plugins are located in the Tools folder:

- [Level Of Detail \(LOD\) Manager](#)

14.8.1 Level Of Detail (LOD) Manager



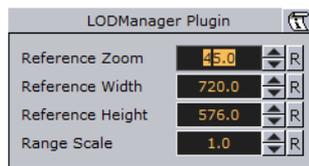
The LOD Manager scene plugin works in conjunction with the [Level Of Detail \(LOD\)](#) container plugin which is rendering objects with different details depending on the camera range; however, an integration with the LOD Manager is not required as the LOD plugin is capable of initializing the parameter values by itself. The LOD Manager plugin is used to change the visual view of the complete scene during rendering depending on the camera and scale settings.

Note: This plugin is located in: Built-ins -> Scene Plugins -> Tools

This section contains information on the following topics:

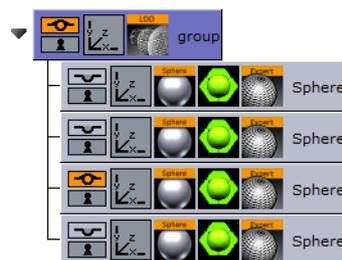
- [Level Of Detail \(LOD\) Manager Properties](#)

Level Of Detail (LOD) Manager Properties



- **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 plugins in the scene.
- **Reference Width:** Not used.
- **Reference Height:** Not used.

To Use the LOD Manager Plugin



1. Start by adding the LOD Manager scene plugin 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.

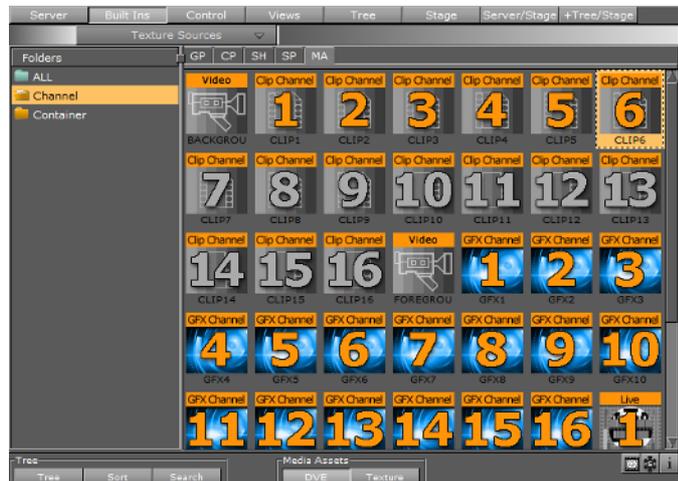
5. Open the Sphere editor for the all sub containers and set different Tessellation values for each to change the number of triangles and therefore the level of detail of an object.

Note: All spheres should, for the sake of example, be placed in the same position.

6. Add the [Level Of Detail \(LOD\)](#) plugin to the group container.
7. Open the LOD editor and change the Range parameter values provided by the LOD plugin. 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 Media Assets

Media Assets are used to add input sources to Scenes as video clips, live video, as a video stream or as graphics channels. The workflow and each type of Media Asset are described in this section.



The Media Assets are located in the Built Ins panel (see [Main Menu Left](#)), and are in two folders (Channel and Container folders).

This section contains the following topics:

- [Media Asset Workflow](#)
- [Channel Folder Media Assets](#)
- [Container Folder Media Assets](#)

See Also

- [Video Playback](#)
- [Video Clip Playback Considerations](#)
- [Media Asset Properties](#)
- [Media Assets \[Manager\]](#)

15.1 Media Asset Workflow

Media Assets can be added to a Scene as [DVE](#) or [Texture](#) through the Media Assets manager.



This section contains the following topics:

- [Media Assets as DVE](#)
- [Media Assets as Texture](#)
- [Media Assets as a Background or Foreground](#)

See Also

- [Channel Folder Media Assets](#) Media Assets
- [Container Folder Media Assets](#) Media Assets
- [Video Playback](#)

15.1.1 Media Assets as DVE

To add a Media Asset as [DVE](#) it must be placed into the Media Asset Manger in the Scene Tree panel.

Tip: If a Scene is dragged to the Media Asset Manager a [GFX Channel Media Asset](#) is automatically added as DVE.



This section contains the following topics:

- [To Add a Media Asset as DVE](#)
- [To Set a DVE Media Asset to Texture](#)
- [Order DVE Media Assets](#)

To Add a Media Asset as DVE

- Right click on a **Media Asset**, and select **Add <Media Asset> as DVE**.
 - The **DVE** button in the Media Asset manger is automatically selected
 - The selected Media Asset is added to the DVE view



Or:

- With DVE view selected, drag the Media Asset to the DVE panel
 - The selected Media Asset is added to the DVE view

Tip: A **Media Asset** can be set to active or inactive. Toggle the active/inactive button (1) or use the **Media Asset** context menu. If Media Assets are multi selected use the context menu to set active or inactive (2).



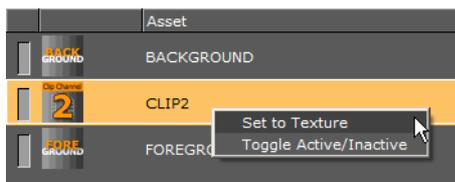
Tip: Click on the **Media Asset**, to open its properties panel (see [Media Asset Properties](#)).

Tip: Click the  icon (top right of the Scene Tree panel) to view Media Asset properties at any time (see [Media Asset Properties](#)).

To Set a DVE Media Asset to Texture

At any time, right click on the Media Asset to set it to Texture. The selected Media Asset will then be set as a texture and will show the in the Media Asset Manager Texture panel.

Tip: Media Assets can also be multi selected and set to Texture.



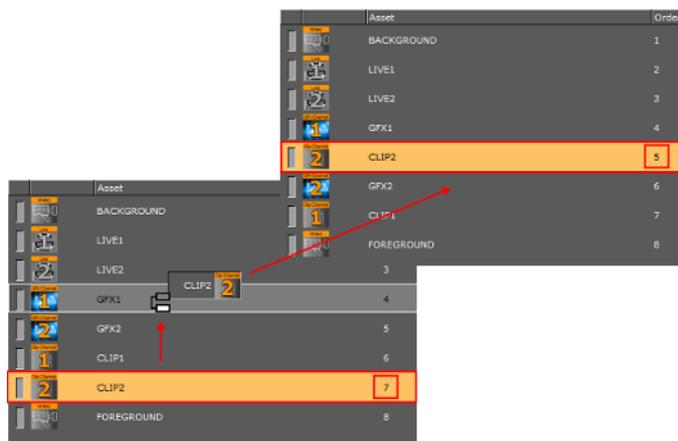
Order DVE Media Assets

Each Media Asset, added as DVE, can be sorted in a render order, as required.

Note: If the **Video Input Layer Targets and Priority** settings are made **Active** in **Scene Default Values**, each active Media Asset will be added to either the DVE or Texture section of the Media Asset Manager. They will be set in order of priority as set in **Scene Default Values** (see [Viz Configuration](#) in the [Viz Engine Administrator Guide](#)).

The Background and Foreground Media Assets are, by default, added as DVE layers. The Background Media Asset is always the first Media Asset to be rendered. The Foreground Media Asset can be ordered the same as other Media Assets (see [Background and Foreground Media Assets](#)).

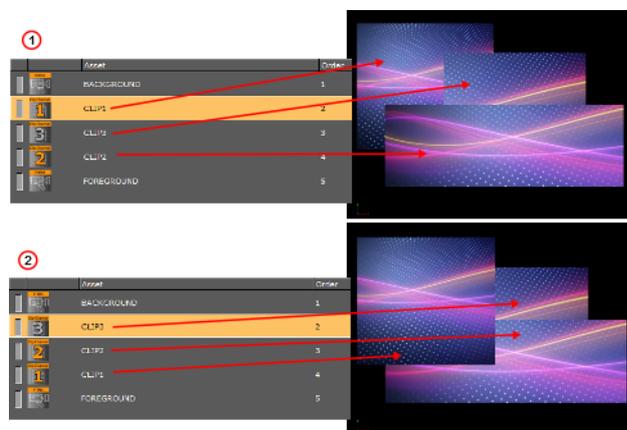
To sort a render order, drag a Media Asset to a new position.



The order of each DVE Media Asset determine the order each Media Asset is rendered. The first Media Asset in the list is rendered first, the second is rendered next, and so on. See the image below:

Example 1: the Clip order is 1, 2 and 3. So Clip 1 is rendered first and Clip 3 is rendered last

Example 2: the Clip order is 3, 2 and 1. So Clip 3 is rendered first and Clip 1 is rendered last.



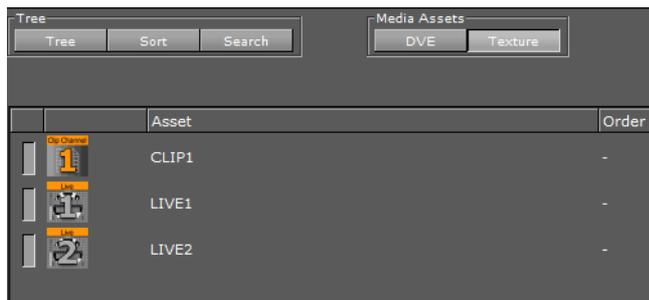
See Also

- [Video Playback](#)
- [Media Assets \[Manager\]](#)

15.1.2 Media Assets as Texture

To add a Media Asset as a [Texture](#) it must be placed into the Media Asset Manger in the [Scene Tree](#).

Tip: If a Scene is dragged to the Media Asset Manager a [GFX Channel Media Asset](#) is automatically added as a Texture.



This section contains the following topics:

- [To Add a Media Asset as Texture](#)
- [To Set a Texture Media to DVE](#)
- [To View the Media Asset Container](#)
- [Order Texture Media Assets](#)

To Add a Media Asset as Texture

- Right click on a **Media Asset**, and select **Add <Media Asset> as Texture**:
 - The **Texture** button in the Media Asset manager is automatically selected
 - The selected Media Asset is added to the Scene Tree and a new Container is created (click **Tree** to view)



Or:

- Drag the Media Asset to a Container or the Scene Tree:
 - If a Media Asset is dragged to a Container in the Scene Tree, or it is dragged to the Scene Tree (which creates a Container), then the Media Asset is automatically created as a Media Asset Texture

Or:

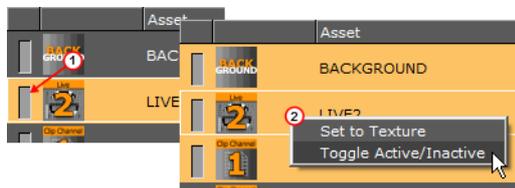
- Drag the Media Asset to the Media Asset Manager:

- With **Texture** selected in the Media Asset Manager, drag the Media Asset to the Texture panel
- If a Media Asset is dragged directly to the Media Asset Manager, a Container is not created automatically. To use the Media Asset in a Scene make sure that the same Media Asset is added to a Container in the Scene Tree

Or:

- Drag a Media Asset to the Background or Foreground drop zone in [Global Settings](#).
 - If a Media Asset is dragged to the Background or Foreground drop zones it is applied in the Scene as a full screen texture

Tip: A **Media Asset** can be set to active or inactive. Toggle the active/inactive button (1) or use the **Media Asset** context menu. If Media Assets are multi selected, use the context menu to set active or inactive (2).



Tip: Click on a **Media Asset**, in the Media Asset Manager, to open its properties panel (see [Media Asset Properties](#)).

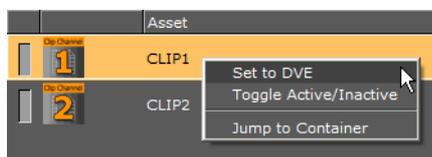
Tip: <Right-click> the **Media Asset** icon in a Container. Select **Show settings of <Media Asset>** (see [Media Asset Properties](#)).

Tip: Click the  icon (top right in the Scene Tree panel) to view Media Asset properties at any time.

To Set a Texture Media to DVE

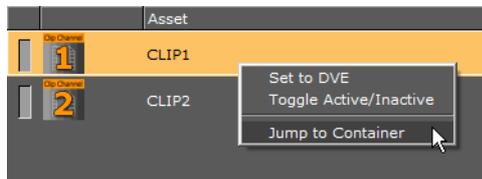
At any time, right click on the Media Asset to set it to DVE. The selected Media Asset will then be set to DVE and will show the Media Asset Manager DVE panel.

Tip: Media Assets can also be multi selected and set as DVE.



To View the Media Asset Container

At any time, right click on the Media Asset and select **Jump to Container** to view the **first** Container in the Scene Tree, which holds the selected Media Asset (if a Container was created for the Media Asset (see [To Add a Media Asset as Texture](#))).



Order Texture Media Assets

Each Media Asset, added as Texture, will be ordered only by its addition to a Container in the Scene Tree.

If the **Video Input Layer Targets and Priority** settings are made active in the **Scene Default Values**, each active Media Asset will be added either the DVE or Texture view of the Scene Tree. Set in order of priority as set in **Scene Default Values** (see [Viz Configuration](#) in the [Viz Engine Administrator Guide](#)).

A Texture added to the Texture view is not added to a Container. To use a Media Asset as a Texture it must be also added to a container as well.

See Also

- [Video Playback](#)
- [Media Assets \[Manager\]](#)

15.1.3 Media Assets as a Background or Foreground

This section details how to add a Media Asset as a background or foreground in a Scene.

Note: If the foreground Media Asset has alpha, the Scene can be shown through the Media Asset.

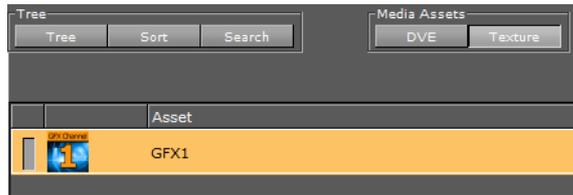
Note: The Foreground Media Asset can be reordered (see [Order DVE Media Assets](#)).

To Add a Media Asset as a Background or Foreground

1. Click *Scene Settings* -> *Global Settings*.
2. Drag a Media Asset to the **Background** or **Foreground** drop zone.
 - The Media Asset is added to the Media Asset Manager as a Texture

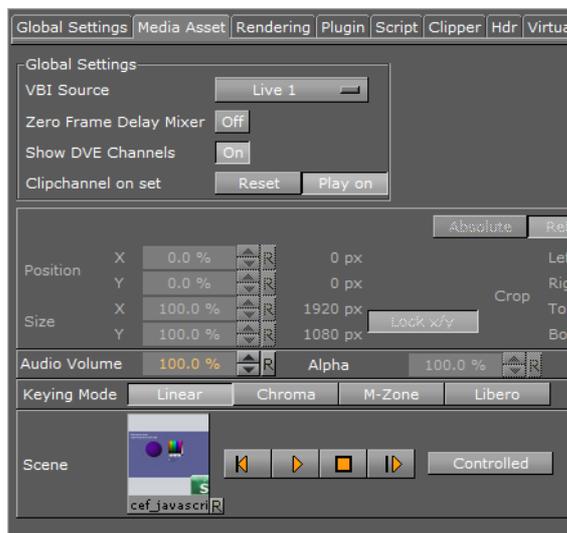


- Click on the Media Asset in the Media Asset Manager.



- In the Media Asset properties panel configure the Media Asset as required:

- [Clip Properties](#)
- [Live Properties](#)
- [Stream Properties](#)
- [GFX Properties](#)

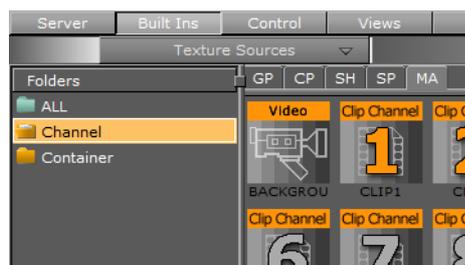


See Also

- [Background and Foreground Media Assets](#)
- [Media Asset Properties](#)
- [Media Asset Workflow](#)
- [Video Playback](#)

15.2 Channel Folder Media Assets

The following Media Assets are located in the Channel folder of the Built-ins panel:



- [Background and Foreground Media Assets](#)
- [Clip Channel Media Asset](#)
- [GFX Channel Media Asset](#)
- [Live Video Media Asset](#)
- [Video Stream Media Asset](#)

For the available Media Asset properties see [Media Asset Properties](#).

See Also

- [Media Asset Workflow](#)
- [Container Folder Media Assets](#)

Background and Foreground Media Assets



The Background and Foreground Media Assets represent the rendered background and foreground graphics on the DVE Compositor.

Note: By default a Background and Foreground Media Asset is applied to each new Scene.



The Foreground Media Asset can be ordered as required (see [Order DVE Media Assets](#)), but the Background Media Asset is always the first in the render order.

Both the Background and Foreground Media Assets can be deleted. If they have been deleted, drag the Background and, or the Foreground Media Asset icon, from the Server Area, to the Media Asset manager, if required.

Note: Only one Background and one Foreground Media Asset can be used in the DVE Media Asset Manager at a time.

See Also

- [Media Assets as a Background or Foreground](#)

Clip Channel Media Asset



Clip Channels represent clip players. The clip players are either hardware bound native players (Matrox) or a DirectShow based software player. Clips can be used either as [DVE](#) or [Texture](#) effect (see [Clip Properties](#)).

Note: **DVE** is not available with DirectShow based software players.

Note: With the Clip Channel Media Asset it is possible to play images as DVE or Texture with the Matrox clip player. Supported image containers are jpg and png. The images have to be available on a physical disk for this feature to work. One advantage of an image played as a clip, is that the same features available for a video clip can be used on the image.

GFX Channel Media Asset



Graphics (GFX) Channels allow a designer to compose several independent Scenes, either as a Texture or DVE effect, into another Scene.

With a GFX Channel a Scene, with animation, can be added to the current Scene as a [DVE](#) or [Texture](#) (see also [GFX Properties](#)).

Note: GFX channels can only be used in the main layer.

Live Video Media Asset



Live Media Assets integrate live video sources either as [DVE](#) or [Texture](#) Effect (see [Live Properties](#)).

Video Stream Media Asset



Video Streams represent IP-based live Video streams (IP Engine). They can be used either as [DVE](#) or [Texture](#) Effect (see also [Stream Properties](#)).

Stream Media Assets visualize a video being streamed into the engine.

See Also

- [Video Playback](#)
- [Media Asset Properties](#)
- [Media Assets](#)
- Media Assets [Container Folder Media Assets](#)

15.3 Container Folder Media Assets

The following Media Assets are located in the Container folder of the Built-ins panel:



- [Dynamic Scene Media Asset](#)
- [Dynamic Texture Media Asset](#)
- [Viewport Tile Media Asset](#)

See Also

- [Media Asset Workflow](#)
- [Channel Folder Media Assets](#)

15.3.1 Dynamic Scene Media Asset



The Dynamic Scene is a Media Asset that is able to render a different scene, local scene or parts of a local scene into a texture and show the texture in the current scene being rendered.

Showing a different scene as 'a window' in the current scene can be used in a virtual studio 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 show dynamic images in camera one's view. This way you can show more than one camera at the same time.

For visual effects, the whole scene or parts of it can be grabbed into an image, and have shaders applied to the whole scene or parts of it.

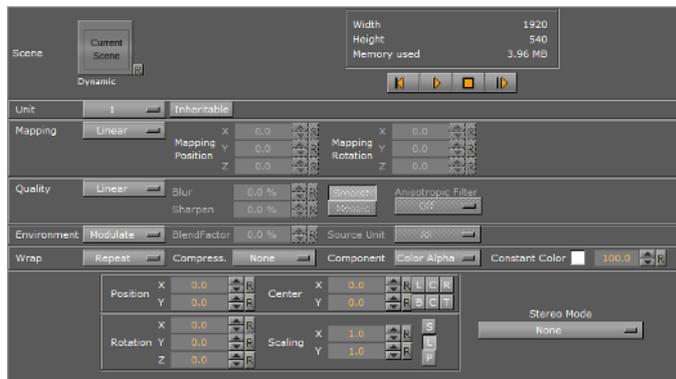
To transition from one scene (A) to another (B) both scenes are rendered into dynamic images and a third scene is used to hold both dynamic images to create a transition between them (see [Create Transition Effects](#)).

Note: This Media Asset is located in: Built-ins -> Media Assets -> Container

This section contains information on the following topics:

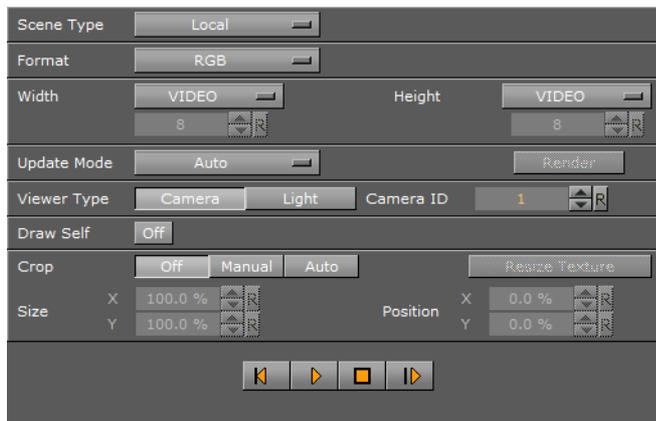
- [Dynamic Scene Texture Properties](#)
- [Dynamic Scene Dynamic Properties](#)

Dynamic Scene Texture Properties



- **Scene:** Placeholder for the rendered scene. If it is empty, the current scene will be rendered.

Dynamic Scene Dynamic Properties



- **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 show 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](#)
- [Create Transition Effects](#)

15.3.2 Dynamic Texture Media Asset



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 pixilated as you zoom in or out of it.

Note: This Media Asset is located in: Built-ins -> Media Assets -> Container

See Also

- [Texture Editor](#)
- [GFX Channel Media Asset](#)
- [Image FX](#) and Noise options for the [Trigger](#) tab.
- Pixel FX plug-ins (separate plugin installer)

15.3.3 Viewport Tile Media Asset



Viewport Tile renders a scene into a part of the current viewport without using a texture like [GFX Channel Media Asset](#). The benefit is that the rendering can be done faster but the disadvantage is that the Viewport Tile is always orthogonal to the camera.

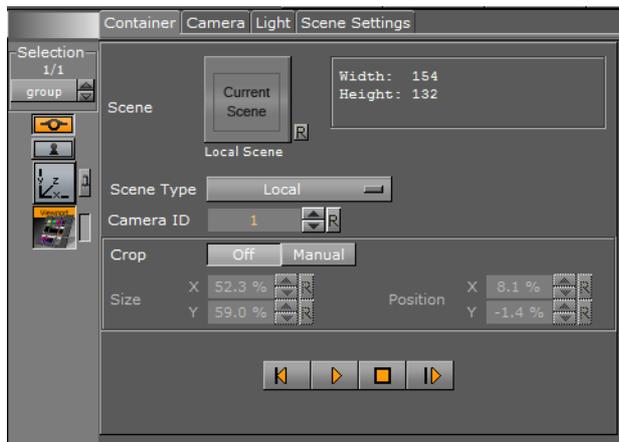
Viewport tiles can also be interactive (use the regular mouse and multi-touch events).

Note: This Media Asset is located in: Built-ins -> Media Assets -> Container

This section contains information on the following topics:

- [Viewport Tile Properties](#)
- [To Render a Scene in a Viewport Tile](#)

Viewport Tile Properties



- **Scene:** Placeholder for the rendered scene.
- **Scene Type:**
 - **Local:** Renders the current scene.
 - **Other:** Renders the Scene put in the Scene drop zone (drag and drop a scene).
 - **Main Layer:** Renders the scene of the main layer.
- **Camera ID:** Select a camera for the Viewport Tile scene.
- **Crop:**
 - **Off:** No cropping.
 - **Manual:** Define the new viewport **Size** and **Position** with the parameters X and Y.
- **Play controls:** 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** to the Scene Tree.
2. Open the Viewport Tile editor and add a Scene to the **Scene** placeholder.
3. Open the **Transformation Editor**
4. Set the **Screen Size (width and height)** of the Viewport Tile.
5. Open the Viewport Tile editor
6. Click the **Play** button.

16.1 Control Buttons

This section contains information on the Controls Buttons, which include the Play and the On Air buttons.



-  **Clapper Board:** Show or hide the [Director Control Panel](#) window
-  **Back:** Jump to the beginning of the animation in the scene
-  **Play:** Start the animation of the scene
-  **Stop:** Stop the animation of the scene
-  **Continue:** Continue the animation after it stopped at a stop point (see [Working with Events in a Director](#))
-  **Render Preview:** Show or hide the VGA Preview window (only available on machines with video cards).

Note: Preview is always enabled on Viz Artist machines without a video card.

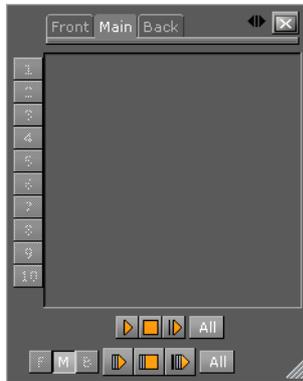
-  **Performance Editor:** Show or hide the [Performance Editor](#) .
-  **System Information:** Show or hide the [On Air Information](#) window.
-  **Lens File Editor:** Show or hide the **Lens File Calibration Editor**. Use to adjust the lens files for virtual studios. Adjust the field of view, lens deformation, mobile point and centership. It is useful in combination with lens calibration.

See Also

- [Director Control Panel](#)
- [Viz Artist Performance](#)
- [On Air Information](#)
- [License Information](#)

16.2 Director Control Panel

The Director Control Panel window can be used to select and animate one, multiple or all directors in the front, main or back layer. In addition it can be used to set slots and to animate a combination of director(s).



Click the  button, when in On Air mode, to open the Director Control Panel window.

See [Director Control Panel](#) for a detailed description of the Director Control Panel.

See Also

- [Viz Artist Performance](#)
- [Director Control Panel](#)
- [Viz Artist Performance](#)
- [On Air Information](#)
- [License Information](#)

16.3 Viz Artist Performance

The performance of Viz Artist can be analyzed with either:

- The [Performance Editor](#): This closely monitors a range of parameters to analyze the real-time performance of Viz Artist/Engine
- The [Performance Analyzer](#): This monitors key performance and camera parameters as a head-up display in the renderer window. It also initiates the log of statistic, command and event information to the log file(s).

This section contains information on the following topics:

- [Performance Editor](#)
- [Performance Analyzer](#)

See Also

- [Control Buttons](#)
- [Director Control Panel](#)
- [On Air Information](#)
- [License Information](#)

16.3.1 Performance Analyzer

The performance analyzer enables key performance and camera information to be shown in the renderer view as a head-up display. Additionally the performance analyzer can be used to write statistics, command and event information to the log files.

Note: Log files are located at <viz data folder>.

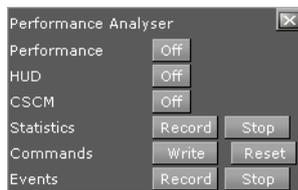
This section contains information on the following topics and procedures:

- [To Open the Performance Analyzer](#)
- [Performance Analyzer Properties](#)

To Open the Performance Analyzer

- Press and hold <Ctrl>, right-click on the X (close) button in Viz Artist.

Performance Analyzer Properties



- **Performance:** Show or hide the [Performance Bar](#) (Current (CUR) and Maximum (MAX) parameters) in the [Scene Editor](#) and Render Preview ().
- **HUD:** Enable the Head-up Display (HUD). The HUD shows the following parameters in the renderer view:
 - **Camera 1-n:** Shows the currently selected camera.
 - **Position:** Shows the camera's X, Y and Z position.
 - **Pan/Tilt/Twist:** Shows the camera's pan, tilt and twist parameters.
 - **FovX/FovY:** Shows the camera's field of view (fov) for the horizontal (X) and vertical (Y) plane.
 - **Center Shift:** Shows the X and Y position of the camera's center shift.
- **CSCM:** Shows the center shift as a cross hair in the renderer.
- **Statistics:** Write statistics to the log file.
- **Commands:** Write command to the log file.
- **Events:** Write events to the log file.

Note: These write to log file buttons are useful if there are ongoing problems with Viz Artist. Record and send the log file to Vizrt (see [Customer Support Request](#)).

See Also

- [Viz Artist Performance](#)
- [Performance Editor](#)
- [Performance Bar](#)

16.3.2 Performance Editor

The performance Editor gives an idea of the current scene rendering performance (frames per second).

Metric	Value	Unit
Current (CUR)	50.01	fps (Frame Rate: 50.0)
Maximum (MAX)	184.10	fps
Vertices (VER)	2	k
AllocTexSize (TET)	944.55	MB
TexSize (TEC)	10.55	MB
Animation (ANI)	5.13	us
Matrix (MAT)	50.96	us
Z-Sort (Z&C)	21.65	us
Video (VID)	4256.95	us
Rendering (REN)	1021.20	us
Script (SCR)	75.54	us
Plugin (PLU)	0.37	us
Idle	14568.20	us

This section contains information on the following topics and procedures:

- [To Open the Performance Editor](#)
- [Performance Editor Properties](#)

To Open the Performance Editor

1. Click .
2. Click  (Eject button) to show or hide the performance parameters.

Performance Editor Properties

Current (CUR)	50.01	fps (Frame Rate: 50.0)
Maximum (MAX)	184.10	fps
Vertices (VER)	2	k
AllocTexSize (TET)	944.55	MB
TexSize (TEC)	10.55	MB
Animation (ANI)	5.13	us
Matrix (MAT)	50.96	us
Z-Sort (Z&C)	21.65	us
Video (VID)	4256.95	us
Rendering (REN)	1021.20	us
Script (SCR)	75.54	us
Plugin (PLU)	0.37	us
Idle	14568.20	us

- **Current (CUR):** 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 **Output Format** (see [Viz Configuration](#) in the [Viz Engine Administrator Guide](#)).
- **Maximum (MAX):** 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):** The number of vectors in the scene.
- **AllocTexSize (TET):** The total allocated size of texture memory.
- **TexSize (TEC):** The size of the currently used texture memory.
- **Animation (ANI):** 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):** 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):** 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):** Consumed time in microseconds from all active scripts. This indicator is linked to the dark green bar.
- **Plugin (PLU):** Time, in microseconds, all active plugins spend in each render cycle. This indicator is linked to the orange bar.
- **Idle:** Available resources, in microseconds, the renderer has available. This indicator is linked to the light green bar.

See Also

- [Viz Artist Performance](#)
- [Performance Analyzer](#)
- [Performance Bar](#)

16.4 On Air Information

The On Air Information shows details about the parameters needed to send external control commands, and shows all connected clients with the IP address, host name and Viz Port.

This section contains information on the following topics and procedures:

- [To Show/Hide the On Air Information Panel](#)
- [Basic Tab](#)
- [Clients Tab](#)

Note: Polling for On Air information can decrease the performance. For information on how to adjust the On Air Update Interval see User Interface in the [Viz Engine Administrator Guide](#).

See Also

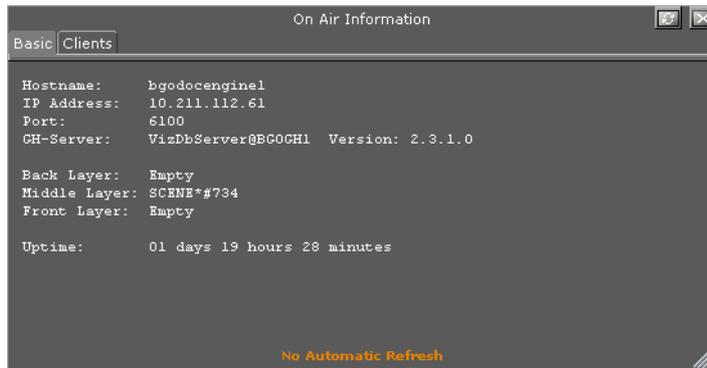
- [Control Buttons](#)
- [Director Control Panel](#)
- [Viz Artist Performance](#)
- [License Information](#)

To Show/Hide the On Air Information Panel

- Click  .

16.4.1 Basic Tab

In the Basic tab, the parameters needed to send external control commands are shown.



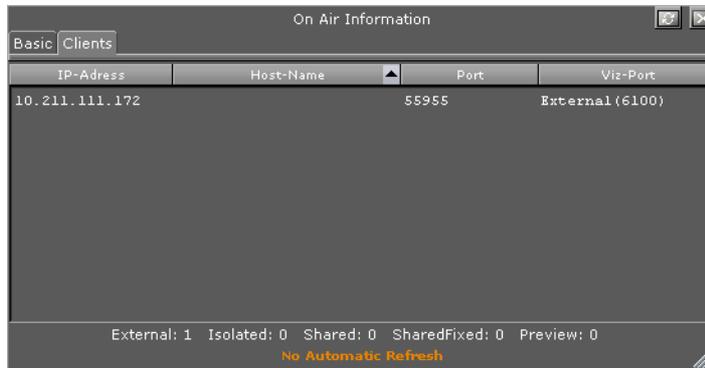
- **Refresh button:** Refreshes the status information.
- **Hostname:** Shows the name external control programs can use to communicate with Viz Artist.
- **IP Address:** Shows the IP address external control commands can communicate with Viz Artist.
- **Port:** Shows the port Viz Artist is using. Default port is 6100 (see the [Viz Engine Administrator Guide](#)).
- **GH-Server:** Shows the Graphic 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 Artist/Engine was started.

See Also

- [On Air Information](#)
- [Clients Tab](#)

16.4.2 Clients Tab

In the Clients Tab, all connected clients are shown, with the IP address, host name and Viz Port.



See Also

- [On Air Information](#)
- [Basic Tab](#)

16.5 License Information

The license information panel shows the licensing information such as licensed features and how many days the license has left before it must be renewed.



See Also

- [Control Buttons](#)
- [Director Control Panel](#)
- [Viz Artist Performance](#)
- [On Air Information](#)

17 Transition Logic

Transition Logic is first and foremost about:

- Having background graphics that are constantly on air to provide an artifact-free show 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.

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 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 show one foreground scene at a time.

Note: 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 Pilot and Viz Multichannel.

This section contains information on the following topics:

- [Standalone versus Transition Logic Scene Design](#)
- [Toggle Layer](#)
- [State Transition Animation](#)
- [Cross Animation](#)
- [Geometry Animation](#)
- [Background Scene](#)
- [Foreground Scene](#)
- [Tutorial](#)

17.1 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 images, below, show the difference in results when executing commands towards Transition Logic scenes (1) compared to Standalone scenes (2). The images show 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.

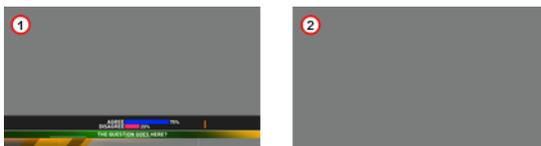
Take Lower Third



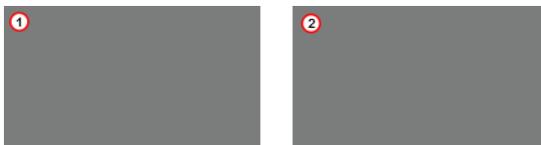
Take OTS



Take Out OTS



Take Out Lower Third

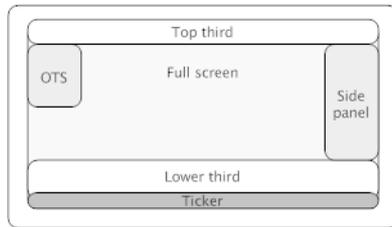


The first action, *take lower third*, results in a lower third graphics being placed on-screen. The second action, *take over the shoulder (OTS)*, places the OTS on-screen; however, with standalone scene design this will also cause the lower third to be taken out (i.e. a hard cut).

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).

17.2 Toggle Layer

A toggle layer acts as a placeholder for one or several graphical elements. Having multiple versions of a lower third on the same layer, 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](#). 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](#) plugin. 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](#)

17.3 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.

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](#)

17.3.1 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 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 state 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 shown 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

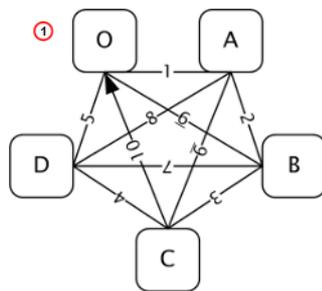
Transitioning from a SMALL to an O state is achieved by using the animation from the O to the SMALL state. In such a case the animation will be played in reverse. If this is not required, you simply add another O state after the SMALL state, and create a new animation.

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.

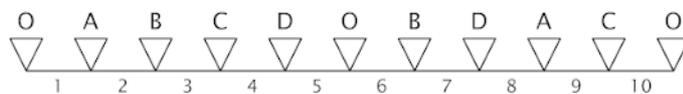
17.3.2 Identifying States

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 (1).



The Euler Walk, although not always possible, is defined as the optimal route taken to visit each point exactly once. However, the theory can be applied to Transition Logic to identify the minimum required states needed to connect each state to the others exactly once.

Straightening the Euler walk out onto the layer's director will show the required transition animations:



Note: When performing transition animations from one state to the other Viz Artist/Engine 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 Artist/Engine 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 Artist/Engine will perform the transition animation.

Looking at the example director, above, 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 Artist/Engine 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 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.

Detailed below are the minimum required number of transition animations for a graphics layer with a given number of states:

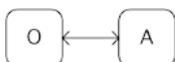
- **1 State:**

- Animations: Not possible.



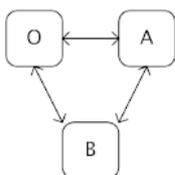
- **2 States:**

- Animations: Requires a minimum of 1 animation



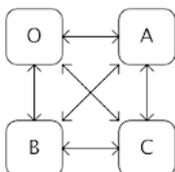
- **3 States:**

- Animations: Requires a minimum of 3 animations



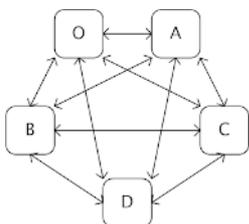
- **4 States:**

- Animations: Requires a minimum of 6 animations



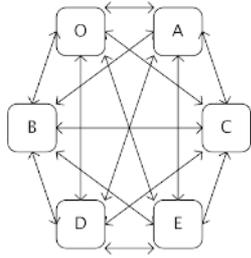
- **5 States:**

- Animations: Requires a minimum of 10 animations



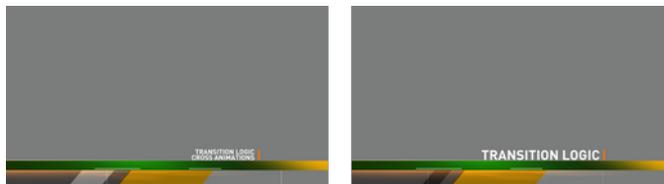
- **6 States:**

- Animations: Requires a minimum of 15 animations

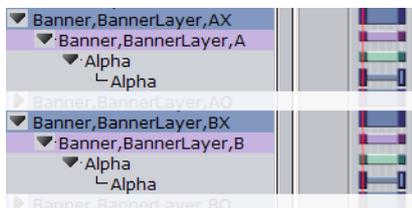


17.4 Cross Animation

Cross animations are used in Transition Logic scene design to animate one object in and another out. The example below shows a lower third that has an alpha cross fade fading from a two lines of text to a one line of text.



Cross animation directors are added by the [Toggle](#) plugin. 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](#)

17.5 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.



17.6 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.

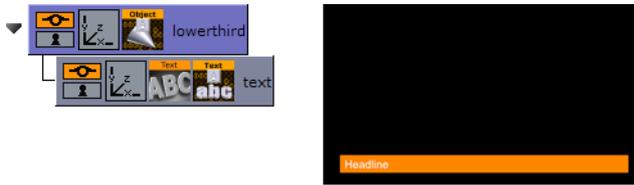
See Also

- [State Transition Animation](#)
- [Cross Animation](#)
- [Foreground Scene](#)

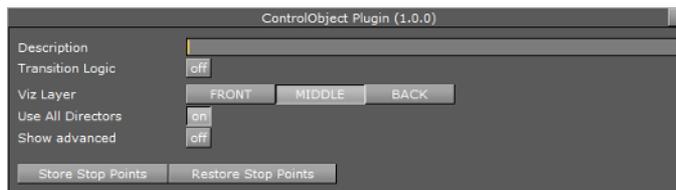
17.7 Foreground Scene

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.



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](#) plugin 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 Artist/Engine 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 Artist/Engine 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 plugins' 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](#) plugins

17.8 Tutorial

This section contains information on the following topics:

- [Create a Project](#)
- [Create a Background Scene](#)
- [Create a Foreground Scene](#)

17.8.1 Create a Project

It is good practice to create a project with a folder for each new design. This will make sure that a project with more than one design does not get mixed up, and potentially overwrite other files.

See Also

- [Folders](#)

17.8.2 Create a Background Scene

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.

Note: It is recommended to create the background scene before [Create a Foreground Scene](#) as the background scene is needed to complete the foreground scene(s).

This section contains information on the following topics:

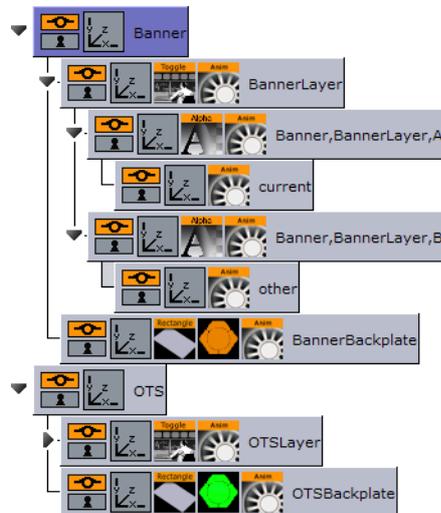
- [To Define the Toggle Layers](#)
- [To Create a Toggle Layer](#)
- [To Add Key](#)

To Define the Toggle Layers

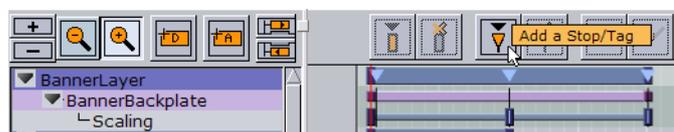
1. Create a list of all graphical elements to be shown. For example:
 - A large two lines lower third.
 - A small one line lower third.
 - A small one line top third.
 - A left side over the shoulder (OTS).
 - A right side OTS.
 - A logo bug.
 - A "coming up next" bug.
 - A news ticker.
 - A clock.
2. 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

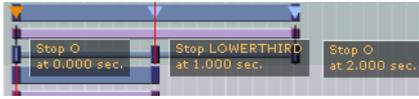


1. Create a new scene, and name it **default**. This will be the background scene.
2. Add a group container to the scene tree, and name it **Banner**.
 - This will be the placeholder for the banner's toggle layer and backplate graphics for the banner foreground scenes.
3. Add a **sub-container** to the Banner container, and name it **BannerLayer**.
 - This will be the placeholder for the [Toggle](#) plugin.
4. Add the **Toggle plugin** 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)

- Open the **Stage** and check that the animation resides under the BannerLayer director. If not, place it (drag and drop) under the BannerLayer director.



- Select the Banner director, and the director's Time Marker to 0.000 seconds, and **add a tag**.
- Select the tag and name it **O** (uppercase O, not zero).
- 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.
- Open the [Toggle](#) plugin editor, and click the **Default Key Frames** button.
 - Adding Key Frames adds a 1 second default alpha cross fade effect to objects placed on that layer.

Note: Adding default Key Frames is not possible until the layer's director has at least two states defined where one has to be the O state.

- 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)
- 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](#)
- [Create a Project](#)
- [Create a Foreground Scene](#)
- [Toggle](#)

17.8.3 Create a Foreground Scene

Before creating foreground scenes you should start by [Create 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 plugin. The Control Object plugin 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 plugins (e.g. Control Text and Control Image). Additionally the foreground scene can be used for adding preview tags such that the scene can be correctly previewed by newsroom users that work with the Viz Pilot Newsroom client.

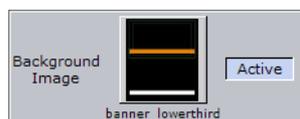
This section contains information on the following procedures:

- [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** plugin 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** plugin 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.



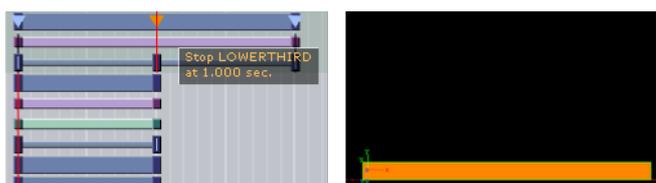
13. **Repeat** steps 2-11 for the OTS foreground scene with the following parameters, transformation properties and animations.
 - Scene name: 2000
 - Description: Text
 - Transition Logic: On
 - Layer Identifier: OTSLayer
 - State Identifier: OTS_LEFT
 - Background scene: default
 - Position: X: -330, 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.

1. **Save** the OTS scene.

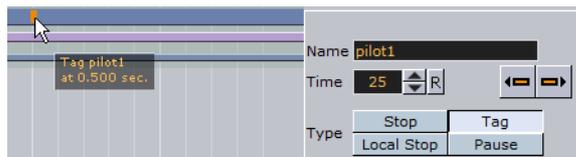
To Create a Background Image for the Foreground Scene

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

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 Artist/Engine preview, either from a newsroom system, or when connected to a Media Sequencer with a control application such as Viz Trio, the **pilot1** tag will be the default preview point for the foreground scene.



Note: It is possible to have several stop-points, in which case pressing next in a control application used for preview will advance to the next stop-point. When reaching the last stop-point in a director, pressing next will not rewind. In a TL scene you can have separate stop-points in sub-directors, and the stop-points in the various sub-directors need not be at the same point in the timeline.

See Also

- [Foreground Scene](#)
- [Container Plugins](#)
- [Viz Pilot User's Guide](#) on remote Viz Artist/Engine preview.

18 Scripting

To learn and use the computer language **Viz 3 Script**, first the basic principles of programming must be understood.

The basic principles of programming are described 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, here is also a quick introduction on how to create and run scripts, how to create reusable plugins of scripts and other example scripts that will help to script functionality:

- [Create and Run Scripts](#)
- [Create Script-based Plugins](#)
- [Control 3D Stereoscopic Clip Playback](#)
- [Program Examples](#)

18.1 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 were 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 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.

18.2 Program Structure

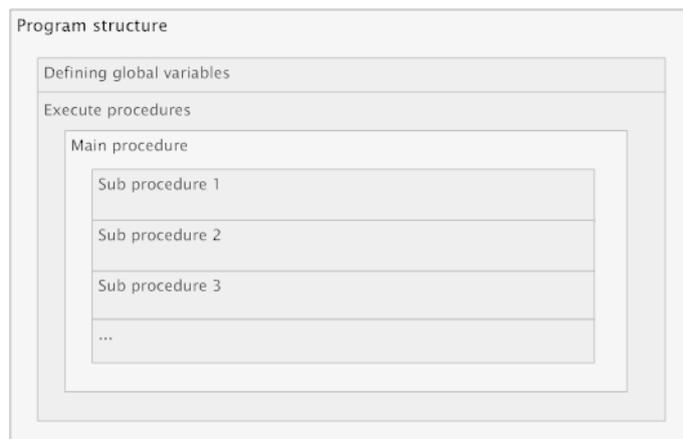
The certain text commands, which are written within the defined programming syntax, will lead to specific actions, done by the computer. Those commands and definitions are written with an adequate Text Editor and are called "source code". This code will then be sent to the compiler for processing.

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 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 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 required result of changing states between 0 and 1.



18.3 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 programming spelling or syntax, so the compiler will not recognize the word. So before using such a variable, it has to be defined 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.

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 programming 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
```

18.3.1 Basic Data Types

This section describes and gives examples of the basic data types:

- [Integer](#)
- [Double](#)
- [String](#)
- [Boolean](#)
- [Array](#)
- [Structure](#)

Integer

Integer numbers

Example: 0,1,2,3 ...

Double

Floating point numbers

Example: 4,213415

String

Text

Example: "Hello", "Florian"

Boolean

State 1/0

Example: true, false

Example:	<code>'String values are enclosed in double quotes Dim name as String name = "Florian" Dim state as Boolean state = true</code>
----------	---

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 programming 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).

To describe a soccer player with programming 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](#)

18.4 Procedures

Like variables, procedures get defined with an appropriate name to manage them more easily in the code.

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 programming languages, there are two types of procedures in Viz 3 Script:

- [Sub-routines](#) (process certain defined commands)
- [Functions](#) (sub-routines that return a variable)

18.4.1 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.

18.4.2 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 is possible to handle the function as a standard variable.

```
sub tmp()
    new_glass = multiply(wineglass) + 10
end sub
```

18.5 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.

Operator	Description
=	Assigning integer, double, string, etc.
==	Boolean assignment (true/false)

Comparison Operator(s)

To compare certain variables to execute different commands (see [Conditions](#)).

Arithmetic Operator(s)

Arithmetic operators are used for executing basic mathematical calculations.

Operator	Description
+	Addition
-	Substraction
*	Multiplication
/	Division

Logical Operator(s)

Allows it to combine two Boolean states.

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)

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.

Operator	Description
<code>+=</code>	Addition of a value to the actual variable
<code>-=</code>	Subtraction of a value from the actual variable
<code>*=</code>	Multiplication of a value of the actual variable
<code>/=</code>	Division of a value on the actual variable
<code>&=</code>	Extension of a string with another one

18.6 Conditions

To make decisions 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:

Operator	Literal	Description
<code>=</code>	equals	Identical values
<code><</code>	smaller than	Value is smaller than the referenced value
<code>></code>	greater than	Value is greater than the referenced value
<code><=</code>	less than or equal	Value is less than or equal to the referenced value
<code>>=</code>	greater than or equal	Value is greater than or equal to the referenced value

Operator	Literal	Description
<>	unequal	Value is different from that of the referenced value

18.7 Loops

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](#)

18.7.1 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
```

18.7.2 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.

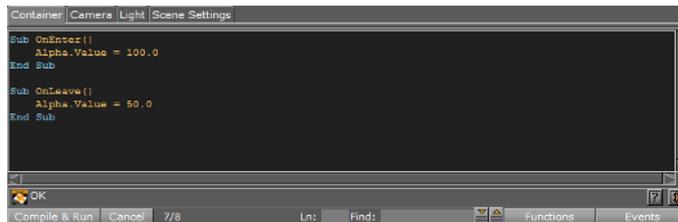
```
For i = 0 to 10
  Println i
```

Next

The output should be the following: 1 2 3 4 5 6 7 8 9 10 (one number per line)

18.8 Script Editor

This section contains details how to open, create and show scripts in the Script Editor.



This section contains the following procedures:

- [To Open the Container Script Editor](#)
- [To Create a Script in the Container Script Editor](#)
- [To Show Script Plugins in the Container Script Editor](#)

See Also

- [Script Editor Properties](#)
- [Create and Run Scripts](#)
- [Create Script-based Plugins](#)
- [Script Plugins](#)
- [Script](#) Plugin

To Open the Container Script Editor



1. Add the [Script](#) plugin to a Scene.

.....
Note: This plugin is located in: Built-ins -> Container Plugins -> Global

2. Click on the Script plugin icon.

To Create a Script in the Container Script Editor

1. Add the [Script](#) plugin to the target Container.

.....
Note: This plugin is located in: Built-ins -> Container Plugins -> Global

2. In the Script Editor type in the new script (see also [Script Editor Properties](#)).

.....
Note: Also, copy and paste a script from another source, if required.

3. Click **Compile & Run** .

To Show Script Plugins in the Container Script Editor

1. Add the [Script](#) plugin to the target Container.

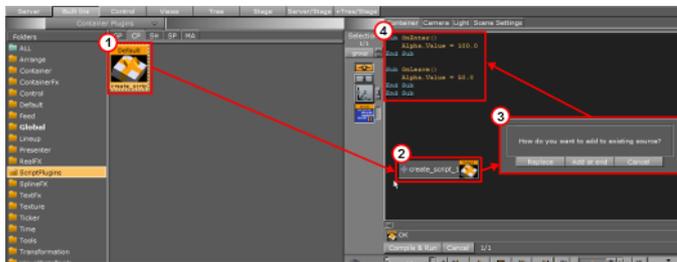
Note: This plugin is located in: Built-ins -> Container Plugins -> Global

2. Drag and drop a created Script Plugin (1 (see [Script Plugins](#))) icon to the Script Editor (2).

Note: These plugins are located in: Built-ins -> Container Plugins -> ScriptPlugins

3. Click **Replace** or **Add to End** (3).

Example: Replaced text (4).



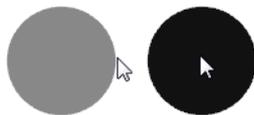
18.9 Create and Run Scripts

This section contains two script examples:

- [Example I](#): This example shows how to write a simple script that controls the alpha value of an object
- [Example II](#): This example shows how to write a simple script that scales the x-scaling value of a container, based on x-scaling value of another container.

Example I

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 [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 immediately 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 [Compile & Run](#).

Click **Edit** to return to editing mode.

To interact with the scene click the [E](#) button ([Scene Editor Buttons](#)) in the [Scene Editor](#).

Note: The [E](#) button offers a good way for designers to test touch based interaction points added to a scene. The script is then the logic which triggers what happens next (i.e. events).

Example II

This 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
```

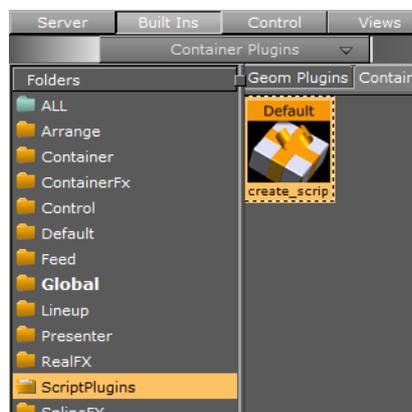
Note: The expression `Scaling.x` refers to the current container while `cube.Scaling.x` refers to the container referenced by the variable cube.

See Also

- [Alpha](#)
- [Container](#)
- [Scaling](#)

18.10 Create Script-based Plugins

A created container script can be used again in other Containers. To do this create a plugin of that script.



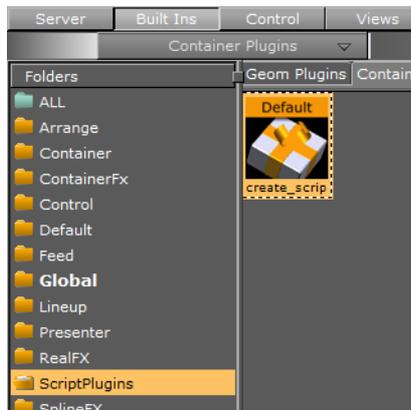
Script-based plugins can be used the same as ordinary plugins and, in addition, allow the following operations:

- Can be deleted: Drag and drop the onto the trash can.
- The source code of a script-based plugin can be copied into a script editor: Drag the plugin from [Script Plugins](#) folder to the [Script Editor](#).

This section contains the following procedures:

- [To Create a Plugin From a Script](#)
- [To Edit a Script Plugin](#)

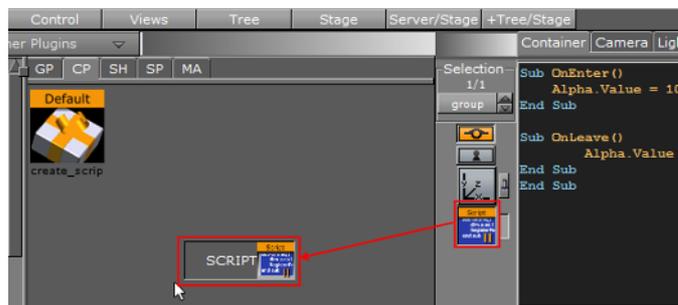
To Create a Plugin From a Script



1. Create a script (see [To Create a Script in the Container Script Editor](#)).
2. Click **Compile & Run**.

Note: If the script is bad a Script Plugin can not be created. Correct any bad script first.

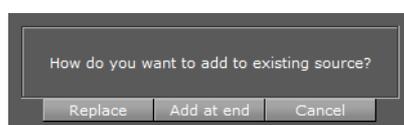
3. Click **Built Ins**.
4. Click **Container Plugins**.
5. Select the **ScriptPlugins** folder.
6. Drag and drop the Script icon to the ScriptPlugins folder.



7. Enter a name for the Script Plugin.
8. Click **OK**.

To Edit a Script Plugin

1. Add a Container to the Scene Tree.
2. Add a [Script](#) plugin.
3. Open the [Script Editor](#) (*Scene Settings -> Script*).
4. Drag and drop the required script plugin from the **ScriptPlugins** folder to the Script Editor.
5. Click **Replace**.



6. Edit the script as required.
7. Click the **Compile & Run** to save.

See Also

- [Script](#) Plugin
- [Script Editor](#)

18.11 Control 3D Stereoscopic Clip Playback

If you want to playback 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: This is for clips only.

The playback 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.
- **HSoft / 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** in [Viz Configuration](#) (see the [Viz Engine Administrator Guide](#)) for setting stereo modes
- [Clip Channel: Stereo](#) script example

18.12 Program Examples

To explain the described terms and definitions, this chapter shows some source code examples and explains the important steps and commands.

Most of the code samples detailed here are written in an easy and understandable form for programming beginners. Some are basic, while others are more advanced and specific for their use case. More efficient code structures and improvements are possible.

This section contains information on the following topics:

- [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](#)
- [General Script Examples](#)

18.12.1 Initializing Scripts

This 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:

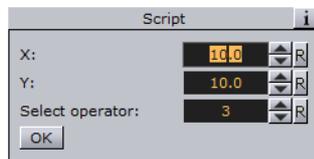
```

The variable a has the following value: 10000
The variable b has the following value: 5000
The addition of the variables are: 15000

```

18.12.2 Calculating Values

This example shows a calculator that is able to calculate with two values, which use basic arithmetic operators + , - , * , / (see [Arithmetic Operator\(s\)](#)).



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 (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 mathematics 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("y")
    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 mathematics 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
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.

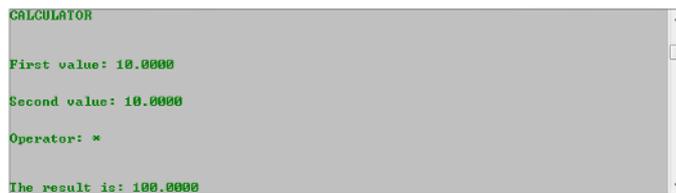
The final printing function takes the 2 variables “`x`” and “`y`” and the result and prints it formatted on the screen. If the Boolean variable “`valid`” is set to false, the error message will be shown.

```

`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

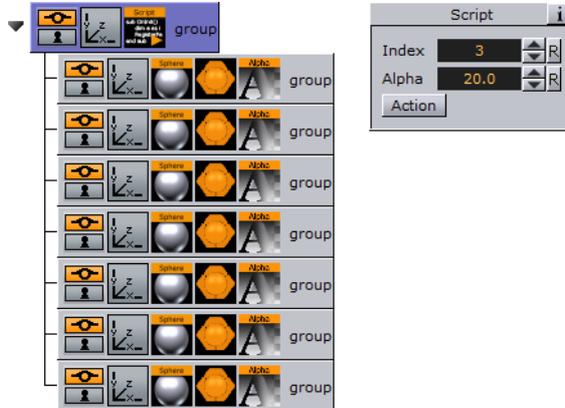
```

See Also

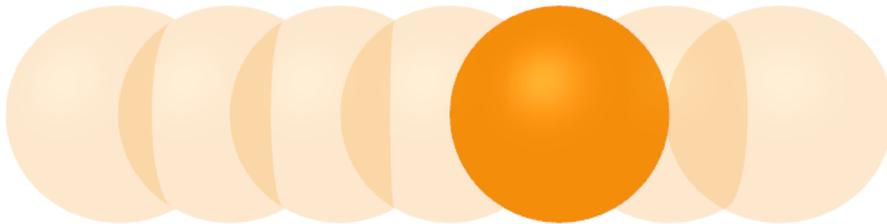
- [Callback Procedures](#)
- [Global Procedures](#)
- [Boolean](#)
- [Double](#)
- [Integer](#)
- [String](#)
- [Structure](#)

18.12.3 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 whereas the alpha value is set for all other containers.



```
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

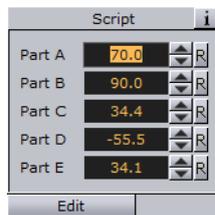
- [Callback Procedures](#)

- [Global Procedures](#)
- [Boolean](#)
- [Container](#)
- [Double](#)
- [Integer](#)

18.12.4 Registering Parameters and Moving Objects

This script example initializes a set of registered parameters that enables the `OnParameterChanged` procedure to rotate the objects accordingly.

Registered parameters



```
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
```

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
end sub
```

```

    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](#)

18.12.5 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

```

18.12.6 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

```

```
end sub
```

18.12.7 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 will 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("Default")
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](#)

18.12.8 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 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
end sub

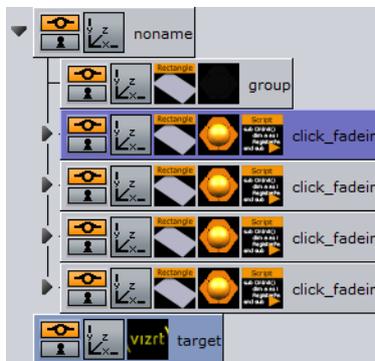
```

See Also

- [Callback Procedures](#)
- [Global Procedures](#)
- [Boolean](#)
- [Scaling](#)

18.12.9 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.

```

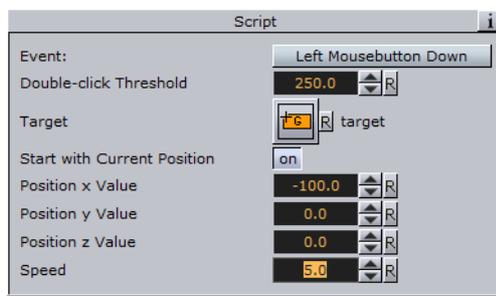
dim events as Array[String]
dim action as Array[String]
dim str_action as string
dim str_event as string
dim doubleclick_tracker as Integer
dim dbl_threshold as Double
dim target_container as container
dim p_anim as Boolean = false
dim p_step as Vertex
dim p_dest as Vertex
dim p_speed,p_time as Double
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_threshold=GetParameterDouble("dbl_threshold")
end sub

```



```

sub OnInitParameters()
    `create GUI
    RegisterRadioButton("do_event", "Event:",2,events)
    RegisterRadioButton("do_action", "Action:",0,action)
    RegisterParameterDouble("dbl_threshold","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

```

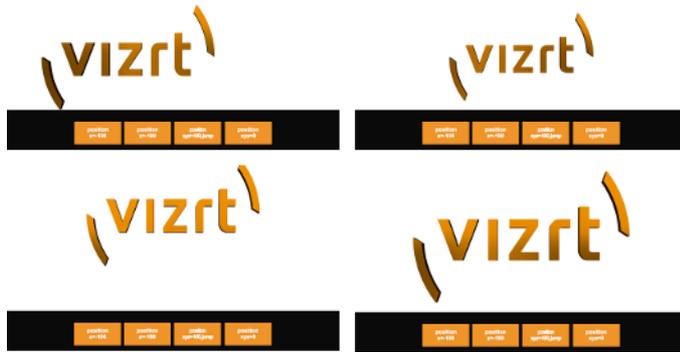
```

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_threshold) 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_threshold) then
      animate()
    end if
  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("position_y"),getPa
    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
    end if
end sub
```

```

        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](#)

18.12.10 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

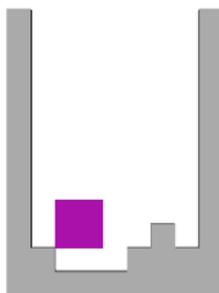
```

See Also

- [Callback Procedures](#)
- [Global Procedures](#)
- [Array\[Type\]](#)
- [Boolean](#)
- [Container](#)
- [Integer](#)

18.12.11 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](#)

18.12.12 Clip Channel

```

\ -----
\
\ -----
structure SClipInstance
    oClipChannel          as ClipChannel
    iClipChannelID        as Integer
    bLiveUpdate           as Boolean
    szShmKeyModeChanged  as String
    szShmKey FrameChanged 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].szShmKey FrameChanged 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].szShmKey FrameChanged =
oClip[iClip].oClipChannel.GetRegisterChangedClipFrame()
    println( "[" & iClip & "]GetRegisterChangedClipFrame()" &
oClip[iClip].szShmKey FrameChanged )
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].szShmKey FrameChanged.length == 0 then
        oClip[iClip].szShmKey FrameChanged = "com.vizrt.clip" &
IntToString( oClip[iClip].iClipChannelID , 1 ) & ".frame"
    end if
    dim oShmMap as SharedMemory
    if Scene.Map.ContainsKey( oClip[iClip].szShmKey FrameChanged ) then
        oShmMap = Scene.Map
    elseif System.Map.ContainsKey( oClip[iClip].szShmKey FrameChanged )
then
        oShmMap = System.Map
    else
        oShmMap = Scene.Map
        oShmMap.CreateKey( oClip[iClip].szShmKey FrameChanged )
    end if
\
oClip[iClip].oClipChannel.RegisterChangedClipFrame( oClip[iClip].szShmKey
FrameChanged )
oShmMap.RegisterChangedCallback( oClip[iClip].szShmKey FrameChanged )
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 sub

```

```

        end if
    end sub
\ -----
\
\ -----
sub ClipPlayerRegisterChanges()
    oClip[iClip].oClipChannel.SetRegisterChangedClipMode
    ( oClip[iClip].szShmKeyModeChanged )

    oClip[iClip].oClipChannel.SetRegisterChangedClipFrame( oClip[iClip].szShmKey
    FrameChanged )
end sub
\ -----
\
\ -----
sub ClipPlayerUnRegisterChanges()
    oClip[iClip].oClipChannel.SetRegisterChangedClipMode ("" )
    oClip[iClip].oClipChannel.SetRegisterChangedClipFrame ("" )
end sub

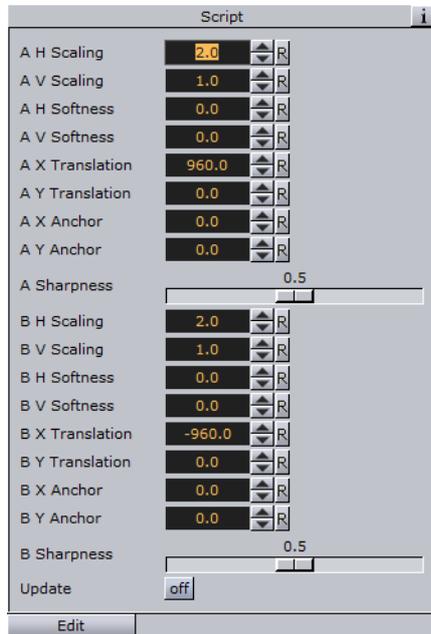
```

See Also

- [Callback Procedures](#)
- [Global Procedures](#)
- [Array\[Type\]](#)
- [Boolean](#)
- [ClipChannel](#)
- [Double](#)
- [Integer](#)
- [String](#)

18.12.13 Clip Channel: Stereo

This 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 [Control 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

```

18.12.14 General Script Examples

Detailed below are several script examples to try.

This section contains information on the following topics:

- [Calculate the Days of a Week and Month](#)
- [Have a Viz Script Update Scene from a Text File](#)

Calculate the Days of a Week and Month

This script example calculates the day of the week and the number of days of the month.

```

DIM smap AS Array[String]

smap.size = 7

smap[0] = "Sun"

smap[1] = "Mon"

smap[2] = "Tue"

smap[3] = "Wed"

smap[4] = "Thu"

smap[5] = "Fri"

smap[6] = "Sat"

```

```

function CalcDayOfWeek(iYear AS Integer,
iMonth AS Integer, iDay AS Integer) AS String

    DIM dt AS DateTime

    dt.year = iYear

    dt.month = iMonth

    dt.dayofmonth = iDay

    dt.normalize()

    CalcDayOfWeek = smap[dt.dayofweek]

end function

function CalcNumOfDaysOfMonth(iYear AS
Integer, iMonth AS Integer) AS Integer

    DIM dt AS DateTime

    dt.year = iYear

    dt.month = iMonth+1

    dt.dayofmonth = 0

    dt.normalize()

    CalcNumOfDaysOfMonth = dt.dayofmonth

end function

sub OnInit()

    DIM dt AS DateTime

    dt.year = 2012

    dt.month = 2

    dt.dayofmonth = 23

    println("Date: " & CalcDayOfWeek(dt.year,
dt.month, dt.dayofmonth) & ", " & dt.year & "" & dt.month & "" &
dt.dayofmonth)

    println("Days: " &
CalcNumOfDaysOfMonth(dt.year, dt.month))

end sub

```

Have a Viz Script Update Scene from a Text File

This script example updates a Scene from a text file

```

        dim timer,limit as integer

        timer = 0

        sub OnInitParameters()

            RegisterInfoText("Text File Grabber
1.0" & chr(10) & chr(10) & "Grabs text file and puts optional
prefix/suffix around the contents of the text file." & chr(10) &
"Filename supports paths with drive letters and UNC paths." & chr(10)
& chr(10) & "Text in \"Default Text\" box is used if file specified in
File box cannot be opened or accessed." & chr(10) & chr(10) & "Listens
to scene map variable \"text_update\" -- text files will be re-read when
this variable changes.")

            RegisterFileSelector("file", "File", "c:
\test", "1.txt", " *.*")

            RegisterParameterString("default","Default
text","",40,50,"")

            RegisterParameterString("prefix","Prefix
(optional)", "",40,50,"")

            RegisterParameterBool("prefixIfNull","Include prefix if file not
found",FALSE)

            RegisterParameterString("suffix","Suffix
(optional)", "",40,50,"")

            RegisterParameterBool("suffixIfNull","Include suffix if file not
found",FALSE)

            RegisterParameterInt("updateFields","Auto-
Update interval (fields)",300,60,99999999999)

            RegisterParameterBool("debug","Debug output
to console",FALSE)

        end sub

        sub OnParameterChanged(parameterName As
String)

            're-populate text when a parameter changes
            if parameterName <> "updateFields" then
                UpdateText

            if parameterName = "updateFields" then

                limit = GetParameterInt("updateFields")

```

```

end if

end sub

sub OnExecPerField()
timer = timer + 1
if timer >= limit then
if GetParameterBool("debug") then println
"Timer limit reached"

UpdateText
timer = 0
end if
end sub

sub OnInit()

Scene.Map.RegisterChangedCallback("text_update")

limit = GetParameterInt("updateFields")

UpdateText
end sub

sub OnSharedMemoryVariableChanged(map As
SharedMemory, mapKey As String)
if mapKey = "text_update" then
UpdateText
end if
end sub

sub UpdateText()
geometry.text = GetTextFile()
end sub

function GetTextFile() as string
dim content, path as string
path = GetParameterString("file")

```

```

"File path: " & path
then
"Successfully loaded file"
content.trim
if len(GetParameterString("prefix")) > 0
then content = GetParameterString("prefix") & " " & content
if len(GetParameterString("suffix")) > 0
then content = content & " " & GetParameterString("suffix")
else
if GetParameterBool("debug") then println
"Could not load file; using default text"
content = GetParameterString("default")
if GetParameterBool("prefixIfNull") then
content = GetParameterString("prefix") & " " & content
if GetParameterBool("suffixIfNull") then
content = content & " " & GetParameterString("suffix")
end if
GetTextFile = content
if GetParameterBool("debug") then println
"Returning '" & GetTextFile & "'"
end function

```

19 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 information on the following high-level topics:

- [Callback Procedures](#)
- [Data Sharing](#)
- [Data Types and Procedures](#)

See Also

- [Shared Memory \(SHM\)](#)

19.1 Callback Procedures

This section lists 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 <i>Compile and Run</i> 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 <i>true</i> by default, is set to <i>false</i> , 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 (<i>subContainerId</i> as Integer, <i>pos</i> 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 (<i>subContainerId</i> as Integer, <i>pos</i> 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 (<i>buttonId</i> as Integer)	Called when the user clicks on any push button defined by RegisterPushButton . If the script defines several push buttons, use <i>buttonId</i> 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 correct place where the plugin can set the UI state to enabled/disabled with SendGuiStatus or shown/hidden with SendGuiParameterShow .
Sub OnKeyDown (<i>keyCode</i> as Integer)	Called whenever the user presses a key. <i>keyCode</i> 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 (<i>keyCode</i> as Integer)	Called whenever the user releases a key.
Sub OnButtonDown6DOF (<i>buttonId</i> as Integer, <i>pos</i> as Vertex, <i>rot</i> as Vertex)	Called when the user clicks on a scene grid. <i>buttonId</i> defines the ID of the pressed button, <i>pos</i> specifies the 3D world position of the cursor and <i>rot</i> gives you the actual rotation.

Sub OnButtonUp6DOF (<i>button</i> as Integer, <i>pos</i> as Vertex, <i>rot</i> as Vertex)	Called whenever the user releases a mouse button.
Sub OnMove6DOF (<i>button</i> as Integer, <i>pos</i> as Vertex, <i>rot</i> as Vertex)	Called when the user moves the cursor on a scene grid. <i>button</i> defines the ID of the pressed button, <i>pos</i> specifies the 3D world position of the cursor and <i>rot</i> gives you the actual rotation.
Sub OnMoveRelative6DOF (<i>button</i> as Integer, <i>pos</i> as Vertex, <i>rot</i> as Vertex)	Called when the user moves the cursor on a scene grid. <i>button</i> defines the ID of the pressed button, <i>pos</i> specifies the offset vector to the last 6DOF position of the cursor and <i>rot</i> gives you the actual rotation.
Sub OnScale6DOF (<i>button</i> as Integer, <i>scale</i> as Vertex)	Called when the user performs a scale operation (e.g.: with a multi-touch device) on a scene grid. <i>button</i> defines the ID of the pressed button and <i>scale</i> specifies the actual scale factor.
Sub OnMTHit (<i>stroke</i> as Integer, <i>x</i> as Integer, <i>y</i> as Integer)	Called when the user touches this object (where this scriptplugininstance resides on) at a multi-touch device. <i>stroke</i> gives you the stroke ID of the multi-touch operation and <i>x</i> and <i>y</i> 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 (<i>x</i> as Integer, <i>y</i> as Integer)	Called when the user performs a menu gesture on the multi-touch device. <i>x</i> and <i>y</i> 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 (<i>x</i> as Integer, <i>y</i> as Integer, <i>rot</i> as Vertex, <i>scale</i> as Vertex, <i>pressure</i> as Double)	Called when a PZR2D control was registered. <i>x</i> and <i>y</i> specify the hit position in screen coordinates, <i>rot</i> gives you the object rotation, <i>scale</i> holds the scale factor and <i>pressure</i> contains the current mean contact pressure.
Sub OnMTControlButton (<i>strokes</i> as Integer, <i>pressure</i> as Double)	Called when a Button control was registered. <i>strokes</i> contains the current hit count on the button object and <i>pressure</i> 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 (<i>x</i> as Integer, <i>y</i> as Integer)	Called whenever the mouse cursor is moved within the output window. <i>x</i> and <i>y</i> specify the cursor position in screen coordinates.

Sub OnLButtonDown()	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 OnMButtonDown()	
Sub OnRButtonDown()	
Sub OnLButtonUp()	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 OnMButtonUp()	
Sub OnRButtonUp()	
Sub OnMouseWheel (<i>distance</i> as Integer)	Called whenever the mouse wheel is rotated within the output window.
Sub OnSharedMemoryVariableChanged (<i>map</i> as SharedMemory, <i>mapKey</i> as String)	Called when a variable in a <i>SharedMemory</i> map is changed. The <i>SharedMemory</i> 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 <i>SharedMemory</i> map by calling its <i>RegisterChangedCallback</i> procedure.
Sub OnSharedMemoryVariableDeleted (<i>map</i> as SharedMemory, <i>mapKey</i> as String)	Called when a variable in a <i>SharedMemory</i> map is deleted. The <i>SharedMemory</i> 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 <i>SharedMemory</i> map by calling its <i>RegisterChangedCallback</i> procedure.
sub OnGeometryChanged (<i>geom</i> as Geometry)	Called when a geometry changed which was registered with <i>RegisterChangedCallback</i> or <i>RegisterTextChangedCallback</i> before.

19.2 Data Sharing

This section describes the data sharing mechanism provided by **Shared Memory Maps** (SHM). A SHM is a map which holds user-defined variables indexed by a string. It is exposed to the script interface via the [Shared Memory](#) data type. The variables are stored as [Variant](#) objects and thus can hold objects of any other type.

There are three types of SHM and the difference between them is their distribution locations:

- [Scene](#) .Map: Referenced in plugins as Scene. This is the map local to the current Scene only. Every Scene has one map that can be used to exchange data among the scripts in the Scene.

- [System](#) .Map: Referenced in plugins as Global. The system-wide map allows for data sharing among the scenes currently loaded into memory (e.g. data sharing between scenes in different layers).
- [VizCommunication](#) .Map: Referenced in plugins as Distributed. A distributed map that enables data sharing among the Viz Engines connected to the same Graphic Hub. The Viz Engines must be logged in with users from the same Graphic Hub group.

This section contains information on the following topics:

- [Script Examples](#)
- [Receiving Notification on Map Changes](#)
- [VizCommunication.Map](#)
- [Map Access Through a TCP or UDP Interface](#)
- [Map Access through the Command Interface](#)
- [Map Access from Within Plugins](#)

See Also

- [Shared Memory \(SHM\)](#)
- [Apply Shared Memory](#) Plugin
- [Multi-Touch Plugins](#), see [Plugin Event and Notification System](#)
- [Visual Data Tools](#)
- [Control 3D Stereoscopic Clip Playback](#)

19.2.1 Script Examples

This section contains two script examples for Shared Memory (SHM):

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).

This 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, see also [Sub VizBoldOnEnterSubContainer\(subContainer as Container\)](#).

Example II

This example shows how external data can be sent to all Viz Engines connected to a Graphic Hub, and how they can utilize the same data. The example also shows how to group together related external data into an array inside the Viz Engine. This is not the only way of managing variables. They could just as well be sent individually, which will then not demand extra variable splitting in a script.

Note: Vizrt do not recommend this as a method for the import of large amounts of data. For more information see [External Data Input](#).

The image below shows a command sent to a Viz Engine which assigns a value to a variable named `my_variable`.

To create and set a value shared by all Viz Engines, that are connected to the same Graphic Hub, send the following command through 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>
```

The below image shows a script which uses the 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 a Viz Engine the shared memory can be accessed by a connection to the Viz Engine listener port 6100. For more information on ways to access shared memory, see [Map Access through the Command Interface](#).

Inside the Viz Engine also set and retrieve data from the SHM through scripting, see:

- [To Set a Value](#)
- [To Retrieve a Value](#)
- [To Split the Values into an Array](#)
- [To Assign a Value to a Text Container](#)
- [To Automatically Update Graphics](#)

Address the different areas as follows:

- [Scene.map](#)
- [System.map](#)
- [VizCommunication.map](#)

To Set a Value

Through a Command Console:

```
send VIZ_COMMUNICATION*MAP SET_STRING_ELEMENT "my_variable"
  1.2;10.8;20.3;15.9
```

or through Script:

```
VizCommunication.map["my_variable"] = 1.2;10.8;20.3;15.9
```

To Retrieve a Value

```
Dim MyVariable as string
MyVariable = (string)VizCommunication.map["my_variable"]
```

string is here used to format the data from the SHM 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 four:

```
\ 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 SHM 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](#).

19.2.2 Receiving Notification on Map Changes

Sometimes you want to be notified when a value in an SHM 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

Functions `OnExecPerField()`, `OnSharedMemoryVariableChanged()` and `OnSharedMemoryVariableDeleted()` in [Callback Procedures](#).

19.2.3 VizCommunication.Map

The map exposed by the [VizCommunication](#) type is used to share data among the client computers connected to a Graphic 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.

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. See [Synchronization](#) in [Shared Memory \(SHM\)](#).

See Also

- [Shared Memory \(SHM\)](#)

19.2.4 Map Access Through a TCP or UDP Interface

Elements in the distributed memory map (`VizCommunication.Map`) can be set (but not retrieved) through a TCP or UDP interface, exposed by Viz Artist. This option is particularly useful when writing external control applications.

For more information about map access through a TCP or UDP Interface, see [External Data Input](#) in [Shared Memory \(SHM\)](#).

See Also

- **Shared Memory Properties** in [Viz Configuration](#) (see the [Viz Engine Administrator Guide](#))
- [Shared Memory \(SHM\)](#)

19.2.5 Map Access through the Command Interface

There is limited support for getting and setting map elements through a 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

A full list of supported commands is at: <viz install folder>\Documentation\\CommandInterface\index.html

Note: From the list of commands, the commands, CLEAR, DELETE_ELEMENT and PURGE_ELEMENT will only work when sent through the command interface of Viz Artist.

IMPORTANT! The command CLEAR must be run on each Engine where the MAP is to be reset (VIZ_COMMUNICATION*MAP CLEAR).

For command examples see [Command Examples](#) in [Shared Memory \(SHM\)](#)

See Also

- **Shared Memory Properties** in [Viz Configuration](#) (see the [Viz Engine Administrator Guide](#))
- [Shared Memory \(SHM\)](#)

19.2.6 Map Access from Within Plugins

Functions for getting and setting map elements are provided by the plugin interface. For more information, see [Plugin API](#) in [Shared Memory \(SHM\)](#).

See Also

- [Shared Memory \(SHM\)](#)

19.3 Data Types and Procedures

This section contains a complete reference of the Viz Artist's supported 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	Key Frame	Shared Memory	Global Procedures

19.3.1 Alpha

Base type

[Base](#)

Description

An alpha object. This is used to control a container's alpha value.

Properties

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

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 FindKey FrameOfObject (Key FrameName as String) as Key Frame	(Inherited from Base) Finds a Key Frame 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\$Key FrameName" for

Name	Description
	the Key FrameName argument. Nested directors may be specified like this: "directorName1\$directorName2\$Key FrameName".
Function FindOrCreateChannelOfObject (channelName as String) as Channel	<i>(Inherited from Base)</i> This function works like FindChannelOfObject except that if no animation channel with the specified name exists, a new one is created.
Function GetChannelsOfObject ([out] v as Array[Channel]) as Integer	<i>(Inherited from Base)</i> 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	<i>(Inherited from Base)</i> 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 ()	<i>(Inherited from Base)</i> Mark this object as changed.

19.3.2 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

Property	Description
LBound as Integer [read-only]	The index of the first element (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 element (-1 if the array is empty)

Member Procedures

Name	Description
Sub Clear ()	Erases all elements
Sub Erase (index as Integer)	Erases the element at 'index'.

Name	Description
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. <i>comparisonFunction</i> must be the name of a user-defined comparison function that takes two arguments of type <i>Type</i> and returns an integer. The return value must be -1 if the first argument is smaller than the second, 1 if the first argument is greater than the second, and 0 if the arguments are equal. <i>object</i> is the this pointer passed to the comparison function. It may be <i>null</i> if the comparison function does not access any member variables. The comparison function must be defined in the same script as the <i>Type</i> (if it is a structure) or in the script where <i>Sort</i> is called.

19.3.3 Base

Description

A Viz object. This is the base class of all other Viz types.

Properties

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.
<i>NumChannels</i> as <i>Integer</i> [read-only]	Deprecated. Use <i>IsAnimated</i> or <i>GetChannelsOfObject</i> instead.

Member Procedures

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 <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$channelName" for the <i>channelName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$channelName".
Function FindKey FrameOfObject (Key FrameName as String) as Key Frame	Finds a Key Frame acting on the object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$Key FrameName" for the <i>Key FrameName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$Key FrameName".
Function FindOrCreateChannelOfObject (channelName as String) as Channel	This function works like FindChannelOfObject except that if no animation channel with the specified name exists, a new one is created.
Function GetChannelsOfObject ([out] v as Array[Channel]) as Integer	Fills the array <i>v</i> 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 <i>Director</i>), <i>v</i> 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 <i>Director</i>), it always returns <i>false</i> .
Sub SetChanged ()	Mark this object as changed.
Function <i>FindChannel</i> (channelName as <i>String</i>) as <i>Channel</i>	Deprecated. Use <i>FindChannelOfObject</i>
Function <i>FindKey Frame</i> (Key FrameName as <i>String</i>) as <i>Key Frame</i>	Deprecated. Use <i>FindKey FrameOfObject</i>
Function <i>GetChannels</i> ([out] v as Array[<i>Channel</i>]) as <i>Integer</i>	Deprecated. Use <i>GetChannelsOfObject</i>
Function <i>GetNumberOfAnimationChannels</i> () as <i>Integer</i>	Deprecated. Use <i>IsAnimated</i> instead.

19.3.4 BezierHandle

Base type

[Base](#)

Description

A spline handle of a Key Frame.

Properties

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]	(<i>Inherited from Base</i>) Gets the current scene.
Stage as Stage [read-only]	(<i>Inherited from Base</i>) Gets the current stage.
System as System [read-only]	(<i>Inherited from Base</i>) Gets system wide data.
VizCommunication as VizCommunication [read-only]	(<i>Inherited from Base</i>) Gets the VizCommunication object.
VizId as Integer [read-only]	(<i>Inherited from Base</i>) Gets or sets the internal id of this object.

Member Procedures

Name	Description
Function FindChannelOfObject (channelName as String) as Channel	(<i>Inherited from Base</i>) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> .

Name	Description
	You can limit the search to a particular director by using the syntax "directorName\$channelName" for the <i>channelName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$channelName".
Function FindKey FrameOfObject (Key FrameName as String) as Key Frame	<i>(Inherited from Base)</i> Finds a Key Frame acting on the object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$Key FrameName" for the <i>Key FrameName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$Key FrameName".
Function FindOrCreateChannelOfObject (channelName as String) as Channel	<i>(Inherited from Base)</i> This function works like FindChannelOfObject except that if no animation channel with the specified name exists, a new one is created.
Function GetChannelsOfObject ([out] v as Array[Channel]) as Integer	<i>(Inherited from Base)</i> Fills the array <i>v</i> 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 <i>Director</i>), <i>v</i> will be empty.
Function IsAnimated () as Boolean	<i>(Inherited from Base)</i> Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it always returns <i>false</i> .
Sub SetChanged ()	<i>(Inherited from Base)</i> Mark this object as changed.

19.3.5 Boolean

Description

A flag that is either true or false.

19.3.6 Camera

Base type

[Base](#)

Description

A camera object. Camera objects are returned by the Scene's CurrentCamera and GetCamera members.

Properties

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
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]	<i>(Inherited from Base)</i> Gets the current scene.

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.
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

19.3.7 Center

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 <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$channelName" for the <i>channelName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$channelName".
Function FindKey FrameOfObject (Key FrameName as String) as Key Frame	(Inherited from Base) Finds a Key Frame acting on the object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$Key FrameName" for the <i>Key FrameName</i> argument. Nested directors may be specified like this:

Name	Description
	"directorName1\$directorName2\$KeyFrameName".
Function FindOrCreateChannelOfObject (channelName as String) as Channel	(<i>Inherited from Base</i>) This function works like FindChannelOfObject except that if no animation channel with the specified name exists, a new one is created.
Function GetChannelsOfObject ([out] v as Array[Channel]) as Integer	(<i>Inherited from Base</i>) 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 <i>Director</i>), v will be empty.
Function IsAnimated () as Boolean	(<i>Inherited from Base</i>) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it always returns <i>false</i> .
Sub ResetPositionTracking ()	
Sub ResetRotationTracking ()	
Sub SetChanged ()	(<i>Inherited from Base</i>) Mark this object as changed.

19.3.8 Channel

Base type

[Base](#)

Description

A channel of a director.

Properties

Property	Description
Actor as Base [read-only]	
AnimationObject as Base [read-only]	
CommandName as String [read-only]	
Director as Director [read-only]	
FirstKey Frame as Key Frame [read-only]	Gets the left most Key Frame.
Key FrameCount as Integer [read-only]	Gets the number of Key Frames.

Property	Description
LastKey Frame as Key Frame [read-only]	Gets the right most Key Frame.
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.
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.
<i>NumKey Frames</i> as Integer [read-only]	<i>Deprecated. Use Key FrameCount .</i>

Member procedures

Name	Description
Function AddKey Frame (time as Double) as Key Frame	
Function CopyToDirector (destination as Director) as Boolean	
Sub Delete ()	
Function DeleteKey Frame (kf as Key Frame) as Boolean	
Function DeleteKey Frame (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 <i>Director</i>), it returns <i>null</i> .

Name	Description
	You can limit the search to a particular director by using the syntax "directorName\$channelName" for the <i>channelName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$channelName".
Function FindKey Frame (Key FrameName as String) as Key Frame	Searches for a Key Frame named Key FrameName.
Function FindKey FrameOfObject (Key FrameName as String) as Key Frame	<i>(Inherited from Base)</i> Finds a Key Frame acting on the object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$Key FrameName" for the <i>Key FrameName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$Key FrameName".
Function FindOrCreateChannelOfObject (channelName as String) as Channel	<i>(Inherited from Base)</i> This function works like FindChannelOfObject except that if no animation channel with the specified name exists, a new one is created.
Function GetChannelsOfObject ([out] v as Array[Channel]) as Integer	<i>(Inherited from Base)</i> 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 <i>Director</i>), v will be empty.
Function GetKey Frame (n as Integer) as Key Frame	Returns the n'th Key Frame of the channel.
Function IsAnimated () as Boolean	<i>(Inherited from Base)</i> Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), 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 ()	<i>(Inherited from Base)</i> Mark this object as changed.

19.3.9 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. See also examples section ClipChannel.

Properties

Property	Description
AutoRun as Boolean	Get/Set the autorun mode. If <i>true</i> 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 <i>In</i> frame.
FrameInActive as Integer	Get/Set the <i>In</i> frame on the active clip player.
FrameInPending as Integer	Get/Set the <i>In</i> frame on the pending clip player.
FrameOut as Integer	Get/Set the <i>Out</i> frame.
FrameOutActive as Integer	Get/Set the <i>Out</i> frame on the active clip player.
FrameOutPending as Integer	Get/Set the <i>Out</i> frame on the pending clip player.
LoopMode as Boolean	Get/Set the loop mode. If <i>true</i> (=default), then the clip will be looped.
LoopModeActive as Boolean	Get/Set the loop mode on the active clip player. If <i>true</i> (=default), then the clip will be looped.

Property	Description
LoopModePending as Boolean	Get/Set the loop mode on the pending clip player. If <i>true</i> (=default), then the clip will be looped.
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 <i>mode on load error</i> error flush.
ModeOnLoadError_NONE as Integer [read-only]	Constant for <i>mode on load error</i> none.
ModeOnLoadError_PAUSE as Integer [read-only]	Constant for <i>mode on load error</i> pause.
ModeOnLoadError_STOP as Integer [read-only]	Constant for <i>mode on load error</i> 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 <i>current mode</i> ERROR.
PlayMode_NONE as Integer [read-only]	Constant for <i>current mode</i> NONE.
PlayMode_PLAY as Integer [read-only]	Constant for <i>current mode</i> PLAYING.
PlayMode_PLAY_PENDING as Integer [read-only]	Constant for <i>current mode</i> PLAYING_PENDING.
PlayMode_SCRUB as Integer [read-only]	Constant for <i>current mode</i> SCRUBBING.
PlayMode_SCRUB_EOF as Integer [read-only]	Constant for <i>current mode</i> SCRUBBING_EOF.
PlayMode_STOP as Integer [read-only]	Constant for <i>current mode</i> STOP.
PlayModePending as Integer [read-only]	Get the current mode of the pending clip player.
RepeatMode as Integer	Get/Set the <i>repeat mode</i> for playback and pause. none field frame.

Property	Description
RepeatMode_FIELD as Integer [read-only]	Constant for <i>repeat mode</i> field.
RepeatMode_FRAME as Integer [read-only]	Constant for <i>repeat mode</i> frame.
RepeatMode_NONE as Integer [read-only]	Constant for <i>repeat mode</i> 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 <i>true</i> swap fields when playing interlaced material backwards.
ReverseFieldsActive as Boolean	If <i>true</i> swap fields when playing interlaced material backwards on the active clip player.
ReverseFieldsPending as Boolean	If <i>true</i> swap fields when playing interlaced material backwards on the pending clip player.
Scene as Scene [read-only]	(<i>Inherited from Base</i>) Gets the current scene.
Stage as Stage [read-only]	(<i>Inherited from Base</i>) Gets the current stage.
System as System [read-only]	(<i>Inherited from Base</i>) Gets system wide data.
UpscaleLuma as Boolean	If <i>true</i> then upscale the luma value in the separate key file.
UpscaleLumaActive as Boolean	If <i>true</i> then upscale the luma value in the separate key file on the active clip player.
UpscaleLumaPending as Boolean	If <i>true</i> then upscale the luma value in the separate key file on the pending clip player.
VizCommunication as VizCommunication [read-only]	(<i>Inherited from Base</i>) Gets the VizCommunication object.
VizId as Integer	(<i>Inherited from Base</i>) Gets or sets the internal id of this object.

19.3.10 Color

Description

A color object consisting of 4 members: red, green blue, alpha.

Properties

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

19.3.11 Container

Base type

[Base](#)

Description

A container object.

Properties

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.
Expert as Expert	The expert object
FirstChildContainer as Container [read-only]	The first child container
Geometry as Geometry	The geometry object

Property	Description
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 container.
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 sub-containers 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]	<i>(Inherited from Base)</i> 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
ScriptSettings as ScriptSettings [read-only]	The script's local settings
Stage as Stage [read-only]	<i>(Inherited from Base)</i> Gets the current stage.

Property	Description
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.
<i>NumChildContainers</i> as IntegerVizCursiv [read-only]	<i>Deprecated. Use ChildContainerCount instead.</i>
<i>Transform</i> as <i>Transform</i>	<i>Deprecated.</i>

Member procedures

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 <i>ScreenOffsetToContainerOffset</i> 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 cursor 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.
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	

Name	Description
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 ()	
Function FindChannelOfObject (channelName as String) as Channel	<p>(<i>Inherited from Base</i>) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i>.</p> <p>You can limit the search to a particular director by using the syntax "directorName\$channelName" for the <i>channelName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$channelName".</p>

Name	Description
Function FindKey FrameOfObject (Key FrameName as String) as Key Frame	<i>(Inherited from Base)</i> Finds a Key Frame acting on the object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$Key FrameName" for the <i>Key FrameName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$Key FrameName".
Function FindOrCreateChannelOfObject (channelName as String) as Channel	<i>(Inherited from Base)</i> This function works like FindChannelOfObject except that if no animation channel with the specified 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").
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 <i>v1_out</i> and <i>v2_out</i> 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	<i>(Inherited from Base)</i> Fills the array <i>v</i> 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 <i>Director</i>), <i>v</i> will be empty.

Name	Description
Function GetChildContainerByIndex (index as Integer) as Container	Returns the child container specified by <i>index</i> .
Function GetChildContainerContainingMouseCursor (<i>cursor</i>) as Container	Returns the first child container containing <i>cursor</i> 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.
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 IntegerVizCursiv	Returns the 0-based index of the container within its parent.
Sub GetScreenBounds (v1_out as VertexVizCursiv , v2_out as VertexVizCursiv)	Returns in <i>v1_out</i> and <i>v2_out</i> 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 IntegerVizCursiv , y as IntegerVizCursiv , pos as VertexVizCursiv) 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 IntegerVizCursiv , y as IntegerVizCursiv) 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.

Name	Description
Function GetScreenSizeOrtho (x as IntegerVizCursiv , y as IntegerVizCursiv) 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 <i>v1_out</i> and <i>v2_out</i> the left-bottom-back and right-top-front vertices of the container's bounding box with applied transformation matrix.
Function GetTransformedBoundingBoxDimensions (x as VertexVizCursiv)	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.)
Function GetUVCoordinatesOfScreenPoint (x as IntegerVizCursiv , y as IntegerVizCursiv) as VertexVizCursiv	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	<i>(Inherited from Base)</i> Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it always returns <i>false</i> .
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 IntegerVizCursiv , y as IntegerVizCursiv) 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 VertexVizCursiv) 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.)

Name	Description
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 VertexVizCursiv) as VertexVizCursiv	Converts the local position to a world coordinate space position.
Sub MoveAllChannelsToDirector (director as Director)	Moves the animation channels to the specified director.
Function MoveTo (destination as Container , where as IntegerVizCursiv) as Container	Moves the container; where = TL_NEXT TL_PREVIOUS TL_DOWN.
Function MoveToKeepTransformation (destination as Container , where as IntegerVizCursiv) 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-screen OffsetToContainerOffset (dx as IntegerVizCursiv , dy as IntegerVizCursiv) as VertexVizCursiv	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. <i>Position.Xyz</i>) to make the container move (<i>dx</i> , <i>dy</i>) pixels on the screen.
Function-screen PosToLocalGrid (dX as Double , dY as Double) as VertexVizCursiv	Converts the screen position to current grid intersection point. The returned coordinates are in local container coordinate space.
Function-screen PosToLocalPos (dX as Double , dY as Double) as VertexVizCursiv	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 VertexVizCursiv)	This procedure sets the container's center point without affecting its position.

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 ()	(<i>Inherited from Base</i>) Mark this object as changed.
Function SetScreenPositionOrtho (x as IntegerVizCursiv , y as IntegerVizCursiv) 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 IntegerVizCursiv , y as IntegerVizCursiv) 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 IntegerVizCursiv)	Makes the child container specified by <i>index</i> 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 VertexVizCursiv) as VertexVizCursiv	Converts the world position to a local coordinate space position.
<i>Function-screenOffsetToWorldOffset(dx as IntegerVizCursiv , dy as IntegerVizCursiv) as VertexVizCursiv</i>	<i>Deprecated.</i>

19.3.12 DateTime

Description

A type storing date/time information.

Properties

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)

Property	Description
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

Name	Description
Sub Normalize ()	Computes <i>DayOfWeek</i> and <i>DayOfYear</i> members and forces the other members to standard ranges.
Function ToString () as String	Returns a date+time string of the form <i>Wed Jan 02 02:03:55 1980</i> .

19.3.13 Director

Base type

[Base](#)

Description

A director in the stage.

Properties

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

Property	Description
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 IntegerVizCursiv [read-only]	Deprecated. Use <i>ActionChannelCount</i> .
NumAnimationChannels as Integer [read-only]	Deprecated. Call <i>GetAnimationChannels</i> to get a list of animation channels.

Member procedures

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 ChannelVizCursiv	Searches for an action channel named <i>channelName</i> .
Function FindChannel (channelName as String) as Channel	Searches for an animation channel named <i>channelName</i> .

Name	Description
Function FindChannelOfObject (channelName as String) as Channel	<i>(Inherited from Base)</i> Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$channelName" for the <i>channelName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$channelName".
Function FindKey Frame (Key FrameName as String) as Key Frame	Searches for a Key Frame named Key FrameName.
Function FindKey FrameOfObject (Key FrameName as String) as Key Frame	<i>(Inherited from Base)</i> Finds a Key Frame acting on the object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$Key FrameName" for the <i>Key FrameName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$Key FrameName".
Function FindOrCreateChannelOfObject (channelName as String) as Channel	<i>(Inherited from Base)</i> This function works like FindChannelOfObject except that if no animation channel with the specified name exists, a new one is created.
Function FindSubDirector (name as String) as Director	Searches for a sub-director named <i>name</i> .
Function GetActionChannel (index as Integer) as Channel	Returns the action channel at <i>index</i> .
Sub GetAnimationChannels ([out] v as Array[Channel])	Fills the specified array with the director's animation channels.
Function GetChannelsOfObject ([out] v as [Channel]) as Integer	<i>(Inherited from Base)</i> Fills the array <i>v</i> 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 <i>Director</i>), <i>v</i> will be empty.
Sub GetKey Frames ([out] v as Array[Key Frame])	Fills the specified array with the director's Key Frames.
Function HasAnimationChannels () as Boolean	Determines whether this director contains any animation channels (not counting channels in sub-directors).

Name	Description
Function IsAnimated() as Boolean	<i>(Inherited from Base)</i> Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it always returns <i>false</i> .
Function IsAnimationRunning() as Boolean	Returns true if the animation of this director is active.
Sub MoveTo (destination as Director , where as Integer)	Moves a director; where = TL_NEXT TL_PREVIOUS TL_DOWN
Sub SetChanged()	<i>(Inherited from Base)</i> 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.
<i>Function GetAnimationChannel(index as IntegerVizCursiv) as ChannelVizCursiv</i>	<i>Deprecated. Use GetAnimationChannels instead.</i>

19.3.14 Double

Description

A double precision floating-point number (64 bits).

19.3.15 Expert

Base type

[Base](#)

Description

A container's Expert property. This type provides access to the parameters of the Expert plugin.

Properties

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.
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]	<i>(Inherited from Base)</i> 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]	<i>(Inherited from Base)</i> Gets the current stage.
System as System [read-only]	<i>(Inherited from Base)</i> Gets system wide data.
TransparencyMode as Integer	Get/Set transparency 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]	<i>(Inherited from Base)</i> Gets the VizCommunication object.
VizId as Integer	<i>(Inherited from Base)</i> Gets or sets the internal id of this object.

Property	Description
Wireframe as Boolean	Get/Set drawing in wireframe mode.
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

Name	Description
Function FindChannelOfObject(channelName as String) as Channel	<i>(Inherited from Base)</i> Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$channelName" for the <i>channelName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$channelName".
Function FindKey FrameOfObject(Key FrameName as String) as Key Frame	<i>(Inherited from Base)</i> Finds a Key Frame acting on the object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$Key FrameName" for the <i>Key FrameName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$Key FrameName".
Function FindOrCreateChannelOfObject(channelName as String) as Channel	<i>(Inherited from Base)</i> This function works like FindChannelOfObject except that if no animation channel with the specified name exists, a new one is created.
Function GetChannelsOfObject([out] v as Array[Channel]) as Integer	<i>(Inherited from Base)</i> Fills the array <i>v</i> 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 <i>Director</i>), <i>v</i> will be empty.
Function IsAnimated() as Boolean	<i>(Inherited from Base)</i> Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it always returns <i>false</i> .

Name	Description
Sub SetChanged()	(Inherited from Base) Mark this object as changed.

19.3.16 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

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).
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)

Property	Description
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]	(<i>Inherited from Base</i>) Gets the VizCommunication object.
VizId as Integer	(<i>Inherited from Base</i>) Gets or sets the internal id of this object.
WordSpacing as Double	The word spacing

Member procedures

Name	Description
Function FindChannelOfObject (channelName as String) as Channel	(<i>Inherited from Base</i>) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$channelName" for the <i>channelName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$channelName".
Function FindKey FrameOfObject (Key FrameName as String) as Key Frame	(<i>Inherited from Base</i>) Finds a Key Frame acting on the object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$Key FrameName" for the <i>Key FrameName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$Key FrameName".
Function FindOrCreateChannelOfObject (channelName as String) as Channel	(<i>Inherited from Base</i>) This function works like FindChannelOfObject except that if no animation channel with the specified name exists, a new one is created.
Function GetChannelsOfObject ([out] v as Array[Channel]) as Integer	(<i>Inherited from Base</i>) 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

Name	Description
	cannot be animated (such as <i>Director</i>), <i>v</i> 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	<i>(Inherited from Base)</i> Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it always returns <i>false</i> .
Sub RegisterChangedCallback ()	Causes <i>OnGeometryChanged</i> to be invoked whenever the geometry changes.
Sub RegisterTextChangedCallback ()	Causes <i>OnGeometryChanged</i> to be invoked whenever the geometry's text changes.
Sub SetChanged ()	<i>(Inherited from Base)</i> 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.

Name	Description
Sub UnregisterChangedCallback()	Removes the binding set up by <i>RegisterChangedCallback</i> .

19.3.17 Grid

Base type

[Base](#)

Description

A grid object.

Properties

Property	Description
Locked as Boolean	Gets or sets grid lock.
Scene as Scene [read-only]	(<i>Inherited from Base</i>) Gets the current scene.
Stage as Stage [read-only]	(<i>Inherited from Base</i>) Gets the current stage.
System as System [read-only]	(<i>Inherited from Base</i>) 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]	(<i>Inherited from Base</i>) Gets the VizCommunication object.
VizId as Integer	(<i>Inherited from Base</i>) Gets or sets the internal id of this object.

Member procedures

Name	Description
Function FindChannelOfObject (channelName as String) as Channel	(<i>Inherited from Base</i>) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> .

Name	Description
	You can limit the search to a particular director by using the syntax "directorName\$channelName" for the <i>channelName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$channelName".
Function FindKey FrameOfObject (Key FrameName as String) as Key Frame	<i>(Inherited from Base)</i> Finds a Key Frame acting on the object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$Key FrameName" for the <i>Key FrameName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$Key FrameName".
Function FindOrCreateChannelOfObject (channelName as String) as Channel	<i>(Inherited from Base)</i> This function works like FindChannelOfObject except that if no animation channel with the specified name exists, a new one is created.
Function GetChannelsOfObject ([out] v as Array[Channel]) as Integer	<i>(Inherited from Base)</i> 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 <i>Director</i>), v will be empty.
Function IsAnimated () as Boolean	<i>(Inherited from Base)</i> Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it always returns <i>false</i> .
Sub MoveToContainer (container as Container)	Moves grid to the specified container and aligns it.
Sub SetChanged ()	<i>(Inherited from Base)</i> Mark this object as changed.

19.3.18 Image

Base type

[Base](#)

Description

An image object. This is used to retrieve an image's properties.

Properties

Property	Description
Name as String [read-only]	The name of the image
Scene as Scene [read-only]	(<i>Inherited from Base</i>) Gets the current scene.
Stage as Stage [read-only]	(<i>Inherited from Base</i>) Gets the current stage.
System as System [read-only]	(<i>Inherited from Base</i>) Gets system wide data.
Uuid as Uuid [read-only]	The UUID of the image
VizCommunication as VizCommunication [read-only]	(<i>Inherited from Base</i>) Gets the VizCommunication object.
VizId as Integer	(<i>Inherited from Base</i>) Gets or sets the internal id of this object.

Member procedures

Name	Description
Function FindChannelOfObject (channelName as String) as Channel	(<i>Inherited from Base</i>) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$channelName" for the <i>channelName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$channelName".
Function FindKey FrameOfObject (Key FrameName as String) as Key Frame	(<i>Inherited from Base</i>) Finds a Key Frame acting on the object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$Key FrameName" for the <i>Key FrameName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$Key FrameName".
Function FindOrCreateChannelOfObject (channelName as String) as Channel	(<i>Inherited from Base</i>) This function works like FindChannelOfObject except that if no animation channel with the specified name exists, a new one is created.

Name	Description
Function GetChannelsOfObject ([out] v as Array[Channel]) as Integer	(<i>Inherited from Base</i>) 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 <i>Director</i>), v will be empty.
Function IsAnimated () as Boolean	(<i>Inherited from Base</i>) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it always returns <i>false</i> .
Sub SetChanged ()	(<i>Inherited from Base</i>) Mark this object as changed.

19.3.19 InfoText

Base type

[Base](#)

Description

A container's InfoText property. This type provides access to the parameters of the InfoText plugin.

Properties

Property	Description
Scene as Scene [read-only]	(<i>Inherited from Base</i>) Gets the current scene.
Stage as Stage [read-only]	(<i>Inherited from Base</i>) Gets the current stage.
System as System [read-only]	(<i>Inherited from Base</i>) Gets system wide data.
Text as String	Get/Set info text.
VizCommunication as VizCommunication [read-only]	(<i>Inherited from Base</i>) Gets the VizCommunication object.
VizId as Integer	(<i>Inherited from Base</i>) Gets or sets the internal id of this object.

Name	Description
Function FindChannelOfObject (channelName as String) as Channel	(<i>Inherited from Base</i>) Finds an animation channel acting on this object. If this function is called on a type that cannot

Name	Description
	<p>be animated (such as <i>Director</i>), it returns <i>null</i>.</p> <p>You can limit the search to a particular director by using the syntax "directorName\$channelName" for the <i>channelName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$channelName".</p>
Function FindKey FrameOfObject (Key FrameName as String) as Key Frame	<p>(<i>Inherited from Base</i>) Finds a Key Frame acting on the object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i>.</p> <p>You can limit the search to a particular director by using the syntax "directorName\$Key FrameName" for the <i>Key FrameName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$Key FrameName".</p>
Function FindOrCreateChannelOfObject (channelName as String) as Channel	<p>(<i>Inherited from Base</i>) This function works like FindChannelOfObject except that if no animation channel with the specified name exists, a new one is created.</p>
Function GetChannelsOfObject ([out] v as Array[Channel]) as Integer	<p>(<i>Inherited from Base</i>) Fills the array <i>v</i> 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 <i>Director</i>), <i>v</i> will be empty.</p>
Function IsAnimated () as Boolean	<p>(<i>Inherited from Base</i>) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it always returns <i>false</i>.</p>
Sub SetChanged ()	<p>(<i>Inherited from Base</i>) Mark this object as changed.</p>

19.3.20 Integer

Description

A 32-bit signed integer.

19.3.21 Key

Base type

[Base](#)

Description

A key object. This is used to control a container's key object.

Properties

Property	Description
Active as Boolean	
AlphaValue as Double	
AutoAlpha as Boolean	
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

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 <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$channelName" for the <i>channelName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$channelName".
Function FindKey FrameOfObject (Key FrameName as String) as Key Frame	(Inherited from Base) Finds a Key Frame acting on the object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$Key FrameName" for the <i>Key FrameName</i> argument. Nested

Name	Description
	directors may be specified like this: "directorName1\$directorName2\$Key FrameName".
Function FindOrCreateChannelOfObject (channelName as String) as Channel	(<i>Inherited from Base</i>) This function works like FindChannelOfObject except that if no animation channel with the specified name exists, a new one is created.
Function GetChannelsOfObject ([out] v as Array[Channel]) as Integer	(<i>Inherited from Base</i>) 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 <i>Director</i>), v will be empty.
Function IsAnimated () as Boolean	(<i>Inherited from Base</i>) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it always returns <i>false</i> .
Sub SetChanged ()	(<i>Inherited from Base</i>) Mark this object as changed.

19.3.22 Key Frame

Base type

[Base](#)

Description

An animation Key Frame.

Properties

Property	Description
ActionString as String	Gets or sets the command string of an action Key Frame.
BoolValue as Boolean	Gets or sets the value of a boolean Key Frame.
Channel as Channel [read-only]	Gets the channel that contains this Key Frame.
ColorValue as Color	Gets or sets the value of a color Key Frame.
EventDirection as Integer	Get or sets the direction of an event Key Frame. Possible values are: ED_BOTH, ED_NORMAL, ED_REVERSE

Property	Description
EventDuration as Double	Get or sets the duration of an event Key Frame (in seconds).
EventValue as Integer	Get or sets the type of an event Key Frame. Possible values are: ET_STOP, ET_LOCAL_STOP, ET_TAG.
FloatValue as Double	Gets or sets the value of a float Key Frame.
IntValue as Integer	Gets or sets the value of an integer Key Frame.
Locked as Boolean	Gets or sets the locked status of a path point Key Frame.
MaterialValue as Material [read-only]	Gets or sets the value of a material Key Frame.
Name as String	Gets or sets the Key Frame's name.
NextKey Frame as Key Frame [read-only]	Gets the next Key Frame.
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 Key Frame. If not, Null is returned.
PreviousKey Frame as Key Frame [read-only]	Gets the previous Key Frame.
Scene as Scene [read-only]	<i>(Inherited from Base)</i> Gets the current scene.
Stage as Stage [read-only]	<i>(Inherited from Base)</i> Gets the current stage.
System as System [read-only]	<i>(Inherited from Base)</i> Gets system wide data.
Time as Double	Gets or sets the Key Frame's time (in seconds).
VizCommunication as VizCommunication [read-only]	<i>(Inherited from Base)</i> Gets the VizCommunication object.
VizId as Integer	<i>(Inherited from Base)</i> Gets or sets the internal id of this object.
XyzValue as Vertex	Gets or sets the value of a a rotation or scaling Key Frame or a path point Key Frame.
<i>NumAxes</i> as Integer [read-only]	<i>Deprecated. Use NumberOfAxes.</i>

Member procedures

Name	Description
Function Delete() as Boolean	Deletes the Key Frame.
Function FindChannelOfObject (channelName as String) as Channel	<i>(Inherited from Base)</i> Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$channelName" for the <i>channelName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$channelName".
Function FindKey FrameOfObject (Key FrameName as String) as Key Frame	<i>(Inherited from Base)</i> Finds a Key Frame acting on the object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$Key FrameName" for the <i>Key FrameName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$Key FrameName".
Function FindOrCreateChannelOfObject (channelName as String) as Channel	<i>(Inherited from Base)</i> This function works like <code>FindChannelOfObject</code> except that if no animation channel with the specified name exists, a new one is created.
Function GetBezierHandle (axis as Integer) as BezierHandle	Returns the bezier handle for the specified axis. The <i>axis</i> parameter is only used if the Key Frame has multiple splines, which is the case for rotation, scaling, and material Key Frames. When called on a position Key Frame, this procedure returns the handle for the distance spline. To get the path handle, use the <i>PathHandle</i> property instead.
Function GetChannelsOfObject ([out] v as Array[Channel]) as Integer	<i>(Inherited from Base)</i> Fills the array <i>v</i> 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 <i>Director</i>), <i>v</i> will be empty.
Function IsAnimated () as Boolean	<i>(Inherited from Base)</i> Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it always returns <i>false</i> .

Name	Description
Sub SetChanged()	(Inherited from Base) Mark this object as changed.
Sub UpdateAnimation()	Update animation to reflect changes to Key Frame. (This update isn't performed in On Air mode where the update is done as soon as the animation is started.)

19.3.23 Light

Base type

[Base](#)

Description

A light object. Light objects are returned by the Scene's *GetLight* member.

Properties

Property	Description
Attenuation as Double	
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

Property	Description
VizCommunication as VizCommunication [read-only]	(<i>Inherited from Base</i>) Gets the VizCommunication object.
VizId as Integer	(<i>Inherited from Base</i>) Gets or sets the internal id of this object.

Member procedures

Name	Description
Function FindChannelOfObject (channelName as String) as Channel	(<i>Inherited from Base</i>) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$channelName" for the <i>channelName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$channelName".
Function FindKey FrameOfObject (Key FrameName as String) as Key Frame	(<i>Inherited from Base</i>) Finds a Key Frame acting on the object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$Key FrameName" for the <i>Key FrameName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$Key FrameName".
Function FindOrCreateChannelOfObject (channelName as String) as Channel	(<i>Inherited from Base</i>) This function works like FindChannelOfObject except that if no animation channel with the specified name exists, a new one is created.
Function GetChannelsOfObject ([out] v as Array[Channel]) as Integer	(<i>Inherited from Base</i>) Fills the array <i>v</i> 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 <i>Director</i>), <i>v</i> will be empty.
Function IsAnimated () as Boolean	(<i>Inherited from Base</i>) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it always returns <i>false</i> .
Sub ResetPositionTracking ()	

Name	Description
Sub ResetRotationTracking()	
Sub SetChanged()	(<i>Inherited from Base</i>) Mark this object as changed.

19.3.24 Material

Base type

[Base](#)

Description

A material object. This is used to control a container's material.

Properties

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]	(<i>Inherited from Base</i>) Gets the current scene.
Shininess as Double	The shininess [0.0, 1.0]
Specular as Color	The specular color
Stage as Stage [read-only]	(<i>Inherited from Base</i>) Gets the current stage.
System as System [read-only]	(<i>Inherited from Base</i>) Gets system wide data.
VizCommunication as VizCommunication [read-only]	(<i>Inherited from Base</i>) Gets the VizCommunication object.
VizId as Integer	(<i>Inherited from Base</i>) Gets or sets the internal id of this object.

Member procedures

Name	Description
Function FindChannelOfObject (channelName as String) as Channel	<i>(Inherited from Base)</i> Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$channelName" for the <i>channelName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$channelName".
Function FindKey FrameOfObject (Key FrameName as String) as Key Frame	<i>(Inherited from Base)</i> Finds a Key Frame acting on the object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$Key FrameName" for the <i>Key FrameName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$Key FrameName".
Function FindOrCreateChannelOfObject (channelName as String) as Channel	<i>(Inherited from Base)</i> This function works like FindChannelOfObject except that if no animation channel with the specified name exists, a new one is created.
Function GetChannelsOfObject ([out] v as Array[Channel]) as Integer	<i>(Inherited from Base)</i> Fills the array <i>v</i> 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 <i>Director</i>), <i>v</i> will be empty.
Function IsAnimated () as Boolean	<i>(Inherited from Base)</i> Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it always returns <i>false</i> .
Sub SetChanged ()	<i>(Inherited from Base)</i> Mark this object as changed.

19.3.25 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

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)	
Sub Transpose ()	

19.3.26 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

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]	<i>(Inherited from Base)</i> 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]	<i>(Inherited from Base)</i> Gets the current stage.
System as System [read-only]	<i>(Inherited from Base)</i> Gets system wide data.
VizCommunication as VizCommunication [read-only]	<i>(Inherited from Base)</i> Gets the VizCommunication object.
VizId as Integer	<i>(Inherited from Base)</i> Gets or sets the internal id of this object.

Member procedures

Name	Description
Function FindChannelOfObject (channelName as String) as Channel	<i>(Inherited from Base)</i> Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> .

Name	Description
	You can limit the search to a particular director by using the syntax "directorName\$channelName" for the <i>channelName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$channelName".
Function FindKey FrameOfObject (Key FrameName as String) as Key Frame	<i>(Inherited from Base)</i> Finds a Key Frame acting on the object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$Key FrameName" for the <i>Key FrameName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$Key FrameName".
Function FindOrCreateChannelOfObject (channelName as String) as Channel	<i>(Inherited from Base)</i> This function works like FindChannelOfObject except that if no animation channel with the specified name exists, a new one is created.
Function GetChannelsOfObject ([out] v as Array[Channel]) as Integer	<i>(Inherited from Base)</i> Fills the array <i>v</i> 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 <i>Director</i>), <i>v</i> 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	<i>(Inherited from Base)</i> Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it always returns <i>false</i> .

Name	Description
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 <i>MTRegister</i> 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 <i>parName</i> .
Sub SetChanged ()	(<i>Inherited from Base</i>) Mark this object as changed.
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.

19.3.27 Position

Base type

[Base](#)

Description

A position object. This is used to control the position of a container, camera, or light.

Properties

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

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 <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$channelName" for the <i>channelName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$channelName".
Function FindKey FrameOfObject (Key FrameName as String) as Key Frame	(Inherited from Base) Finds a Key Frame acting on the object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$Key FrameName" for the <i>Key FrameName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$Key FrameName".

Name	Description
Function FindOrCreateChannelOfObject (channelName as String) as Channel	(<i>Inherited from Base</i>) This function works like FindChannelOfObject except that if no animation channel with the specified name exists, a new one is created.
Function GetChannelsOfObject ([out] v as Array[Channel]) as Integer	(<i>Inherited from Base</i>) Fills the array <i>v</i> 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 <i>Director</i>), <i>v</i> will be empty.
Function IsAnimated () as Boolean	(<i>Inherited from Base</i>) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it always returns <i>false</i> .
Sub SetChanged ()	(<i>Inherited from Base</i>) Mark this object as changed.

19.3.28 Rotation

Base type

[Base](#)

Description

A rotation object. This is used to control the position of a container or light.

Properties

Property	Description
Scene as Scene [read-only]	(<i>Inherited from Base</i>) Gets the current scene.
Stage as Stage [read-only]	(<i>Inherited from Base</i>) Gets the current stage.
System as System [read-only]	(<i>Inherited from Base</i>) Gets system wide data.
VizCommunication as VizCommunication [read-only]	(<i>Inherited from Base</i>) Gets the VizCommunication object.
VizId as Integer	(<i>Inherited from Base</i>) 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.

Property	Description
y as Double	Gets or sets the y component.
z as Double	Gets or sets the z component.

Member procedures

Name	Description
Function FindChannelOfObject (channelName as String) as Channel	<i>(Inherited from Base)</i> Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$channelName" for the <i>channelName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$channelName".
Function FindKey FrameOfObject (Key FrameName as String) as Key Frame	<i>(Inherited from Base)</i> Finds a Key Frame acting on the object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$Key FrameName" for the <i>Key FrameName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$Key FrameName".
Function FindOrCreateChannelOfObject (channelName as String) as Channel	<i>(Inherited from Base)</i> This function works like FindChannelOfObject except that if no animation channel with the specified name exists, a new one is created.
Function GetChannelsOfObject ([out] v as Array[Channel]) as Integer	<i>(Inherited from Base)</i> Fills the array <i>v</i> 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 <i>Director</i>), <i>v</i> will be empty.
Function IsAnimated () as Boolean	<i>(Inherited from Base)</i> Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it always returns <i>false</i> .
Sub SetChanged ()	<i>(Inherited from Base)</i> Mark this object as changed.

19.3.29 Scaling

Base type[Base](#)**Description**

A scaling object. This is used to control the scaling of a container.

Properties

Property	Description
Scene as Scene [read-only]	(<i>Inherited from Base</i>) Gets the current scene.
Stage as Stage [read-only]	(<i>Inherited from Base</i>) Gets the current stage.
System as System [read-only]	(<i>Inherited from Base</i>) Gets system wide data.
VizCommunication as VizCommunication [read-only]	(<i>Inherited from Base</i>) Gets the VizCommunication object.
VizId as Integer	(<i>Inherited from Base</i>) 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

Name	Description
Function FindChannelOfObject (channelName as String) as Channel	(<i>Inherited from Base</i>) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$channelName" for the <i>channelName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$channelName".
Function FindKey FrameOfObject (Key FrameName as String) as Key Frame	(<i>Inherited from Base</i>) Finds a Key Frame acting on the object. If this function is

Name	Description
	called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$Key FrameName" for the <i>Key FrameName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$Key FrameName".
Function FindOrCreateChannelOfObject (channelName as String) as Channel	<i>(Inherited from Base)</i> This function works like <code>FindChannelOfObject</code> except that if no animation channel with the specified name exists, a new one is created.
Function GetChannelsOfObject ([out] v as Array[Channel]) as Integer	<i>(Inherited from Base)</i> Fills the array <i>v</i> 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 <i>Director</i>), <i>v</i> will be empty.
Function IsAnimated () as Boolean	<i>(Inherited from Base)</i> Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it always returns <i>false</i> .
Sub SetChanged ()	<i>(Inherited from Base)</i> Mark this object as changed.

19.3.30 Scene

Base type

[Base](#)

Description

A scene object. The `Scene` member of the `Base` type provides access to the current scene.

Properties

Property	Description
AlphaClearValue as Integer	
AutoKeyActive as Boolean	
AutoKeyPosition as Vertex	
BackgroundColor as Color	

Property	Description
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 Shared Memory [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]	<i>(Inherited from Base)</i> Gets the current scene.
Script as Script [read-only]	The script object: it can be used to dynamically call member procedures of the scene
ScriptPluginInstance as PluginInstance [read-only]	The script's plugin instance
Stage as Stage [read-only]	<i>(Inherited from Base)</i> Gets the current stage.
System as System [read-only]	<i>(Inherited from Base)</i> Gets system wide data.
Uuid as Uuid [read-only]	The UUID of the scene
VizCommunication as VizCommunication [read-only]	<i>(Inherited from Base)</i> Gets the VizCommunication object.
VizId as Integer	<i>(Inherited from Base)</i> Gets or sets the internal id of this object.

Member procedures

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	<i>(Inherited from Base)</i> Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax

Name	Description
	"directorName\$channelName" for the <i>channelName</i> 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 <code>Container.FindSubContainer</code> into a single call by concatenating the individual container names using \$ as a separator. For example, <code>scene.FindContainer("name1\$name2")</code> has the same effect as <code>scene.FindContainer("name1").FindSubContainer("name2")</code>
Function FindKey FrameOfObject (Key FrameName as String) as Key Frame	<i>(Inherited from Base)</i> Finds a Key Frame acting on the object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$Key FrameName" for the <i>Key FrameName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$Key FrameName".
Function FindOrCreateChannelOfObject (channelName as String) as Channel	<i>(Inherited from Base)</i> This function works like <code>FindChannelOfObject</code> except that if no animation channel with the specified name exists, a new one is created.
Function GetCamera (cameraIndex as Integer) as Camera	
Function GetChannelsOfObject ([out] v as Array[Channel]) as Integer	<i>(Inherited from Base)</i> Fills the array <i>v</i> 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 <i>Director</i>), <i>v</i> 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.

Name	Description
Function IsAnimated() as Boolean	<i>(Inherited from Base)</i> Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it always returns <i>false</i> .
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
Function IsMainlayer() as Boolean	Returns true, if the script is executed within Mainlayer
Function-screen PosToWorldGrid (dX as Double , dY as Double) as Vertex	Converts the screen position to current grid intersection point.
Function-screen PosToWorldPos (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()	<i>(Inherited from Base)</i> 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.

19.3.31 Script

Description

The container's script object. This is used to dynamically call member procedures of another container.

19.3.32 ScriptSettings

Description

A script's local settings.

Properties

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 = <i>true</i> .
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 = <i>true</i> .
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 = <i>false</i> .
<i>CallEnterLeaveForBackgroundObjects</i> as Boolean	<i>Deprecated. Use CallEnterLeaveForTopObjectsOnly instead.</i>
<i>CallMouseMoveForBackgroundObjects</i> as Boolean	<i>Deprecated. Use CallMouseMoveForTopObjectsOnly instead.</i>

19.3.33 Shared Memory

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

Property	Description
Scene as Scene [read-only]	<i>(Inherited from Base)</i> Gets the current scene.
Stage as Stage [read-only]	<i>(Inherited from Base)</i> Gets the current stage.
System as System [read-only]	<i>(Inherited from Base)</i> Gets system wide data.

Property	Description
VizCommunication as VizCommunication [read-only]	(<i>Inherited from Base</i>) Gets the VizCommunication object.
VizId as Integer	(<i>Inherited from Base</i>) Gets or sets the internal id of this object.

Member procedures

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	(<i>Inherited from Base</i>) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$channelName" for the <i>channelName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$channelName".
Function FindKey FrameOfObject (Key FrameName as String) as Key Frame	(<i>Inherited from Base</i>) Finds a Key Frame acting on the object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$Key FrameName" for the <i>Key FrameName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$Key FrameName".
Function FindOrCreateChannelOfObject (channelName as String) as Channel	(<i>Inherited from Base</i>) This function works like FindChannelOfObject except that if no animation channel with the specified name exists, a new one is created.
Function GetChannelsOfObject ([out] v as Array[Channel]) as Integer	(<i>Inherited from Base</i>) Fills the array <i>v</i> 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 <i>Director</i>), <i>v</i> will be empty.

Name	Description
Sub GetKeys (keys as Array[String])	Fills the passed array with all defined keys.
Function IsAnimated () as Boolean	<i>(Inherited from Base)</i> Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it always returns <i>false</i> .
Function LoadFromDb (path as String) as Boolean	Loads the contents of the map from a database file. The file is specified by <i>path</i> , which must be an absolute path, such as <i>"/User/mem_map"</i> .
Sub RegisterChangedCallback (key as String)	Causes <i>OnSharedMemoryVariableChanged</i> to be invoked whenever the variable identified by <i>key</i> is changed and <i>OnSharedMemoryVariableDeleted</i> in response to the key being deleted. If you pass an empty string for <i>key</i> , the callbacks will be invoked whenever <i>any</i> variable in the map is changed or deleted.
Function SaveToDb (path as String) as Boolean	Saves the contents of the map into a database file. The file is specified by <i>path</i> , which must be an absolute <i>path</i> , such as <i>"/User/mem_map"</i> .
Sub SetChanged ()	<i>(Inherited from Base)</i> Mark this object as changed.
Sub UnregisterChangedCallback (key as String)	Removes the binding set up by <i>RegisterChangedCallback</i> .

19.3.34 Stage

Base type

[Base](#)

Description

A stage object. The *Stage* member of the *Base* type provides access to the current scene's stage.

Properties

Property	Description
RootDirector as Director [read-only]	Gets the topmost director in the stage.
Scene as Scene [read-only]	<i>(Inherited from Base)</i> Gets the current scene.

Property	Description
Stage as Stage [read-only]	(<i>Inherited from Base</i>) Gets the current stage.
System as System [read-only]	(<i>Inherited from Base</i>) Gets system wide data.
VizCommunication as VizCommunication [read-only]	(<i>Inherited from Base</i>) Gets the VizCommunication object.
VizId as Integer	(<i>Inherited from Base</i>) Gets or sets the internal id of this object.

Member procedures

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	(<i>Inherited from Base</i>) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$channelName" for the <i>channelName</i> 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 FindKey FrameOfObject (Key FrameName as String) as Key Frame	(<i>Inherited from Base</i>) Finds a Key Frame acting on the object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$Key FrameName" for the <i>Key FrameName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$Key FrameName".
Function FindOrCreateChannelOfObject (channelName as String) as Channel	(<i>Inherited from Base</i>) This function works like FindChannelOfObject except that if no animation channel with the specified name exists, a new one is created.

Name	Description
Function GetChannelsOfObject ([out] v as Array[Channel]) as Integer	(<i>Inherited from Base</i>) 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 <i>Director</i>), v will be empty.
Function IsAnimated () as Boolean	(<i>Inherited from Base</i>) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it always returns <i>false</i> .
Function IsAnimationRunning () as Boolean	Returns true if the animation active.
Sub SetChanged ()	(<i>Inherited from Base</i>) Mark this object as changed.
Sub StartAnimation ()	Starts the animation in the directors selected in the director tree.
Sub StopAnimation ()	Stops the animation in all directors.

19.3.35 String

Description

A character string. Use the '&' operator to concatenate strings.

Properties

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 proper length, you have to convert it using <code>Utf8ToAnsi()</code>

Member procedures

Name	Description
Sub AnsiToUtf8 ()	Converts an ANSI encoded string (using the system code-page) to UTF-8.

Name	Description
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 a 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 in the string
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'

Name	Description
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: <pre>dim a as string = "Monday Tuesday Wednesday" dim b as Array[String] a.split(" ",b) b will be ["Monday", "Tuesday", "Wednesday"]</pre>
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 occurrences of 'regex' are replaced. Otherwise, only the first occurrence 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 whitespace characters from both ends of the string
Sub Utf8ToAnsi ()	Converts a UTF-8 encoded string to ANSI (using the system code-page).

19.3.36 StringMap

Description

An associative array that maps strings to Variant objects. Use the index operator [] to access individual elements.

Member procedures

Property	Description
Sub Clear ()	

Property	Description
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.

19.3.37 System

Base type

[Base](#)

Description

This type provides access to system-wide parameters and procedures.

Properties

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.
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 Shared Memory [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).

Property	Description
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.
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.

Property	Description
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]	<i>(Inherited from Base)</i> Gets the current stage.
System as System [read-only]	<i>(Inherited from Base)</i> 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]	<i>(Inherited from Base)</i> Gets the VizCommunication object.
VizId as Integer	<i>(Inherited from Base)</i> Gets or sets the internal id of this object.
<i>RefreshRate</i> as DoubleVizCursiv [read-only]	<i>Deprecated.</i>

Member procedures

Name	Description
Function DeleteFile (filename as String) as Boolean	Deletes the file <i>filename</i> , 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 <i>activate</i> 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	<i>(Inherited from Base)</i> Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$channelName" for the <i>channelName</i> argument. Nested

Name	Description
	directors may be specified like this: "directorName1\$directorName2\$channelName".
Function FindKey FrameOfObject (Key FrameName as String) as Key Frame	<i>(Inherited from Base)</i> Finds a Key Frame acting on the object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$Key FrameName" for the <i>Key FrameName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$Key FrameName".
Function FindOrCreateChannelOfObject (channelName as String) as Channel	<i>(Inherited from Base)</i> This function works like FindChannelOfObject except that if no animation channel with the specified name exists, a new one is created.
Function GetChannelsOfObject ([out] v as Array[Channel]) as Integer	<i>(Inherited from Base)</i> Fills the array <i>v</i> 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 <i>Director</i>), <i>v</i> 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 <i>dirname</i> matching the <i>pattern</i> , 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 <i>device</i> with <i>type</i> .
Function GetTimecodeByIndex (Index as Integer) as Timecode	Returns a timecode object by its <i>index</i> .
Function IsAnimated () as Boolean	<i>(Inherited from Base)</i> Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it always returns <i>false</i> .

Name	Description
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 <i>filepath</i> into <i>result</i> . Returns false if the file does not exist or is not accessible.
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	Undefine the VizMouseServer with the address <i>serverAddress</i> , the server port <i>serverPort</i> and the local listening port <i>localPort</i>
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

Name	Description
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 <i>activate</i> is false, the application loading the file is minimized.
Sub PlaySound (filename as String)	Plays a sound file.
Function SaveTextFile (filepath as String , text as String) as Boolean	Saves the contents of <i>text</i> into the file identified by <i>filepath</i> . 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 <i>timeoutInMs</i> > 0, the procedure waits at most <i>timeoutInMs</i> milliseconds for an answer, and subsequently returns that answer. If <i>timeoutInMs</i> = 0, the procedure returns immediately, ignoring any answer.
Sub SetChanged ()	<i>(Inherited from Base)</i> Mark this object as changed.
Function StartApplication (filename as String , parameters as String , activate as Boolean) as Boolean	Launches an application. If <i>activate</i> is false, the application window is minimized.
Sub StopSound ()	Stops sound playback. See <i>PlaySound</i> .
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 <i>SendRemoteCommand</i> , this procedure sends the command string as is, that is without a Viz command identifier attached. The <i>timeoutInMs</i> has the same meaning as for <i>SendRemoteCommand</i> .
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 <i>TcpSend</i> , this procedure works asynchron and the return result is stored in a shared memory map, at your defined <i>sharedMemoryKey</i> . The defined <i>sharedMemoryKey</i> or its RegisterChangedCallback will be searched in the maps in the following

Name	Description
	order: <i>Scene.Map</i> , <i>System.Map</i> and <i>VizCommunication.Map</i> If <i>timeoutInMs</i> is set to 0 then <i>TcpSendAsync</i> will wait until a answer is received or as long as no error occurs (e.g. host doesn't exist).
Sub TcpSendAsyncStatus (sharedMemoryKey as String , hostname as String , port as Integer , commandString as String , timeoutInMs as Integer , sharedMemoryKeyStatus as String)	Same as <i>TcpSendAsync</i> but with additional status Shared Memory Key which informs you about the transmission process. The following Messages (Strings) will be returned in <i>sharedMemoryKeyStatus</i> : SENDING RECEIVED ERROR TIMEOUT The <i>timeoutInMs</i> has the same meaning as for <i>TcpSendAsync</i> .

19.3.38 Texture

Base type

[Base](#)

Description

A container's texture object.

Properties

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

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]	<i>(Inherited from Base)</i> 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]	<i>(Inherited from Base)</i> Gets the current stage.
System as System [read-only]	<i>(Inherited from Base)</i> Gets system wide data.
VizCommunication as VizCommunication [read-only]	<i>(Inherited from Base)</i> Gets the VizCommunication object.
VizId as Integer	<i>(Inherited from Base)</i> Gets or sets the internal id of this object.
WrapType as Integer	The wrap type: TX_WRAP_REPEAT TX_WRAP_CLAMP

Member procedures

Name	Description
Function FindChannelOfObject (channelName as String) as Channel	<i>(Inherited from Base)</i> Finds an animation channel acting on this object. If this function is called on a type that cannot

Name	Description
	<p>be animated (such as <i>Director</i>), it returns <i>null</i>.</p> <p>You can limit the search to a particular director by using the syntax "directorName\$channelName" for the <i>channelName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$channelName".</p>
Function FindKey FrameOfObject (Key FrameName as String) as Key Frame	<p>(<i>Inherited from Base</i>) Finds a Key Frame acting on the object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i>.</p> <p>You can limit the search to a particular director by using the syntax "directorName\$Key FrameName" for the <i>Key FrameName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$Key FrameName".</p>
Function FindOrCreateChannelOfObject (channelName as String) as Channel	<p>(<i>Inherited from Base</i>) This function works like FindChannelOfObject except that if no animation channel with the specified name exists, a new one is created.</p>
Function GetChannelsOfObject ([out] v as Array[Channel]) as Integer	<p>(<i>Inherited from Base</i>) Fills the array <i>v</i> 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 <i>Director</i>), <i>v</i> will be empty.</p>
Function IsAnimated () as Boolean	<p>(<i>Inherited from Base</i>) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it always returns <i>false</i>.</p>
Sub SetChanged ()	<p>(<i>Inherited from Base</i>) Mark this object as changed.</p>

19.3.39 Timecode

Base type

[Base](#)

Properties

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

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 <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$channelName" for the <i>channelName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$channelName".
Function FindKey FrameOfObject (Key FrameName as String) as Key Frame	(Inherited from Base) Finds a Key Frame acting on the object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$Key FrameName" for the <i>Key FrameName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$Key FrameName".
Function FindOrCreateChannelOfObject (channelName as String) as Channel	(Inherited from Base) This function works like FindChannelOfObject except that if no animation channel with the specified 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 <i>Hours</i> , <i>Minutes</i> , <i>Seconds</i> and <i>Frames</i> .
Function GetChannelsOfObject ([out] v as Array[Channel]) as Integer	(Inherited from Base) Fills the array <i>v</i> with the animation channels acting on this

Name	Description
	object, returning the number of channels. If this function is called on a type that cannot be animated (such as <i>Director</i>), <i>v</i> 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	(<i>Inherited from Base</i>) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it always returns <i>false</i> .
Sub SetChanged()	(<i>Inherited from Base</i>) Mark this object as changed.

19.3.40 Uuid

Description

A Universally Unique Identifier.

Properties

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

Name	Description
Sub GenerateNewUuid()	Generates a new UUID.

19.3.41 Variant

Description

A variant type. Objects of this type can hold objects of any other type.

19.3.42 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

Property	Description
X as Double	The x component
Y as Double	The y component
Z as Double	The z component

Member procedures

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	

Name	Description
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)	
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 <i>GINormal</i> ()	<i>Deprecated.</i>
Sub <i>GIVertex</i> ()	<i>Deprecated.</i>

19.3.43 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 Shared Memory member (Map) that shares its data among all engines connected to the database.

Properties

Property	Description
Map as Shared Memory [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]	<i>(Inherited from Base)</i> Gets the current scene.
Stage as Stage [read-only]	<i>(Inherited from Base)</i> Gets the current stage.
System as System [read-only]	<i>(Inherited from Base)</i> Gets system wide data.
VizCommunication as VizCommunication [read-only]	<i>(Inherited from Base)</i> Gets the VizCommunication object.

Property	Description
VizId as Integer	(<i>Inherited from Base</i>) Gets or sets the internal id of this object.

Member procedures

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	(<i>Inherited from Base</i>) Finds an animation channel acting on this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$channelName" for the <i>channelName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$channelName".
Function FindKey FrameOfObject (Key FrameName as String) as Key Frame	(<i>Inherited from Base</i>) Finds a Key Frame acting on the object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it returns <i>null</i> . You can limit the search to a particular director by using the syntax "directorName\$Key FrameName" for the <i>Key FrameName</i> argument. Nested directors may be specified like this: "directorName1\$directorName2\$Key FrameName".
Function FindOrCreateChannelOfObject (channelName as String) as Channel	(<i>Inherited from Base</i>) This function works like FindChannelOfObject except that if no animation channel with the specified name exists, a new one is created.
Function GetChannelsOfObject ([out] v as Array[Channel]) as Integer	(<i>Inherited from Base</i>) Fills the array <i>v</i> 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 <i>Director</i>), <i>v</i> will be empty.
Function IsAnimated () as Boolean	(<i>Inherited from Base</i>) Returns true if there are animation channels for this object. If this function is called on a type that cannot be animated (such as <i>Director</i>), it always returns <i>false</i> .

Name	Description
Sub SendVizCommandToHostGroup (command as String)	Sends a Viz command to the group of hostnames (which was previously filled by calls to <i>AddHostToGroup</i>).
Sub SetChanged ()	(Inherited from Base) Mark this object as changed.

19.3.44 Global Procedures

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.
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.

Name	Description
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.
Function Exp (x as Double) as Double	
Function Fix (x as Double) as Double	Returns <i>Floor</i> (x) if x larger than or equal to 0.0, otherwise <i>Ceil</i> (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.

Name	Description
Function GetParameterBool (parameterName as String) as Boolean	Returns the value of the Boolean parameter <i>parameterName</i> .
Function GetParameterColor (parameterName as String) as Color	Returns the value of the Color parameter <i>parameterName</i> .
Function GetParameterContainer (parameterName as String) as Container	Returns the value of the Container parameter <i>parameterName</i> .
Function GetParameterDouble (parameterName as String) as Double	Returns the value of the Double parameter <i>parameterName</i> .
Function GetParameterInt (parameterName as String) as Integer	Returns the value of the Integer parameter <i>parameterName</i> .
Function GetParameterString (parameterName as String) as String	Returns the value of the String parameter <i>parameterName</i> .
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 <i>System.GetElapsedTime()</i> 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 <i>addLeadingZeros</i> is true) with zeros.
Function Len (s as String) as Integer	Returns the length of the specified string.
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.

Name	Description
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 new line.
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 <i>Variant</i> .
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.
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	Registers a parameter of type Color.

Name	Description
as String , label as String , defaultValue as 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 drop-down box. <i>width</i> defines the parameter's width in number of characters and <i>height</i> sets the drop-down 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. <i>separatorChar</i> defines the character which is used to separate the values. <i>width</i> and <i>height</i> 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.
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. <i>width</i> and <i>height</i> 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. <i>width</i> 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. <i>width</i> specifies the parameter's width in pixels.

Name	Description
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 <i>charSet</i> 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. <i>width</i> and <i>height</i> 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
Sub SendGuiParameterShow (parameterName as String , state as Integer)	Shows or hides a parameter in the GUI. state can be set to <i>SHOW</i> or <i>HIDE</i> .
Sub SendGuiParameterUpdate (parameterName as String , state as Integer)	Enables or disables GUI parameter updates. state can be set to <i>ENABLED</i> or <i>DISABLED</i> .
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 <i>ENABLED</i> or <i>DISABLED</i> .
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 $x*x$.

Name	Description
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 ComboBox
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.
Function <i>IsKeyDown</i> (keyCode as Integer) as BooleanVizCursiv	<i>Deprecated. Use System.IsKeyDown.</i>
Function <i>IsLButtonDown</i> () as BooleanVizCursiv	<i>Deprecated. System.LButtonState.</i>
Function <i>IsMButtonDown</i> () as BooleanVizCursiv	<i>Deprecated. System.MButtonState.</i>
Function <i>IsRButtonDown</i> () as BooleanVizCursiv	<i>Deprecated. System.RButtonState.</i>
Function <i>SendCommand</i> (command as String) as StringVizCursiv	<i>Deprecated. Use System.SendCommand.</i>
Function <i>SendRemoteCommand</i> (hostname as String , port as Integer , commandString as StringVizCursiv , timeoutInMs as Integer) as StringVizCursiv	<i>Deprecated. Use System.SendRemoteCommand.</i>
Function <i>TcpSend</i> (hostname as String , port as Integer , commandString as StringVizCursiv , timeoutInMs as Integer) as StringVizCursiv	<i>Deprecated. Use System.TcpSend.</i>

20 Shared Memory (SHM)

The information in this section relates to the [VizCommunication.Map](#).

A local [VizCommunication.Map](#) in each Viz Engine (as part of a cluster), collects and stores data. This data can be internal data, like a scene script pushing data to the map, or data from external control applications through TCP or UDP.

This section contains information on the following topics:

- [External Data Input](#)
- [Internal Data \(Interactive Scene\)](#)
- [Synchronization](#)
- [Snapshot](#)

See Also

- [Data Sharing](#)
- [Visual Data Tools](#)

20.1 External Data Input

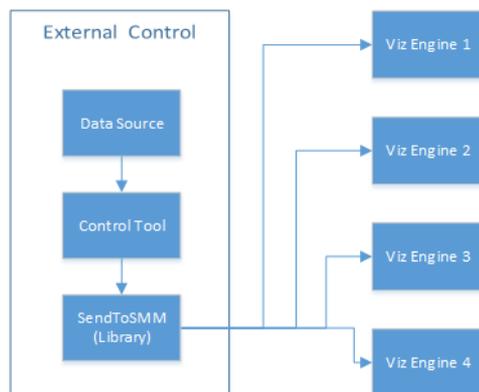
Data feed into Shared Memory (SMM) should be done through the dedicated UDP or TCP IP ports for the SMM. Vizrt provides a set of components, `SendToSMM`, to make this task easier.

Note: Go to `<viz install folder> -> Tools -> SendToSMM`, for more information about `SendToSMM`.

Data can also be sent to SHM through a Command Interface. Data sent through the Command Interface may be seen as a good option because the data would need to be sent to one Viz Engine only, and this Viz Engine would then distribute the data to the other Viz Engines. But Data sent to SHM through the Command Interface has problems:

- Data sent through the Command Interface will block the render queue of the receiving engine causing potential frame drops. Since the data needs to be sent through a command significant more bytes are transferred over the Network.
- This Viz Engine is also a single point of failure
- The data will arrive at this one Viz Engine sooner than on all other Viz Engines
- The notification method of the Graphic Hub is used to distribute the data and can cause additional load for the Graphic Hub

The preferred method to send data is to use the 'SendToSMM' library (or an equivalent) to send the data to the individual Viz Engines.



The communication protocol for the import of Shared Memory data depends on the type and final output of the data. There are set protocols to use with large amounts of data, in which all of the data must reach its destination graphic, and also where large amounts of data must be received, quickly, but some loss of data is acceptable.

For use cases see:

- [TCP Protocol](#)
- [UDP Protocol](#)
- [Plugin API](#)
- [Command Interface](#)

Note: It is also possible to import data through Multicast. This method is not recommended as it can pollute the network.

This section contains information on the following topics:

- [TCP Protocol](#)
- [UDP Protocol](#)
- [Plugin API](#)
- [Command Interface](#)

20.1.1 TCP Protocol

The SHM TCP communication protocol guarantees a reliable delivery of packages. It is a much more efficient than the [Command Interface](#), but not as fast as the [UDP Protocol](#) protocol.

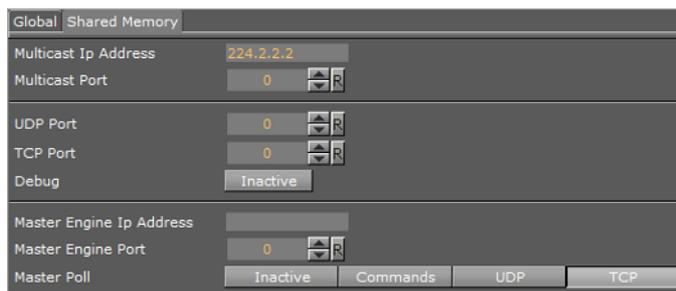
Use cases for a TCP connection could be finance stocks and currencies, or election result information, where the requirement is to deal with large amounts of information, and all of this data must reach its destination graphic. A single piece of lost data can have economic consequences, falsify charts, show mathematically wrong results, etc.

A TCP connection to a Viz Engine can be held open for a long time (this is recommended), and should not be opened and closed between sending variables.

Note: The default maximum number of TCP connections is limited to 255. Within this number of connections a User defined limit of maximum connections can be set (see **To Limit the Number of TCP Connections** in [Viz Configuration](#) (see the [Viz Engine Administrator Guide](#))).

IMPORTANT! The external program which provides the data, must connect and send the data to each Viz Engine individually. Vizrt provides a C# library, SendToSMM (part of the Viz install), for this purpose.

To Use TCP for SHM



1. Open Viz Configuration.
2. Click on **Communication**.
3. Click on the **Shared Memory Properties** tab.
4. In the Shared Memory panel set these parameters:
 - **TCP Port:** Vizrt does not recommend a specific port. Always make sure that the selected port is not in use by any other program on the same subnet.
5. Click **Save**.

The syntax for the key-value pairs is:

- key|value\0

Multiple key-value pairs can be sent, at once, as well. To do this make sure that each pair is terminated with 0.

- key1|value1\0key2|value2\0key...

See Also

- [UDP Protocol](#)
- [Plugin API](#)
- [Command Interface](#)

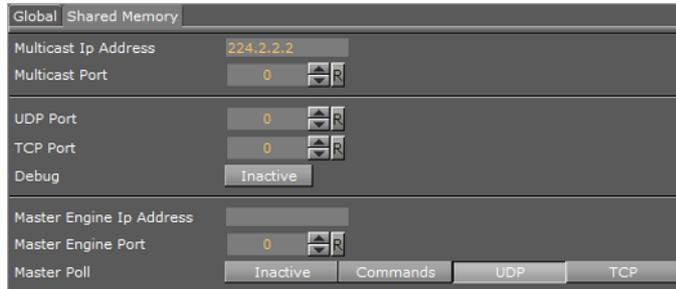
20.1.2 UDP Protocol

The SHM UDP communication protocol should be used for the delivery of volatile data. It is quicker than the [TCP Protocol](#) protocol, but less reliable, and is much more efficient than the [Command Interface](#).

A use case for UDP would be Motor Sports, where data like speed, velocity, etc., is required. This is where there is a requirement to deal with large amounts of data, but

not all of this data must reach its destination. A single piece of data lost will not affect the constant data update.

To Use UDP for SHM



1. Open Viz Configuration.
2. Click on **Communication**.
3. Click on the **Shared Memory Properties** tab.
4. In the Shared Memory panel set these parameters:
 - **UDP Port:** Vizrt does not recommend a specific port. Always make sure that the selected port is not in use by any other program on the same subnet.
5. Click **Save**.

The syntax for sending key-value pairs is the same as for [TCP and UDP Synchronization](#).

See Also

- [TCP Protocol](#)
- [Plugin API](#)
- [Command Interface](#)

20.1.3 Plugin API

An option to manipulate data in SHM is by a plugin interface.

A use case would be where a TCP or UDP connection cannot be used, or is not to be used. It is possible to write a plugin to import data (e.g. from an XML file, another database, etc.) and push it to SHM.

Another use case would be an interactive Scene (see [Internal Data \(Interactive Scene\)](#)).

Note: The Plugin API is documented in the plugin API documentation and comes with the Viz installer (go to, *Start -> All Programs -> vizrt -> Viz 3.x -> Documentation -> Plugin SDK Documentation -> Classes -> Class List -> Shared_Memory*).

See Also

- [TCP Protocol](#)
- [UDP Protocol](#)
- [Command Interface](#)

20.1.4 Command Interface

For small and single value changes the Command Interface of Viz Artist can be used. For example, to update a headline in a Scene.

IMPORTANT! A command operation can block the renderer. If there are too many commands, within a small time, or commands containing a large amount of data, are sent, this can result in not rendering real-time anymore.

Note: Vizrt do not recommend this as a method for data import.

Any external program should consider the performance of the single or all connected Viz Engines. If there is a burst of thousands of SHM variables this can have implications on the Viz Engine rendering performance (Current (CUR) and Maximum (MAX)).

A full list of commands is at: `<viz install folder>\Documentation\CommandInterface\index.html`

Note: From the list of commands, the commands, CLEAR, DELETE_ELEMENT and PURGE_ELEMENT will only work when sent through the command interface of Viz Artist.

IMPORTANT! The command CLEAR must be run on each Engine where the MAP is to be reset (VIZ_COMMUNICATION*MAP CLEAR).

Whenever a new entry is made in the map (a new key-value pair) or values are changed, then the change is propagated to the other Viz Engines through a database messaging service to update the local copy of each Viz Engine's map (this only works when sent over the general communication port of Viz Engine (default port 6100)).

Command Examples

VIZ_COMMUNICATION*MAP can be used to access the map.

SET_DOUBLE_ELEMENT and GET_DOUBLE_ELEMENT

Example: VIZ_COMMUNICATION*MAP SET_DOUBLE_ELEMENT "my_double" 1.2

Example: VIZ_COMMUNICATION*MAP GET_DOUBLE_ELEMENT "my_double"

See Also

- [TCP Protocol](#)
- [UDP Protocol](#)
- [Plugin API](#)

20.2 Internal Data (Interactive Scene)

When data is modified on one Viz Engine through a script or through a plugin, the data change will get reflected on the other Viz Engines automatically.

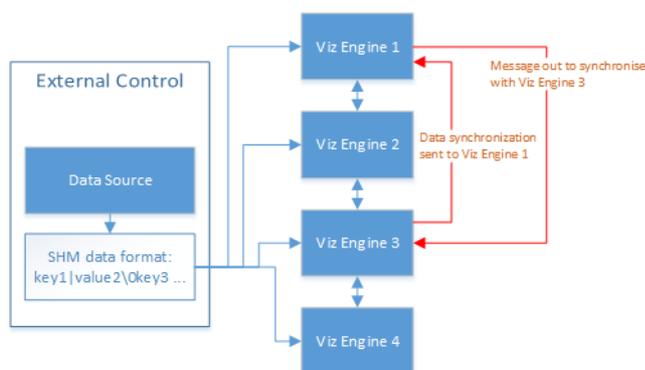
A use case could be a touch screen scene which modifies data, which is also used for HD-SDI Viz Engines or Viz Engines driving a Video Wall.

This synchronization uses the Graphic Hub as a relay. Therefore it is important that all Viz Engines, which are to receive the data, are connected to the same Graphic Hub, which use the same user or at least the same group.

20.3 Synchronization

If a Viz Engine is restarted or added to the cluster of Viz Engines for playback, the VizCommunication.Map data will not be available on this Viz Engine. The local map on a new or restarted Viz Engine can be synchronized with the local map on another running Viz Engine in the same cluster.

Example: If Viz Engine 1 restarts it will look to, for example, Viz Engine 3 to update its local VizCommunication.Map.



A startup Viz Engine can be synchronized through:

- TCP, or
- UDP, or
- An External Control Application, or
- Commands

This section contains information on the following topics:

- [TCP and UDP Synchronization](#)
- [External Control Synchronization](#)
- [Command Synchronization](#)

20.3.1 TCP and UDP Synchronization

Use the procedure detailed below to synchronize a restarted or added Viz Engines, in a cluster, with the TCP or UDP protocol:

- **TCP:** The recommended communication protocol to synchronize maps on startup because it is reliable and efficient (see [TCP Protocol](#)).
- **UDP:** For fast communication, but has drawbacks. For instance, packets can get lost (see [UDP Protocol](#)).

To Synchronize a Viz Engine with TCP or UDP

The screenshot shows the 'Shared Memory' configuration window. It includes the following fields and controls:

- Multicast Ip Address:** 224.2.2.2
- Multicast Port:** 0
- UDP Port:** 0
- TCP Port:** 0
- Debug:** Inactive
- Master Engine Ip Address:** (empty)
- Master Engine Port:** 0
- Master Poll:** Inactive, Commands, UDP, TCP

1. Open Viz Configuration.
2. Click on **Communication**.
3. Click on the **Shared Memory Properties** tab.
4. Enter the port number for:
 - **UDP Port**, or
 - **TCP Port**

Note: Vizrt does not recommend a specific port number. Always make sure that the selected port number is not in use by any other program on the same subnet.

5. In **Master Engine Ip Address** type the IP address of a running Viz Engine to synchronize with. Example <IP/hostname of Viz Engine 3>.

Note: Must be the IP address of a running Viz Engine in the same cluster system.

6. In **Master Engine Port** type the port number to be used (default 6100). This port sends the single startup synchronization command to start synchronizing. It should be the same as the general communication port of the Viz Engine set in **Master Engine Ip Address**.
7. In **Master Poll** click on:
 - **UDP**, or
 - **TCP**

Note: Must be the same protocol as selected in step 4.

8. Click **Save**.

See Also

- [Command Synchronization](#)

20.3.2 External Control Synchronization

Synchronization can also be done from an External Control Application. The following command has to be sent to the Viz Engine which has the memory map populated:

- **From a Command Interface**

```
VIZ_COMMUNICATION SYNCHRONIZE_TO <engine port>
```

where the Viz Engine is the engine which receives the data through the Command Interface. The port is usually 6100 (standard command interface port).

Note: Vizrt do not recommend this as a method for synchronization.

- **Through TCP Communication**

```
VIZ_COMMUNICATION SYNCHRONIZE_SMMTCP <engine port>
```

where the Viz Engine is the engine which receives the data, and the port is the one that was specified for incoming TCP key|value pairs on the Viz Engine which is to receive the data (see [TCP Protocol](#)).

- **Through UDP Communication**

```
VIZ_COMMUNICATION SYNCHRONIZE_UDP <engine port>
```

where the Viz Engine is the engine which receives the data, and the port is the one that was specified for incoming UDP key|value pairs on the Viz Engine which is to receive the data (see [UDP Protocol](#)).

20.3.3 Command Synchronization

Another communication option is by Commands. Here each map entry is synchronized through commands.

It is reliable, but very slow and blocks the engine for quite an amount of time, if the map is large.

Note: Vizrt do not recommend this as a method for synchronization.

To synchronize a Startup Viz Engine with Commands

1. Open Viz Configuration.
2. Click on **Communication**.
3. Click on the **Shared Memory Properties** tab.
4. In the Shared Memory panel set these parameters:
 - **Master Engine Port:** Enter the communication port of the Master engine (default is 6100).
 - **Master Poll:** Click on **Commands**.
5. Click **Save**.

See Also

- [TCP and UDP Synchronization](#)

20.4 Snapshot

It is also possible to take a data 'snapshot' (save the whole content of the map) at any time. One Viz Engine can save the whole content of the map by calling the map's SaveToDb procedure, and another client can read it by calling LoadFromDb.

The downside of this approach is that these functions block the renderer and can cause poor performance of the Graphic Hub database, if the map is stored repeatedly.

IMPORTANT! The resulting SHM map objects are replicated as well. So saving the map each field or every minute can result in serious problems for the Graphic Hub database (replication failing, etc.).

Note: For more information see the script function documentation. Go to, *Start -> All Programs -> vizrt -> Viz 3.x -> Documentation -> Script Documentation -> Data Types and procedures -> Shared Memory*, or *<viz install folder> -> ScriptDoc -> DataTypeSharedMemory*.

See Also

- [External Data Input](#)
- [Synchronization](#)

21 Integration with 3rd Party Applications and Files

This section details the 3rd Party application that can be integrated with Viz Artist.

This section contains the following topics:

- [Working with Adobe After Effects Files](#)
- [Working with CINEMA 4D Files](#)
- [Working with FBX Files](#)
- [Working with TriCaster](#)

21.1 Working with Adobe After Effects Files

This section describes how to work with Adobe After Effects and Viz Artist. For the system requirements and integration setup see [Adobe After Effects Integration](#).

The Vizrt / After Effects Import is capable of reading *.aep files (After Effects Project) and store its compositions as scenes.

All elements that are not yet able to be interpreted by the importer or cannot be rendered in real-time (like special effects) are rendered as clip (by the After Effects renderer). These clips are automatically imported into the Graphic Hub and composed in the imported scene. Like any other file in the Graphic Hub the clips are referenced with the scene as well.

This section contains the following topics:

- [Import of Adobe After Effect Files](#)
- [Import Limitations](#)

21.1.1 Import of Adobe After Effect Files

This section details how to import Adobe After Effects Scenes.

1. Create an Adobe After Effects project.
2. Save the project.
3. Import the project into Viz Artist as a Scene (see [Import Files](#)).



If an already imported scene is to be replaced, Viz Artist asks if the images and clips should be imported again. If there are no changes in these media elements, continue with **No**. The import will be much faster in this case because no output from the Adobe After Effects Renderer is required

Note: The Viz Artist output format setting should be the same as the Adobe After Effects Composition setting, and activate the used clip channels (activate Alpha as well). Otherwise the clip player cannot play the rendered video clips.

Note: The animation (director) is bound to the video clip (new director option called master clip). To establish of a frame accurate animation and clip playback set the Texture Delay to 0 in **Video Input: Clip Input** (see [Viz Configuration](#) in the [Viz Engine Administrator Guide](#)).

Note: Imported images, fonts and sub-scenes are imported into a subdirectory of the import target location called 'import'. If a required font was already imported into the Graphic Hub the font will not be imported again, because the import searches the Graphic Hub for this font and uses it.

See Also

- [Import Limitations](#)

21.1.2 Import Limitations

Currently the import of Adobe After Effects compositions is limited to these Adobe After Effects elements:

- Cameras
- Texts
- Solids
- Images
- Single compositions and their basic settings (e.g. resolution, frequency and background-color). Nested compositions are not (currently) supported.

IMPORTANT! Two licenses are required for the After Effects import: One license from Adobe for After Effects CS6 and the AE Import license issued by vizrt.

These elements are imported with their animations (Key Frames are not baked, they are converted directly to Viz Artist Key Frames).

All elements and or effects from Adobe After Effects, which cannot be converted to Viz Artist objects, are rendered as video clips and imported into the Graphic Hub. The import tries to combine multiple objects, which cannot be imported, and render them into one video (MPEG2 iFrame HD + Alpha).

Note: This can lead to several video clips if there are many alternations of importable and not importable layers.

Adobe After Effects offers a 2D/3D layer option. In Viz Artist there is one camera per scene. This means that each time 2D and 3D layers are mixed in the composition the import needs to create a new sub-scene (-> GFX Channel) to set a camera accordingly and still maintain the correct layer z-sorting.

See Also

- [Adobe After Effects Integration](#)
- [Import Files](#)

21.2 Working with CINEMA 4D Files

This section details how to work with CINEMA 4D and Viz Artist. For the system requirements and integration setup see [CINEMA 4D Integration](#).

The CINEMA 4D and Viz Artist integration imports complete geometries, lights and animations from CINEMA 4D into Viz Artist/Engine. Scenes from CINEMA 4D are imported directly into Viz Artist without the need for CINEMA 4D to be installed on the Viz Artist workstation.

3D objects are imported either as their direct primitive object counterpart in Viz Artist or as meshes.

A CINEMA 4D LiveLink tool, integrated into Viz Artist, lets designers make changes to a scene in CINEMA 4D and see the changes updated live when Viz Artist is set to On-Air mode.

The CINEMA 4D LiveLink tool is able to update the position, rotation and scaling properties of all objects in the scene.

This section contains the following topics:

- [Using the CINEMA 4D LiveLink](#)
- [Current Limitations](#)

See Also

- [CINEMA 4D Integration](#)
- [Import Files](#)

21.2.1 Using the CINEMA 4D LiveLink

This section details the requirements and how to connect to Viz Artist through the CINEMA 4D LiveLink.

This section contains the following procedures:

- [To Create a CINEMA 4D Scene for Import](#)
- [To Open a LiveLink](#)

To Create a CINEMA 4D Scene for Import

1. Create a scene in CINEMA 4D.
2. Select *File -> Save Project With Assets...* (this will save the CINEMA 4D Scene together with its textures).
3. Import the CINEMA 4D scene (.c4d) in to Viz Artist (see [Import Files](#)).

To Open a LiveLink

1. Open Viz Artist and CINEMA 4D.
2. Open the same scene in Viz Artist and CINEMA 4D.
3. Set Viz Artist to **On Air** mode.

4. In CINEMA 4D:
 1. Click on any entry in the CINEMA 4D Scene Tree.
 2. Click on **Tags**.
 3. Click on **Vizrt LiveLink**.
 4. In the Vizrt Realtime Demo properties panel, click on **Enable**.
 5. Enter the Viz Artist/Engine **Host** name.
 6. Enter the Viz Artist/Engine **Port** number.
 7. Click on **Reconnect**.

Changes made to the Scene in CINEMA 4D will now be automatically reflected in the scene in Viz Engine.

IMPORTANT! The same files must be opened in Viz Artist and CINEMA 4D, since they communicate through their UUIDs. If a different saved version of the file is open in CINEMA 4D and Viz Artist the live link will not work.

Tip: The Tag properties panel always shows the current connection status.

Note: The live Link works only one way, CINEMA 4D to Viz Artist. Objects moved in Viz Artist, will not transfer back to CINEMA 4D.

21.2.2 Current Limitations

These are the current limitations of the CINEMA 4D and Viz Artist integration, when a CINEMA 4D Scene is imported into Viz Artist:

- [Cameras](#)
- [Lights](#)
- [Materials & Textures](#)
- [General Limitations](#)
- [CINEMA 4D LiveLink Tool](#)

Cameras

- Maximum number of cameras in the scene: 16 (including the editor camera)

Lights

- Maximum number of lights in the scene: 8
There is no support for a scale value other than 1.0 for X,Y,Z
- Supported CINEMA 4D light types:
 - Omni
 - Spot
 - Distant
- Unsupported CINEMA 4D light types:
 - Parallel
 - Rectangular Spot

- Parallel Spot
- Parallel Spot Rectangular
- Photometric
- Tube
- There is no support for:
 - light intensity parameter
 - Shadows
 - near/far clipping
 - Noise
 - Caustics
 - Lens Effects
 - Reflection
 - Visible Light

Materials & Textures

- Currently no multi-texturing is supported
- The texture that will be imported into Viz Artist is only the one on the Color channel of the last texture tag of an object (the one which shows on the top layer)
- Textures generated by a CINEMA 4D shader (and CINEMA 4D shaders in general) are not supported
- There is no support for:
 - Polygon selections
 - Sides
 - Seamless or Repetitions settings

General Limitations

- By default, Viz Artist/Engine do not render the backfaces of objects. If this is required, add the Expert plugin to the required containers manually.
- The setting **Save Animation for Melange** generates Key Frames for animations produced by CINEMA 4Ds MoGraph toolset (among others). This setting will bake the animations as Key Frames into the timeline, potentially leading to an enormous amount of Key Frames that can slow down the import and playback of the scene in Viz Artist. If Save Animation for Melange is set to Off, in CINEMA 4D, this will eliminate the MoGraph animations and the baking of those Key Frames (while keeping other animation Key Frames intact).
- Viz Engine does not support negative values on the scale parameter, therefore there is no support for negative scale values on import or over the **Vizrt LiveLink**. If negative scale values are used, objects may show mirrored in Viz.
- Sky, Environment, Background, Foreground and Physical Sky Objects, are not supported.
- The Floor Object is imported as a rectangle of the size 2000x2000, just as shown in the Editing mode in CINEMA 4D (as opposed to a infinitely large rectangle in the Render Mode of CINEMA 4D). After import, its size can be scaled by the designer as required.

CINEMA 4D LiveLink Tool

Currently only changes in the position, rotation and scale of objects are updated over the LiveLink. Changes in other parameters are not currently supported.

21.3 Working with FBX Files

Viz Artist's built-in 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 features of a scene, that can be carried over to a Viz Scene, depends on:

- **FBX support of a 3D modeling application**

The FBX file format supports many common features of a 3D scene. But not every feature offered by the major 3D modeling packages can be covered by an interchange format.

Refer to the documentation of the 3D modeling application to discover which features can be exported to the FBX format.

- **Availability of the contained features in Viz Engine**

Viz Engine's feature is optimized for real-time. Features which are not available in Viz Engine cannot be imported and are ignored.

Note: Certain effects can be technically possible in Viz Engine but would require an elaborate conversion which may not be available.

To import an FBX file to Viz Artist, see [Import Files and Archives](#).

This chapter contains information on the following topics:

- [Scene Graph Traversal](#)
- [Transformations](#)
- [Export FBX Files from 3ds Max and Maya](#)

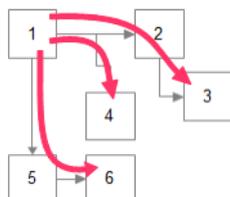
21.3.1 Scene Graph Traversal

The Scene representation in the form of a Scene Graph is common to all 3D applications, the FBX file format is also built this way.

When an FBX file is imported to Viz the Scene Graph is traversed node by node. All child nodes are traversed before advancing to the next sibling node.

This is how FBX nodes are transformed into Viz containers:

Scene graph traversal, child nodes first



FBX node types and how they are translated into Viz scene tree elements

FBX Node	Child Nodes	Viz Tree
Empty	Imported	Group
Camera	Ignored	Imported into global camera
Light	Ignored	Imported into global light
Mesh	Imported	Geometry
Other	Imported	Group

21.3.2 Transformations

The computation of transformations is more streamlined in Viz Engine than in FBX because the focus is on real-time graphics. To limit computing time spent on transformations the number of parameters incorporated in computing the final transformation matrix has deliberately been kept low in Viz.

In contrast, in a 3D modeling application the number of parameters incorporated in computing the final transformation matrix is high, to give maximum flexibility.

Note: If unsupported transformation features are contained in a FBX scene, Viz Engine will try to provide them by combining several transformations into one (maintaining the transformation order). This limits animation and can only work to a certain degree, after which the engine will fall back to baking the imported transformation onto the imported vertices.

- **Baking transformations:** This is the fallback which may be chosen for imported nodes as necessary. It implies computing the transformation matrix with the FBX formula and to apply the result to the imported vertices.
 - **The good:** To Bake the imported transformations onto the imported vertices basically means that arbitrarily complex transformations can be imported into Viz Engine.
 - **Drawback:** Any information on how the result of a complex transformation has been obtained is discarded through baking. In particular individual elements of the transformation cannot be animated anymore.

This section also contains information on the following topics:

- [Avoid Transformations Being Baked](#)
- [Avoid Transformations Being Combined](#)

Avoid Transformations Being Baked

If unsupported features are detected the transformations of those nodes are Baked, which means that individual transformations are not retained in a Viz Scene. The transformed vertices will be in the Viz Scene, but with all transformation parameters reset to their default values.

To avoid this make sure that unsupported features are not used in a Scene and, ideally, that the FBX properties which import into the same transformation in a Viz Scene are not used at the same time:

- **Do not use a scaling offset**

An extra offset applied after rotation and before scaling. It depends on the modeling application if it is available and under which name. To be on the safe side try to limit this to the regular translation and the rotation pivot.

- **Do not use a separate scaling pivot**

This can be tricky, depending on the 3D modeling application. In Maya the two pivots move together by default, which keeps the possibility to map them both to a single axis center in Viz Engine.

If the modeling application moves the rotation pivot separately by default, and objects are scaled afterwards, there will be two transformations based on different pivots. Viz Engine will have to bake the results to reflect them correctly.

FBX Import: Avoid separate scaling pivot

Rotation and scaling pivot identical	Object rotated and/or scaled	Ok, fully retained in a Viz Scene
Rotation and scaling pivot different	Object not scaled	Ok
	Object scaled	Baked

Note: More complex transformation hierarchies can be built and multiple pivots simulated by placing the objects in groups and apply additional transformations to the group or groups. This is reproduced in the scene tree when importing to Viz Engine.

See Also

- [Transformations](#)
- [Avoid Transformations Being Combined](#)

Avoid Transformations Being Combined

If, during import, two FBX properties have to be combined into a single property in Viz, they cannot be animated separately anymore. Limitations are not fully specified yet, since animation import is still under development, but to keep 100% the same control over transformations as in the original scene it is a good idea to consider the these tips:

- Do not use a rotation offset
- Do not use geometric translation/rotation/scaling in 3ds Max
- Do not use unit conversion (see [Export FBX Files from 3ds Max and Maya](#))
- Do not use axis conversion
 - FBX uses a Y-up axis system, this is also the default setting for the FBX export. If the modeling application uses a different axis system it will apply pre-rotation of $\pm 90^\circ$ to all objects to change the orientation of the Scene to match the FBX axis system. This can interfere with animation.
 - Maya uses a Y-up axis system by default so all pre- and post-rotations will be set to zero, which is perfect. If the modeling application uses another axis system (e.g. 3ds Max uses Z-up) and problems occur it is possible to select a different option in the 'Axis Conversion' section in the FBX export settings, so that no corrective rotation has to be applied.

See Also

- [Transformations](#)
- [Avoid Transformations Being Baked](#)

21.3.3 Export FBX Files from 3ds Max and Maya

To Export FBX Files from 3ds or Maya

1. From the File menu in Maya 2013, select Export (All)... or Export Selection....
2. Select a directory and enter a filename.

Note: Make sure that FBX is selected as the export file type to make export options show.

This section also contains information on the following topics:

- [Units](#)
- [Materials](#)
- [Other Materials](#)
- [Textures](#)
- [Camera](#)
- [Lights](#)
- [Spotlights](#)

Units

It depends on each modeling application how units in virtual 3D space correspond to real world units. The default working unit is centimeters. When a Scene is exported choose a scale factor in the FBX export dialog, the option is named **File units converted to** and can be found in section **Advanced Options > Units**.

The table below shows how units import from Maya into Viz Artist when different settings are used:

Maya units imported in to Viz Artist

Maya Working Units	Object size in Maya	File units converted to	Size in Viz Artist	Scaling factor
Centimeter	6	Centimeters	6	1
Centimeter	6	Millimeters	60	10

Caution: It is recommended that Scenes are built so that a scaling factor of 1 will do or, if this is not possible, scale the Scene up in Viz Artist after it is imported.

The reason: If scaling is enabled during export Maya will automatically alter the Scene to preserve the hierarchical transformation results. In some cases this will cause problems, e.g. create scaling pivot points, which are not supported in Viz.

Materials

Viz Engine supports one material per mesh node. The first material found on a FBX node is imported into a Material object, which is then applied to the Viz Container.

Note: Multiple or layered materials are not supported.

Supported FBX material classes are:

- [Lambert Shading](#)
- [Phong Shading](#)

Both types of shading have their own set of parameters which are described in more detail in the tables below:

Lambert Shading

FBX	3ds	Maya	Viz
Ambient Color	Ambient	Ambient Color	Ambient
Diffuse Color	Diffuse	Color	Diffuse
Diffuse Factor	-	Diffuse	Diffuse Color Attenuation
-	-	-	Specular =0,1%
Emissive Color	Self-Illumination	Incandescence	Emission
Transparency Factor/Color	Opacity (%)	Transparency (Color)	Alpha

Phong Shading

FBX	3ds	Maya	Viz
Ambient Color	Ambient	Ambient Color	Ambient Color
Diffuse Color	Diffuse	Color	Diffuse Color
Diffuse Factor	-	Diffuse	Diffuse Color Attenuation
Emissive Color	Self-Illumination	Incandescence	Emission Color
Specular Color	Specular	Specular Color	Specular Color
Specular Factor	Specular Level	-	Specular Color Attenuation
Shininess	Glossiness	Cosine Power	Shininess
Transparency Factor/Color	Opacity (%)	Transparency (Color)	Alpha

Other Materials

Other kinds of material which the modeling application supports may be mapped to a FBX Lambert or Phong material by the exporter, at least the parameters which are compatible to those material types.

Textures

FBX can associate a texture with each color channel on a material.

Note: Currently one texture for the diffuse color channel is supported.

Textures are referenced by the FBX file through their path and filename by default. This feature is broken in the current FBX SDK provided by Autodesk, i.e. the exported file may contain an absolute path and in some cases this path will not even be correct on the machine where the file was created.

Also, texture images can be embedded into the FBX file which is a lot more robust. Tick the check-box Embed Media in the export dialog to enable this.

Note: Regardless of the export parameters chosen, the FBX file can still contain incorrect texture references. If texturing problems occur use a free FBX viewer (e.g. the QuickTime plugin) to check if the texture import really failed or if there is a problem with the Viz import.

Camera

Currently up to 16 cameras are imported. The import is limited and can give little more than the correct camera position and direction. Dependant on the transformations that have been applied to a camera, and additional parameters unique to the modeling application, the camera position and viewing parameters can deviate a lot after import.

Lights

The lights detailed below are supported, but not all their parameters are imported:

Supported lights

	Lights
3ds	Target Spot
	Free Spot
	Target Directional (Switch to 'Overshoot' to make them resemble Viz Artist's 'Infinite' lights)
	Directional (Switch to 'Overshoot' to make them resemble Viz Artist's 'Infinite' lights)
	Omni (Switch to 'Overshoot' to make them resemble Viz Artist's 'Infinite' lights)
Maya	Directional Light
	Spot Light

	Lights
	Point Light

The only animation properties imported for Lights are:

- **Translation**
- **Cutoff / Cone Angle** (because hardware accelerated OpenGL lights are limited, this is unlikely to produce the exact same effect produced in the modeling software)

Color animation is not imported because it requires separate channels for red, green and blue components.

Spotlights

For performance reasons Viz Engine only supports the standard OpenGL light parameters.

OpenGL spotlight parameters and corresponding light parameters in Viz

OpenGL	Viz	Description
Cutoff	Cone Angle	Specifies the angle between the axis of the light cone and a ray along the edge of the cone. Can also be named 'outer cone angle'.
Exponent	Penumbra	Controls how concentrated the light is. The light's intensity is highest in the center of the cone. It's attenuated toward the edges of the cone by the cosine of the angle between the direction of the light and the direction from the light to the vertex being lit, raised to the power of the spot exponent.

FBX translates a modeling application's spotlight parameters to an inner and outer cone angle. During import into Viz the exponent is approximated based on the inner angle of the spotlight. Results can be different from what is seen in Maya or 3ds.

Note: Exponent in Maya: Maya has an alternative spotlight parameter, 'Dropoff', which corresponds directly to the OpenGL parameter 'Exponent'. With the Viz Artist VizFbxExtension plugin this parameter can be exported. This allows work with real GL-spotlight parameters in Maya and gives a viewport lighting much more faithful to the actual rendering results in Viz. Without the VizFbxExtension plugin Maya does not export that parameter, because no other Autodesk product supports importing it.

21.4 Working with TriCaster

Newtek's TriCaster is a popular choice for a vision mixer for small to medium budget operations. For the system requirements see [TriCaster Integration](#).

Top of the range is the TriCaster 8000 with eight simultaneous live inputs. If a customer wanted to use Vizrt graphics, two of these eight inputs needed to be sacrificed, or a

converter from HD-SDI to IP, such as iVGA Connect, was needed. With the Viz Engine and TriCaster integration this is no longer necessary.

Also, now Trio can be controlled directly from the TriCaster through macros, which can be mapped to keyboard and control panel shortcuts.

The integration of Viz Artist and TriCaster provides a fill and key IP stream, which is automatically detected by TriCaster as a source for graphics overlay.

Note: Two different graphic streams can be assigned in the TriCaster (net1 and net2). Each stream is a fill key combination.

As part of the Viz Artist and TriCaster integration package, Viz Trio is included. If the integration package runs on a Trio One Box, animated preview is available within Trio.

A second integration package includes Viz Trio on a HP ZBook, with still preview to create an ultra-portable graphics and video production solution for mobile production trucks (see [Viz Trio and Viz Engine for a TriCaster Production on the Same Laptop](#)).

TriCaster can control Viz Trio through macros, which can be mapped to the keyboard or the panel (see [To Set TriCaster Macros](#)).

This section contains the following topics and procedures:

- [To Connect a Viz Engine to TriCaster](#)
- [To Set TriCaster Macros](#)
- [Viz Trio and Viz Engine for a TriCaster Production on the Same Laptop](#)

To Connect a Viz Engine to TriCaster

1. Download **Processing.AirSend.x86.dll** (from ftp.vizrt.com directory /products/Vizrt Drivers/Newtek) to the Viz Engine folder.

IMPORTANT! Make sure to download the 32-bit **Processing.AirSend.x86.dll driver**, currently only the 32-bit Viz Engine/Artist is tested/supported with TriCaster integration.

2. Open the Viz Config file.
3. Set the value for **ignore_all_boards** to **1**
4. Make sure the Viz Artist dongle has the TriCaster feature enabled.
5. In the TriCaster user interface, click on the drop-down menu (next to **net1** or **net2**).
6. Using the drop-down menu from step 5, connect either net1 or net2.

To Set TriCaster Macros

1. Open the macro configuration panel from the TriCaster Macros menu.
2. Create a new macro called **Trio Take**.
3. Edit the macro:
 - **Shortcut:** net1 (tells TriCaster that the network address is the one where Viz Trio is installed)
 - **Value:** vizrt
 - **Key0:** trio

- **Value0:** take
4. Assign a keyboard shortcut to the macro.
 5. Start streaming from the Viz Engine, which is on the same host as Viz Trio.
 6. Connect TriCaster **net1** to it.

Viz Trio and Viz Engine for a TriCaster Production on the Same Laptop

On a laptop there is only one GPU and it must be shared between Viz Trio preview and Viz Engine program out.

Note: Since this will use one GPU to create the preview and the program, it is advisable to use a ringbuffer on the Viz Engine output. Depending on how heavy the Scenes are, this can go up to 15 frames.

1. Start Viz Trio client with the `-vizpreview localhost:50010` command line option.

Note: The 50010 port is a dedicated preview port (check in the Viz Config file (<viz data folder>)).

2. In the Viz Config file set:
 - **create_default_renderer** to **1**
3. In Viz Trio select **Single Engine** Mode.
4. Start Viz Engine.

22 Keyboard and Mouse Shortcuts

This section lists various keyboard and mouse shortcuts which can be used in Viz Artist.

When a keyboard shortcut is used, always make sure that the keyboard 'Caps Lock' is set to Off, unless otherwise stated.

This section contains information on the following topics:

Application Controls and Shortcuts	Stage Shortcuts
Integer/Float Controls	Import Shortcuts
Server Area Shortcuts	On Air Shortcuts
Scene Tree Shortcuts	Polygon Plugin Editor Shortcuts
Scene Editor Shortcuts	Script Editor Shortcuts

22.1 Application Controls and Shortcuts

Function	Key and/or 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 Quit button
Show Import Window	F9
Show Archive Window	F10
Show Viz Configuration Panel	F11
Show On Air Window	F12
Quit Viz Engine	Double-Esc Alt+Q Alt+F4

See Also

- [Integer/Float Controls](#)
- [Server Area Shortcuts](#)
- [Scene Tree Shortcuts](#)
- [Scene Editor Shortcuts](#)
- [Stage Shortcuts](#)

- [Import Shortcuts](#)
- [On Air Shortcuts](#)
- [Polygon Plugin Editor Shortcuts](#)
- [Script Editor Shortcuts](#)

22.2 Integer/Float Controls

Function	Key and/or Mouse
Increase/Decrease very large steps (100)	Shift+Alt, or Hold left mouse button and move
Increase/Decrease large steps (10)	Alt, or Hold left mouse button and move
Increase/Decrease normal steps (1)	Hold left mouse button and move
Increase/Decrease small steps (0.1)	Ctrl, or 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)

See Also

- [Application Controls and Shortcuts](#)
- [Server Area Shortcuts](#)
- [Scene Tree Shortcuts](#)
- [Scene Editor Shortcuts](#)
- [Stage Shortcuts](#)
- [Import Shortcuts](#)
- [On Air Shortcuts](#)
- [Polygon Plugin Editor Shortcuts](#)
- [Script Editor Shortcuts](#)

22.3 Server Area Shortcuts

This section contains the following topics:

- [General Shortcuts](#)
- [General Element Area Shortcuts](#)
- [Server View Tree Shortcuts](#)
- [Built-in View Shortcuts](#)

General Shortcuts

Function	Key and/or Mouse
Jump to top of tree	Home
Jump to bottom of tree	End
Move a page down in the tree	PageDown
Move a page up in the tree	PageUp
Scroll up the tree	Cursor up
Scroll down the tree	Cursor down
Open or close the selected tree	Return, or the numeric keyboard Return
Delete a folder	Delete (after confirmation)

General Element Area Shortcuts

Function	Key and/or Mouse
Jump to Top Element	Home
Jump to Last Element	End
Move a page down	PageDown
Move a page up	PageUp
Scroll up the Element list	Cursor up
Scroll down the Element list	Cursor down
Scroll left in the Element list	Cursor left
Scroll right in the Element list	Cursor right
Open or close the selected tree	Return, or the numeric keyboard Return
Delete an item	Delete (confirmation required)

Server View Tree Shortcuts

Function	Key and/or Mouse
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

Function	Key and/or Mouse
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 <i>This works only in Material or Scene View.</i>
Copy File(s)	Ctrl+C
Cut File(s)	Ctrl+X
Paste File(s)	Ctrl+V <i>This will create a link. File will not be duplicated.</i>
Paste File(s)	Ctrl+Shift+V <i>This will duplicate the file(s).</i>
Export	Ctrl+E <i>This works only in Image View.</i>
Rename File	F2
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

Built-in View Shortcuts

Function	Key and/or Mouse
Jump to plugin which starts with a given letter	Press the first letter of the plugin name.
Jump to Default folder for a category of plugins	Double-click the plugin folder tab.
Scroll up the plugin list	Cursor up
Scroll down the plugin list	Cursor down
Scroll left in the plugin list	Cursor left
Scroll right in the plugin list	Cursor right

See Also

- [Application Controls and Shortcuts](#)
- [Integer/Float Controls](#)
- [Scene Tree Shortcuts](#)
- [Scene Editor Shortcuts](#)
- [Stage Shortcuts](#)
- [Import Shortcuts](#)
- [On Air Shortcuts](#)
- [Polygon Plugin Editor Shortcuts](#)
- [Script Editor Shortcuts](#)

22.4 Scene Tree Shortcuts

This section contains the following topics:

- [General Shortcuts](#)
- [Basic View Controls and Shortcuts](#)
- [Sort View Shortcuts](#)

General Shortcuts

Function	
Show to Tree View	Ctrl+1
Show to Sort View	Ctrl+2
Show to Search View	Ctrl+F
Show DVE Media Asset Manager	Ctrl+4
Show Texture Media Asset Manager	Ctrl+5
Refresh	Ctrl+R
Select all Containers	Ctrl+A

<Ctrl-Shift-A> toggles the auto grab focus. This is toggles the auto grab focus to and from the Property Editor. When auto grab focus is on the Property Editor the shortcuts below, do not work.

Tip: This enables a user to work with objects without the need to click into the property again.

Function	Key and/or Mouse
Go to Top Container	Home
Go to Last Container	End

Function	Key and/or Mouse
Go to top of page	PageUp
Go to bottom of page	PageDown
Scroll up the Container list	Cursor up
Scroll down the Container list	Cursor down
Open or close the Container tree	Return, or the numeric keyboard Return

Basic View Controls and Shortcuts

Function	Key and/or Mouse
Insert New Container	Ctrl+Insert <i>The new Container will be created above the first selected Container.</i>
Multi select Containers	Ctrl+Right click
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
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 click on container name and drag
Copy Selected Container(s) to Clipboard	Ctrl+C
Paste Containers from Clipboard	Ctrl+V <i>They will be created one hierarchical level underneath selected container.</i>
Cut Selected Container(s)	Ctrl+X

Function	Key and/or Mouse
Move Selected Container(s)	Left click on container name and drag
Move Selected Container and retain transformation properties	Alt+Left click on container name and drag
Copy Properties of Container	Left click on container Properties and drag
Merge Group	Ctrl+M
Split Container	Ctrl+Shift+M

Sort View Shortcuts

Function	Key and/or Mouse
Sort by Vertices	Ctrl+Q
Sort by Render Time	Ctrl+W
Sort by Texture Size	Ctrl+E
Toggle Unique for Texture Size	Ctrl+T

See Also

- [Application Controls and Shortcuts](#)
- [Integer/Float Controls](#)
- [Server Area Shortcuts](#)
- [Scene Editor Shortcuts](#)
- [Stage Shortcuts](#)
- [Import Shortcuts](#)
- [On Air Shortcuts](#)
- [Polygon Plugin Editor Shortcuts](#)
- [Script Editor Shortcuts](#)

22.5 Scene Editor Shortcuts

This section contains the following topics:

- [General Shortcuts](#)
- [Camera Controls and Shortcuts](#)
- [Orthogonal View Controls and Shortcuts](#)
- [Animation Controls and Shortcuts](#)

General Shortcuts

Function	Key and/or Mouse
Scene Editor Shortcut Help	Ctrl+F1

Function	Key and/or Mouse
Zoom in / out	Ctrl + (numpad) Ctrl - (numpad)
Multi select Containers	Ctrl+Left-click
Deselect Container or Containers	SPACE, or Left-click
Deselect one of multiple selected Containers	Ctrl+Left-click
Delete container	Ctrl+Delete, or Backspace
Copy container(s)	Ctrl+Left-click and drag
Move Container on the z axis	Hold left and right button down
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+click on Snap, or Alt+click on Snap
Add new Key Frame in Stage of selected container	Return
Enable expansive Z-Sort mode	S

Camera Controls and Shortcuts

Function	Key and/or Mouse
Switch Camera from 1 to 10	1 - 0
Switch to Front Camera	Numeric Keypad 7
Switch to Back Camera	Numeric Keypad 0
Switch to Top Camera	Numeric Keypad 8
Switch to Bottom Camera	Numeric Keypad 2
Switch to Left Camera	Numeric Keypad 4

Function	Key and/or Mouse
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

Orthogonal View Controls and Shortcuts

Function	Key and/or 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

Animation Controls and Shortcuts

Function	Key and/or Mouse
Reset handle of selected Key Frame 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

See Also

- [Application Controls and Shortcuts](#)
- [Integer/Float Controls](#)
- [Server Area Shortcuts](#)
- [Scene Tree Shortcuts](#)
- [Stage Shortcuts](#)
- [Import Shortcuts](#)
- [On Air Shortcuts](#)
- [Polygon Plugin Editor Shortcuts](#)
- [Script Editor Shortcuts](#)

22.6 Stage Shortcuts

This section contains the following topics:

- [General](#)
- [Stage Tree](#)
- [Stage Views](#)

General

Function	Key and/or Mouse
Show Obj. Editor	Ctrl+1
Dopesheet View	Ctrl+2, or Shift+PageDown
Spline View	Ctrl+3, or Shift+PageUp
Dope/Spline View	Ctrl+4

Stage Tree

Function	Key and/or Mouse
Jump to Top Container	Home
Jump to Last Container	End
Move a page down in the tree	PageDown
Move a page up in the tree	PageUp
Scroll up the tree	Cursor up
Scroll down the tree	Cursor down
Jump to an item (Director, Actor, or Channel), in the tree	Press the first letter of the item name

Function	Key and/or Mouse
Open or close the selected tree	Return, or the numeric keyboard Return

Stage Views

Function	Key and/or Mouse
Start/Continue Animation Forward	Space
Start Animation Backwards	Shift+Enter
Set Timeline Marker/Value to Start	Ctrl+Home
Set Timeline Marker/Value to End (not when infinity is selected)	Ctrl+End
Set View to 0	Home
Set View to End director	End
Move the Timeline Marker one field	<Ctrl> and the left or right arrow key
Move the Timeline Marker more than one field at a time	Left or right arrow key. This, for example, in 1080i50 format, moves the timeline marker 5 fields, left or right, and in 1080i60 the movement is 6 fields, left or right
	<Alt> and the left or right arrow key. This, for example, in 1080i50 format, moves the timeline marker 50 fields, left or right, and in 1080i60 the movement is 60 fields, left or right
Delete Selected Key Frame(s)	Delete
Pan View	Middle mouse click and drag
Zoom Time Axis In or Out	Mouse wheel
Zoom Time Axis to Selected Channel	Shift+Z, Shift+Y
Zoom Time Axis to whole Animation	Shift+X
Zoom Spline/Dopesheet View In	Mouse wheel, or Z, or Y, or Ctrl + (Numpad)
Zoom Spline/Dopesheet View Out	Mouse wheel, or X, or Ctrl - (Numpad)
Zoom Spline View to whole Spline	Shift+A

Function	Key and/or Mouse
Zoom Time Axis and Spline View to view all Key Frames of selected Channel	Shift+Q
Zoom Time and Value Axis to View All Key Frames	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 Key Frame	C+Left mouse click and drag
Bring View to Timeline Marker	M
Reverse the Order of Selected Key Frames	Ctrl+R
Snap to Grid on/off	Ctrl+G
Set Time Axis to Seconds	Ctrl+8
Set Time Axis to Frames	Ctrl+9
Set Time Axis to Fields	Ctrl+0
Move Timeline Marker to nearest Key Frame	J
Scale a Portion of Selected Director	T+Left mouse click on Director Bar <i>This shows a region where Key Frames can be scaled.</i>
Rubber band 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 (Rotation Channel Only)	Ctrl+Z
View Spline 2 (Rotation Channel Only)	Ctrl+X
View Spline 3 (Rotation Channel Only)	Ctrl+C
Jump to next Key Frame	Tab (a Key frame must be highlighted first)
Jump to previous Key Frame	Shift+Tab
Escape from moving a Key Frame	Esc

See Also

- [Application Controls and Shortcuts](#)
- [Integer/Float Controls](#)
- [Server Area Shortcuts](#)
- [Scene Tree Shortcuts](#)
- [Scene Editor Shortcuts](#)
- [Import Shortcuts](#)
- [On Air Shortcuts](#)
- [Polygon Plugin Editor Shortcuts](#)
- [Script Editor Shortcuts](#)

22.7 Import Shortcuts

This section contains the following topics:

- [File Shortcuts](#)
- [Archive Shortcuts](#)

File Shortcuts

Function	Key and/or Mouse
Select Fonts for import	Ctrl+1
Select Images for import	Ctrl+2
Select Geometries for import	Ctrl+3
Select Scenes for import	Ctrl+4
Select Audio for import	Ctrl+5
Select Archives for import	Ctrl+6

Archive Shortcuts

Function	Key and/or Mouse
View Scenes for import	Ctrl+1
View Geometries for import	Ctrl+2
View Materials for import	Ctrl+3
View for Materials Advanced for import	Ctrl+4
View Image for import	Ctrl+5
View fonts for import	Ctrl+6
View Video Clips for import	Ctrl+7
View All files	Ctrl+8

Function	Key and/or Mouse
Entry Point View	Ctrl+7
Folder View	Ctrl+8

See Also

- [To Import Selected Files](#)
- [Application Controls and Shortcuts](#)
- [Integer/Float Controls](#)
- [Server Area Shortcuts](#)
- [Scene Tree Shortcuts](#)
- [Scene Editor Shortcuts](#)
- [Stage Shortcuts](#)
- [On Air Shortcuts](#)
- [Polygon Plugin Editor Shortcuts](#)
- [Script Editor Shortcuts](#)

22.8 On Air Shortcuts

Function	Key and/or Mouse
Return to Server View	Esc
Show or hide the VGA Preview window (not available on VGA version)	Hold Shift+Backspace

See Also

- [On Air Mode](#)
- [Application Controls and Shortcuts](#)
- [Integer/Float Controls](#)
- [Server Area Shortcuts](#)
- [Scene Tree Shortcuts](#)
- [Scene Editor Shortcuts](#)
- [Stage Shortcuts](#)
- [Import Shortcuts](#)
- [Polygon Plugin Editor Shortcuts](#)
- [Script Editor Shortcuts](#)

22.9 Polygon Plugin Editor Shortcuts

Function	Key and/or Mouse
Open Shortcut list	<Ctrl+F1>

Function	Key and/or Mouse
Sharpen Edge of Selected Point or Points	<Shift> + Left click
Delete Selected Point(s)	<Delete>
Add New Anchor Point	Left click
Add new Bezier point	Hold Left mouse button down and drag
Create New Spline	<Insert> + Left click
Close/Open Active Spline	C
Close Spline	Click on the first point to toggle open / closed
Enable/Disable Spline Info	<H>
Move Active Spline	<Alt>
Copy	<Ctrl+C>
Cut	<Ctrl+X>
Paste	<Ctrl+V>
Select Point or Spline	<S>
Move Anchor point	<Ctrl> + Left mouse button
Move handles	<Ctrl> + Left mouse button
Move single handle	<Shift> + Left mouse button
Remove an Anchor point	Click on existing point with Left mouse button
Rectangular selection	Hold <S> + Left mouse button down, and drag
Convert straight point to Bezier	<Shift> + Left mouse button drag on point

See Also

- [Application Controls and Shortcuts](#)
- [Integer/Float Controls](#)
- [Server Area Shortcuts](#)
- [Scene Tree Shortcuts](#)
- [Scene Editor Shortcuts](#)
- [Stage Shortcuts](#)
- [Import Shortcuts](#)
- [Script Editor Shortcuts](#)

22.10 Script Editor Shortcuts

Function	Key and/or Mouse
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

See Also

- [Application Controls and Shortcuts](#)
- [Integer/Float Controls](#)
- [Server Area Shortcuts](#)
- [Scene Tree Shortcuts](#)
- [Scene Editor Shortcuts](#)
- [Stage Shortcuts](#)
- [Import Shortcuts](#)