

USER GUIDE

VERSION 17.0v3

FOUNDRY.

Nuke™ 17.0v3 Copyright © 2026 The Foundry Visionmongers Ltd. All Rights Reserved. Use of this document and the Nuke software is subject to an End User License Agreement (the "EULA"), the terms of which are incorporated herein by reference. This document and the Nuke software may be used or copied only in accordance with the terms of the EULA. This document, the Nuke software and all intellectual property rights relating thereto are and shall remain the sole property of The Foundry Visionmongers Ltd. ("Foundry") and/or Foundry's licensors.

The EULA is available in this Nuke User Guide, End User License Agreement (EULA).

Foundry assumes no responsibility or liability for any errors or inaccuracies that may appear in this document and this document is subject to change without notice. The content of this document is furnished for informational use only.

Except as permitted by the EULA, no part of this document may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, recording or otherwise, without the prior written permission of Foundry. To the extent that the EULA authorizes the making of copies of this User Guide, such copies shall be reproduced with all copyright, trademark and other proprietary rights notices included herein. The EULA expressly prohibits any action that could adversely affect the property rights of Foundry and/or Foundry's licensors, including, but not limited to, the removal of the following (or any other copyright, trademark or other proprietary rights notice included herein):

Nuke™ compositing software © 2026 The Foundry Visionmongers Ltd. All Rights Reserved.

Nuke™ is a trademark of The Foundry Visionmongers Ltd.

Digital Domain ® is a registered trademark of Digital Domain, Inc.

Primatte™ keyer tool © 1997-2026 Photron USA, Inc. All Rights Reserved.

Primatte™ is a trademark of IMAGICA Corp.

Primatte™ patent is held by IMAGICA Corp.

In addition to those names set forth on this page, the names of other actual companies and products mentioned in this User Guide (including, but not limited to, those set forth below) may be the trademarks or service marks, or registered trademarks or service marks, of their respective owners in the United States and/or other countries. No association with any company or product is intended or inferred by the mention of its name in this User Guide.

ACADEMY AWARD ® is a registered service mark of the Academy of Motion Picture Arts and Sciences.

Linux ® is a registered trademark of Linus Torvalds.

Windows ® is the registered trademark of Microsoft Corporation.

Mac, macOS, Mojave, Catalina, Big Sur, Shake, Final Cut Pro, and QuickTime are trademarks of Apple, Inc., registered in the U.S. and other countries.

Adobe ® and Photoshop ® are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Maya ® is a registered trademark of Autodesk, Inc., in the USA and other countries.

Houdini ® is a registered trademark of Side Effects Software, Inc.

Boujou is a trademark of 2d3 Ltd.

3D-Equalizer is a trademark of Science.D.Visions.

RenderMan ® is a registered trademark of Pixar.

Cineon™ is a trademark of Eastman Kodak Company.

Stereoscopic images courtesy of Mr. Henry Chung, HKSC (http://www.stereoscopy.com/henry/). Images illustrating warping and morphing courtesy of Ron Brinkmann (http://www.digitalcompositing.com). Images from “The Day After Tomorrow” ©2004 courtesy of and copyright by 20th Century Fox. Images from "Stealth" courtesy of and copyright by Sony Pictures Inc. Images from "xXx" ©2002 courtesy of and copyright by Columbia Pictures Industries. Images illustrating GridWarpTracker and EdgeExtend courtesy of Little Dragon Studios. All rights reserved by their respective owners in the United States and/or other countries.

Thank you to Diogo Girondi for providing icons for the Nuke user interface and Tim Baier for proofreading.

The Foundry Visionmongers Ltd.
5 Golden Square
London
W1F 9HT
UK

Rev: Tuesday, June 2, 2026

Contents

Getting Started 33

Meet the Nuke Product Family 34

About Nuke Indie 36

About Nuke Assist 38

About Nuke Non-commercial 42

Key Concepts 43

Installation and Licensing 48

Windows 48
Installing on Windows 49
Launching on Windows 51
Uninstalling Apps on Windows 56

macOS 57
Installing on Mac 58
Launching on Mac 59
Uninstalling Apps on Mac 63

Linux 64
Installing on Linux 65
Launching on Linux 66
Uninstalling Apps on Linux 71

Licensing Foundry Applications 72

Nuke Studio Environments 88

Using the Compositing Environment 99

Using the Toolbar 100

Working with Nodes 102
Using the Tab Menu 117

Navigating Inside the Node Graph 122

Properties Panels 126

Using the Color Controls 133

Load Custom Settings with Node Presets 140
Animating Parameters 141
Compositing Viewers 164
Using the Viewer Controls 168
Viewer Selection Modes 177
Controlling Zoom and Resolution 181
Viewer Overlays and Input Processes 185
Using the File Browser 196
Undoing and Redoing 201
Progress Bars 201
Handling Errors 202

Using Nuke Studio's Timeline Environment 203
About Clips and Shots 205
Clip and Shot Properties 208
Ingesting Media 212
Using Drag-and-Drop 214
Using the File Browser 215
Sorting and Searching Media 220
Color-coding Source Clips and Shots 225
Reconnecting and Refreshing Clips 229
Localizing Media 230
Using the Timeline Viewer 239
Timeline Playback Tools 240
Using In and Out Markers 252
Working with Colorspaces 254
Monitor Output in Nuke Studio (NDI - SDI/HDMI) 256
Working with HDR Images 263
Streaming Viewer Output Over the Internet with NDI® 267
Using Scopes 279

About Anamorphic Media 285
About QuickTime Media 285

Compositing with Nuke 287

Managing Scripts 289
Loading Image Sequences 296
Reformatting Image Sequences 303
Image Caching 304
The Cache Directory 305
Using the DiskCache Node 306
Localizing Files for Better Performance 308
Saving Scripts and Recovering Back-Ups 315
Closing Scripts 318
Loading Scripts 319
Defining Frame Ranges 319

Reformatting Elements 321

Adjusting the Bounding Box 327

Channels 332
Understanding Channels and Layers 334
Creating Channels and Layers 335
Calling Channels 337
Linking Channels Using the Link Menu 341
Tracing Channels 342
Renaming Channels 342
Removing Channels and Layers 343
Shuffling Channels 344

Merging Images 351

Merge Operations 354
Generating Contact Sheets 365
Copying a Rectangle from one Image to Another 368
Removing Noise with Denoise 372
Connecting Denoise 373
Analyzing and Removing Noise 373
Reviewing the Results 375
Fine Tuning 376
Keying with ChromaKeyer 380
Picking the Screen Color 381
Improving Mattes 383
Despilling and Color Replacement 386
Multi-Pass Keying 388
Keying with Cryptomatte 389
Keying with Keylight 400
Basic Keying 401
Advanced Keying 404
View 407
Screen Color 409
Clip Black and White 415
Screen Gain 416
PreBlur and Tuning 417
Screen Processing 418
Mattes 421
Inside and Outside Masks 421
Source Alpha 422
Color Replacement 423
Keying with Primatte 426
Connecting the Primatte Node 426

Primatte Basic Operation Tutorial 427
Smart Select BG Color 429
Clean BG Noise 431
Clean FG Noise 432
Spill Removal - Method #1 433
Spill Removal - Method #2 434
Spill Removal - Method #3 435
Sampling Tools 436
Replacing Spill 437
Primatte Controls 439
Primatte Viewer Tools 441
Degrain Section 443
Actions Section 446
Adjust Lighting 451
Hybrid Matte 452
Fine Tuning 453
Spill Process Section 454
Output Section 455
The Primatte Algorithm 458
Explanation of How Primatte Works 459
Contact Details 467
Keying with Ultimatte 468
Connecting the Ultimatte Node 468
Sampling the Screen Color 469
Using Overlay Tools and Screen Correct 470
Adjusting the Density of the Matte 472
Adjusting Spill Controls 473
Retaining Shadows and Removing Noise 474
Adjusting Color Controls 476
Adjusting Film Controls 476
Choosing an Output Mode 477
Using RotoPaint 478
Connecting the RotoPaint Node 479

Working with the Toolbars 479

Working with the Stroke/Shape List 481

Drawing Paint Strokes 483
- Using the Brush tool 484
- Using the Eraser Tool 485
- Using the Clone Tool 486
- Using the Reveal Tool 487
- Using the Blur Tool 489
- Using the Sharpen Tool 490
- Using the Smear Tool 490
- Using the Dodge Tool 491
- Using the Burn Tool 492

Drawing Shapes 493
- Using the Bezier and Cusped Bezier Tools 494
- Using the B-Spline Tool 496
- Using the Ellipse, Rectangle, and Cusped Rectangle Tools 498
- Using the Open Spline Tool 499

Setting Default RotoPaint Tools and Settings 501

Selecting the Output Format and Channels 504

Selecting Existing Strokes/Shapes for Editing 505

Editing Existing Stroke/Shape Attributes 508
- Transforming Strokes/Shapes/Groups 511
- Adjusting Mask Controls 514
- Editing Shape-Specific Attributes 514
- Editing Stroke-Specific Attributes 515
- Editing Clone or Reveal Attributes 518

Editing Existing Stroke/Shape Timing 519

Editing Existing Stroke/Shape Stack Order 520

Editing Existing Stroke/Shape Splines 520
- Copying, Cutting, and Pasting Stroke Attributes 522
- Point Cusping, Smoothing, Expressions, and Links 524

Animating Strokes/Shapes 525
- Viewing Spline Keyframes 527
- Deleting or Rippling Keyframes 527
- Copying, Cutting, and Pasting Animations 529

Adding Motion Blur 531

Viewing Points in the Curve Editor and the Dope Sheet 535

Copying, Pasting, and Cutting Stroke Positions 536

Copying, Pasting, and Cutting Point Positions 537
Pasting Point Positions 538
Cutting Point Positions 538

RotoPaint and Stereoscopic Projects 539
Reproducing Strokes/Shapes in Other Views 540
Editing Strokes/Shapes in One View Only 541
Applying a Stereo Offset 542

Where Are the Bezier and Paint Nodes? 542

Tracking and Stabilizing 544

Connecting the Tracker Node 544

Adding Track Anchors 545

Tracking Preferences and Viewer Tools 547

Automatic vs. Keyframe Tracking 549

Automatic Tracking 549

Keyframe Tracking 553

Applying Tracking Data 558

Warping with GridWarpTracker 566

Drawing Grids 567

Tracking Using SmartVectors 569

Tracking Using the Tracker Node 571

Correcting Grid Deformation 573

Warping Shots with GridWarpTracker 575

Morphing Shots with GridWarpTracker 579

Transforming Elements 580

Transforming in 2D 580
Choosing a Filtering Algorithm 587
How Nodes Concatenate 594

Applying Core Transformations in 2.5D 599

Adding Motion Blur 603
Replicating the Input Image Across the Output 607
Warping Images 610
Warping 610
Morphing 629
Transforming and Animating Warps and Morphs 637
Temporal Operations 641
Distorting Time 642
Simple Retiming 642
Interpolation 643
OFlow Retiming 646
Warping Clips 650
Global Frame Range and Speed 652
Applying a TimeBlur Filter 653
Editing Clips 654
Working with Color 660
Using Scopes 660
Using the Pixel Analyzer 667
Making Tonal Adjustments 672
Making Hue, Saturation, and Value Adjustments 680
Masking Color Corrections 685
Applying Grain 687
Applying Mathematical Operations 690
Transforming the Color Space 694
Making Other Colorspace Conversions 696
Changing the Viewer Color Space 698
Filtering and Spatial Effects 700
Simulating Depth-of-Field Blurring 703
Removing Elements Using Inpaint 716

Simple Element Removal Using Inpaint 717
Inpainting Textures and Detail 722

Correcting Foreground Color Using EdgeExtend 725
Using EdgeExtend 726
Working with Edge Masks 727

Bilateral Filtering 729

Creating Effects 732

Background Reflections on Foreground Elements 732

Creating Star Filter Effects on Image Highlights 736

Creating Text Overlays 739
Entering Text 740
Fonts and Font Properties 745
Updating the Font Cache 751
Transforming Text 751
Animating Transforms 753
Changing the Text Color 754
Masking Regions of the Viewer 755
Adding Shadows to Text 756

Adding a Drop Shadow 758

Analyzing and Matching Clips 761

Cropping Black Edges 762

Analyzing the Intensity of a Frame Sequence 763

Removing Flicker 764

Analyzing Exposure Differences 764

Tracking the Brightest and Darkest Pixels 765

3D Compositing with the New 3D System 766

USD Basics: Stage, Prims, and Layers 770
Primitive Types and Primitive Attributes 773

USD Concepts in the New 3D System 776

3D System Nodes: From Classic to New 779

Importing into Your Scene 786
Importing with GeoImport 787

Referencing with GeoReference 793
Payloads and Deferred Loading 797

Creating Prims in Your Scene 799
Creating Geometry Prims 799
Creating Scope and Xform Prims 801
Creating Points Clouds 803

Viewing Your Scene 805
Look Through Camera 808
Selection in the 3D Viewer 809
Hydra in the 3D Viewer 822

Configuring Your Scene 824
New 3D Scene Basics 825
New 3D Scenes in the Scene Graph 827
Using Paths and Masks 835
Merging 3D Data 843
Using Collections 844

Working with Cameras in the New 3D System 846
Using Axis Node in Your Scene 854

Projections in the New 3D System 857

Transforming and Modifying 861
Transformations in 3D Scenes 861
Constraining Prims 864
Using Transformation Tools 867
Manipulating Geometry Data 873
Using Geometry Data 879
Instancing Geometry 884

Assigning Materials and Shaders 885
Using GeoBindMaterial to Assign Materials 887
Shading Geometry 888
ReflectiveSurface: Generating Reflections 891
BasicSurface: Shader Usage 893
Project3DShader: Projecting Textures 895
FillShader: Cut Out 897
ConstantShader: Color 898
WireframeShader: Applying Wireframe Overlay 899
MergeLayerShader: Merging Shaders 902
MtlXStandardSurface: MaterialX Shader 903
PreviewSurface: USD Workflows 906

Light Your Scene 908
Using Nuke Lights 909
Using USD Lights 917
Lights and Shadows 923

Lights in the Viewer 925
Rendering Lights 928

Rendering in the New 3D System 930
Ray-tracing with ScanlineRender 934
Motion Blur with ScanlineRender 936
Outputting AOVs with ScanlineRender 942

Export From Your Scene 944

Debugging Your Scene 946
Inspecting with GeoViewScene Node 947
Editing with GeoPython Node 948

Working with Gaussian Splats 951

Importing Gaussian Splats 952

Manipulating Gaussian Splats 954

Rendering Gaussian Splats 958

Classic 3D Compositing 960

Setting Up a Scene 961

Using the 3D Viewer 964

3D Scene Geometry 970
Using Built-in Primitive Geometry 970
Importing Objects from Other Applications 977
USD in Nuke's Classic 3D System 980
Importing ABC Meshes and Point Clouds 993
Creating Point Clouds and Geometry from Scratch 996
Creating a Position Pass Using the DepthToPosition Node 996
Creating a Dense Point Cloud Using the PositionToPoints Node 998
Creating a Point Cloud Using the DepthToPoints Node 999
Object Display Properties 1000
3D Selection Tools 1002
Merging Objects 1010
Modifying Object Shapes 1010
Modifying Objects Using Lookup Curves 1013
Modifying Objects Using a Power Function 1015
Modifying Objects Using an Image - Method 1 1016
Modifying Objects Using an Image - Method 2 1018
Modifying Objects Using a Perlin Noise Function 1020
Modifying Objects Using a Distortion Function 1022
Modifying Objects Using a Trilinear Interpolation 1023

Materials and Textures 1023

Object Material Properties 1025
Merging Two Shader Nodes 1036
Merging a Material with the Objects Behind 1037
Replacing Material Channels with a Constant Color 1041
Projecting Textures onto Objects 1043
Importing UDIM Patches 1044

Lighting 1046
Working with Lights 1048
Casting Shadows 1058
Manipulating Object Normals 1065
Relighting a 2D Image Using 3D Lights 1066

Cameras 1067
Working with Cameras 1068
Projection Cameras 1074
Importing Cameras from FBX Files 1077
Importing Cameras from Alembic Files 1079
Importing Cameras from Boujou 1081
Locating a 3D Point from an Animated Camera 1082

Transforming Geometry, Cameras, and Lights 1084
Using the Transform Handles 1085
Transforming from the Node Properties Panel 1088
Transformations and the Pivot Point 1089
Parenting to Axis Objects 1094
Using the TransformGeo Node 1096
Applying Tracks to an Object 1099
Importing Transforms from FBX Files 1100
Importing Transforms from Alembic Files 1102
Transforming and Projecting with SphericalTransform 1104
Compositing in 360 Footage 1106
Reviewing 360 Footage 1109

Adding Motion Blur to the 3D Scene 1110
Adding Motion Blur Using a Renderer 1111
Adding Motion Blur Using VectorBlur 1113

Exporting Geometry, Cameras, Lights, Axes, or Point Clouds 1119

Rendering a 3D Scene 1120

Stereoscopic Scripts 1122

Setting Up Views for the Script 1123

Loading Multi-View Images 1126

Displaying Views in the Viewer 1129

Selecting Which Views Display Changes 1131
Performing Different Actions on Different Views 1134
Reproducing Changes Made to One View 1135
Swapping Views 1136
Converting Images into Anaglyph 1137
Changing Convergence 1139
Previewing Stereoscopic Images 1143
Rendering Stereoscopic Images 1148

Working with Graph Scope Variables 1149

Using Graph Scope Variables in Your Workflows 1151
Using Variables with the Root Node 1162
Using Python with Graph Scope Variables 1164
Specifying Variables for Command Line Render 1167
Configuring Visibility with the Variables Panel 1168
Using the VariableSwitch Node 1170
Using the VariableGroup Node 1173
Using Graph Scope Variables in the New 3D System 1176
Using Link Nodes 1178

Deep Compositing 1187

Reading in Deep Footage 1188
Importing DTEX Files 1188
Creating Deep Data 1189
Merging Deep Images 1194
Creating Holdouts with the DeepMerge Node 1195
Creating 2D and 3D Elements from Deep Images 1196
Modifying Deep Data 1198
Cropping, Reformatting, and Transforming Deep Images 1199
Sampling Deep Images 1201

Writing Deep Data 1201

Working with File Metadata 1203

Viewing Metadata 1204

Modifying Metadata 1205

Copying and Filtering Metadata Between Inputs 1208

Adding a Time Code to Metadata 1209

Rendering Metadata 1210

Accessing Metadata Using Tcl Expressions 1210

Accessing Metadata Using Python 1211

Audio in Nuke 1212

Reading Audio Files into the Node Graph 1212

Creating and Editing Audio Curves 1214

Flipbooking the Audio Track 1215

Previews and Rendering 1216

Previewing Output 1217
- Previewing in a Nuke Viewer 1217
- Reviewing Your Work with Monitor Out in Nuke (NDI - SDI/HDMI) 1218
- Working with HDR Images 1231
- Streaming Viewer Output Over the Internet with NDI® 1235
- Flipbooking Sequences 1247

Rendering Output 1254
- Render Resolution and Format 1255
- Output (Write) Nodes 1256
- Rendering Using the Frame Server 1271
- Bypassing Nodes During Renders 1277
- File Name Conventions for Rendered Images 1285
- Changing the Numbering of Rendered Frames 1286
- Using a Write Node to Read in the Rendered Image 1288
- Render Farms 1289

Organizing Scripts 1291

Displaying Script Information 1291

Using Visual Diagnostics 1292
Filtering Profile Data 1295
Exporting and Importing Profile Data 1297
Using Performance Timing 1299
File Name Search and Replace 1302
Grouping Nodes in the Node Graph 1302
Adding Notes to the Node Graph 1309
Using the Precomp Node 1310
Creating Precomp Nodes 1311
Using a Precomp Node to Speed-up Rendering 1313
Precomp Revisions 1314
Collaborative Workflow Example 1315
Using the LiveGroup Node 1316
Collaborative Workflow Examples 1319
Creating LiveGroups 1322
Managing LiveGroup Controls 1327
Editing and Publishing LiveGroups 1329
Overriding LiveGroup Controls 1333
Configuring Nuke 1336
What Is a Terminal and How Do I Use One? 1337
Command Line Operations 1338
Environment Variables 1349
Nuke Environment Variables 1353
Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts 1360
Defining the Nuke Plug-in Path 1362
Loading OFX Plug-ins 1363
Defining Common Favorite Directories 1364
Handling File Paths Cross-Platform 1366
Setting Default Values for Controls 1367
Defining Custom Menus and Toolbars 1368
Defining Common Image Formats 1373
Creating and Accessing Gizmos 1374
Managing Gizmo Controls 1376

Adding Knobs Manually 1383
Accessing Gizmos in Nuke 1394
Sourcing Custom Plug-ins and Generic Tcl Scripts 1396
Template Scripts 1397
Defining Common Preferences 1398
Altering a Script's Lookup Tables (LUTs) 1399
Managing Nuke's Native LUTs 1400
Selecting the LUT to Use 1403
Example Cases 1404
OCIO Color Management 1405
Creating Custom Viewer Processes 1412
Using a Gizmo as a Custom Viewer Process 1414
Applying Custom Viewer Processes to Images 1417
Expressions 1419
Linking Expressions 1419
Adding Mathematical Functions to Expressions 1424
Converting Expressions Between Scripting Languages 1431
The Script Editor and Python 1432
Using the Script Editor 1432
Automating Procedures 1435
Nuke as a Python Module 1437
Getting Help 1438
Timeline Editing in Nuke Studio and Hiero 1441
Using Tags 1442
Tagging Using the Viewer 1443
Tagging Shots 1444
Adding Notes to Tags 1445
Removing Tags 1446
Creating Custom Tags 1447
Sharing Custom Tags 1448

Filtering and Flagging Media Using Tags 1449

Viewing Metadata 1451

Conforming Sequences 1452

Timeline Environment Project Settings 1453

Creating and Using Project Templates 1460

Importing Sequences 1461

Conforming Sequences 1466
Conforming Using a Browser 1466
Conforming with Pre-ingested Media 1470
About the Media Spreadsheet 1472
Adjusting Timecodes 1475

Renaming Shots on the Timeline 1477

Saving and Loading Projects 1479
Autosaved Projects 1480

Managing Timelines 1482

Adding Tracks to the Timeline 1486

Adding Clips to the Timeline 1488

Audio and the Timeline 1491
WAV Shots 1493
Displaying Audio Waveforms 1494
Audio Scrubbing 1495
Synchronizing Audio and Video 1495
PulseAudio on Linux 1496

Using Reference Media 1497
Comparing Media 1499

Caching Frames in the Playback Cache 1503

Caching Frames in the Disk Cache 1504
Caching Sequence Ranges 1507
Caching Selected Shot Ranges 1508
Caching In/Out Ranges 1510
Clearing Cached Frames 1511

Viewing Multi-Format Timelines 1514

Refreshing and Replacing Shots 1518

Setting Soft Trims 1519
Enabling and Disabling Shots 1520
Adding Transitions Between Shots and Audio Clips 1521
Invalid Transitions 1525
Retiming Shots 1526
Using Freeze Frames 1530
Blending Tracks on the Timeline 1531
Stereoscopic and Multi-View Projects 1535
Creating Views in a Project 1536
Importing Source Clips 1538
Displaying Views in the Viewer 1542
Displaying Views in the Timeline 1547
Applying Changes to Selected Views 1550
Adding Effects on the Timeline 1554
Using Multichannel Soft Effects 1572
Correct Shots Using LensDistortion 1576
Apply Pre-Trained Models to Shots Using Inference 1582
Create Comp 1585
Annotations 1598
Sync Review 1616
Timeline Editing Tools 1623
Using the Multi Tool 1625
Using the Move/Trim Tool 1626
Using the Selection Tools 1633
Using the Slip Clip Tool 1636
Using the Slide Clip Tool 1638

Using the Roll Edit Tool 1640
Using the Retime Clip Tool 1642
Using the Razor and Join Tools 1643
Insert, Overwrite, and 3-Point Editing 1646

Versions and Snapshots 1653

Using Versions 1653
Version Linking 1655
Versions in Bins 1658
Versions in Sequences 1660

Using Snapshots 1661
Creating Snapshots 1661
Restoring Snapshots 1663

Exporting from Nuke Studio 1664

Introduction to the Export Dialog 1665
Exporting Sequences with Quick Export 1667
Using Local and Project Presets 1671
Adding Burn-in Text to Exports 1677
Adding Additional Nodes During Export 1679
Using the Frame Server on Render Machines 1680

Exporting Sequences and Shots 1683
Nuke Project File Settings 1685
Nuke Write Node Settings 1687
Tracks, Range, and Handles Settings 1689
Building VFX Tracks and Comp Clips 1691
Exporting Multi-View Source Clips 1696

Transcoding 1698
Transcoding a Sequence as Shots 1702
Transcoding Timeline Selections 1704
Transcoding from the Bin View 1705

Ad Hoc Exports 1705
Exporting Sequences as EDL, OTIO, and XML 1706
Using the Audio Exporter 1710
Using the Copy Exporter 1715
Using the SymLink Generator 1717

Contact Sheet in Nuke Studio 1718
Using Contact Sheet: Manual Pick Mode 1722
Using Contact Sheet: Playback 1731
Using Contact Sheet: Tag Filtering 1736
Using Contact Sheet: Workflows & Soft Effects 1741
Using Contact Sheet: Preferences 1745

Advanced Compositing with NukeX and Nuke Studio 1747

Retiming and Motion Blur 1749

VectorGenerator 1749
Quick Start 1750
Connecting VectorGenerator 1751
Viewing and Refining the Results 1751

Kronos 1753
Quick Start 1755
Connecting Kronos 1755
Retiming a Sequence 1757
Refining the Results 1760
Adding Motion Blur 1763

MotionBlur 1764
Quick Start 1764
Connecting MotionBlur 1765
Adjusting MotionBlur Controls 1765

Working with Lens Distortion 1768

Estimating Lens Distortion Using a Grid 1769
Adjusting Grid Detection Parameters 1773
Removing Lens Distortion from an Image 1776

Estimating and Removing Lens Distortion Using Lines 1778

Applying Lens Distortion to an Image 1782

Working with STMaps 1783

Tracking with PlanarTracker 1786

Tracking a Plane 1786

Reusing a Track Result 1795
Placing an Image on the Planar Surface 1795

Camera Tracking 1800
Connecting the CameraTracker Node 1801
Masking Out Regions of the Image 1802
Working with Multi-View Scripts 1803
Setting Camera Parameters 1803
Tracking in Sequence Mode 1804
Viewing Track Data 1807
Troubleshooting Sequence Tracks 1809
Extending Existing Camera Tracks 1812
Retracking Partial Frame Ranges 1813
Tracking in Stills Mode 1814
Still Photography Guidelines 1814
Tracking Still Frames 1816
Viewing Reference Frames and Track Data 1820
Disconnected Frame Sets 1822
Troubleshooting Still Tracks 1823
Working with User Tracks 1824
Adding and Positioning User Tracks 1825
User Tracking Methods 1827
Tracking Assists 1830
Tracking a Scene Manually 1831
Linking Still Reference Frames 1831
Assigning 3D Survey Points 1832
Solving the Camera Position 1833
Viewing Solve Data 1834
Troubleshooting Solves 1840
Updating Solves with Extended Tracking Data 1843
Adjusting the Scene 1843
Setting the Ground Plane and Axes 1844
Transforming the Scene Manually 1846
Using Scene Constraints 1847
Using Solve Data 1849
Creating Camera Nodes 1849
Creating Scenes 1851
Creating Point Clouds 1853
Creating Cards 1854

Combining Solves 1855
Placing Objects in the Scene 1856
Accounting for Lens Distortion 1857
About the Lens Model 1858
Match Grades between Shots with MatchGrade 1860
Extracting a Baked-In Grade 1861
Matching a Grade Between Two Different Clips 1866
Using the Smart Vector Toolset 1871
Generating Motion Vectors 1871
Adding Paint to the Source 1874
Adding an Image to the Source 1875
Applying Motion Vectors to the Source 1876
Improving Warps 1878
Warping Images Using VectorCornerPin 1880
Warping Multiple Reference Frames 1884
Generating Depth Maps 1886
Connecting DepthGenerator 1887
Selecting What to Output 1888
Analyzing Depth 1890
Refining the Results 1891
Using the Results 1894
Creating Dense Point Clouds 1896
Connecting the PointCloudGenerator Node 1897
Masking Out Regions of the Image 1897
Setting Keyframes in a Sequence 1898
Tracking a Dense Point Cloud 1899

Filtering Your Point Cloud 1900
Removing Rejected Points 1901
Grouping, Labeling, and Baking Points 1902
Creating a Mesh from a Point Cloud 1904
Adding Texture to a Mesh 1905
Using the PoissonMesh Node 1906
Adding Texture to the PoissonMesh 1907
Using ModelBuilder 1908
Connecting the ModelBuilder Node 1909
Creating Shapes 1910
Editing Shapes' Display Characteristics 1913
Positioning Shapes 1915
Editing Shapes 1917
Editing Vertices 1918
Editing Edges and Edge Loops 1920
Editing Faces 1924
Editing Objects 1928
Setting the Initial Transform Action Center 1930
Applying Textures 1933
Projecting Textures onto Your Shapes 1934
UV Unwrapping 1936
Exporting Shapes to Separate Geometry Nodes 1947
Creating 3D Particles 1948
Connecting Particle Nodes 1949
Emitting Particles 1950
Spawning Particles with ParticleSpawn 1955
Adjusting the Speed and Direction of Particles 1956
Modifying the Particles' Movement 1958
Adjusting Controls Common to Several Particle Nodes 1961
Adjusting Particle Properties Using Curves 1963
Adjusting Particles Using Expressions 1964

Adjusting Particle Simulation Settings 1968
Merging Particle Streams 1969
Controlling Particles by Channel 1969
Caching Particles 1971

PrmanRender 1974

Using The PrmanRender Node 1974
Render Quality 1975
Shadows, Reflections, Refractions and Depth of Field 1976
Motion Blur Parameters 1976
Shader Parameters 1977
RIB Parameters 1977
Using the ModifyRIB Node 1978
Using the Reflection Node 1979
Using the Refraction Node 1979

FurnaceCore Nodes 1980

Global Motion Estimation 1981
What is Global Motion Estimation? 1981
Global Motion Estimation Effects 1982
Controls 1983
Widgets 1985
Using F_Align 1986
Quick Start 1987
Parameters 1989
Using F_DeFlicker2 1991
Quick Start 1991
Parameters 1992
Using F_ReGrain 1993
Quick Start 1994
Response 1995
Checking the Result 1996
Proxy Resolutions 1997
Parameters 1997
Color Space in FurnaceCore Plug-ins 2000
Using F_RigRemoval 2001
Quick Start 2002
Occlusions 2004

Parameters 2005

Using F_Steadiness 2007
Quick Start 2008
Parameters 2009

Using F_WireRemoval 2012
Background 2012
Reconstruction Methods 2013
Tracker 2014
Quick Start 2015
Positioning the On-Screen Wire Tool 2015
Tracking 2016
User Keyframes and Track Keyframes 2017
Indicators on the On-Screen Wire Tool 2017
Tracker Controls 2018
Parameters 2020

Train Neural Networks to Replicate Effects Using Machine Learning 2022

Create a Data Set for Training 2024

Train and Monitor the Network 2028

Train Networks from the Command Line 2038

Share the Load with Distributed CopyCat Training 2041

Apply and Improve the Results from Your Network 2047

Using BigCat 2054

Import Pre-Trained PyTorch Models 2059

Visualize Unreal Scenes in NukeX and Nuke Indie 2065

Using the BlinkScript Node 2077

Quick Start 2078

Connecting the BlinkScript Node 2078

Using and Writing Kernels 2078

Setting Kernel Parameters 2085

Loading Library Files 2086

Baking and Protecting Your Kernels 2088
Limitations and Known Issues 2090

Using the ParticleBlinkScript Node 2091
ParticleBlinkScript Attributes and Parameters 2094
Example ParticleBlinkScript Kernels 2097
Coloring Particles 2108
Constraining Particles 2111
Flowing Particles 2113
Image Input Projection 2117

Stitching Rigs with CaraVR 2119
Stitching Camera Rigs 2122
C_CameraSolver 2122
Preparing Camera Rigs 2123
Matching and Solving Cameras 2127
Troubleshooting Matches and Solves 2132
Exporting to Preset Nodes 2137
C_CameraIngest 2140
C_Stitcher 2143
Stitching Images Together 2144
Stitching Pre-Projected Images Together 2148
Troubleshooting Stitches 2150
Exporting to Preset Nodes 2152
Stitching Stereographic Rigs 2154
Rotoscoping to Correct Errors in the Source 2157
C_GlobalWarp 2160
Matching and Solving Warps 2160
Troubleshooting Matches and Warps 2162
Constraining Warps 2165
C_ColourMatcher 2172
Matching Colors Across All Cameras 2173
Exporting to Preset Nodes 2175
Compositing Workflows 2175
Transforming and Projecting with SphericalTransform 2176
Compositing in 360 Footage 2179
Transforming Using Metadata 2181
Reviewing 360 Footage 2182
Tracking and Stabilizing 2183
Automatic Tracking 2184
Manual Tracking 2190
Solving Cameras 2193
Stabilizing Using C_Tracker 2195

Warping Using STMaps 2198
Re-applying Corrections to a Warp 2200
Generating Stitch and PPass Maps 2202
Applying LatLong Blur 2203
Split and Join Selectively 2205
Generating Alpha Masks 2206
Blending Multiple Views Together 2207
Rendering Using RayRender 2209
Spherical Projection with RayRender 2209
Using Slit Scan with RayRender 2211
Compositing Using Facebook Surround Data 2213
Compositing Using Google Jump Data 2216
Compositing Using Nokia OZO Data 2218
Reviewing on VR Headsets (HMD) 2221
Written Tutorials 2224
The Projects 2225
Installing the Project Files 2225
Tutorial 1: Compositing Basics 2227
Starting Nuke 2227
Using the Toolbar 2229
Using the Menus 2229
Customizing Your Workspace 2230
Saving Files and File Backup 2232
Setting Up the Project 2233
Working with Nodes 2235
Connection Tips 2238
Importing Image Sequences 2241
Navigating Inside the Windows 2244
Working with Viewers 2245
Displaying the Images in a Viewer Window 2246
Viewing Multiple Inputs 2248
Reformatting Images 2250

Using Proxies and "Down-res" 2250
Compositing Images 2252
Color-Correcting Images 2254
Masking Effects 2255
Creating Flipbook Previews 2256
Rendering Final Output 2257
Epilogue 2260

Tutorial 2: 2D Point Tracking 2261
One-Point, Two-Point, Three-Point, Four 2262
Open the Tutorial Project File 2263
Tracking a Single Feature 2264
Tracking Obscured Features 2269
Stabilizing Elements 2271
Match-Moving Elements 2273
Epilogue 2275

Tutorial 3: Keying and Mattes 2277
Open the Tutorial Project File 2277
Keying with Primatte 2279
Image-Based Keying 2284
Rotoscoping 2290
Epilogue 2295

Tutorial 4: 3D Integration 2296
The Basic 3D System 2296
Open the Tutorial Project File 2299
Setting Up a 3D System 2300
Making a Scene 2305

Merging and Constraining Objects 2308
Animating a Scene 2310
Working with Geometry 2314
Lighting and Surface Properties 2320
Epilogue 2322
Appendices 2323
Appendix A: Preferences 2324
Appendix B: Keyboard Shortcuts 2366
Appendix C: Supported File and Camera Formats 2395
Supported Audio Formats 2410
End User License Agreement (EULA) 2411

Getting Started

An overview of installation and licensing, the default Nuke environments and workspaces, and the **Preferences** dialog.

The Nuke Family

Learn about the different products and modes of Nuke as well as some key concepts, everything you should know before using Nuke products.

Installation & Licensing

Find out how to install and license Nuke on Windows, Mac, and Linux to the point where you have the application in front of you and are ready to start work.

Comp Environment

Discover the basics of Nuke's compositing environment, designed to streamline day-to-day workflow. Flexible, efficient and feature packed, this toolset delivers film-grade results, fast.

Timeline Environment

Explore Nuke Studio's timeline environment allowing you to conform, review, edit, and even create and render compositions from the timeline.

Meet the Nuke Product Family

These pages introduce the different products and modes of Nuke and along with some Key Concepts you should know before using Nuke products.



Multi-shot management, editorial, compositing, and review. Nuke Studio adds an editorial environment to a full NukeX, giving artists working on multiple shots more context and control along with the familiar compositing Node Graph.



Multi-shot management, conform, editorial and review, Hiero has the same customizable timeline as Nuke Studio, but without Node Graph. Create and review timelines and generate Nuke scripts giving you greater creative control.



Advancing the art of digital compositing, NukeX adds advanced tools for tracking, clean up, and refining 3D. Putting more power and control in the hands of compositors and reducing the need to round-trip to other packages.



Artist's desktop player. Review multi-shot timelines in context with seamless timeline sharing with Hiero and Nuke Studio. Flexible and customizable, HieroPlayer caters to your workflow.

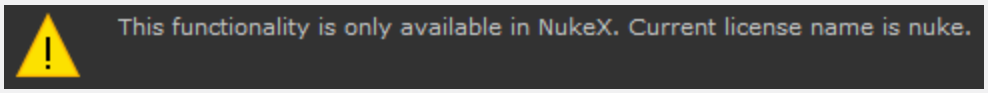


Fast and powerful industry standard node-based compositing. Nuke is the shot-based compositing toolset. From stereo to deep compositing, Nuke includes all of the essential compositing tools.

Nuke Products

When you purchase Nuke, it contains all the Nuke products as different modes. For example, you can choose to run Nuke in NukeX mode, or Nuke Non-commercial mode.

All the Nuke products listed below can also be licensed as separate products, except for Nuke Assist, which is included with NukeX and Nuke Studio.

Mode	Description
Nuke Studio	Includes the Timeline environment and Compositing environment, and has some additional Nuke Studio specific features, such as the ability to create a Nuke Comp, add soft effects to the timeline, and the annotations menu. Nuke Studio combines the tools and abilities of Nuke, NukeX, and Hiero. For more information about Nuke Studio, see Nuke Studio Environments.
Hiero	Multi-shot conform, timeline editorial, and review. Hiero is Nuke Studio without the Nuke and NukeX compositing capabilities. See Timeline Editing in Nuke Studio and Hiero for more details.
HieroPlayer	Timeline editorial and review. HieroPlayer employs the same timeline environment as Nuke Studio and Hiero, but has a few restrictions including no conform workflow or export capability. HieroPlayer is a free annual subscription with your Nuke or NukeX license. Simply renew your existing Nuke or NukeX license to receive your HieroPlayer entitlement or fill in the on-demand form. See HieroPlayer Free with Nuke for more information.
NukeX	Just like Nuke, a single shot compositor (what is referred to as the Compositing environment in Nuke Studio), with additional tools. Nuke Studio and NukeX are fully compatible, with Nuke capable of viewing and rendering - but not editing - NukeX features. If you open a script containing NukeX-only nodes in standard Nuke, a warning displays:  You can still work on the script, but you cannot change any NukeX node properties, such as ParticleEmitter or SmartVector, and they are grayed-out in the **Properties** panel. See Advanced Compositing with NukeX and Nuke Studio for more information.
Nuke	A single shot compositor (what is referred to as the Compositing environment in Nuke Studio), offering a Node Graph, Viewer, animation tools, and so on.
Nuke Indie	Nuke Indie is a single user version of Nuke using a log-in licensing model. Nuke

	Indie is designed for personal, small-scale commercial use and includes most of the features of the commercial version of Nuke. See About Nuke Indie for more information.
Nuke Assist	Provides basic tools including Roto, RotoPaint, Merge, Transform, and so on. The Nuke Assist mode is intended for use as a workstation for artists performing painting, rotoscoping, and tracking. See About Nuke Assist for more information.
Nuke Non-commercial	This mode is intended for personal and educational use. It is similar to the commercial version of Nuke with a few functional restrictions, offering you a chance to explore and learn the application while using it from the comfort of your own home. See About Nuke Non-commercial for more information.

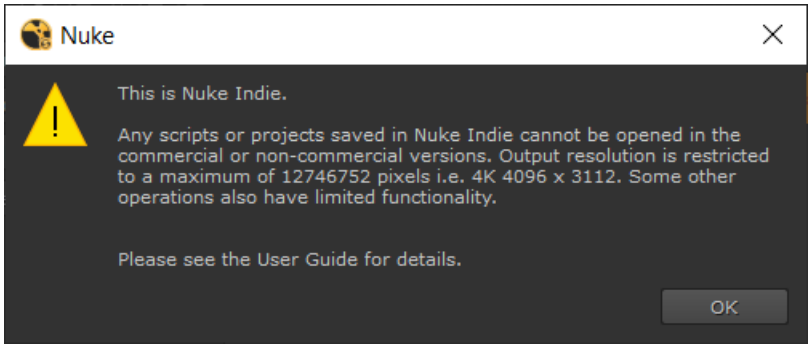
About Nuke Indie

Nuke Indie is a single user version of Nuke Studio using a log-in licensing model. Nuke Indie is designed for personal, small-scale commercial use and includes most of the features of the commercial version of Nuke Studio.

You can run Nuke Studio in Indie mode using the **--indie** command line argument. For example, to launch Nuke Studio in Indie mode on Mac, enter:

```
/Applications/Nuke17.0v3/Nuke17.0v3.app/Contents/MacOS/Nuke17.0 --indie
```

Nuke Indie is a fully functional version of Nuke Studio, but as it's designed for non-pipeline commercial use only, it differs from the full commercial version in some respects. Launching an application in Indie mode displays a warning:



The following table highlights the differences between Nuke Non-commercial, Nuke Indie, and Nuke Studio.

	Nuke Non-commercial	Nuke Indie	Nuke Studio
Usage			
Commercial use	No	Yes*	Yes
* Nuke Indie can be used commercially by artists earning up to $100k (USD) per year.			
Pipeline use	No	No	Yes
Licensing model	Single user login	Single user login	Site and node-locked licensing (multi-user support)
Online access required	Yes	Yes	No
Version timeout after 90 days	Yes	Yes	No
Feature Support			
Supported Export Resolutions	Up to 1080p	Up to 4096x3112	Unlimited
Encryption Free Script/Project Saving	No	No*	Yes
* Nuke Indie scripts (**.nkind**), projects (**.hroxind**), gizmos (**.gzind**), and CopyCat files (**.catind**) can only be read in Nuke Indie sessions.			
Full Python Api Support	No*	No*	Yes
* Nuke Non-commercial and Nuke Indie are limited to 10 node retrievals per script or project.			
Full Suite Of Keying Tools	No	Yes	Yes
Monitor Output Support	No	Yes	Yes
Geometry Export	No	Yes	Yes
Sync Review Support	No	No	Yes
Third Party / Custom OFX node / plug-in support	No	Yes	Yes

	Nuke Non-commercial	Nuke Indie	Nuke Studio
Third Party / Custom C++ node / plug-in support	No	No	Yes
NDK/ C++ SDK Support	No	No	Yes
External Rendering Support	No	No*	Yes
* Nuke Indie does not support external Frame Server renders or render licenses.			
Import Support			
H.264 (AVC) Variants Import Support	No	Yes	Yes
AAC Audio Import Support	No	Yes	Yes
Export Support			
EDL / OTIO / XML Export Support	No	Yes	Yes
LUT Export Support	No	Yes	Yes
H.264 (AVC) Variants Export Support	No	Yes	Yes
AAC Audio Export Support	No	Yes	Yes

About Nuke Assist

Nuke Assist is licensed as part of a NukeX or Nuke Studio maintenance package and is intended for use as a workstation for artists performing painting, rotoscoping, and tracking. Nuke Assist doesn't support any NukeX-or Nuke Studio-specific features apart from PlanarTracker, and has a limited subset of Nuke nodes and features available. Nuke Assist does not support any custom plug-ins, only the following nodes are supported:

Nuke Assist Nodes			
Image			
Checkerboard	ColorBars	ColorWheel	Constant
Read	Viewer		
Draw			
Bezier	Radial	Ramp	Rectangle
Roto	RotoPaint		



Note: Bezier is only available through the **X** menu. Press **X** in the Node Graph and then enter **Bezier** as a Tcl command to add the node.

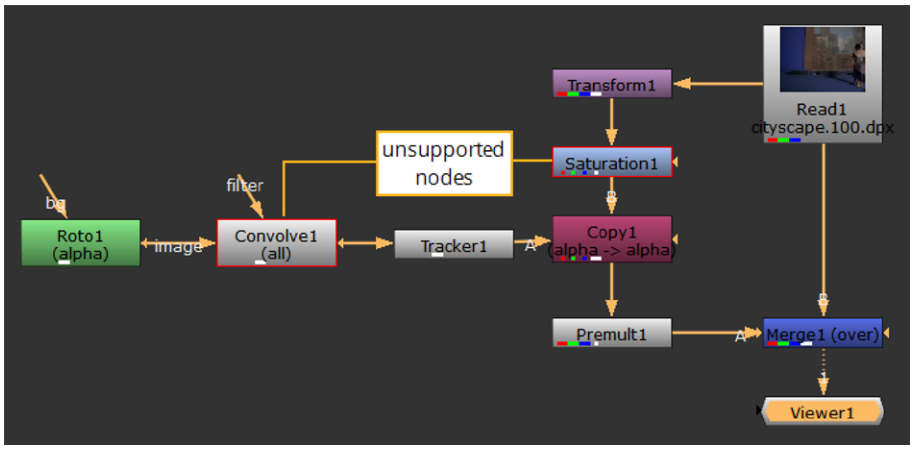
Time			
FrameBlend	FrameHold	FrameRange	TimeEcho
TimeOffset			
Channel			
Add	Copy	ChannelMerge	Remove
Shuffle			
Color			
Grade	Invert	OCIO CDLTransform	OCIO Colorspace
OCIO Display	OCIO FileTransform	OCIO LogConvert	
Filter			
Blur			
Keyer			
Keyer			
Merge			

Nuke Assist Nodes			
AddMix	Dissolve	KeyMix	Merge
Premult	Switch	Unpremult	
Transform			
Crop	CornerPin	PlanarTracker	Reformat
Tracker	Transform	TransformMasked	
Views			
JoinViews	OneView	ShuffleView	Split and Join
Stereo Anaglyph	Stereo MixViews	Stereo ReConverge	Stereo SideBySide
Metadata			
AddTimeCode	CompareMetadata	CopyMetadata	ModifyMetadata
ViewMetadata			
Other			
Backdrop	Dot	Group	Input
Output	PostageStamp	StickyNote	



Note: You cannot render output using Write nodes in Nuke Assist.

You can load projects created in Nuke's other modes and work as normal, within the constraints of Nuke Assist. The Viewer renders the output of the node tree whether the components are supported or not. Any unsupported nodes and plug-ins in the Node Graph are outlined in red and their controls are grayed out. You cannot modify the output of unsupported nodes and plug-ins.



Tip: For more information on node trees and node-based compositing, see Understanding the Workflow.

Using Gizmos, Groups, Precomps, Knobs, and Python

Gizmos, Group, and Precomp nodes can be loaded as normal, but if they contain any nodes that are not supported by Nuke Assist they have a red outline around the node in the Node Graph and the control panel controls are grayed out.

Note: Nuke Assist allows the creation of custom knobs, but they can't be edited.

Python scripts work as usual for nodes that are supported by Nuke Assist, but any attempt to add unsupported nodes displays an error message in the Script Editor output section, or terminal (in **-t** mode).

For example, executing **nuke.createNode('Transform')** adds a Transform node to the Node Graph, but **nuke.createNode('Convolve')** prints the following error:

```
# Result:
Traceback (most recent call last):
File "<string>", line 1, in <module>
RuntimeError: Convolve is not available in Nuke Assist
```

About Nuke Non-commercial

Nuke Non-commercial is a free version of Nuke that runs outside the regular licensing model. Nuke Non-commercial is meant for personal, educational, and other non-commercial use. It is aimed at students, industry professionals, and others interested in Nuke. It includes most of the features of the commercial version of Nuke, offering you a chance to explore and learn the application fully while using it from the comfort of your own home.



Note: Non-commercial mode can only be used for 180 days after a version of Nuke is built and released. For example, if Nuke 13.2v1 was built on March 25th, but released on March 31st, non-commercial mode is active until September 21st. You can find the build date for versions of Nuke in the splash screen, available from the **Help** > **About Nuke** menu.

You can run Nuke, NukeX, and Nuke Studio in non-commercial mode using the **--nc** command line argument. For example, to launch Nuke Studio in non-commercial mode on Mac, enter:

/Applications/Nuke17.0v3/Nuke17.0v3.app/Contents/MacOS/Nuke17.0 --nc --studio

Nuke Non-commercial is a fully functional version of Nuke, but as it's designed for non-commercial use only, it does differ from the commercial version in some aspects. Here are the main differences:

- Certain nodes are disabled in Nuke Non-commercial, including GenerateLUT, Primatte, Ultimatte, and WriteGeo.
- Rendered output is restricted to 1920x1080 HD and the MPEG4 and H.264 formats are disabled.
- The Python API only allows the retrieval of 10 nodes per script.
- Command line renders are restricted to encrypted **.nknc** scripts.
- Frame Server worker rendering is disabled.
- Exporting EDL/OTIO/XML sequences is disabled.
- Exporting LUTs from MatchGrade is disabled.
- Gizmos, clipboard contents, **.hrox** project files, **.nk**, and CopyCat **.cat** files scripts are all encrypted.
- Monitor Output is disabled.

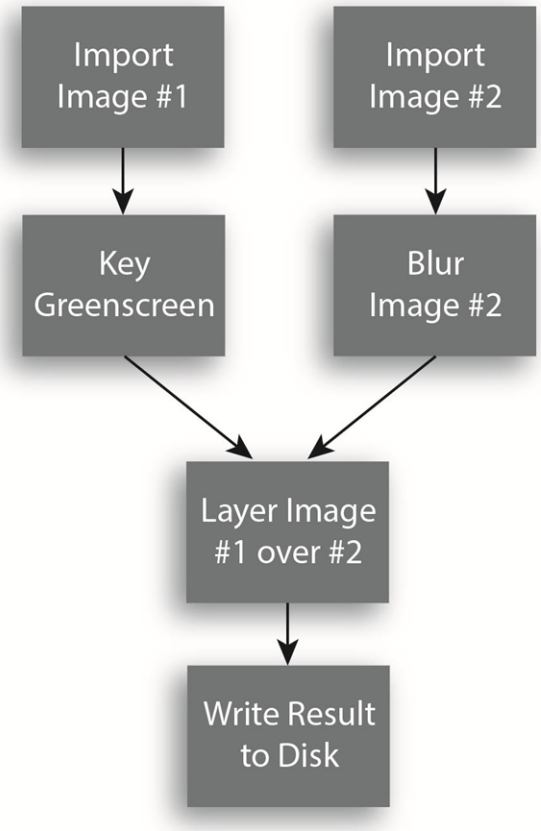
In other respects, Nuke Non-commercial contains all the functionality of the commercial version of Nuke.

Key Concepts

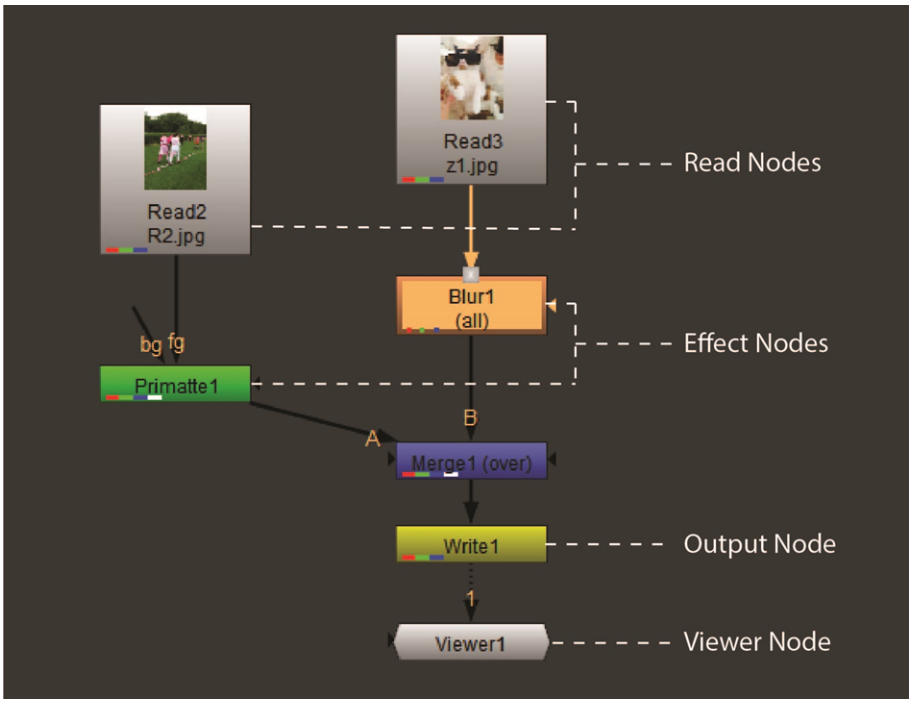
Nuke products are a resolution-independent compositing system, with extensive channel support, powerful image manipulation tools, and a rich 3D compositing environment. This section explains concepts you should know before using Nuke products.

Understanding the Workflow

A Nuke project consists of a network of linked operators called nodes. Nuke's Compositing environment utilizes a node-based workflow, where you connect a series of nodes to read, process, and manipulate images. Each node in the script, or comp, performs an operation and contributes to the output.



You can open a Nuke comp file in a text editor, and a series of sequential commands are displayed, which are interpreted and executed when you render the output.



In the image above, you see an example of a very simple Nuke script. Two Read nodes reference media on disk. Effect nodes extract a matte and blur an image. A Merge node set to over, composites the foreground image (input **A**) over the background image (input **B**). Finally, a Write node renders and outputs the completed composite to disk. There is also a Viewer node, which displays the output of any node in the script.

Note: Nuke Assist does not support Write nodes or render farms. See Nuke Products for more information.

Working with Multiple Image Formats

Nuke products support multiple file formats, such as Cineon, TIFF, OpenEXR, HDRI, and RAW camera data (using the dcraw command line program), and allows you to mix them all within the same composite. By default, Nuke products convert all imported sequences to their native 32-bit linear RGB colorspace. You can, however, use the Colorspace node to force one of several color models, including sRGB, Cineon, rec709, gamma 1.80/2.20, HSV, or HSL. The Log2Lin node lets you convert between logarithmic and linear colorspace (and vice versa).

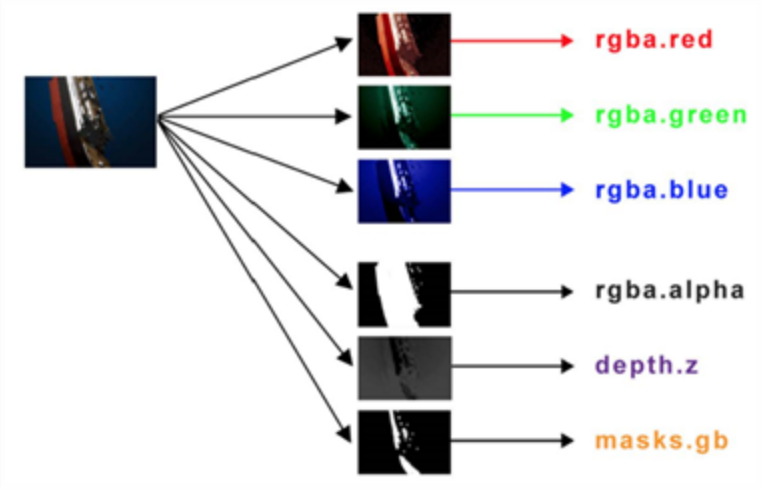
Note: Nuke Assist does not support Colorspace or Log2Lin nodes. See Nuke Products for more information.

There are no restrictions on image resolution - you can freely mix and scale elements of any resolution within the same script. You can, for example, use a 2k film plate as the background for video shot in PAL format, and then output the result in HD1080i. Nuke products automatically adjust their Viewers to accommodate the image you're viewing.

Channel Operations

In Nuke products, you can assign the output of each node as new channels, and pass them to the next node in the script. When you need to re-use a particular channel (say, to apply a color correction to the hair), you simply choose the channel containing the matte from the downstream color-correction node.

Nuke products support up to 1023 channels of image data. This provides additional benefits when working with computer-generated (CG) elements, especially when such elements are rendered out in the OpenEXR format.



For example, your 3D department could render out multiple lighting passes for a particular CG element (beauty, fill, backlight, reflection, shadow, etc.) as an **.exr** sequence, which you could then read into a Nuke script, or comp. You would be able to access all of the render passes stored within the **.exr** sequence from any downstream node in your script.

You might choose to color correct only the CG element's highlights by using the specular pass as a mask to a particular color correction operator. Such an approach again has the advantage of keeping the Nuke comp free of unnecessarily complex branching - virtually all render passes and mattes can be passed through a single pipe in the comp.

The Channels chapter explains how to take full advantage of the 1023-channel workflow.

8-, 16-, and 32-Bit Image Processing

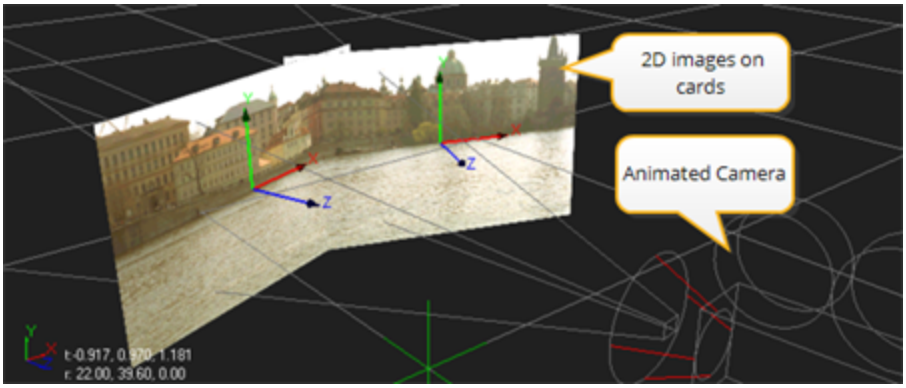
Some digital compositing systems, especially those geared for video work, are optimized for processing exclusively 8-bit elements (that is, images with 256 intensity values per channel). Other systems allow for the mixing of 8, 16, and 32-bit elements.

For Nuke products, which began as a film effects tool, image quality is paramount. Thus, they support the processing of exclusively 32-bit-per channel elements (elements with lower bit depths are converted to 32 bits per channel upon import). Thirty-two bit support allows for a much richer palette of colors and floating point precision in all script calculations. In practice, this means that Nuke products carry out every operation - from an increase in gamma to a transform - with much greater accuracy than a lower-bit-depth system.

Compositing in 3D

Some digital compositing systems support a strictly two-dimensional workflow. Nuke products, by contrast, offer a robust 3D workspace that lets you create and render complex scenes composed of polygonal models, cards (planes textured with images), cameras, lights, and textures.

This 3D workspace has countless uses, the simplest of which is generating pan-and-tile scenes. These are scenes with 2D image planes arranged into a curved shape, and then rendered out through an animated camera to give the illusion of a seamless environment.



Simple pan-and-tile scene.

The Classic 3D Compositing chapter explains how to make use of Nuke's Classic 3D System workspace.

The 3D Compositing with the New 3D System chapter explains how to make use of Nuke's New 3D System workspace.

Render Farms and Frame Servers

Nuke products support virtually all third-party and proprietary render-queuing software. By integrating Nuke products with such a system, the render load can be distributed across all the Nuke- or NukeX-licensed machines on your network, whether Windows, Mac, or Linux-based. See Render Farms for more information.

In addition, Nuke Studio ships with an internal Frame Server, which also allows you to setup external worker machines to process renders faster. See Using the Frame Server on Render Machines for more information.



Note: Nuke Assist does not support Write nodes, render farms, or the Frame Server. See Nuke Products for more information.

Installation and Licensing

We know the installation and licensing of a new application can be a boring task that you just want to be done with as soon as possible. To help you with that, this chapter guides you to the point where you have the application in front of you and are ready to start work.

Operating Systems

To see the installation and licensing instructions for your operating system, go to:

- Windows,
- macOS, or
- Linux.



Note: For more information on the differences between the Nuke applications, see Nuke Products.

Windows

These pages guide you through installation and licensing to the point where you have the application in front of you and are ready to start work. After installation, all applications are run from either desktop icons, the browser, or from the command line using arguments.

System Requirements

Qualified Operating Systems

- Windows 11 (64-bit)

Note: Other operating systems may work, but have not been fully tested.

Installing on Windows

The installation bundle installs the entire Nuke family, including Hiero and HieroPlayer, and icons for the various components appear in your installation folder.

Note: Some modern anti-virus software may wrongly assume that certain files in the Nuke installer are suspicious. Examples of these files include **nuke-17.0.dll**. If you have trouble installing Nuke on your machine, try disabling your anti-virus software before installation. Don't forget to restart your anti-virus software after installation.

1. Download the **.zip** file from our website at https://www.foundry.com/products/nuke/download and extract the contents.
   In the extracted directory, you'll find an **.msi** executable and several **.cab** files, which are required for installation on Windows.
2. Double-click on the **.msi** file to start the installation. Follow the on-screen instructions. By default, the application is installed to:
   `C:\Program Files\Nuke17.0`
3. That's it! Proceed with Launching on Windows.

Installing from the Command Line

Note: If you do not have administrator permissions, you may be prompted for a username and password in order to install the application to the machine. To avoid this, right click on the command prompt in the start menu and click **Run as administrator**.

1. Download the **.zip** file from our website at https://www.foundry.com/products/nuke/download and extract the contents.
2. To open a command prompt window, click **Start**, type **cmd**, and then click **Command Prompt**.
3. Navigate to the directory containing the **.msi** installation file. For example:
   `cd C:\Users\biff\Downloads`

4. Run the following command:

```
start /wait msiexec.exe /i Nuke17.0v3-win-x86_64.msi ACCEPT_FOUNDRY_
EULA=ACCEPT /qb
```



Note: By using the **ACCEPT_FOUNDRY_EULA=ACCEPT** property, you agree to the terms of the End User Licensing Agreement. The installer will display an error message if it is omitted. To see the agreement, please refer to https://www.foundry.com/eula.

Tip: The **/passive** argument can be used in place of **/qb** to suppress popup error messages during the installation. In this case it is advisable to also use **/l** to enable logging so that you can check the log file for errors in the event of a failed installation:

```
start /wait msiexec.exe /i Nuke17.0v3-win-x86_64.msi ACCEPT_
FOUNDRY_EULA=ACCEPT /passive /l log.txt
```


Tip: To install Nuke without any Shortcuts, use the ADDDEFAULT MSI property: `start /wait msiexec.exe /i Nuke17.0v3-win-x86_64.msi ACCEPT_FOUNDRY_ EULA=ACCEPT ADDDEFAULT=ProductFeature /qn`

By default, the application is installed to:

```
<Drive Letter>:\Program Files\Nuke17.0
```

- To install to a specified directory, use the property **INSTALL_ROOT="<Target Directory>"**, for example:

```
start /wait msiexec.exe /i Nuke17.0v3-win-x86_64.msi ACCEPT_FOUNDRY_
EULA=ACCEPT INSTALL_ROOT="E:\Nuke" /qb /l log.txt
```


Tip: You can display a list of install options by running the installer command by itself: `msiexec.exe`

5. Proceed with Launching on Windows.



Note: For versions prior to Nuke 14, use the **/S** and **/D** arguments instead of **/qb** and **INSTALL_ROOT**. For versions prior to Nuke 12, use the **/silent** and **/dir** commands instead of **/qb** and **INSTALL_ROOT**.
See https://learn.foundry.com/nuke/content/releases.html for more information.

Installing Oculus Headsets

Windows support for Oculus Rift CV1 and Oculus Quest 2 requires you to follow the Oculus Setup procedure. Refer to the accompanying documentation supplied by Oculus.

See Reviewing on VR Headsets (HMD) for more information on using headsets.

For the full Rift experience, Oculus recommend the following system:

- NVIDIA GTX 970 / AMD 290 equivalent or greater
- Intel i5-6400 equivalent or greater
- 8 GB+ RAM
- Compatible HDMI 1.3 video output sensor and 1x USB 2.0 port

The latest Oculus drivers come with the Oculus Setup utility, available here: https://www.oculus.com/setup/

Installing HTC Vive/Vive Pro

Windows support for the HTC Vive and Vive Pro requires that you install Steam and SteamVR as well as some additional hardware. Refer to the following setup procedure for more information http://www.vive.com/us/setup/

See Reviewing on VR Headsets (HMD) for more information on using headsets.

For the full Vive experience, Steam recommend the following:

- NVIDIA GeForce® GTX 970 / AMD Radeon™ R9 290 equivalent or greater
- Intel® i5-4590 / AMD FX 8350 equivalent or greater
- 4 GB RAM

Launching on Windows

To launch the application on Windows, do one of the following:

- Double-click the required icon on the Desktop.
- Navigate to **Start** > **All Programs** > **The Foundry** > **Nuke17.0v3** and select the required application.

- Using a command prompt, navigate to the application directory (by default, **\Program Files\Nuke17.0v3**) and enter:
  - **Nuke17.0.exe --studio** to launch Nuke Studio.
  - **Nuke17.0.exe --nukex** to launch NukeX.
  - **Nuke17.0.exe** to launch Nuke.
  - **Nuke17.0.exe --indie** to launch Nuke Studio in Indie mode. See About Nuke Indie for more information.
  - **Nuke17.0.exe --hiero** to launch Hiero.
  - **Nuke17.0.exe --player** to launch HieroPlayer.
  - **Nuke17.0.exe --nukeassist** to launch Nuke Assist.



Note: Nuke Assist licenses are only available as part of the NukeX or Nuke Studio package, and cannot be purchased separately. For more information, see About Nuke Assist.



Article: You can create desktop shortcuts to all versions of Nuke by adding command line flags to the shortcut's properties. See Knowledge Base article Q100412 for more information.

If you already have a valid license, the interface appears, and a command line window opens. If you don't have a license or haven't installed one yet, proceed to Licensing Foundry Applications.

High DPI Scaling

Nuke supports high definition displays, automatically scaling the interface using the operating system's scaling settings. Go to the Appendix A: Preferences **Behaviors** > **UI Scaling** sub-menu to manually set the scaling mode and scale per display, if required.

You can also use the QT_SCALE_FACTOR environment variable to force scaling to **1**, **1.5,** or **2**. The recommended scaling factor is **1.5**.



Note: You can disable automatic scaling by setting the QT_AUTO_SCREEN_SCALE_FACTOR environment variable to **0**.

In multi-monitor setups, you can scale the interface independently by screen using the QT_SCREEN_SCALE_FACTORS variable. Scaling uses the same recommended factors, separated by **;** (semicolon). For example, QT_SCREEN_SCALE_FACTORS="1;1.5" where the first monitor is lower resolution than the second.

See Environment Variables for more information about setting environment variables.

Command Line Startup Options

If you choose to launch the application from a command line, you can append arguments to the command as follows:

Argument	Result
--version	Displays version and copyright information.
-h (--help)	Displays the available arguments with examples.
--usehierolicense	Used with the **-t** option to run Nuke in terminal mode, but uses a Hiero license instead of the standard Nuke license. You can use this option to allow Hiero to run Nuke for render exports. **Note:** It is not possible to use the Hiero Python API in terminal mode.
--nukex	Launch NukeX instead of standard Nuke.
--studio	Launch Nuke Studio instead of standard Nuke.
--nukeassist	Launch Nuke Assist instead of standard Nuke.
--nc	Launch Nuke in Non-Commercial mode. See the notes farther down the page for more information.
--indie	Launch Nuke in Indie mode. See the notes farther down the page for more information.
-q (--quiet)	Launch the application without displaying the splash screen or startup dialog.
--safe	Launch the application without loading any plug-ins, Export presets, and so on.
--topdown	Enables the new top-down rendering project setting to render the full frame at the cost of more memory. Although top-down rendering produces the full frame faster, it disables progressive rendering and uses more memory. You may want to use the legacy method if you're only interested in the first few scan lines in the Viewer.

Argument	Result
Nuke Studio only	
--log-file	Sets the location of any logfiles created. For example: ./Nuke<version> --studio --log-file /Desktop/log.txt
--log-level	Sets the level of logging produced during operation. For example: ./Nuke<version> --studio --log-level warning Log messages are output to screen unless you specify a **--log-file**. There are four levels of detail, on a sliding scale from minimal to verbose: **error** **warning** (default) **message** **verbose**  **Note:** Setting the logging level to **verbose** can produce large log files when **--log-file** is specified.
--workspace	Launch Nuke Studio and apply the specified workspace. Only the workspaces listed in the **Workspace** menu are valid, but this includes any custom workspaces you have saved as **.xml** files in your **.nuke** folder under: /Workspaces/NukeStudio/ See Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts for more information on locating your **.nuke** directory by operating system.
<project path>.hrox	Launch Nuke Studio and open the project specified. The path to the project can be absolute or relative, such as: Nuke<version>.exe --studio C:\Users\mags\myProject.hrox Nuke<version>.exe --studio ..\..\mags\myProject.hrox

Argument	Result
	./Nuke<version> --studio /tmp/myProject.hrox ./Nuke<version> --studio ../../myProject.hrox
<mediaFile path>	Specify the location of media to load on startup. You can import specific files or whole directories: Nuke<version>.exe --studio C:\Users\mags\Media1.mp4 Nuke<version>.exe --studio C:\Users\mags\ ./Nuke<version> --studio /tmp/myfile.mov ./Nuke<version> --studio /tmp/
--script file argv	Run the specified Python script on startup and pass the listed options to the script.

See Command Line Operations for more in-depth information on command line flags.

Nuke Indie

Nuke Indie is a single user version of Nuke Studio using a log-in licensing model. Nuke Indie is designed for personal, small-scale commercial use and includes most of the features of the commercial version of Nuke Studio. See About Nuke Indie for more information.

To launch the application on Windows, do one of the following:

- Double-click the Nuke Indie icon on the Desktop.
- Using a command prompt, navigate to the Nuke application directory and enter:
    - **Nuke17.0.exe --indie** to launch Nuke Indie.

If you already have a valid license, the interface appears, and a command line window opens. If you don't have a license or haven't installed one yet, proceed to Licensing Foundry Applications.

Nuke Non-commercial

If you want to try out or learn Nuke, you can run Nuke Non-commercial. This version allows you to explore most of Nuke's features, but prevents the commercial use of the application. For more information, see About Nuke Non-commercial.

To launch the application on Windows, do one of the following:

- Double-click the NukeNC, NukeXNC, or Nuke StudioNC icon on the Desktop.
- Using a command prompt, navigate to the Nuke application directory and enter:
  - **Nuke17.0.exe --nc --studio** to launch Nuke Studio.
  - **Nuke17.0.exe --nc --nukex** to launch NukeX.
  - **Nuke17.0.exe --nc** to launch Nuke.

If you have already activated Nuke Non-commercial on the current device, the graphical interface appears, and a command line window opens. If you haven't activated the device yet, proceed to Licensing Foundry Applications.

Uninstalling Apps on Windows

To uninstall the application on Windows, there are a few things you need to do:

1. Navigate to the install directory for the required version. By default, the application is installed to:
   `C:\Program Files\Nuke17.0v3`
2. Double-click **uninstall.bat**.

   The **Uninstall** dialog displays.
3. Click **Yes** to uninstall.

Tip: You can also uninstall silently, with no prompts or popups, by running the Command Prompt as an Administrator and entering:
`"C:\Program Files\Nuke17.0v3\uninstall.bat" /passive`

4. Delete, rename, or move your **.nuke** directory, if it exists.

   The **.nuke** directory is usually found under the directory pointed to by the HOME environment variable. If this variable is not set (which is common), the directory is under the directory specified by the USERPROFILE environment variable, which is generally one of the following:
   `C:\Documents and Settings\login name\`
   `C:\Users\login name\`

   To find out if the HOME and USERPROFILE environment variables are set and where they are pointing at, enter %HOME% or %USERPROFILE% into the address bar in Windows Explorer. If the environment variable is set, the folder it's pointing at is opened. If it's not set, you get an error.
5. Delete, rename, or move your cached files, which reside in the following directory by default:

```
~\AppData\Local\Temp\nuke\
```

Where ~ is equal to %HOME% or %USERPROFILE% as detailed above.



Note: If you specified an alternate directory using the NUKE_TEMP_DIR environment variable, purge those files as well as the default location. See Nuke Environment Variables for more information.

6. Remove any application-related environment variables. See Environment Variables for more information.

macOS

These pages guide you through installation and licensing to the point where you have the application in front of you and are ready to start work. After installation, all applications are run from either desktop icons, the browser, or from the command line using arguments.

System Requirements

Qualified Operating Systems

- macOS Sequoia (15.x)
- macOS Tahoe (26.x)



Note: Nuke 15.0 and later includes native support for Apple silicon hardware. Other OS versions may work, but have not been tested or qualified on.



Article: For more information on Foundry products and supported macOS versions, see Foundry Knowledge Base article Q100592.



Note: Other operating systems may work, but have not been fully tested.

Installing on Mac

The installation bundle installs the entire Nuke family, including Hiero and HieroPlayer, and icons for the various components appear in your installation folder.



Note: Some modern anti-virus software may wrongly assume that certain files in the Nuke installer are suspicious. Examples of these files include **libnuke-12.0.0.so** and **geolib-runtime-prof.so**. If you have trouble installing Nuke on your machine, try disabling your anti-virus software before installation. Don't forget to restart your anti-virus software after installation.

Installing with the User Interface (UI)

1. Download the correct **.dmg** installation file from our website https://www.foundry.com/products/nuke/download
2. Double-click on the **.dmg** to start the installation.
3. Follow the on-screen instructions to install the application. By default, the install directory is:
   `/Applications/Nuke<version>`
4. Proceed to Launching on Mac.

Installing from the Terminal

1. Download the correct **.dmg** installation file from our website at https://www.foundry.com/products/nuke/download.
2. Launch a Terminal window.
3. To mount the **.dmg** installation file, use the **hdiutil attach** command with the directory where you saved the installation file. For example, if you saved the installation file in **/Builds/Nuke**, use the following command:
   `hdiutil attach /Builds/Nuke/Nuke<version>-mac-x86_64.dmg`
4. Read and acknowledge the End User License Agreement (EULA) by pressing **Y** at the end of the text.

Tip: If you've already read and agreed to the terms of the EULA, you can skip to the end of the text by pressing **Q**.

The installer is mounted as a disk image.

5. Enter the following command:
   `pushd /Volumes/Nuke<version>-mac-x86_64`

   This stores the directory path in memory, so it can be returned to later.
6. To install the application, copy the **Nuke<version>** directory to the **Applications** directory using the following command:
7. cp -R Nuke<version> /Applications/
8. Enter the following command:
   `popd`

   This changes to the directory stored by the **pushd** command.
9. Finally, use the following command to eject the mounted disk image:
   `hdiutil detach /Volumes/Nuke<version>-mac-x86_64`

Launching on Mac

To launch the application on Mac, do one of the following:

- Double-click the required icon on the Desktop.
- Open the application directory (**/Applications/Nuke17.0v3/**), and double-click the required icon.
- Using the terminal, navigate to the application directory (**/Applications/Nuke17.0v3/Nuke17.0v3.app/Contents/MacOS/**) and enter:
  - **./Nuke17.0 --studio** to launch Nuke Studio.
  - **./Nuke17.0 --nukex** to launch NukeX.
  - **./Nuke17.0** to launch Nuke.
  - **./Nuke17.0 --indie** to launch Nuke Studio in Indie mode. See About Nuke Indie for more information.
  - **./Nuke17.0 --hiero** to launch Hiero.
  - **./Nuke17.0 --player** to launch HieroPlayer.
  - **./Nuke17.0 --nukeassist** to launch Nuke Assist.

Note: Nuke Assist licenses are only available as part of the NukeX or Nuke Studio package, and cannot be purchased separately. For more information, see About Nuke Assist.

Article: You can create desktop shortcuts to all versions of Nuke by adding command line flags to the shortcut's properties. See Knowledge Base article Q100412 for more information.

If you already have a valid license, the interface appears, and a command line window opens. If you don't have a license or haven't installed one yet, proceed to Licensing Foundry Applications.

Command Line Startup Options

If you choose to launch the application from a command line, you can append arguments to the command as follows:

Argument	Result
--version	Displays version and copyright information.
-h (--help)	Displays the available arguments with examples.
--usehierolicense	Used with the **-t** option to run Nuke in terminal mode, but uses a Hiero license instead of the standard Nuke license. You can use this option to allow Hiero to run Nuke for render exports. **Note:** It is not possible to use the Hiero Python API in terminal mode.
--nukex	Launch NukeX instead of standard Nuke.
--studio	Launch Nuke Studio instead of standard Nuke.
--nukeassist	Launch Nuke Assist instead of standard Nuke.
--nc	Launch Nuke in Non-Commercial mode. See the notes farther down the page for more information.
--indie	Launch Nuke in Indie mode. See the notes farther down the page for more information.

Argument	Result
-q (--quiet)	Launch the application without displaying the splash screen or startup dialog.
--safe	Launch the application without loading any plug-ins, Export presets, and so on.
--topdown	Enables the new top-down rendering project setting to render the full frame at the cost of more memory. Although top-down rendering produces the full frame faster, it disables progressive rendering and uses more memory. You may want to use the legacy method if you're only interested in the first few scan lines in the Viewer.
Nuke Studio only	
--log-file	Sets the location of any logfiles created. For example: ./Nuke<version> --studio --log-file /Desktop/log.txt
--log-level	Sets the level of logging produced during operation. For example: ./Nuke<version> --studio --log-level warning Log messages are output to screen unless you specify a **--log-file**. There are four levels of detail, on a sliding scale from minimal to verbose: **error** **warning** (default) **message** **verbose** **Note:** Setting the logging level to **verbose** can produce large log files when **--log-file** is specified.
--workspace	Launch Nuke Studio and apply the specified workspace. Only the workspaces listed in the **Workspace** menu are valid, but this includes any custom workspaces you have saved as **.xml** files in your **.nuke**

Argument	Result
	folder under: /Workspaces/NukeStudio/ See Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts for more information on locating your **.nuke** directory by operating system.
<project path>.hrox	Launch Nuke Studio and open the project specified. The path to the project can be absolute or relative, such as: Nuke<version>.exe --studio C:\Users\mags\myProject.hrox Nuke<version>.exe --studio ..\..\mags\myProject.hrox ./Nuke<version> --studio /tmp/myProject.hrox ./Nuke<version> --studio ../../myProject.hrox
<mediaFile path>	Specify the location of media to load on startup. You can import specific files or whole directories: Nuke<version>.exe --studio C:\Users\mags\Media1.mp4 Nuke<version>.exe --studio C:\Users\mags\ ./Nuke<version> --studio /tmp/myfile.mov ./Nuke<version> --studio /tmp/
--script file argv	Run the specified Python script on startup and pass the listed options to the script.

See Command Line Operations for more in-depth information on command line flags.

Nuke Indie

Nuke Indie is a single user version of Nuke Studio using a log-in licensing model. Nuke Indie is designed for personal, small-scale commercial use and includes most of the features of the commercial version of Nuke Studio. See About Nuke Indie for more information.

To launch the application on Mac, do one of the following:

• Double-click the Nuke Indie dock icon.

- Open the Nuke application directory and double-click the Nuke Indie icon.
- Using the terminal, navigate to the Nuke application directory and enter:
    - **./Nuke17.0 --indie** to launch Nuke Indie.

If you already have a valid license, the interface appears, and a command line window opens. If you don't have a license or haven't installed one yet, proceed to Licensing Foundry Applications.

Nuke Non-commercial

If you want to try out or learn Nuke, you can run Nuke Non-commercial. This version allows you to explore most of Nuke's features, but prevents the commercial use of the application. For more information, see About Nuke Non-commercial.

To launch the application on Mac, do one of the following:

- Double-click the NukeNC, NukeXNC, or Nuke StudioNC dock icon.
- Open the Nuke application directory and double-click the NukeNC, NukeXNC, or Nuke StudioNC icon.
- Using the terminal, navigate to the Nuke application directory and enter:
    - **./Nuke17.0 --nc --studio** to launch Nuke Studio.
    - **./Nuke17.0 --nc --nukex** to launch NukeX.
    - **./Nuke17.0 --nc** to launch Nuke.

If you have already activated Nuke Non-commercial on the current device, the graphical interface appears, and a command line window opens. If you haven't activated the device yet, proceed to Licensing Foundry Applications.

Uninstalling Apps on Mac

To uninstall the application on Mac, there are a few things you need to do:

1. Navigate to **Applications** and delete the application directory.
2. Delete, rename, or move your **.nuke** directory, if it exists.

   The directory is found in your home directory, by default:
   `/Users/<login name>/.nuke`

Note: The directory may be hidden on your machine. To allow your Mac to display hidden files and directories, type the following command in the Terminal application, press **Return**, and then relaunch the Finder application:
`defaults write com.apple.finder AppleShowAllFiles YES`

3. Delete, rename, or move your cached files, which reside in the following directory by default:
`/var/tmp/nuke`

Note: If you specified an alternate directory using the NUKE_TEMP_DIR environment variable, purge those files as well as the default location. See Nuke Environment Variables for more information.

Linux

These pages guide you through installation and licensing to the point where you have the application in front of you and are ready to start work. After installation, all applications are run from either desktop icons, the browser, or from the command line using arguments.

System Requirements

Qualified Operating Systems

• Rocky 9.0 (64-bit)

Warning: Nuke requires **libnuma** to run under Linux distributions, the library is required by the Nablet H264 Codec SDK.

Note: The currently supported version of VFX Reference Platform includes library versions that are only compatible with Rocky 9.0.

Other operating systems may work, but have not been fully tested.

Installing on Linux

The installation bundle installs the entire Nuke family, including Hiero and HieroPlayer, and icons for the various components appear in your installation folder.

Note: Some modern anti-virus software may wrongly assume that certain files in the installer are suspicious. Examples of these files include **libnuke-12.0.0.so** and **geolib-runtime-prof.so**. If you have trouble installing the application on your machine, try disabling your anti-virus software before installation. Don't forget to restart your anti-virus software after installation.

1. Download the installation file from our website at https://www.foundry.com/products/nuke/download
2. Extract the application from the **.tgz** archive and then execute the following terminal command, replacing <version number> with the current version:
   `sudo ./Nuke<version number>-linux-x86_64.run`

   Note: If you leave out **sudo** from the terminal command, you need to ensure that you have sufficient permissions to install the application under your current working directory.

   After the application files have been installed, the installer also runs a post-installation script that creates the following directory:
   `/usr/local/foundry/RLM`

   If you don't have sufficient permissions on the **/usr/local** folder for this directory to be created, the post-installation script prompts you for your **sudo** password as necessary.

   The installer displays the End User Licensing Agreement (EULA) and prompts you to accept it.
3. If you agree with the EULA, enter **y** and press **Return** to continue. (If you don't agree with the EULA and press **N** instead, the installation is canceled.)

   Note: You can skip the EULA step using the **--accept-foundry-eula** option, which means you agree to the terms of the EULA:
   `sudo ./Nuke<version number>-linux-x86_64.run --accept-foundry-eula`
   To see the EULA, please refer to https://www.foundry.com/eula.

By default, Nuke is installed in the current working directory.

4. Proceed with Launching on Linux.

Tip: You can also use the following options after the terminal command when installing the application:
`--prefix=/home/biff/nuke_installs`
Specifies a different install directory, in this case, **nuke_installs**.
`--help`
Displays additional installer options.

Installing Oculus Rift

Linux support for Oculus Rift CV1 and Oculus Quest 2 requires the OpenHMD third-party library, which is installed as part of the CaraVR toolset.



Article: See Knowledge Base article Q100455 for information on how to configure headsets on Linux operating systems.

Installing HTC Vive/Vive Pro

Linux support for HTC Vive requires the OpenHMD third-party library, which is installed as part of the CaraVR toolset.



Article: See Knowledge Base article Q100455 for information on how to configure headsets on Linux operating systems.

Launching on Linux

To launch the application on Linux, do one of the following:

• Double-click the required icon on the Desktop.

- Open the application directory (by default, **/usr/local/Nuke17.0v3**) and double-click the required icon.
- Using a terminal, navigate to the application directory and enter:
  - **./Nuke17.0 --studio** to launch Nuke Studio.
  - **./Nuke17.0 --nukex** to launch NukeX.
  - **./Nuke17.0** to launch Nuke.
  - **./Nuke17.0 --indie** to launch Nuke Studio in Indie mode. See About Nuke Indie for more information.
  - **./Nuke17.0 --hiero** to launch Hiero.
  - **./Nuke17.0 --player** to launch HieroPlayer.
  - **./Nuke17.0 --nukeassist** to launch Nuke Assist.



Note: Nuke Assist licenses are only available as part of the NukeX or Nuke Studio package, and cannot be purchased separately. For more information, see About Nuke Assist.



Article: You can create desktop shortcuts to all versions of Nuke by adding command line flags to the shortcut's properties. See Knowledge Base article Q100412 for more information.

If you already have a valid license, the interface appears. If you don't have a license or haven't installed one yet, proceed to Licensing Foundry Applications.

High DPI Scaling

Nuke supports high definition displays, automatically scaling the interface using the operating system's scaling settings. On Linux operating systems, scaling is currently set to **off** by default. Go to the Appendix A: Preferences **Behaviors** > **UI Scaling** sub-menu to manually set the scaling mode and scale per display, if required.

You can also enable automatic scaling by setting the QT_AUTO_SCREEN_SCALE_FACTOR environment variable to **1**.



Note: Auto-scaling in some multi-screen setups causes the interface to scale incorrectly when moving from lower resolution screens to higher resolution screens. This is possibly a symptom of the way Qt calculates screen scale.
To avoid this issue, we recommend always placing your physical screens with the highest resolution on the far left-hand side of the setup.

Alternatively, you can set the screen scaling factor manually using the QT_SCALE_FACTOR environment variable to force scaling to **1**, **1.5,** or **2**. The recommended scaling factor is **1.5**.

In multi-monitor setups, you can manually scale the interface independently by screen using the QT_SCREEN_SCALE_FACTORS variable. Scaling uses the same recommended factors, separated by **;** (semicolon). For example, QT_SCREEN_SCALE_FACTORS="1.5;1" where the first monitor is higher resolution than the second.

See Environment Variables for more information about setting environment variables.

Command Line Startup Options

If you choose to launch the application from a command line, you can append arguments to the command as follows:

Argument	Result
--version	Displays version and copyright information.
-h (--help)	Displays the available arguments with examples.
--usehierolicense	Used with the **-t** option to run Nuke in terminal mode, but uses a Hiero license instead of the standard Nuke license. You can use this option to allow Hiero to run Nuke for render exports. **Note:** It is not possible to use the Hiero Python API in terminal mode.
--nukex	Launch NukeX instead of standard Nuke.
--studio	Launch Nuke Studio instead of standard Nuke.
--nukeassist	Launch Nuke Assist instead of standard Nuke.
--nc	Launch Nuke in Non-Commercial mode. See the notes farther down the page for more information.
--indie	Launch Nuke in Indie mode. See the notes farther down the page for more information.
-q (--quiet)	Launch the application without displaying the splash screen or startup dialog.

Argument	Result
--safe	Launch the application without loading any plug-ins, Export presets, and so on.
--topdown	Enables the new top-down rendering project setting to render the full frame at the cost of more memory. Although top-down rendering produces the full frame faster, it disables progressive rendering and uses more memory. You may want to use the legacy method if you're only interested in the first few scan lines in the Viewer.
Nuke Studio only	
--log-file	Sets the location of any logfiles created. For example: ./Nuke<version> --studio --log-file /Desktop/log.txt
--log-level	Sets the level of logging produced during operation. For example: ./Nuke<version> --studio --log-level warning Log messages are output to screen unless you specify a **--log-file**. There are four levels of detail, on a sliding scale from minimal to verbose: **error** **warning** (default) **message** **verbose** **Note:** Setting the logging level to **verbose** can produce large log files when **--log-file** is specified.
--workspace	Launch Nuke Studio and apply the specified workspace. Only the workspaces listed in the **Workspace** menu are valid, but this includes any custom workspaces you have saved as **.xml** files in your **.nuke** folder under: /Workspaces/NukeStudio/

Argument	Result
	See Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts for more information on locating your **.nuke** directory by operating system.
<project path>.hrox	Launch Nuke Studio and open the project specified. The path to the project can be absolute or relative, such as: Nuke<version>.exe --studio C:\Users\mags\myProject.hrox Nuke<version>.exe --studio ..\..\mags\myProject.hrox ./Nuke<version> --studio /tmp/myProject.hrox ./Nuke<version> --studio ../../myProject.hrox
<mediaFile path>	Specify the location of media to load on startup. You can import specific files or whole directories: Nuke<version>.exe --studio C:\Users\mags\Media1.mp4 Nuke<version>.exe --studio C:\Users\mags\ ./Nuke<version> --studio /tmp/myfile.mov ./Nuke<version> --studio /tmp/
--script file argv	Run the specified Python script on startup and pass the listed options to the script.

See Command Line Operations for more in-depth information on command line flags.

Nuke Indie

Nuke Indie is a single user version of Nuke Studio using a log-in licensing model. Nuke Indie is designed for personal, small-scale commercial use and includes most of the features of the commercial version of Nuke Studio. See About Nuke Indie for more information.

To launch the application on Linux, do one of the following:

- Double-click the Nuke Indie icon on the Desktop.
- Open the Nuke application directory and double-click the Nuke Indie icon.
- Using a terminal, navigate to the Nuke application directory and enter:
    - **./Nuke17.0 --indie** to launch Nuke Indie.

If you already have a valid license, the interface appears, and a command line window opens. If you don't have a license or haven't installed one yet, proceed to Licensing Foundry Applications.

Nuke Non-commercial

If you want to try out or learn Nuke, you can run Nuke Non-commercial. This version allows you to explore most of Nuke's features, but prevents the commercial use of the application. For more information, see About Nuke Non-commercial.

To launch the application on Linux, do one of the following:

- Double-click the NukeNC, NukeXNC, or Nuke StudioNC icon on the Desktop.
- Open the Nuke application directory and double-click the NukeNC, NukeXNC, or Nuke StudioNC icon.
- Using a terminal, navigate to the Nuke application directory and enter:
  - **./Nuke17.0 --nc --studio** to launch Nuke Studio.
  - **./Nuke17.0 --nc --nukex** to launch NukeX.
  - **./Nuke17.0 --nc** to launch Nuke.

If you have already activated Nuke Non-commercial on the current device, the graphical interface appears, and a command line window opens. If you haven't activated the device yet, proceed to Licensing Foundry Applications.

Uninstalling Apps on Linux

To uninstall the application on Linux, there are a few things you need to do:

1. Navigate to **/usr/local/** and delete the application directory.
2. Delete, rename, or move your **.nuke** directory, if it exists.

   The directory is found in your home directory, by default:
   `/home/<login name>/.nuke`
3. Delete, rename, or move your cached files, which reside in the following directory by default:
   `/var/tmp/nuke`



Note: If you specified an alternate directory using the NUKE_TEMP_DIR environment variable, purge those files as well as the default location. See Nuke Environment Variables for more information.

Licensing Foundry Applications

Foundry applications support two types of license, cloud-based login licensing and local RLM licensing. Login licenses require an internet connection, but local RLM licenses are locked to a single machine or license server and don't require an internet connection.

Login Licensing

Cloud-based login licensing provides quick and easy access to all your licenses regardless of which machine you happen to be using. You can use your login license on any machine with internet access, giving you the freedom to work anywhere.



Note: Local RLM licensing is available alongside login licensing. You are not obliged to switch to the login-based model unless you're using an Indie or Non-Commercial license.

As well as **Individual** licenses, login licensing also supports **Team** licenses that give administrators greater control over who uses the licenses available in the team pool. You can quickly invite a team of artists to collaborate on a project with nothing more than their name and email address.

Individual Login Licenses

Individual Login licenses allow you to run Foundry applications on any machine using your Foundry account email and password. There's no need to manually install a license, the application takes care of all your license administration. If you don't have a Foundry account, you can create one by clicking **Sign up** on this page: https://www.foundry.com/user/register

Once you have a Foundry account, you can request an Individual license for Foundry products:

- from Foundry's Sales Department at sales@foundry.com
- from the product pages on our website, such as https://www.foundry.com/products/nuke
- by launching a Foundry application without a license, selecting **Try or Buy a License**, and then selecting:
    - **Try** - opens a web browser directly to our website's 30-day trial license screen. Enter your Foundry account details or create a new account and follow the on-screen instructions to receive a trial license.
    - **Buy** - opens a web browser directly to our website to purchase a license.

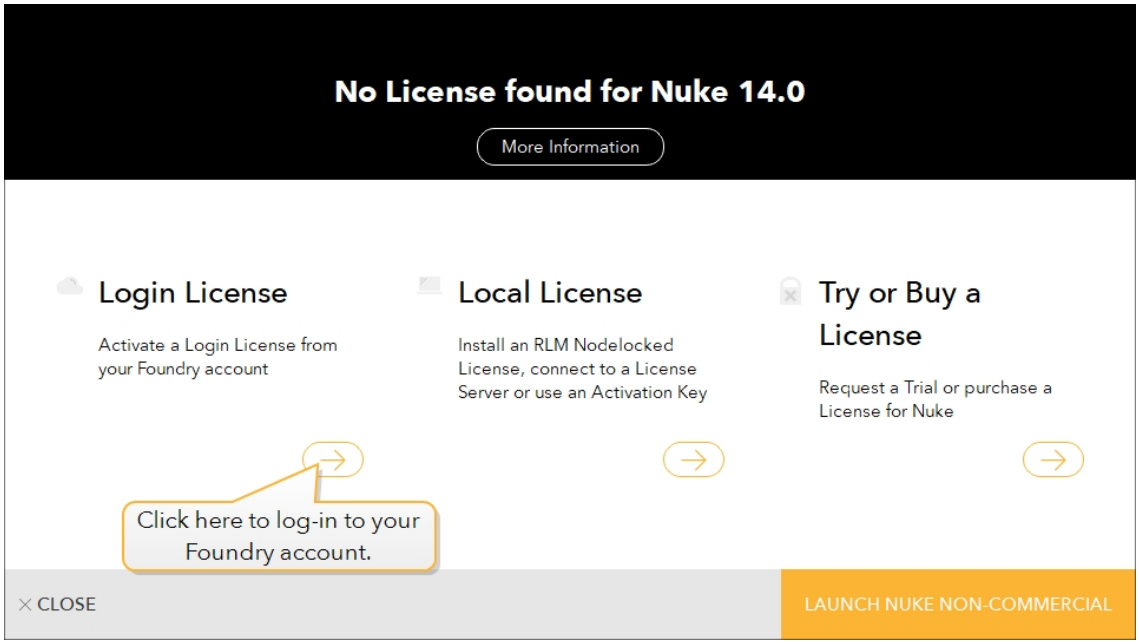
After purchasing a license, download your Foundry application from https://www.foundry.com/product-downloads

Note: If you were given an Activation Key, navigate to https://www.foundry.com/licensing/activate-product, enter the key, and follow in the onscreen instructions.

1. Launch your Foundry application as described in the Install Guide for your Foundry application by operating system.

   The **Licensing** dialog displays.

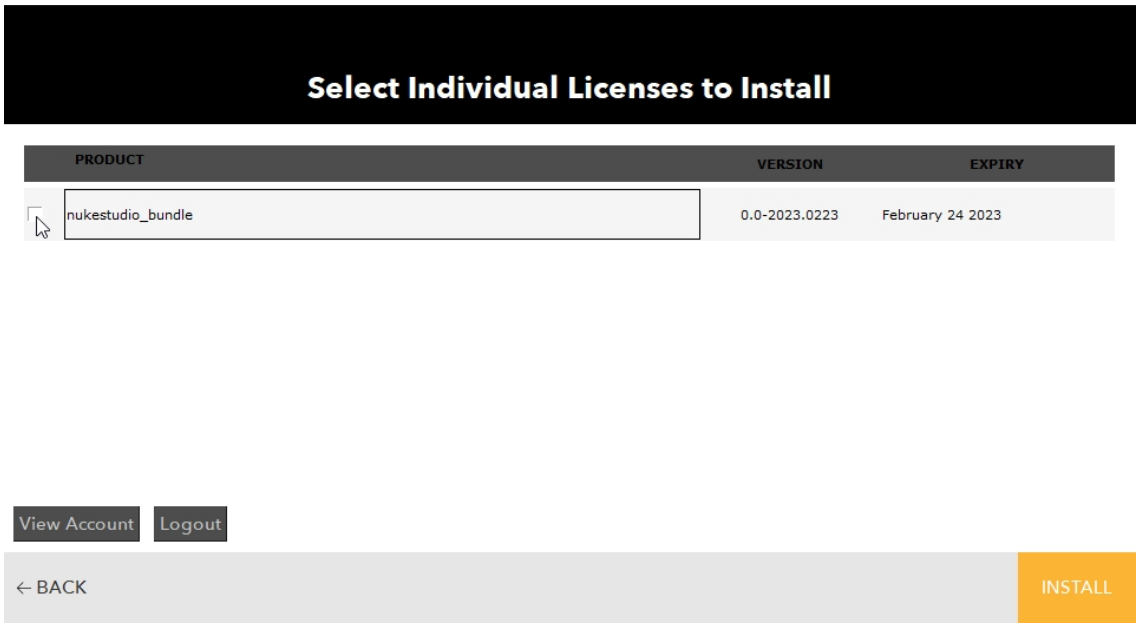
2. Select **Login License** to continue.



3. If this is the first time you've logged in to a Foundry application on this machine, enter your email address and Foundry account password, and then click **Sign in**.

   You don't need to enter your Foundry account information again unless you log out of an application.

4. Select the license you want to install from the list and click **Install**.



5. Click **Done** to close the **Licensing** dialog.

   You can now use your Foundry application.

Note: A license is activated for a period of 37 days. Make sure that you connect to the internet at least once during the activation period to extend your access. You can check the activation period of your license in your Foundry application's **Help** menu.

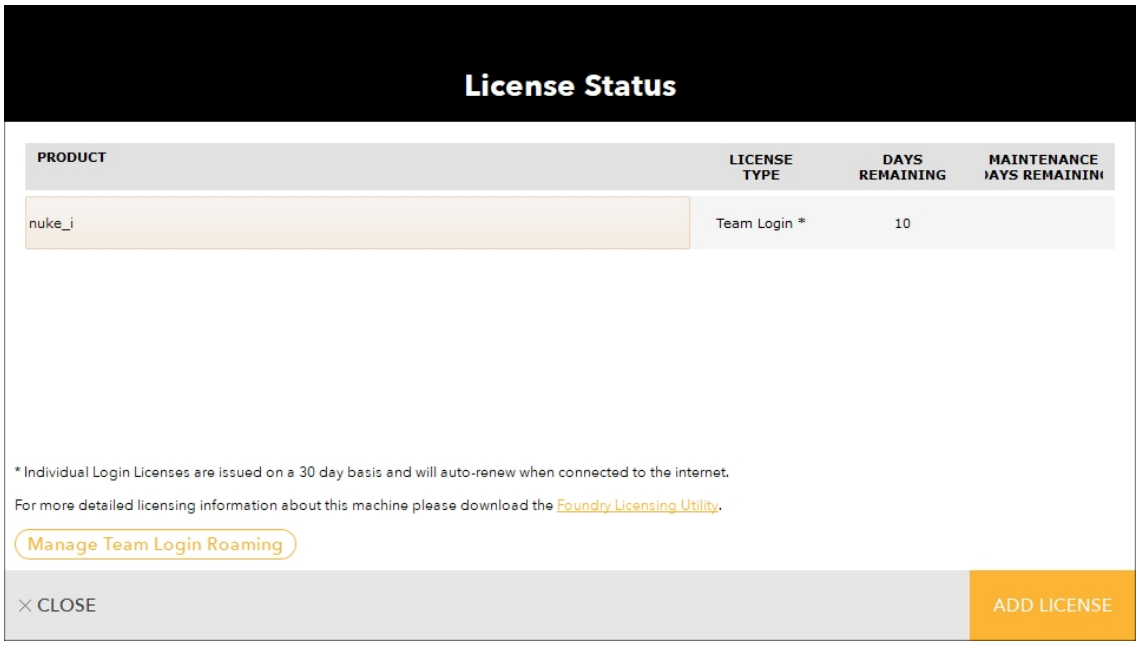
Deactivating an Individual License

An Individual license is active until you deactivate it or sign out of the Foundry Licensing application. Deactivation means you can activate the license again without needing to enter your email address and password.

Note: Individual licenses can be active on up to two machines concurrently. If you want to activate a license on a third machine, you must return an active license first.

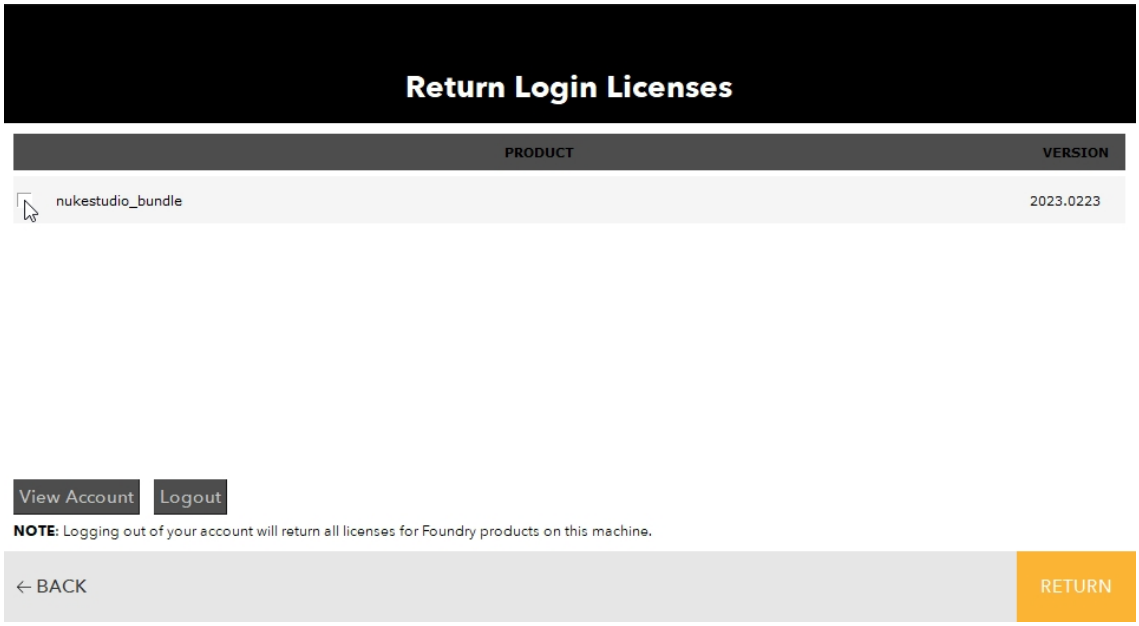
To deactivate a license:

1. Go to your application's **Help** menu and click **Licenses**.
2. The **License Status** dialog displays.



3. Click **Manage Login Licenses**.

   The **Return Login Licenses** dialog displays.



4. Select the license you want to deactivate and then click **Return**,

   OR

   Click **Logout** to return all of the activated licenses on this machine and clear your login details.

Tip: Returning a license allows you to activate it again without entering your Foundry login details.

Team Login Licenses

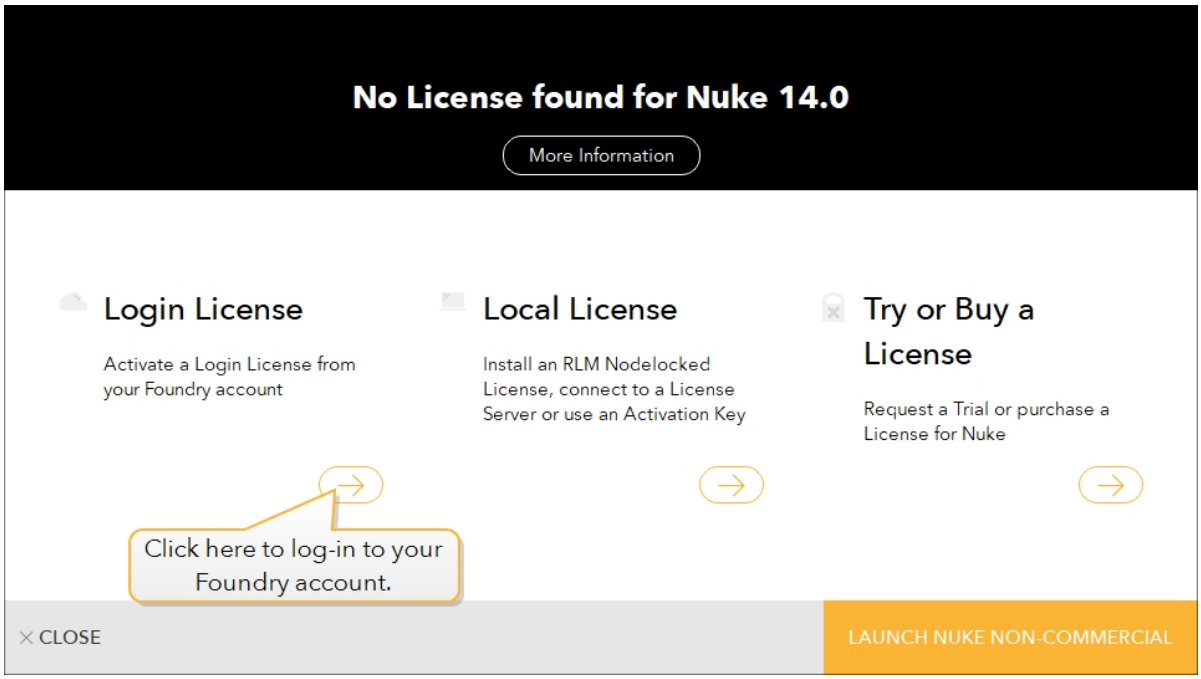
Team licenses also allow you to run Foundry applications using your Foundry account email and password, but the license is tied to team entitlements within the cloud-based login licensing system, rather than an individual's Foundry account. A license is taken from the pool of available licenses when you launch your Foundry application and the license is returned automatically when you close the application. Members of a team don't need to log-in and log-out to take and return licenses.

Team administrators can add and remove users from the team and control license access. See the Foundry Licensing website for more information on Team license administration.

1. Launch your Foundry application as described in the Install Guide for your Foundry application by operating system.

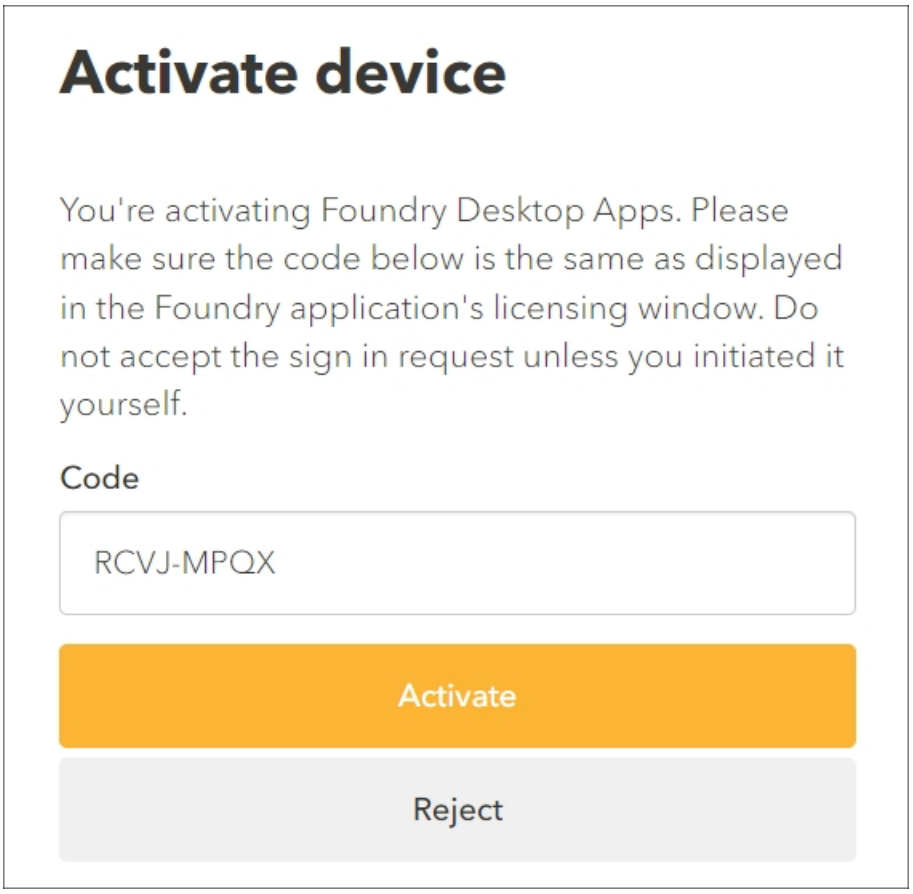
   The **Licensing** dialog displays.

2. Select **Login License** to continue.



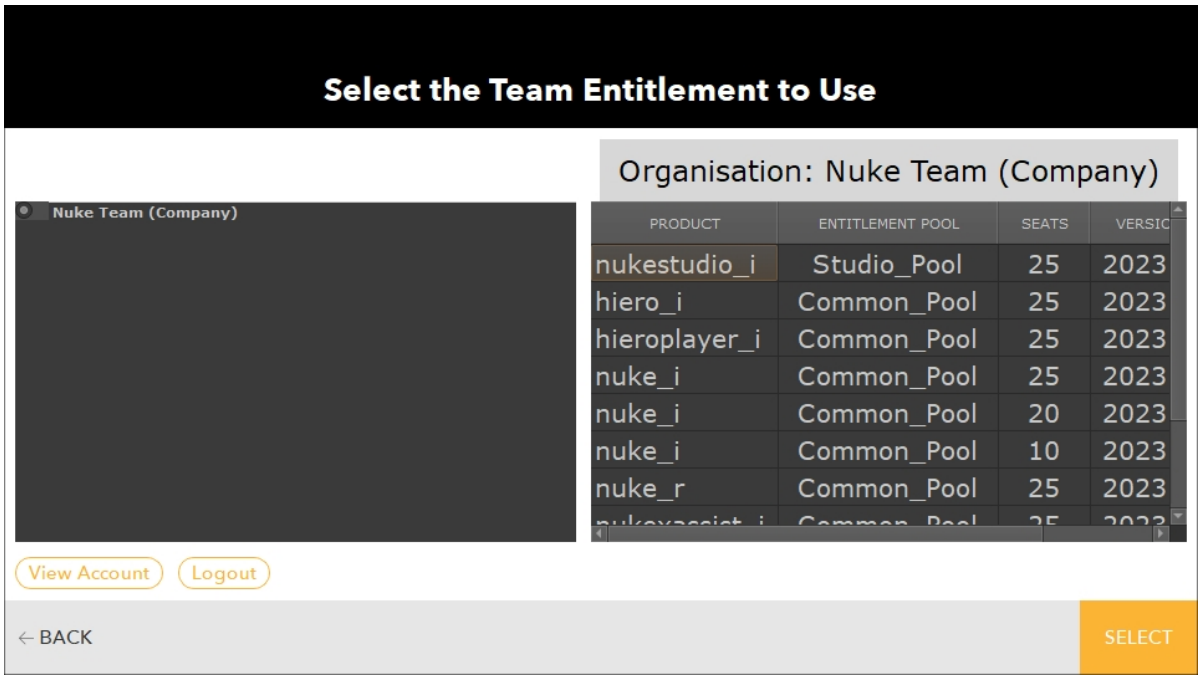
3. Enter your Foundry account information and click **Sign in**.

   You'll be asked to verify an activation code in a web browser.



4. Check that the code in the browser matches the code in the Licensing dialog and then click **Activate**.
5. Select **Team** to access your team entitlements.
6. Click the entitlement from which you want to take a license and click **Select**.

> **Note:** You can use any of the licenses in the selected entitlement, except those reserved by your administrator. See Foundry Licensing for more information on reserving licenses.



7. Click **Done** to close the **Licensing** dialog.

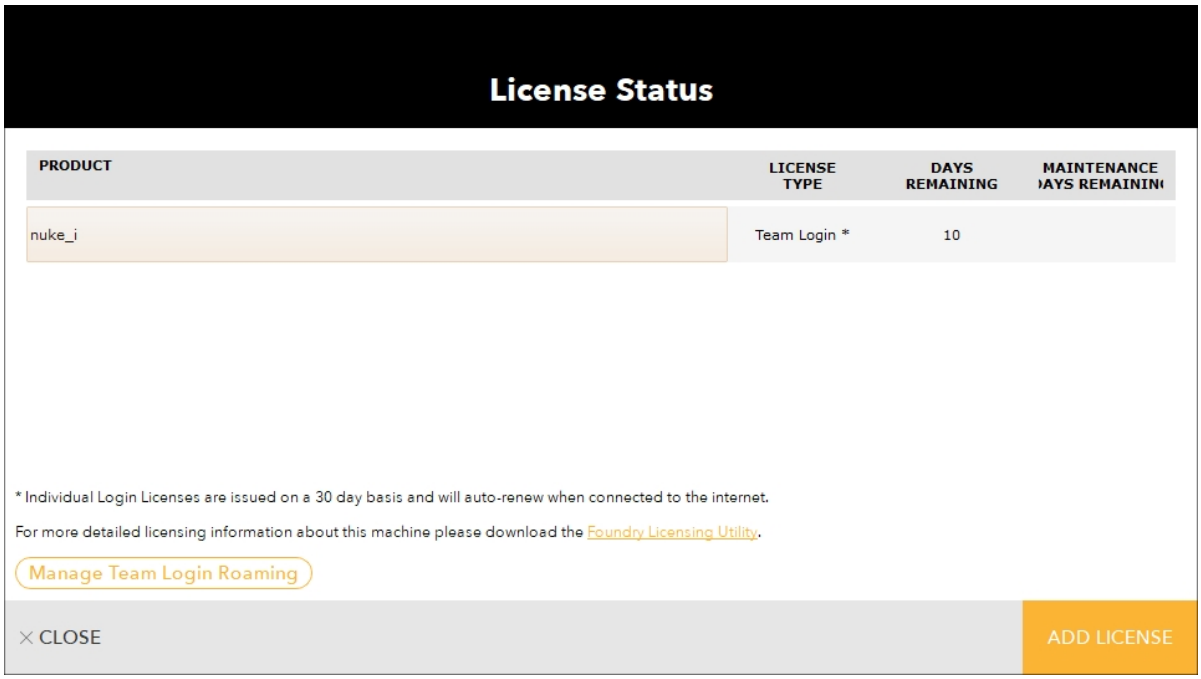
   You can now use your Foundry application.

Using a Different Team License or Logging Out

A Team license is active until you close your Foundry application, you don't need to return the license manually. You can use a license from a different Team on the same machine or you can logout to stop any licenses running on your machine.

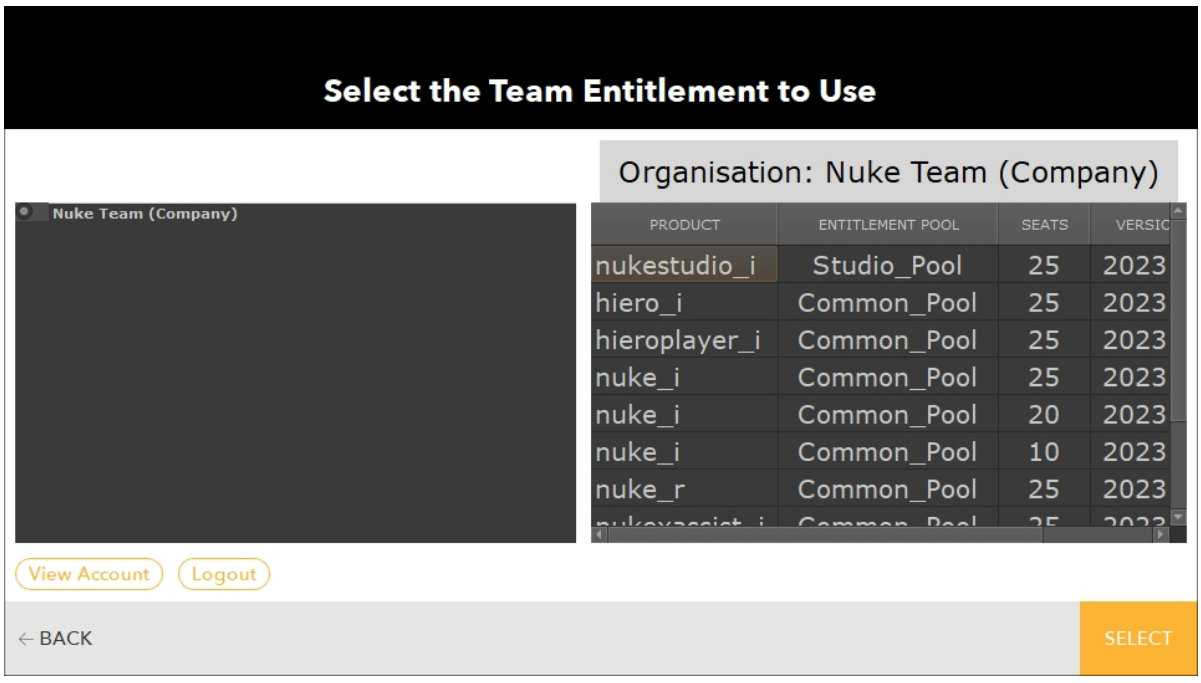
To use a different Team license:

1. Go to your application's **Help** menu and click **Licenses**.
2. The **License Status** dialog displays.



3. Click **Add License** and select **Login License**.

   The **Select the Team Entitlement to Use** dialog displays.



4. Click the entitlement from which you want to take a license and click **Select**.



Note: You can use any of the licenses in the selected entitlement, except those reserved by your administrator. See Foundry Licensing for more information on reserving licenses.

5. Click **Done** to close the **Licensing** dialog.

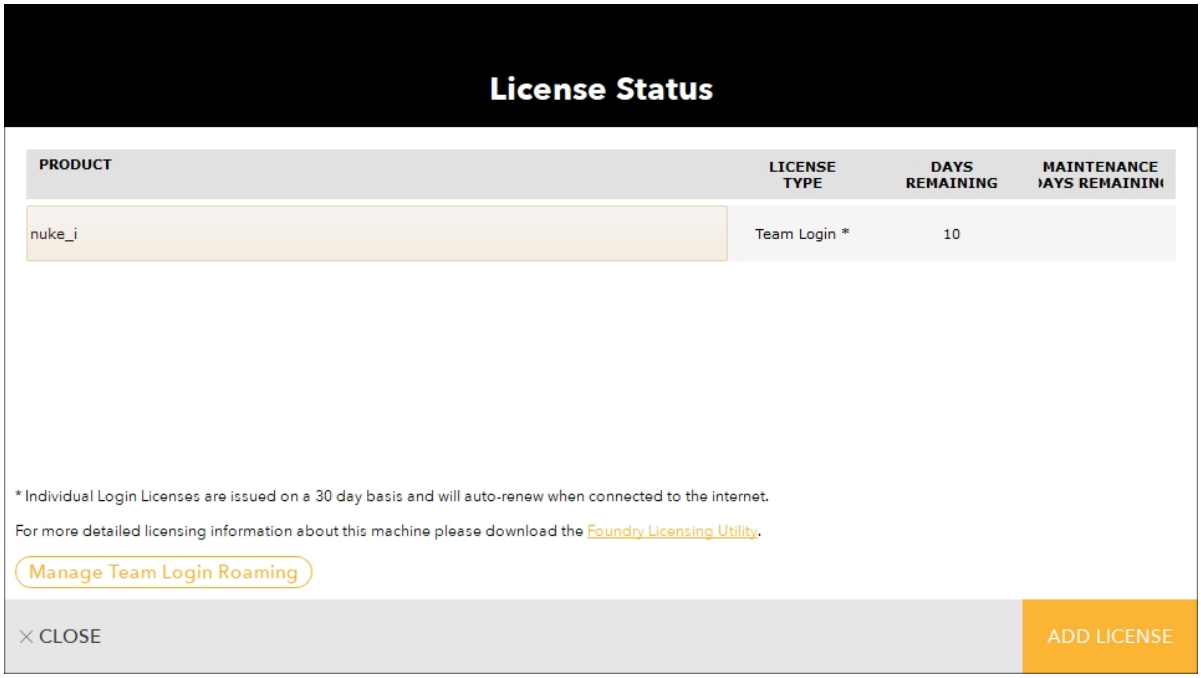
   You can now use your Foundry application with the new Team license.

   When you're finished working, close the application to automatically return the license to the pool.

To logout, return all of the activated licenses on this machine and clear your login details:

1. Go to your application's **Help** menu and click **License**.

   The **License Status** dialog displays.



2. Click **Add License** and then click **Login License**.
3. Click **Team** and then click **Logout** to return all of the activated licenses on this machine and clear your login details.

Local RLM Licensing

Local licensing using RLM software does not required internet access and so it is well suited to individuals that move around a lot and to site licensing where security considerations mean that there is no access to

the internet.



Note: Local RLM licensing is available alongside login licensing. You are under no obligation to switch to the login-based model unless you're using an Indie or Non-Commercial license.

The following local licensing methods are available:

- **Activation Keys** - activation keys allow you to generate your actual product license key, at a later point after purchase, on the machine for which you require the license.

  They are provided for both node locked and floating licenses, and generate the appropriate license type once installed using the application's **Licensing** dialog. Go to the Activate a Product page to use your Activation Key: https://www.foundry.com/licensing/activate-product
- **Node Locked Licenses** - these can be used to license an application on a single machine. They do not work on different machines and if you need them to, you'll have to transfer your license.

  Node locked licenses, sometimes called uncounted licenses, do not require additional licensing software to be installed.
- **Floating Licenses** - also known as counted licenses, enable applications to work on any networked client machine. The floating license is put on the server and is locked to a unique number on that server.

  Floating licenses on a server requires additional software to be installed on the server to manage the licenses and give them out to the client stations that want them. This software is supplied as part of the Foundry Licensing Utility can be downloaded at no extra cost from https://www.foundry.com/licensing/tools

Obtaining Local Licenses

To obtain a local license, you'll need your machine's System ID (sometimes called the MAC address, Host ID, or rlmhostid). Just so you know what a System ID looks like, here's an example: `000ea641d7a1`.



Note: For local floating licenses, you'll need the System ID of the license server, not the machines on which you intend to run the application.

There are a number of ways you can find out your machine's System ID:

- Launch the application without a license, click **Status**, and then scroll down the error report until you see your System ID.
- Download the Foundry License Utility (FLU) from https://www.foundry.com/licensing/tools and run it. Click **System ID** to display your computer's unique identifier.

When you know your System ID, you can request a local license for Foundry products:

- from Foundry's Sales Department at sales@foundry.com
- from the product pages on our website, such as https://www.foundry.com/products/nuke
- by launching the application without a license and selecting **Try or Buy a License** and then selecting:
  - **Try** - opens a web browser directly to our website's 30-day trial license screen. Enter your Foundry account details or create a new account and follow the on-screen instructions to receive a trial license.
  - **Buy** - opens a web browser directly to our website to purchase a license.



Note: By default, if you have installed a temporary license, the application displays a dialog at start-up alerting you to the number of days remaining. If you want to disable this behavior, you can set the FN_DISABLE_LICENSE_DIALOG environment variable to 1 to suppress the warning message about imminent license expiration. See Environment Variables for more information.

To use local license roaming you require a Roaming License. Licenses issued Before July 10th 2020 do not support roaming. New roaming licenses will be issued to customers at Maintenance Renewal time. If you are in Current Maintenance and would like to upgrade your licenses to take advantage of this new feature, please submit a request using this form: https://www.foundry.com/nuke-roaming-license

See Local License Roaming for more information on roaming.

Installing Local Licenses

We recommend using the Foundry Licensing Utility to install local licenses, available from https://www.foundry.com/licensing/tools free of charge. See https://learn.foundry.com/licensing/Content/install.html for more information on installing licenses.

However, you can also install local licenses after starting a Foundry application.

When you start the application before installing a license, a **Licensing** dialog displays an error informing you that no license was available. The installation process is dependent on what type of license you requested:

- **License file** - if you requested a license file, typically **foundry.lic**, this option allows you to browse to the file location and install it automatically. See To Install a Local License from Disk or Local License Text for more information.
- **Activation Key or license text** - if you requested an Activation Key or license by email, this option allows you to paste the key or license text into the Licensing dialog, which then installs the license in the correct directory. See To Install an Activation Key for more information.

- **A floating license** - also known as a server license. if you requested a server license to supply licenses to multiple client machines, this option allows you to enter the server address that supplies the client licenses.



Note: You must install a floating license and additional software on the license server to use this option.

See To Use a Floating Server License for more information.

To Install a Local License from Disk or Local License Text

1. Save the license file to a known location on disk.
2. Launch your Foundry application.

   The **Licensing** dialog displays.
3. Click **Local License** to display the available license installation options and then click **License File**.
4. Click **Open File**, browse to the location of the license file, and then click **Open** to install the license automatically in the correct directory,

   OR

   Copy the license text, paste it over the **Drag & Drop or paste license texta License here** string, and then click **Install License**.

   License text typically looks something like this:

```
LICENSE foundry nuke_i 2015.0929 29-sep-2015 uncounted hostid=000a957bfde5
share=h min_timeout=30 start=29-sep-2015 issued=29-sep-2015 disable=VM _
ck=da32d7372f sig="60P0450MJRP97E3DP
B42C99Y5UAPRMEMGNQ39PG22H4WGH3WFK2KPTXFWJTYR0GYASJBXC0PU8"
```

   The license is automatically installed on your machine in the correct directory.

To Install an Activation Key

1. Launch your Foundry application.

   The **Licensing** dialog displays.
2. Click **Local License** to display the available license installation options.
3. Click **Activation Key** and then enter the Activation Key string into the key field.

   A license key typically looks something like this:

```
nuke-0101-77d3-99bd-a977-93e9-8035
```

4. Click **Activate Key**.

   The license is automatically installed on your machine in the correct directory.



Note: Activation Keys require an internet connection. If you access the internet through a proxy server and cannot connect to the activation server, you may get an error dialog prompting you to either:
Click **Use Proxy** to enter the proxy server name, port number, username, and password. This enables the application to connect to the activation server and obtain a license. The license is then installed automatically, or
Click on the web link in the dialog and use the System ID (also known as hostid) provided to manually activate and install a license.

To Use a Floating Server License

If you requested a server license from Foundry, you receive your license key (foundry.lic) in an email or download. You should also receive a link to the Foundry License Utility (FLU) application to help you install the license key on the license server machine. The server manages licenses for the client machines on your network.

To install a connection to your local license server:

1. Launch your Foundry application.

   The **Licensing** dialog displays.
2. Click **Local License** to display the available license installation options.
3. Click **Server** and then enter the port and server name or IP address of the license server.

   The format for the server name is: `<port>@<servername>`, for example, `30001@red`.
4. Click **Connect**.
5. 

   Note: You must perform these steps on each client machine that requires a license from the server. The client machines do not need the server tools installed.

Local License Roaming

License roaming allows you to temporarily check out a server license so you can work offline. Licenses are checked out and returned through the **Help** > **License** menu or by setting environment variables.



Note: License Roaming is not currently supported by Katana and Mari or with Production Collective and EDU Collective licenses.

There are a few important things to take into consideration with license roaming:

- Roaming is only available to Maintenance customers and requires a new license from Foundry.
- Roaming and returning licenses requires a connection to the license server.
- Roamed licenses are valid for a maximum of 30 days.
- Roamed licenses are only valid on the machine that took the license from the server.
- The number of roaming licenses available on the server decreases by one until you return the license.

Installing the Local Roaming License

To activate roaming licenses you need to request a new roaming license and install the license on your license server.

See Licensing Foundry Applications and To Use a Floating Server License for more details on requesting and installing a server license.

How to Roam a Local License

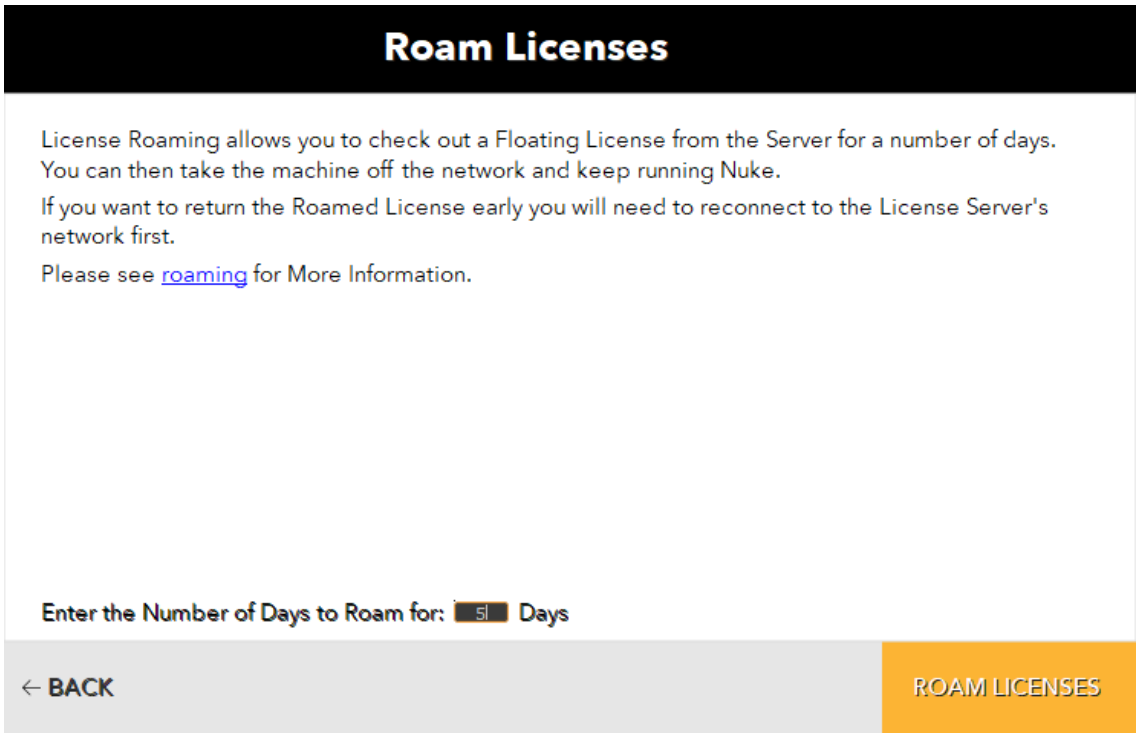
1. Launch the application you want to roam, making sure it has a connection to your license server.
2. Go to **Help** > **License**.

   The **Licensing** dialog displays.
3. Click **Manage Roaming**.

   The **Roam Licenses** panel displays.
4. Enter the number of days you want the roamed license to last, from 1 to 30 days.



Note: The maximum number of roam days may be less than 30 depending on your license.



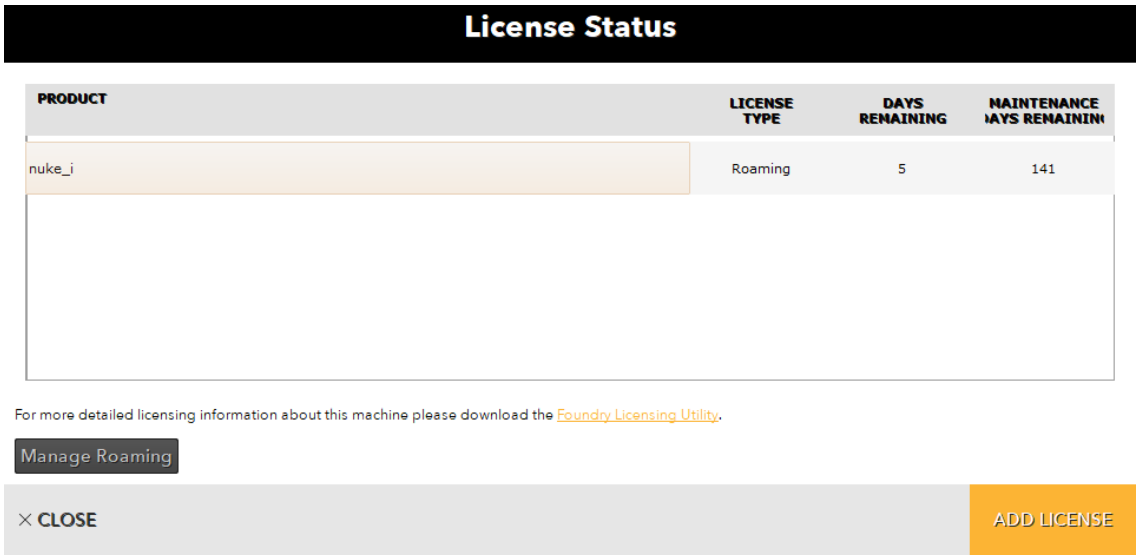
5. Click **Roam Licenses**.

   The panel displays the number of roaming days remaining.
6. Click **Close**.

How to Return a Roamed Local License

1. Launch the application using the roamed license, making sure it has a connection to your license server.
2. Go to **Help** > **License**.

   The **Licensing** dialog displays the license type and number of roam days remaining.



3. Click **Manage Roaming**.
4. Click **Return Roam**.
5. Exit the application to return the roaming license to the server.

Roaming Using Environment Variables

You can automate some of the aspects of roaming licensing using the **RLM_ROAM** environment variable supplied by Reprise Software, the third-party vendor of our licensing administration software.

For more information see *How to use Roaming Licenses* in the *RLM Administration Guide*: https://www.reprisesoftware.com/RLM_License_Administration.pdf

Further Reading on Licensing

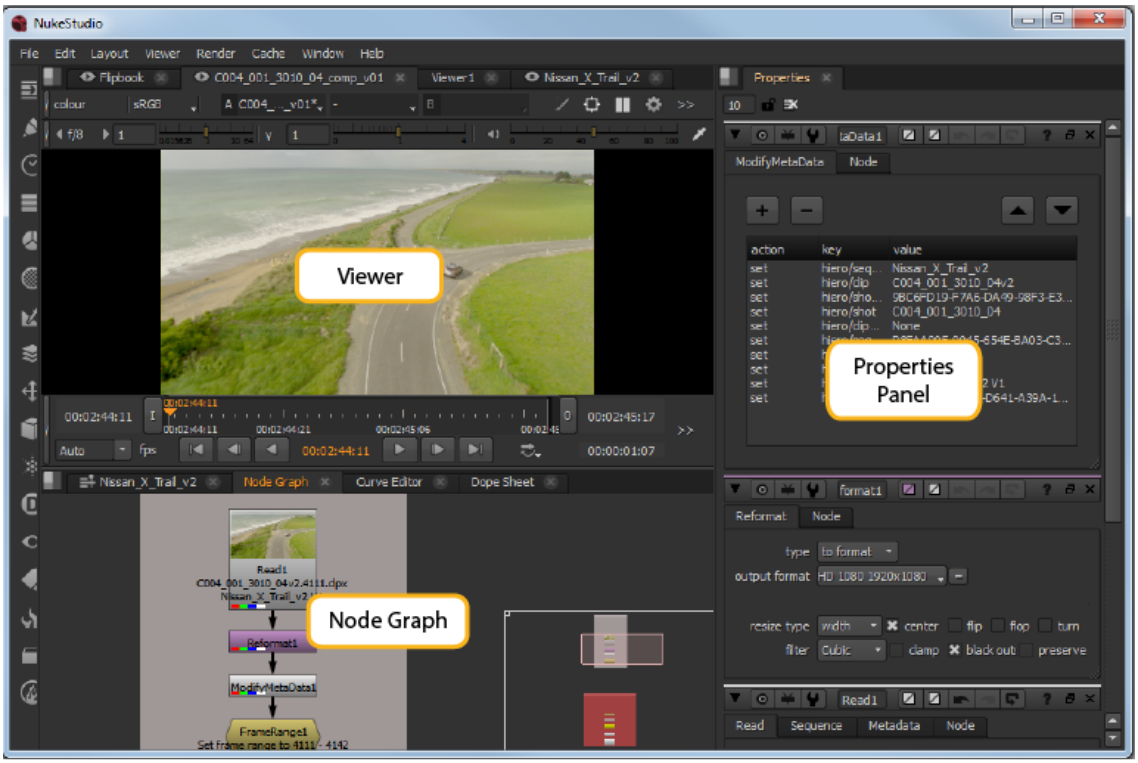
There is a lot to learn about licenses, much of which is beyond the scope of this manual. For more information on cloud-based login licensing and local RLM licensing, go to our Licensing website: https://learn.foundry.com/licensing/

Nuke Studio Environments

The Compositing Environment

You can use the Compositing environment to perform node-based compositing with a choice of different VFX tools, manage color grading, review your script, and render out your script. See Using the Compositing Environment for more information.

By default, there is a Node Graph panel in the lower-left corner, a Viewer panel in the top-left corner, and a **Properties** panel on the right.

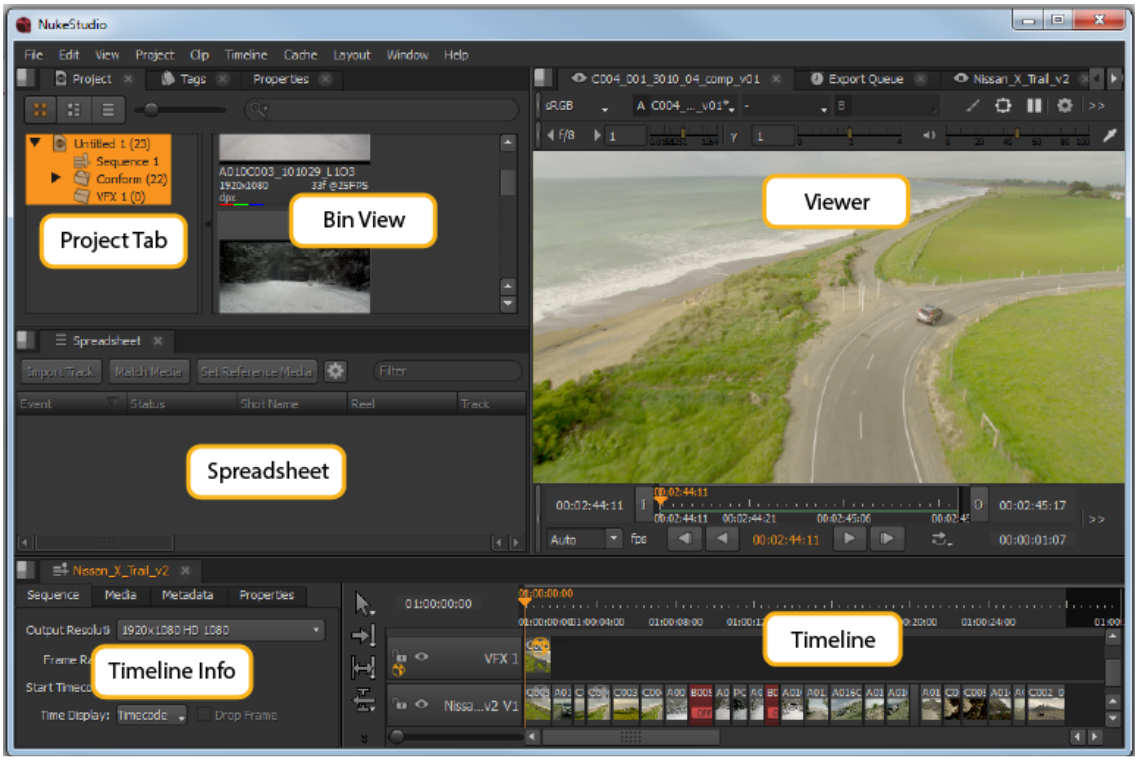


Node Graph	The Node Graph is where you add nodes and build your node tree.

Properties Panel	When you add a node to the Node Graph, its properties appear in the **Properties** panel on the right.
Viewer	To check the result, you can view the output in a Viewer.

The Timeline Environment

The Nuke StudioTimeline environment allows you to conform, create Nuke Comps, add soft effects, perform timeline-based editing, export your project, and view and edit metadata and properties. See Using Nuke Studio's Timeline Environment for more information.



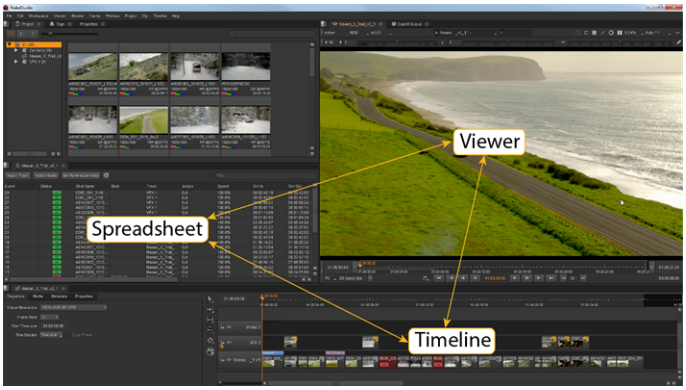
Project Tab	You can manage all aspects of your projects and bins in the **Project** tab.
Menu Bar	Use the Menu bar to access Nuke Studio's dropdown menus.
Bin View	The bin displays the contents of any selected **Project** tab.
Viewer	You can display and review your media in the Viewer.
Timeline Info	The timeline info displays the current timeline's sequence, media, and metadata

	information.
Editing Tools	There is a comprehensive set of editing tools provided in Nuke Studio. See Timeline Editing Tools for more information.
Spreadsheet Tab	Use the **Spreadsheet** tab to display the contents of the timeline in spreadsheet form. Note that the spreadsheet and timeline are linked, mirroring any selections made.
Timeline	The timeline displays the current track, including all shots and any effects that have been added.

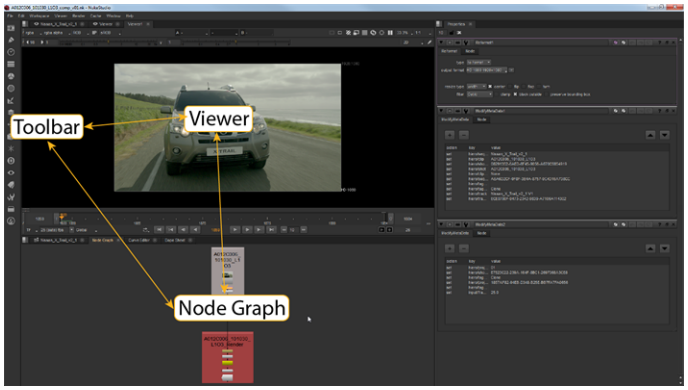
How the Panels Link

In Nuke Studio, some panels are linked together so that the linked panel tabs are automatically displayed at the front of a panel, when one of linked group is selected. This makes it quick and easy to switch between working in the Timeline and Compositing environments.

The Timeline linked group includes the spreadsheet, the Timeline environment Viewer, and the timeline. The Compositing linked group includes the Node Graph, the Compositing environment Viewer, and the node Toolbar.



Timeline Linked Group



Compositing Linked Group

For example, when you are in the Compositing environment and you select the a spreadsheet from the **Properties** content menu. The spread sheet appears as a new tab in the **Properties** pane, and automatically opens the timeline, and the timeline Viewer.



Note: If more than one tab in the linked group is in the same pane, the most recently viewed tab is displayed at the front of the pane.

Shared panels take precedence over unshared panels. Shared panels include Curve Editor, Dope Sheet, Pixel Analyzer, Scopes, and Progress bars. The **Properties** pane and Script Editor are also shared panels, but differ slightly in that they show workspace-specific content. For example, the **Properties** panels shows soft effect keyframes and Node Graph keyframes depending on the currently selected workspace.

Panel Focus and Keyboard Shortcuts

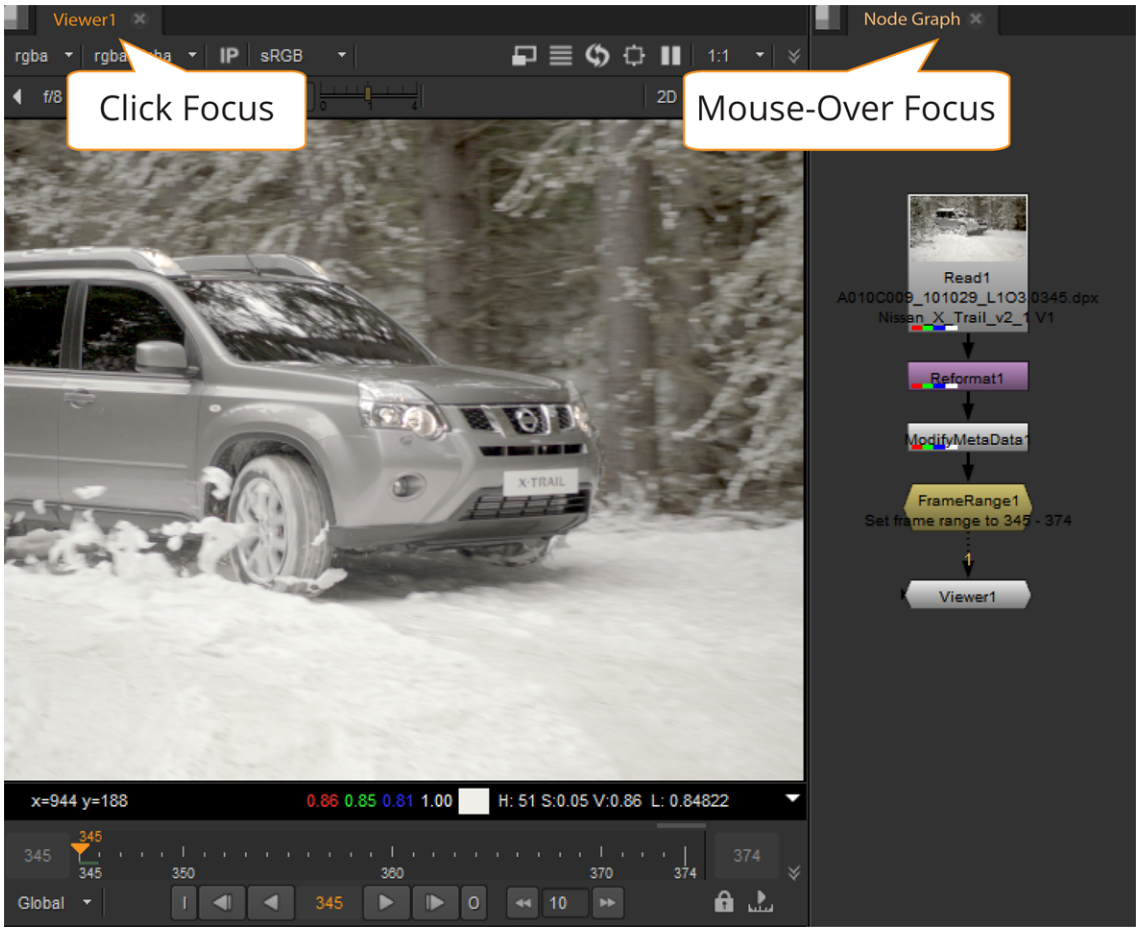
Nuke Studio deals with panel focus in two ways: click focus and mouse-over focus. Click focus defines the main panel where keyboard events are registered and mouse-over focus allows you to temporarily override that focus. If the mouse-over focus is centered on a panel that doesn't recognize a particular keyboard event, the event falls back to the click focus panel.

A good example of click focus versus mouse-over focus is between Viewers and the Node Graph:

- If a Compositing Viewer has click focus and you press **B**, only the blue color channel is displayed in the Viewer. If the mouse-over focus resides on the Node Graph and you press **B**, the click focus is overridden and a Blur node is added to the Node Graph.
- If a Compositing Viewer has click focus and you press **,** (comma), the **gain** is reduced in the Viewer. If the mouse-over focus resides on the Node Graph and you press **,** (comma)**,** the gain is still reduced in the Viewer because the Node Graph has no **,** (comma) equivalent keyboard shortcut.

Some areas of the interface retain focus no matter where the mouse-over focus resides, such as the **message** control in the Text node's **Properties** panel and the **Filter** field in the **Project** panel.

Click focus is shown in the interface with a bright orange highlight in the panel name and mouse-over focus is shown as a muted orange highlight.



Default Workspaces

There are six default workspaces in Nuke/NukeX and Nuke Studio. By default, Nuke/NukeX opens in the **Compositing** workspace and Nuke Studio opens in the **Finishing** workspace. To change the workspace, you can do either of the following:

- Select **Workspace** from the top menu bar and then select the required workspace.

OR

- Use the keyboard shortcuts to open the required workspace. Press **Shift** and the required workspace keyboard shortcut, depending on mode:

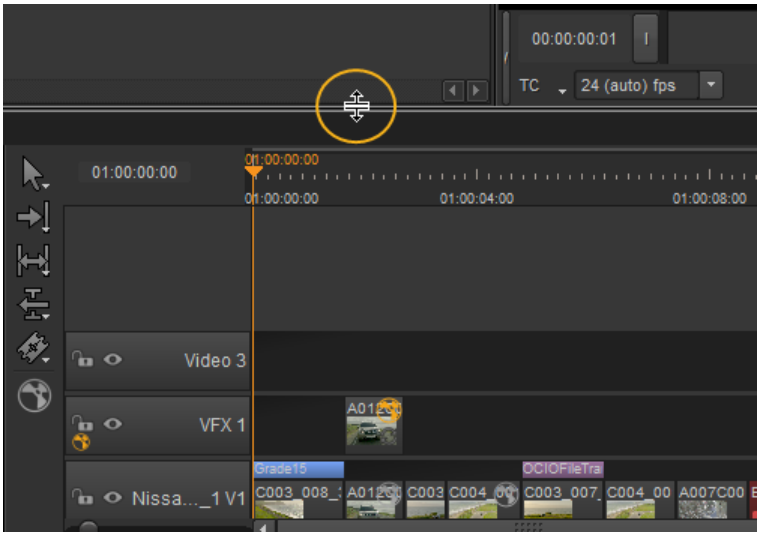
Keyboard Shortcut	Nuke/NukeX	Nuke Studio
Shift+F1	Compositing	Conforming
Shift+F2	Large Node Graph	Editing

Keyboard Shortcut	Nuke/NukeX	Nuke Studio
Shift+F3	Large Viewer	Reviewing
Shift+F4	Scripting	Timeline
Shift+F5	Animation	Finishing
Shift+F6	Floating	Compositing
Shift+F7	Monitor Out	Monitor Out
Shift+F8	N/A	Sync Session (Beta)

Customizing Workspaces

Customizing Panes

You can resize and split panes to make more room for different elements on the screen. To resize a pane, drag the divider line of the pane into a new location.



You can split a pane by clicking on the content menu button  in the top-left corner of the pane, and then selecting **Split Vertical** or **Split Horizontal** from the menu that opens.

Note: Pressing and holding the **spacebar** brings up the right-click menu for that pane, where available.

Moving the Toolbar

You can move the Toolbar into a new position by adding a new panel for it, hiding the panel name and controls, and resizing the panel. For more information on how to do this, see Adding Tabs, and Hiding Tab Names and Controls.

Adding Tabs

When you can't fit more elements into your display, you can use tabs to save space. You can also use tabs to move the Toolbar into a new location.

You can add a tab by clicking on the content menu button  in the top-left corner of the pane, and then selecting the type of tab you want to add. For example, you can add **Node Toolbar**, **Node Graph**, **New Viewer**, or **Script Editor**. The new tab is added on top of the existing tabs.

To move tabs, click on the name of the tab and drag it to a new position inside the same pane or in another pane.

You can close tabs again by clicking the **X** in the top-right corner of the tab you want to close.

Note: Closing a linked tab closes all associated tabs. If you hold **Alt** while closing a linked tab, it only closes that tab.

Soloing Tabs

You can choose to solo a tab by either right-clicking on the tab name or clicking the content menu, and then selecting **Solo Tab**. This automatically closes any other open tabs in the same pane, except for the one you have chosen to solo.

Floating Windows

You can turn tabs and panes into floating windows and vice versa. To turn a tab or pane into a floating window, click the content menu button  in the top-left corner in the tab or pane you want to float, and

then select **Float Tab** or **Float Pane**. You can also float tabs by either clicking **Ctrl**/**Cmd**+click on the tab name, or right-clicking on the tab name and select **Float Tab**.

To change a floating window into a tab or pane, click on the tab or pane name in the floating window and drag it to where you want it to dock. You can close floating windows by clicking the **X** button in the top-right corner of the tab or pane.

Maximizing Windows

To make a window fullscreen, first ensure the window you want to make fullscreen is active, and then press **Alt**+**S**. This could be the main application window or a floating Viewer. Making it fullscreen removes the window borders. You can also maximize tabs and panels by pressing spacebar.

Hiding Tab Names and Controls

You can hide the names and control buttons of tabs, as you may not need them with all panels, such as the Toolbar panel. To hide the names and controls on tabs, click the content menu button  in the top-left corner of the tab, and disable **Show Tabs**.

You can show the names and controls on tabs again by moving the cursor over the top of the pane area until the top edge of the pane highlights, right-click to open the content menu, and select **Show Tabs**.

Saving and Loading Workspaces

After you have customized a workspace and you are happy with it, you can save it by selecting **Workspace** > **Save Workspace...** You are then asked to name it. After saving it, your custom workspace appears in the **Workspace** dropdown under the existing default workspaces. Select it from the dropdown to load it.

Workspaces are saved in your **.nuke** file under **Workspaces** > **Nuke** or **NukeStudio**, depending on which Nuke mode you're currently using.



Note: The location of the **.nuke** file varies by platform. See Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts for more detailed information.

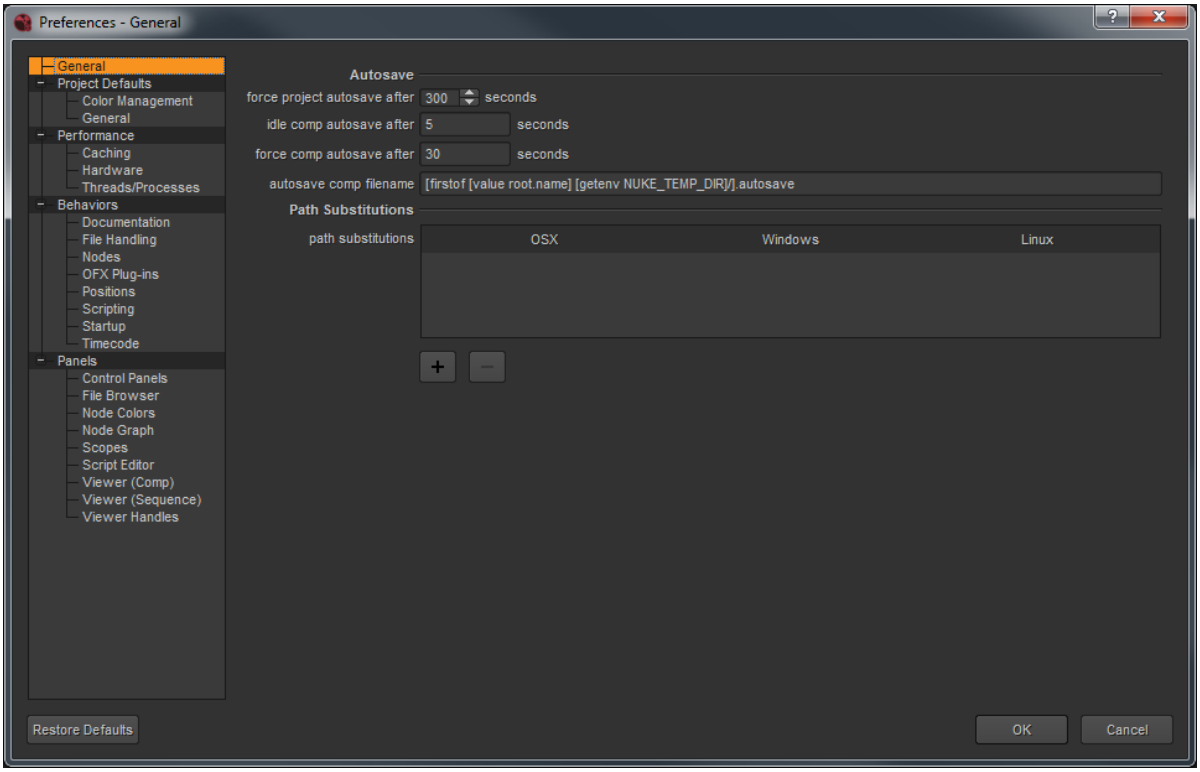
Setting the Startup Workspace

When you launch Nuke, it opens in the workspace set in the Preferences. You can change the startup workspace to any other default workspace or a custom workspace, by doing the following:

1. Open the **Preferences** dialog by pressing **Shift**+**S**.
2. In the **Behaviors** section, select **Startup**.
3. Use the **startup workspace** dropdown to select the workspace you want Nuke to load on startup.

Setting Preferences

Open the **Preferences** dialog by pressing **Shift**+**S**. The preferences available depend on which mode Nuke is launched in.



Changing Preferences

The function of each preference is described under Appendix A: Preferences.

When you make a change to a preference, in most cases, the interface registers the change immediately (for example, an interface element displays in the new color). Some preference changes, such as **Performance** > **Hardware** > **default blink device**, require you to restart Nuke Studio, for the changes to take effect.

Saving Preferences

Nuke stores your preference settings in a file called **preferences17.0.nk**, which resides in your **.nuke** directory. The location of this is dependent on your operating system.

- Linux: /home/login name/.nuke
- Mac: /Users/login name/.nuke
- Windows: drive letter:\Users\login name\.nuke



Note: On Windows, **.nuke** resides in the directory pointed to by the HOME environment variable. If this variable is not set (which is common), the **.nuke** directory is under the folder specified by the USERPROFILE environment variable.

Each Nuke user can maintain his or her own unique settings. After making a change in the **Preferences** dialog, you can simply click **OK** to save and close your preferences. If you click **Cancel**, any changes that you made are not saved.

To save your preferences

Make the desired changes inside the **Preferences** dialog, then click **OK**. Nuke writes the new settings to **preferences17.0.nk** file, which you can find in the .nuke directory:

- **On Windows**: The .nuke directory can be found under the directory pointed to by the HOME environment variable. If this variable is not set (which is common), the .nuke directory is under the folder specified by the USERPROFILE environment variable - which is generally of the form *drive letter*:\Users*login name*\

  To find out if the HOME and USERPROFILE environment variables are set and where they are pointing at, enter %HOME% or %USERPROFILE% into the address bar in Windows Explorer. If the environment variable is set, the folder it's pointing at is opened. If it's not set, you get an error.
- **On Mac**: /Users/*login name*/.nuke
- **On Linux**: /users/*login name*/.nuke

Your new preferences remain in effect for the current and all subsequent sessions.

Resetting Preferences

To reset any changes you made simply click **Restore Defaults** in the bottom-left of the **Preferences** dialog. You can reset the preferences to default by deleting the **preferences17.0.nk** file. After doing this, the next time you launch Nuke, it rebuilds the file with the default preferences.

Using the Compositing Environment

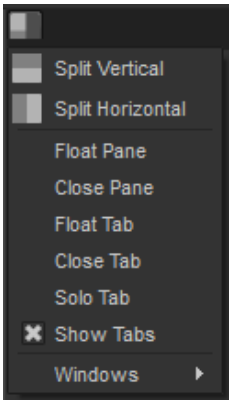
These pages are designed to help you learn how to use the Compositing environment including working with nodes, using the Toolbar, and using the **Properties** panel.

Toolbar, Menu Bar, and Content Menus

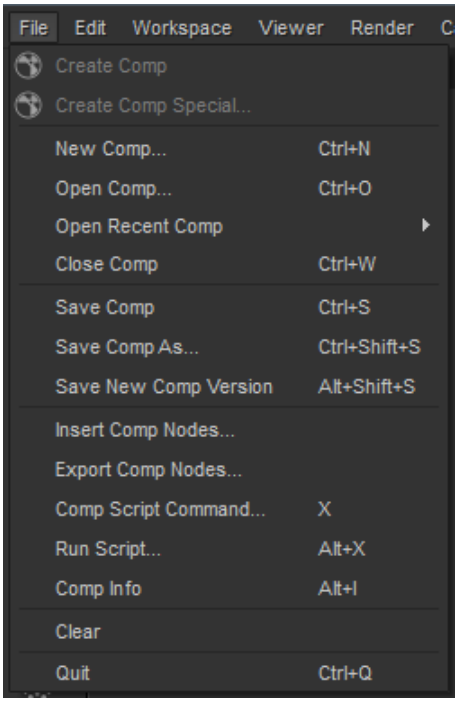
The Toolbar is located on the left-hand side of the Viewer in the Compositing environment. It consists of a number of menu icons. The different nodes are grouped under these icons based on their functions. You use the Toolbar to add nodes to the Node Graph.

The menu bar is located on top of the Nuke window (in all workspaces). Its menus, such as the **File** or **Edit** menu, let you perform more general actions related to the project or script, the Viewers, or editing, rather than certain individual nodes.

In addition to the Toolbar and the menu bar, you should also familiarize yourself with the content menus. They are the gray checkered boxes in the top-left corner of each pane. If you click on the box, a menu opens as shown in the image below. You can use the options in the menu to customize the workspace.



Finally, to work faster, you can right-click on the different panels to display a menu with options related to that particular panel.



Using the Toolbar

The Compositing environment's Toolbar includes the following icons:

Icon		Functions
	Image	Image Read and Write nodes, built-in Nuke elements, and Viewer nodes.
	Draw	Roto shapes, paint tools, film grain, fills, lens flares, sparkles, and other vector-based image tools.
	Time	Retiming image sequences.
	Channel	Channel management.
	Color	Applying color correction effects.

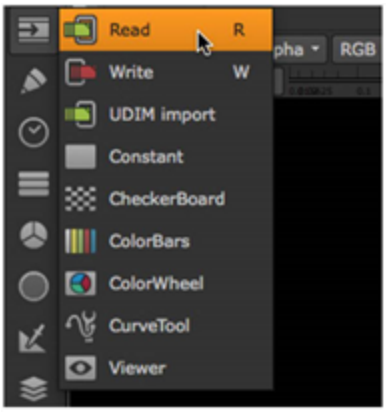
Icon		Functions
	Filter	Applying convolve filters, such as blur, sharpen, edge detect, and erode.
	Keyer	Extracting procedural mattes.
	Merge	Layering background and foreground elements.
	Transform	Translating, scaling, tracking, and stabilizing elements.
	3D	3D compositing nodes and tools.
	Particles	Creating, spawning, and editing particles.
	Deep	Creating, merging, and editing deep images.
	Views	Nodes for working with views and stereoscopic or multi-view material.
	Metadata	Viewing, editing, and comparing image metadata.
	ToolSets	Creating, deleting, and managing tool sets.
	Other	Additional operators for script and Viewer management.
		Any installed plug-ins and custom menus that do not have their own icon.

Note: Nuke Assist does not support all the toolbars available in Nuke. See Nuke Products for more information.

To display a tool tip that explains the icon's function, move the cursor over the icon.



To make selections from the Toolbar, click on an icon and select an option from the menu.



Working with Nodes

Nodes are the basic building blocks of any composite. You can create a new compositing script by inserting and connecting nodes to form a network of operations. These operations concatenate and allow you to manipulate your images.

Note: Not all nodes are supported in all versions of products. For a run down of supported nodes, see Meet the Nuke Product Family.

Adding Nodes

You can add nodes using the Toolbar, the Tab menu, or the right-click menu. When you add a node, Nuke automatically connects it to the currently selected node.

Using the Toolbar

1. Select the existing node that you want the new node to follow by clicking on it.
2. Click an icon on the Toolbar and select a node from the menu that appears. For example, if you want to add a Blur node, click the **Filter** icon and select **Blur**.

Note: You can press the middle-mouse button on a menu icon to repeat the last item used from that menu. For example, if you first select a Blur node from the **Filter** menu, you can then add another Blur node by simply pressing the middle-mouse button on the **Filter** icon.

Using the Tab Menu

Tip: See Using the Tab Menu for more detailed information.

1. Select the existing node that you want the new node to follow by clicking on it.
2. Press the **Tab** key and start typing the name of the node you want to create.
   This opens a prompt displaying a list of matches.
3. To select the node you want to add from the list, you can either click on it, or scroll to it with the **Up** and **Down** arrow keys and press **Return**.

Note: To add the last node created using this method, simply press **Tab** and then **Return**.

Using the Right-Click Menu

To add a node using the right-click menu, do the following:

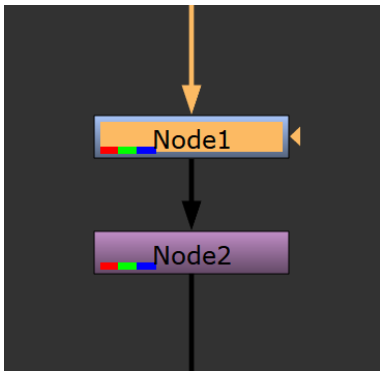
1. Right-click on an existing node that you want the new node to follow.
2. From the menu that opens, select the node you want to add.

Note: You can also add nodes using keyboard shortcuts. Most menus in the Toolbar include a note of the relevant keyboard shortcut next to the item in question.

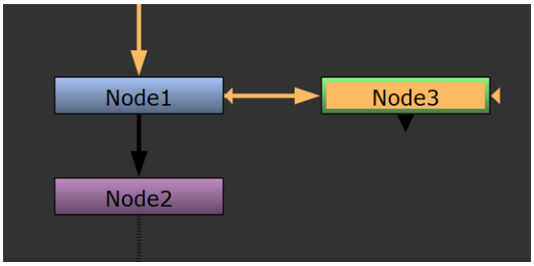
Adding a New Branch of the Node Tree

To add a node in a new branch of the Node Tree, do the following:

1. Select the existing node that you want the new node to follow by clicking on it.



2. Hold down **Shift** and create the node using the Toolbar, Tab menu, or right-click menu. To add a node in a new branch with the Tab menu, press the **Tab** key first, then hold down **Shift** when selecting the new node.



The node is added after the selected node in a new branch of the node tree.

Selecting Nodes

Nuke offers a number of options for selecting nodes. Selected nodes are highlighted in a color that is defined in the **Preferences** dialog. See Appendix A: Preferences for more information.

Type of Selection	How to Make the Selection
Select a single node	To select a single node, simply click on it.
Select multiple nodes	To select multiple nodes, you can either press **Shift** while clicking on each node you want to select, or click and drag in the workspace to draw a marquee around the nodes you want to select.
Select all upstream nodes	Press **Ctrl/Cmd** while dragging a node. Nuke selects all nodes that feed data to the selected node. You can also **Ctrl/Cmd**+**Shift**+click to select more nodes without clearing the current selection.
Select all nodes in a node tree	Click on a node in the Node Graph and select **Edit** > **Select Connected Nodes** (or press **Ctrl/Cmd**+**Alt**+**A**). This selects all nodes in the node tree, whether they are

Type of Selection	How to Make the Selection
	upstream or downstream from the current node. Nodes in any other node trees are not selected.
Select all nodes in a script	Select **Edit** > **Select all** (or press **Ctrl**/**Cmd**+**A**).
Select nodes by name	1. Choose **Edit** > **Search**, or press the forward slash (**/**). A dialog is displayed. 2. Type an alphanumeric string that is included in the names of the nodes you wish to select. Click **OK**. **Tip:** When typing the above alphanumeric search string, you can use asterisks (*****) and question marks (**?**) as wild cards. An asterisk stands for multiple alphanumeric characters. A question mark represents just one character.
Invert a selection	Select **Edit** > **Invert Selection**.

Replacing Nodes

To replace a node in Nuke, simply hold down **Ctrl**/**Cmd** and create a new node. See below for more details.

To Replace One Node with Another

1. In the Node Graph, select the node you want to replace by clicking on it.
2. Hold down **Ctrl**/**Cmd** and create a new node using the Toolbar, a right-click menu, or the Tab menu. To replace a node with the Tab menu, press the **Tab** key first, then hold down **Ctrl**/**Cmd** when selecting the new node.

   The new node replaces the selected node in the Node Graph.

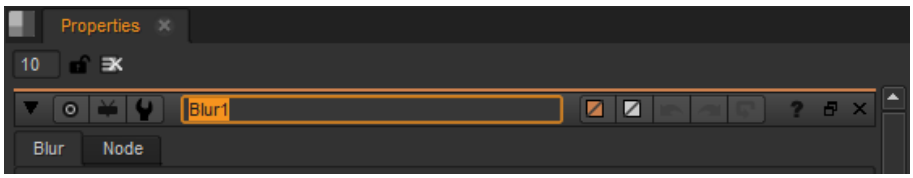
Note: Note that you cannot replace nodes in this manner if you are using a keyboard shortcut (such as **B** for the Blur node) to create the new node.

Renaming Nodes

There are a couple of ways to rename a node in Nuke.

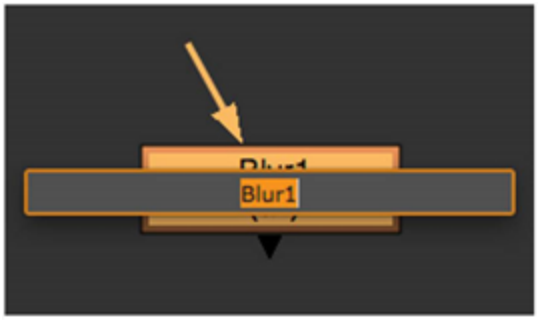
To rename a node, you can either:

1. Double-click on the node to open its properties panel.
2. In the title field on top of the **Properties** panel, you should see the current name of the node. Delete that name and enter a new name in its place,



OR

1. Click on the node in the Node Graph to select it.
2. Press **N**.
3. Enter a new name for the node in the rename field that appears on top of the node.



Editing Nodes

To copy, paste, and perform other editing functions in the Node Graph, you can use the standard editing keys (for example, **Ctrl/Cmd+C** to copy, and **Ctrl/Cmd+V** to paste). You can copy nodes to files or memory. Copied nodes inherit the values of their parent, but these values, unlike those in cloned nodes, are not actively linked. Copied nodes allow you to assign different values to the original and the copy.

When you paste nodes, Nuke automatically connects them to the node that is selected before the paste operation. If you don't want to connect anything, click on a blank area of the workspace to deselect any selected nodes before pasting.

Type of Edit	How to Preform It
Copy nodes to the clipboard	To copy nodes to the clipboard: 1. Select the node or nodes you want to copy. 2. Choose **Edit** > **Copy** (or press **Ctrl**/**Cmd**+**C**).
Copy nodes to files	To copy nodes to files: 1. Select the node or nodes you want to copy. 2. Choose **File** > **Export Comp Nodes...** 3. Navigate to the directory where you want to store the node(s) as a script. 4. Type a name for the script at the end of the pathway, followed by the extension **.nk**.
Cut Nodes	To cut nodes: 1. Select the node or nodes you want to cut. 2. Choose **Edit** > **Cut** (or press **Ctrl**/**Cmd**+**X**). Nuke removes the node(s) from the script and writes the node(s) to the clipboard.
Paste nodes from the clipboard	To paste nodes from the clipboard: 1. Select the node that you want the pasted node(s) to follow. 2. Choose **Edit** > **Paste** (or press **Ctrl**/**Cmd**+**V**). Nuke adds the nodes to the script, connecting them to the selected node.
Load nodes from files	To load nodes from files: 1. Select the node that you want the loaded node to follow. 2. Choose **File** > **Insert Comp Nodes**. 3. Navigate to the directory that stores the script. 4. Select the script, and click **Open**. Nuke adds the nodes described by the file to the selected node.

Cloning Nodes

You can clone nodes directly into the Node Graph or copy clone nodes in preparation for pasting pasting them elsewhere in a script. Cloned nodes inherit the values of their parent, but unlike copied nodes, they also maintain an active link with their parents' values. If you alter the values of a node that has been cloned, the clone automatically inherits these changes.

Clones are helpful for maintaining consistent setups across multiple elements. For example, you might use clones to apply an identical film grain setup to a series of elements shot on the same stock. If you need to make changes to the setup, these changes would automatically ripple throughout the script.



Note: You cannot clone gizmos. This applies to both gizmos created in your facility and the nodes in the Nuke default Toolbar that are, in fact, gizmos, such as the LightWrap node, the Grain node, and the IBK nodes. For more information on gizmos, see the Configuring Nuke chapter.

Cloning a Node

To clone nodes, do the following:

1. Select the node or nodes you want to clone.
2. Click **Edit** > **Clone**.

   Nuke clones the node(s), while maintaining an active link to the parental node(s). The clone has the same name as the original node and is indicated with an orange line connecting the clone to its parent node.



Note: Expression links between nodes are not shown if the node is cloned. For clarity, only the clone link is drawn in the Node Graph. See Linking Expressions for more information.

Copying a Node as a Clone

You can copy a node as a clone and paste it in a different location, by doing the following:

1. Select the node or nodes you want to clone.
2. Click **Edit** > **Copy as Clones** (or press **Ctrl**+**K**).

Decloning a Node

To declone nodes, do the following:

1. Select the node or nodes you want to declone.
2. Click **Edit** > **Declone** (or press **Alt**+**Shift**+**K**).

   Nuke removes the clone status of the selected nodes.

Linking Nodes

A more efficient way to copy nodes is to use Link Nodes.

Link Nodes allow you to make a linked copy of a node, and have changes updated in both nodes, whilst optionally being able to override any of the knobs.

To create a Link Node:

1. Select the node you want to make a link node from and press **Alt+L**, or go to the menu bar and select **Edit** > **Link**.
2. A Link Node will be created.
3. A yellow square indicates that the knob is overridden.

For more information on how to use these nodes see, Using Link Nodes.

Disabling and Deleting Nodes

Nuke allows you you to disable, re-enabe, and delete nodes in your Node Graph.

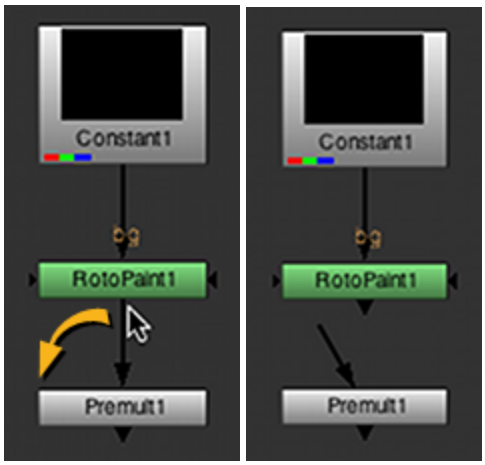
Action	Instructions
Disable node	To disable nodes: 1. Select the node or nodes you want to disable. 2. Select **Edit** > **Node** > **Disable/Enable** (or press **D**). Nuke disbales the node's effect on the data stream.
Re-enable node	To re-enable nodes: 1. Select the node or nodes you want to re-enable. 2. Select **Edit** > **Node** > **Disable/Enable** (or press **D**).
Delete nodes	To delete nodes: 1. Select the node or nodes you want to delete. 2. Select **Edit** > **Delete** (or press **Delete**). Nuke removes the node(s) from the script.

Connecting Nodes

When you add or paste nodes into a script, Nuke automatically generates pipes between the currently selected node and the new nodes. As you build up a script, you'll need to move these pipes, or run new pipes between nodes. In Nuke, you make such modifications by dragging on the back end of the pipe (the end without the arrowhead).

Disconnecting Nodes

You can disconnect nodes by either dragging the head or tail of the connecting arrow to an empty area of the workspace, or selecting the lower node in the tree and pressing **Ctrl**/**Cmd**+**D**.

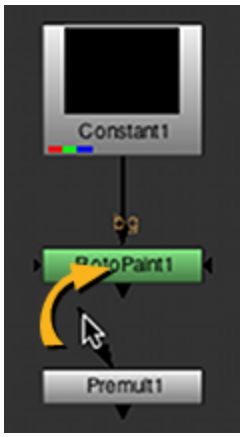


Shake to Disconnect Nodes

You can disconnect a node or selection of nodes by grabbing the selection and shaking the pointer. The sensitivity of the shaking required to disconnect nodes is controlled by a preference under **Panels** > **Node Graph**. You can also disable **shake to disconnect nodes** in the same panel.

Reconnecting Nodes

You can reconnect a node by dragging the head or tail of the connecting arrow and drop it over the center of the node that want to connect.

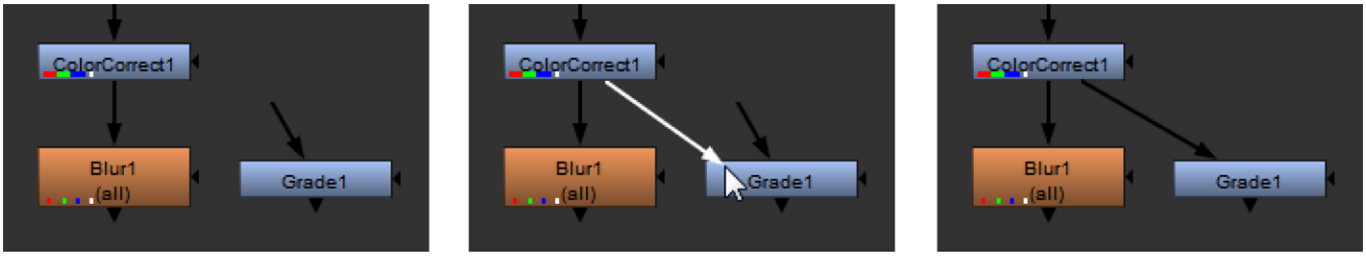


Note: Nuke distinguishes the dual inputs that may run into a Merge node with the labels **A** and **B**. **A** refers to the foreground element, and **B** to the background element. Nuke always copies from the **A** input to the **B**. This means that if you later decide to disable the node associated with an **A** input, the data stream keeps flowing because, by default, it uses the **B** input.

Tip: It's good organization practice to keep all the **B** inputs vertical in your node graph, as well as bringing in all the **A** inputs from the left and any **masks** from the right.

Duplicating a Connecting Arrow

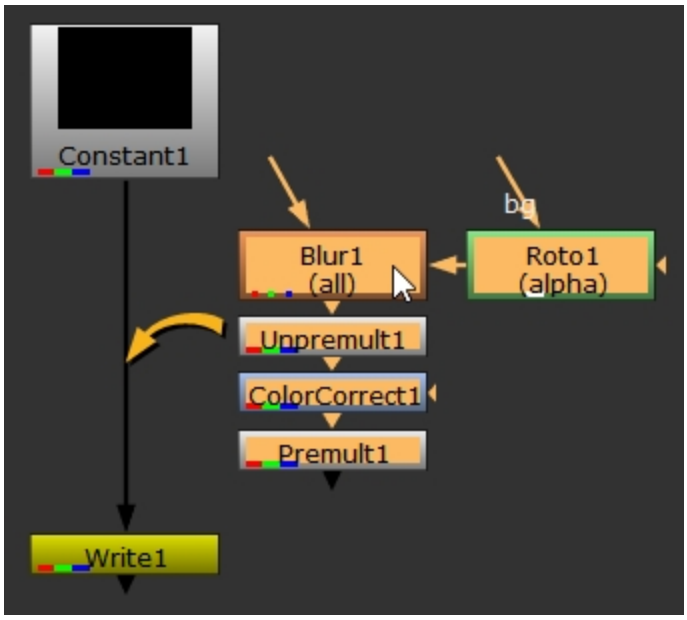
To duplicate a connecting arrow, hold **Shift** and drag the connecting arrow on top of the node you want to create a connection to. Nuke duplicates the connecting arrow, leaving the original connection untouched.

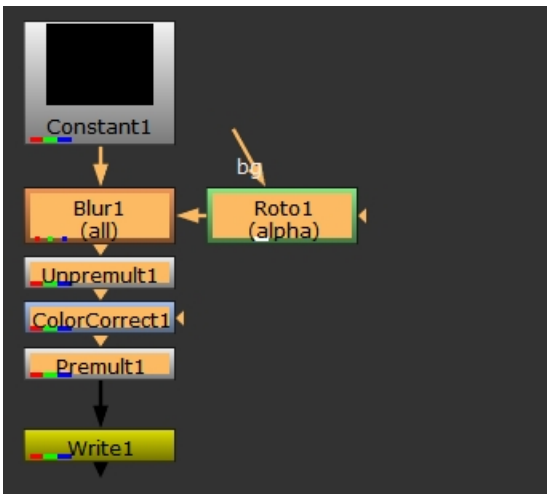


Adding Nodes Between Nodes

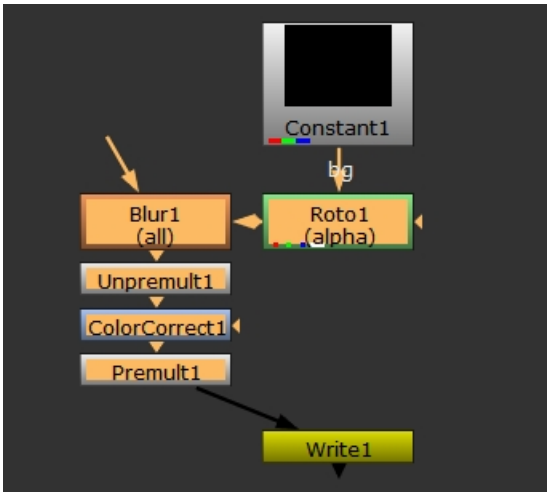
To add a node between two connected nodes, drag the node into the space between the already connected nodes. As you do so, you see the link between these two nodes become active. When that happens, release the node you are dragging and it is automatically placed and connected between the two nodes.

You can also drag a node stack onto an existing connection. This action places the nodes between existing nodes in the pipeline.





The pointer position is important when you connect nodes this way. In the example, the Blur node at the top of the stack is under the pointer. If the Roto node was under the pointer, the connection would look like this.



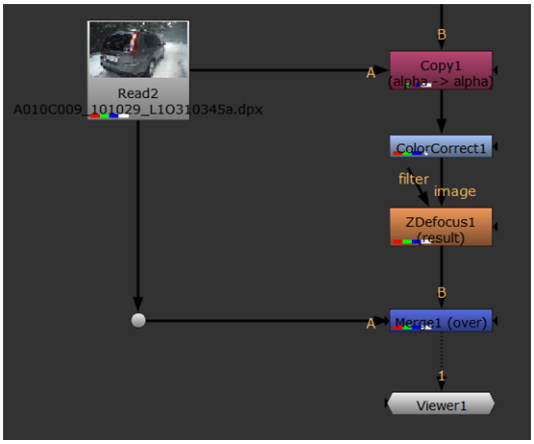
Bending Connecting Arrows

To bend connecting arrows, do the following:

1. Select the node before the connector you want to bend.
2. From the Toolbar, select **Other** > **Dot**.

   A dot appears after the selected node, causing a bend in the connector.
3. Drag the dot as necessary to reposition the bend.

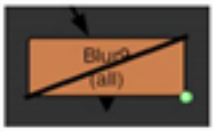
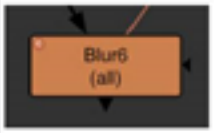
Tip: You can also add a dot to an existing connection by pressing **Ctrl/Cmd** and clicking on the yellow dot that appears on the connecting arrow.



Node Indicators

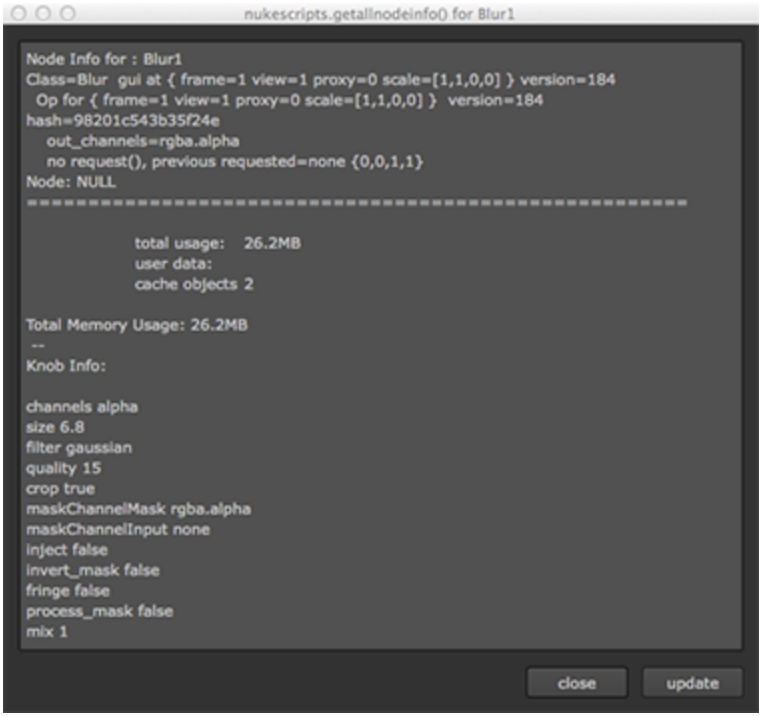
There are several indicators that can appear on the nodes in the Node Graph, depending on what you are doing. The following table describes what each indicator means.

Indicator	Where it appears	What it means
		The wide rectangles indicate the channels the node processes. The thin rectangles indicate the channels that are passed through the node untouched.
		The node's effect is limited by a mask from either the node's primary input or output.
		The node has been disabled by pressing **D** or by selecting **Edit** > **Node** > **Disable/Enable**.

Indicator	Where it appears	What it means
		The node has been disabled using an expression.
		The node has been cloned. The indicator appears on both the parent and the child node.
		One or more of the node parameters are animated over time.
		One or more of the node parameters are being driven by an expression.
		You are working with a multi-view project and have split off one or more views in the node's controls.
		You are working with a multi-view project and have split off one or more views in the node's controls, dots also appear on the node to indicate which views have been split off. For example, if you are using red for the left view and split off that view, a red dot appears on the node.
		The full effect of the node is not in use, because you have adjusted the **mix** slider in the node's controls.
		The node is a linked copy of another node.
		The linked node has an override.

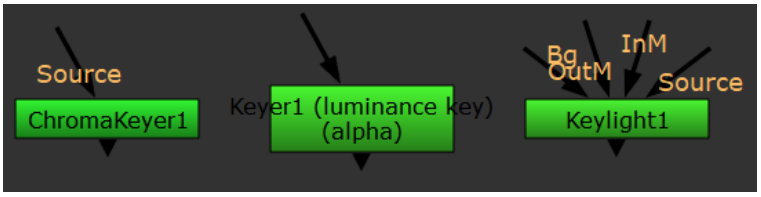
Displaying Node Information

You can obtain more detailed information from any node by selecting that node and then pressing the **I** key. This displays an information window associated with that node, particularly useful when troubleshooting.



Customizing the Node Display

You can modify the color, name, and notes that a particular node displays. Doing so can make it easier for other artists to decipher the intent of your script. For example, you might color all nodes green that relate to keying.

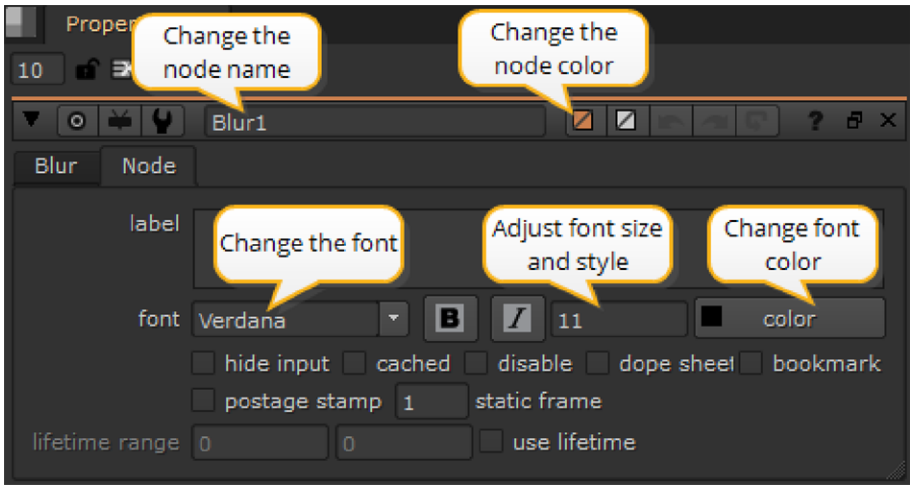


Modifying Node Display Characteristics

To modify a node's display characteristics, do the following:

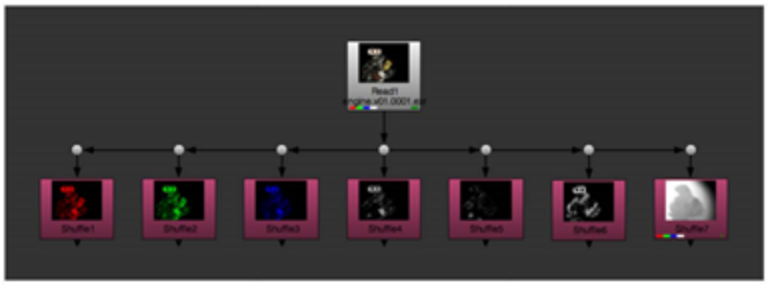
1. Double-click on the node in the Node Graph to display its properties in the **Properties** pane.

2. Click the **Node** tab at the top of the dialog in the **Properties** pane. Its attributes are displayed:



In the node's properties, you can also do the following:

- Select the **hide input** checkbox to conceal the node's incoming pipe. This can enhance the readability of large scripts.
- Select the **postage stamp** checkbox to display a thumbnail render of the node's output on its surface. You can also press **Alt**+**P** on the **Properties** pane or on the node, to toggle postage stamps.



If you want the postage stamp to display a fixed frame (rather than update to match the current frame), enter the frame to display in the **static frame** field. For this to work, you also need to press **Shift**+**S** to open the **Preferences** dialog, go to **Panels** > **Node Graph**, and ensure the **postage stamp mode** dropdown is set to **Static frame**. Note that if the frame number you use is outside the frame range for the node, it is clamped to the first or last frame in the range.

Tip: You can also have Nuke automatically color code nodes for you based on their function. See Appendix A: Preferences.

Creating Node Tool Sets

If you find yourself creating the same set of nodes repeatedly, you can create a custom tool set in the Nuke toolbar. This allows you to quickly and easily create the set of nodes from the **ToolSets** menu.

Tool sets can be shared between artists if they are using a centralized .nuke folder. This needs to be accessed through a NUKE_PATH environment variable that you can set up (see Configuring Nuke for more information).

To create a tool set do the following:

1. In the Node Graph, select any number of nodes (or just one node). The nodes don't have to be connected to each other.
2. Click  (**ToolSets**) on the toolbar and select **Create**. You can also right-click on the Node Graph and select **ToolSets** > **Create**. The **CreateToolSet** dialog appears. You can also access your tool sets by pressing **Tab** and searching for them in the search field.
3. In the **ToolSets menu** dropdown, select the menu where you'd like to place your new tool set. Then give it a name in the **Menu item** field and click **Create**.

   By default, your new tool set goes under the **ToolSets** menu, but if you'd like to create a subfolder for the tool set, you can do that by specifying the folder name before the tool set name, separated by a forward slash, in the **Menu item** field. For example, entering **Roto/BasicRoto** would create a subfolder called Roto in the **ToolSets** menu, and place a new tool set by the name of BasicRoto in it.

   There are some tool sets for creating particles, for instance, that are built in to Nuke. These can also be found in the **ToolSets** menu. To prevent confusion between built-in tool sets and tool sets that you've created, if you happen to create a tool set folder with the same name as a built-in one, Nuke adds **[user]** in front of the folder name for clarity.
4. To delete a tool set you've created, you can click **ToolSets** > **Delete** in the toolbar and select the tool set you want to remove.

Using the Tab Menu

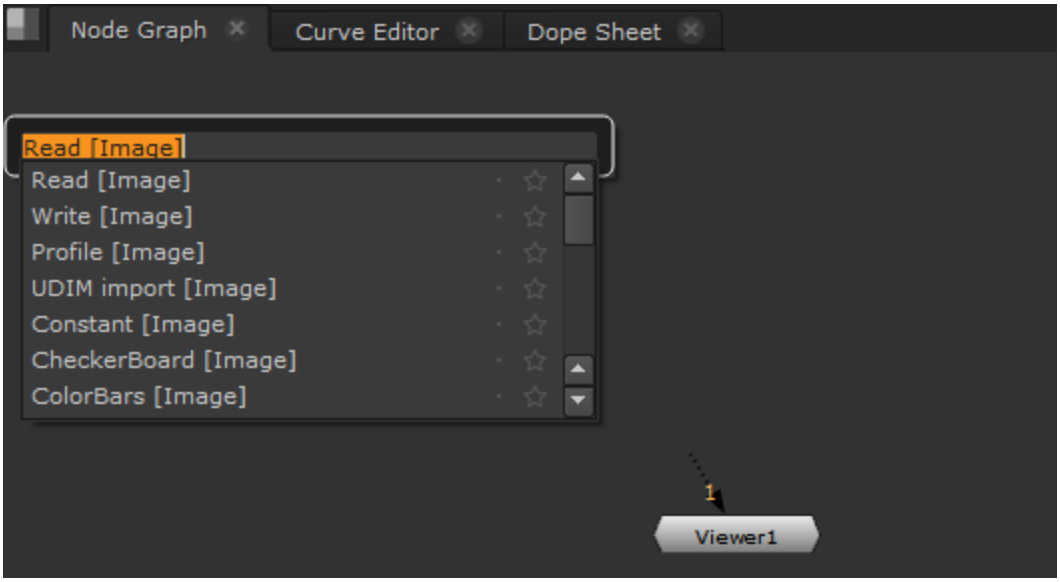
The Node Graph's Tab menu, opened by pressing **Tab** in the Node Graph, enables you to search for and add nodes easily using partial names. Commonly used nodes are weighted so that they appear higher up the list of choices and you can favorite nodes, pinning them to the top of the list. Weights and favorites can be enabled, disabled, and cleared in the **Preferences** under **Behaviors** > **Nodes**.

Adding Nodes by Sub-string

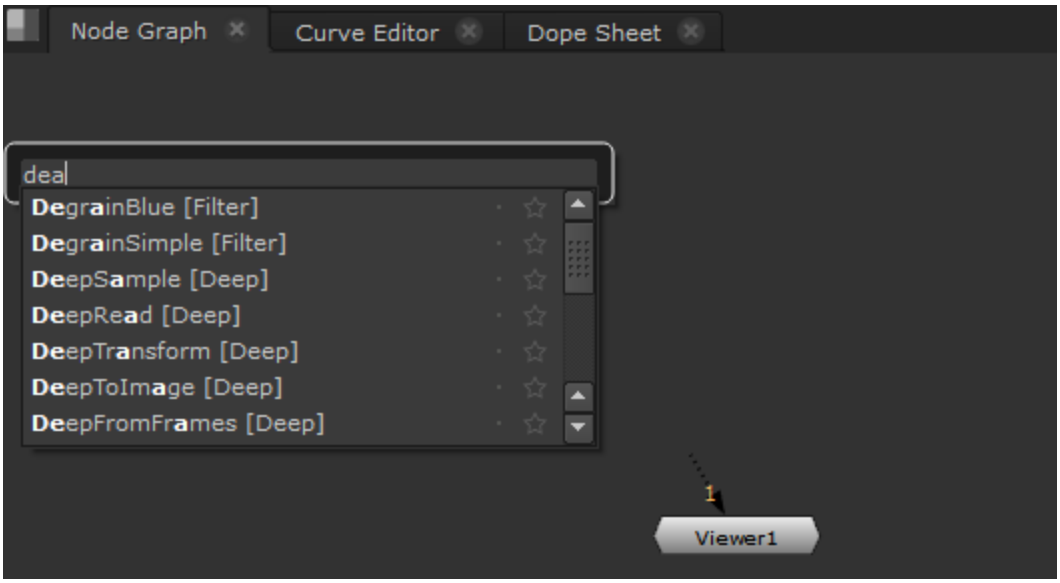
Typing characters into the **Tab** menu displays a list of matches containing those characters. The more characters you enter, the more the list of applicable nodes is refined. To display the menu, press **Tab** when the Node Graph has either click or mouse-over focus.



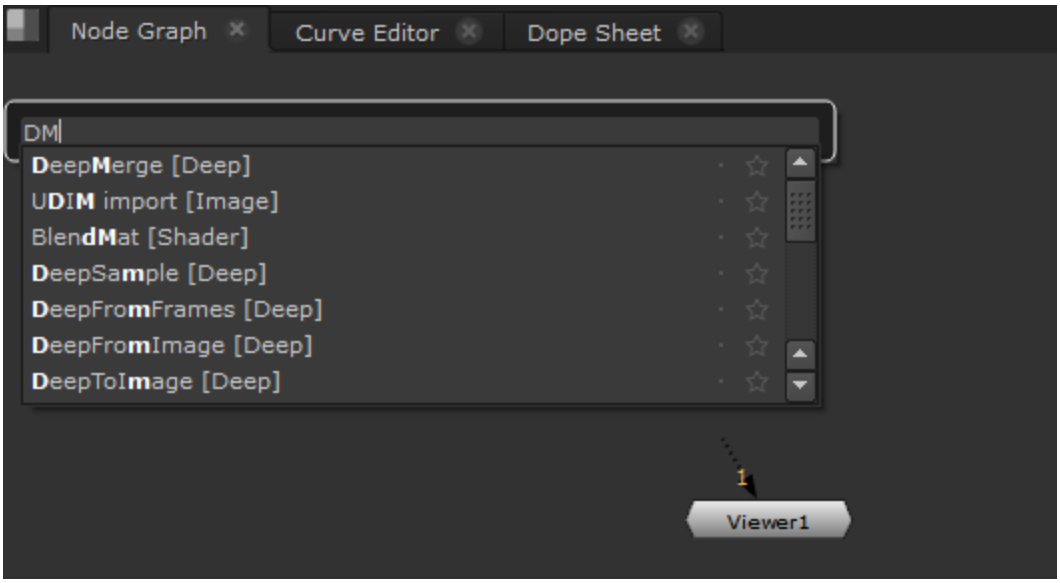
Tip: See Panel Focus and Keyboard Shortcuts for more information about focus.



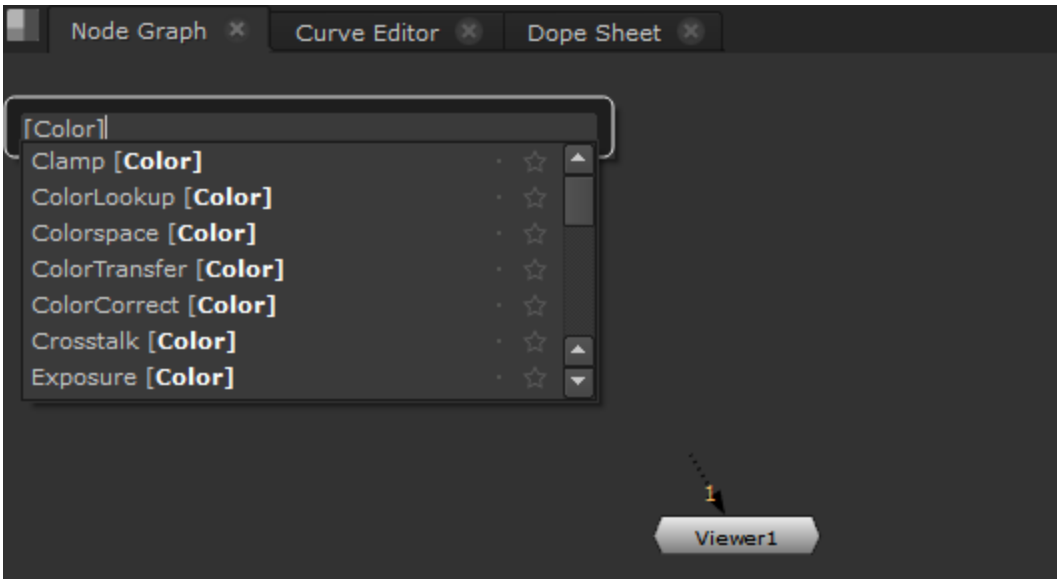
Typing in the **Tab** menu narrows the selection. For example, entering the string **dea** displays all entries beginning with **dea**. In this case, you can see that there are no nodes beginning **dea**, so Nuke looks instead for node beginning **de** with an **a** in the name.



The **Tab** menu parses capitalization as well. For example, if you want a DeepMerge node, start your string **DM** rather than **dm** to refine the results.



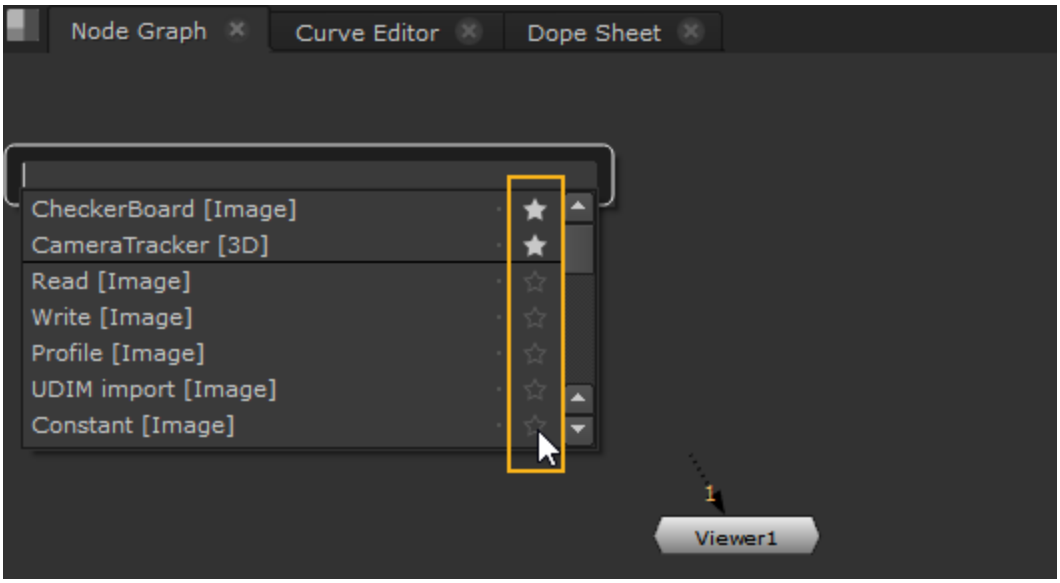
You can also search by toolbar menu name. For example, if you know that you're searching for a color correction node you could enter **[Color]**.



Nuke's Bookmark feature also allows you to use sub-strings to locate the bookmark you need. See Bookmarking Nodes for more information.

Adding and Removing Favorites

Favorites allow you to mark nodes that you use most often so that they are listed first in the **Tab** menu. You can add and remove favorites by clicking the star next to their entry in the node list.

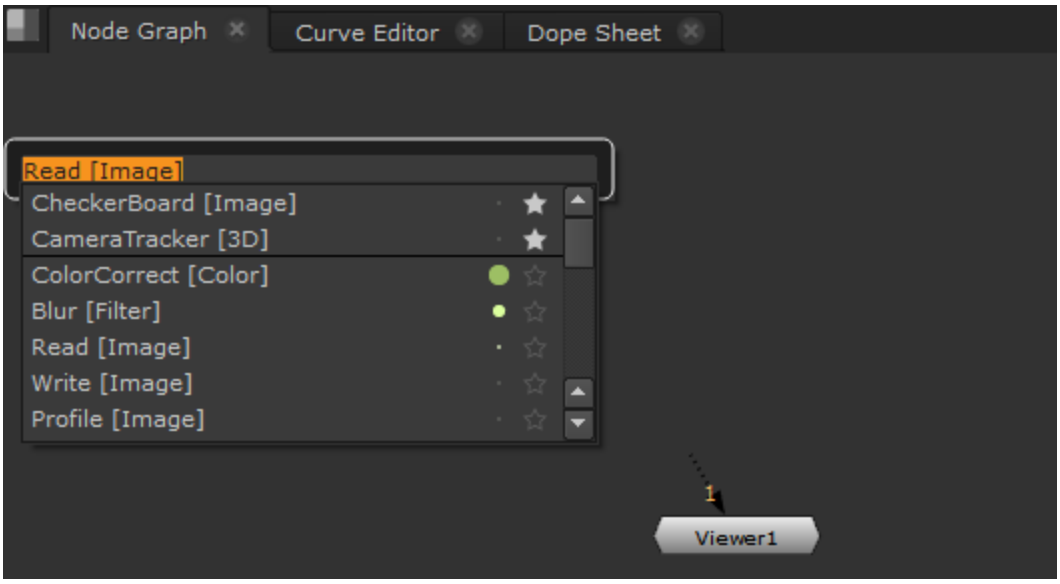


You can enable, disable, and clear all favorites in the **Preferences** under **Behaviors** > **Nodes**.

Managing Weighting

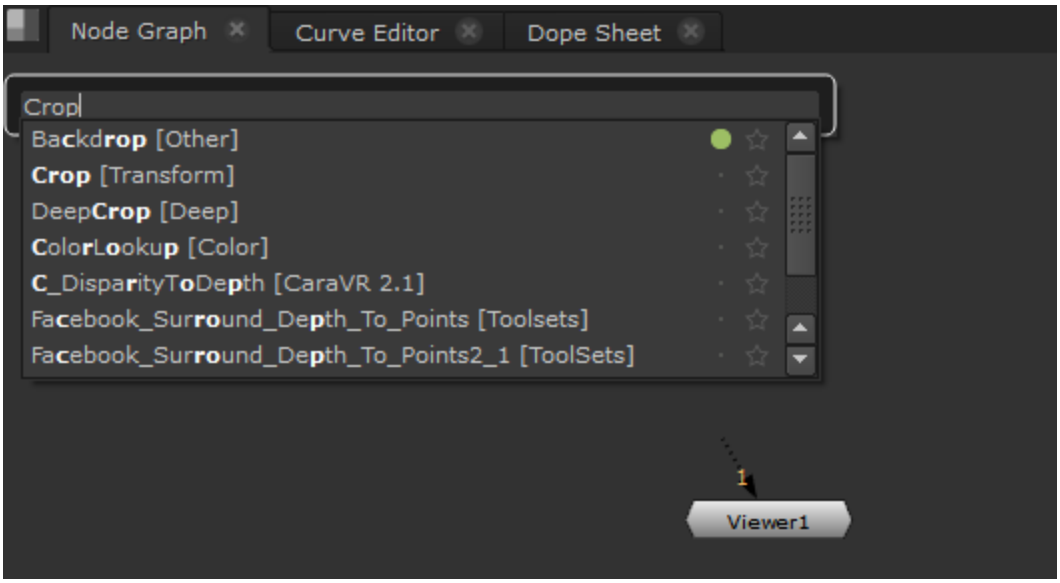
Weighting describes how often you use a node from the **Tab** menu. Commonly added nodes carry more weight and are listed higher up in the node list.

Note: Adding nodes using a keyboard shortcut, such as pressing **B** to add a Blur node, does not affect node weighting.



The more often you add a node form the **Tab** menu, the larger the green dot next to the node name.

Weighting can produce unexpected results when you enter strings in the **Tab** menu. For example, if you use Backdrop nodes often, you may find that weighting causes Backdrop to appear above nodes that start with the first letter of the search string.



You can enable, disable, and clear weighting information in the **Preferences** under **Behaviors** > **Nodes**.

Navigating Inside the Node Graph

As scripts grow in complexity, you need to be able to pan to a particular cluster of nodes quickly. The Node Graph offers a couple of methods for doing so.

Panning

You can pan using the mouse or the navigator map that appears in the lower right corner of the Node Graph.

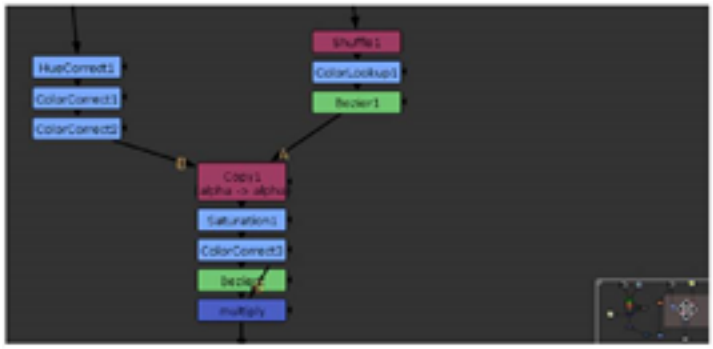
Panning with the Mouse

To pan with the mouse, press the middle mouse button and drag the mouse pointer over the workspace (you can also use **Alt**+drag). The script moves with your pointer.

Note: In many Linux windows managers, the **Alt** key is used by default as a mouse modifier key. This can cause problems in 3D applications where **Alt** is used for camera navigation in 3D environments.
You can use key mapping to assign the mouse modifier to another key, such as the (Super or Windows logo) key, but the method changes depending on which flavor of Linux you're using. Please refer to the documentation on key mapping for your particular Linux distribution for more information.

Panning with the Map

If your script is larger than the visible workspace, a navigator map automatically appears in the bottom-right corner.



The map shows you a miniature view of the entire script and the pink rectangle shows the portion of the script that you see within the workspace borders.

To pan with the map, drag the pink rectangle to pan to a different view of the script.

When the whole script is contained within the window border, then the map automatically disappears.

Tip: The navigator map is resizeable. Drag on its upper-left corner to make it as large or small as you like.

Zooming, Fitting in the Node Graph

Action	How to perform it
Zooming in	To zoom in, you can either: • Move the cursor over the area you want to zoom in on, and press the addition key (+) repeatedly until the workspace displays the script at the desired scale. OR • Press **Alt** and drag right while holding down the middle-mouse button.
Zooming out	You can zoom out by doing one of the following: • Move your mouse pointer over the area you want to zoom out from, and press the subtraction key (-) repeatedly until the workspace displays the script at the desired scale. OR • Press **Alt** and drag left while holding down the middle-mouse button.



Note: In many Linux windows managers, the **Alt** key is used by default as a mouse modifier key. This can cause problems in 3D applications where **Alt** is used for camera navigation in 3D environments.
You can use key mapping to assign the mouse modifier to another key, such as the  (Super or Windows logo) key, but the method changes depending on which flavor of Linux you're using. Please refer to the documentation on key mapping for your particular Linux distribution for more information.

Fitting the selected nodes within the visible workspace	To fit selected nodes in the Node Graph, click the middle-mouse button or press **F**.
Fitting the node tree within the visible workspace	To fit the entire node tree in the Node Graph, click on the Node Graph to make sure no nodes are selected and click the middle-mouse button or press **F**.

Bookmarking the Pan and Zoom Level

To bookmark the pan and zoom level:

1. Pan and/or zoom the script as necessary.
2. To save the current pan and zoom level, navigate to **Edit** > **Bookmark** > **Save Location 1** (or press **Ctrl/Cmd+F7**).

   You can save three more locations in a similar manner, using **Save Location 2**, **Save Location 3**, and **Save Location 4** (or by pressing **Ctrl/Cmd+F8/F9/F10**).

   The saves are temporary and not saved to the script.
3. To restore a saved location later, select **Edit** > **Bookmark** > **Restore Location1**, **Restore Location 2**, **Restore Location 3**, or **Restore Location 4** (or press **Shift+F7/F8/F9/F10**).



Note: Restoring a saved location doesn't work across different Node Graphs - you can't save a location in one Node Graph and then restore it in another.

Bookmarking Nodes

One extremely useful function of Nuke nodes, is their ability to act as jump-to points throughout a project.

1. Double-click on the node you want to bookmark to open its properties.
2. Go to the **Node** tab and select the **bookmark** checkbox (or press **Ctrl/Cmd+Shift+B**).

   This adds the node to the bookmark list.

   By default, Backdrop nodes have their **bookmark** checkbox enabled. All other nodes have it disabled.
3. Select **Edit** > **Bookmark** > **Jump to Bookmarked Node** (or press **J** on the Node Graph) to bring up the bookmarks jump to menu.
4. Start typing the name of the node you wish to navigate to.

   Nuke shows all matchingnodes that have the bookmark flag enabled. They are listed according to the node name or, if a **label** is present, the label followed by the node name in square brackets.
5. To select the node to navigate to, you can either click on it, or scroll to it with the **Up** and **Down** arrow keys, and press **Return**.

Searching for Nodes

Nuke's **Search** feature allows you to search for nodes in your script and select any matches found. As a search string, you can enter all or part of a node name. For example, you can search for all Blur nodes in your script by entering **bl** as the search string.

You can also do more complex searches using regular expressions, such as searching for all the Read and Write nodes in a script.

To Search for Nodes

1. Select **Edit** > **Search** (or press **/**) to bring up the search dialog.
2. In the search field, enter the string you want to search for.

   If you want to search for all nodes in the script, enter ***** (an asterisk).

   If you want to search for all Read nodes in the script, enter **Read**.

   If you want to search for all the Read and Write nodes, enter the following expression:

   (***Read***|***Write***)
3. Click **OK**.

Nuke searches for the nodes in the script and selects all matches it finds and focuses the Node Graph the first of them. If you perform the same search again, the Node Graph view focuses on the next node in order. For instance, if you have a ColorBars node, a ColorWheel node, and a ColorCorrect node in your Node Graph, searching for "Color" three times in a row selects all of these nodes and focuses on each of them in alphabetical order.



Note: When you enter expressions in the search dialog, remember that the search field only takes regular expressions. Any characters that have specific meanings in regular expressions, such as square brackets (**[** and **]**), need to be preceded by the backslash character (****).

Cleaning up the Node Graph

Sometimes you may find that your node tree becomes a bit disorganized as you're creating and removing nodes. There's a quick fix you can use to tidy up your tree:

1. Select all the nodes you want to rearrange in the Node Graph.
2. Click **Edit** > **Node** > **Autoplace**. Alternatively, press **L**.

   The Autoplace function automatically arranges the selected nodes in a neat tree formation. It determines a sensible position for the nodes by ensuring they don't overlap with other nodes and by keeping the input connections horizontal.

Properties Panels

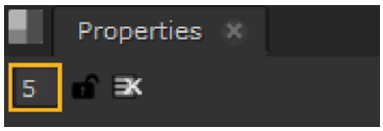
When you insert a node, its properties automatically appear in the **Properties** panel with options to define the node's output. You can also open the properties panel later by doing any of the following:

- Double-click on the node in the Node Graph.
- **Ctrl**/**Cmd**+click on the node in the Node Graph.
- Select the node in the Node Graph and press **Return**.

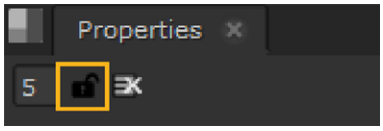
Tip: To open a properties panel in a floating window, **Ctrl**/**Cmd**+double-click or **Ctrl**/**Cmd**+**Alt**+click on the node.

Managing the Properties Panel

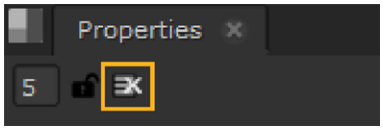
You can limit the number of node panels that can be open in the **Properties** panel. To do so, enter the maximum number of nodes in the field on the Properties panel.



To lock the **Properties** panel and have all new panels appear in floating windows, click the lock button on the **Properties** panel.



To empty the **Properties** panel and close all the node panels in it, click the remove all panels button.



Tip: You can also close all the nodes in the **Properties** panel by **Alt**+clicking on the close (**X**) button of one of the nodes.

Properties Panel Controls

These are the standard controls of every properties panel:

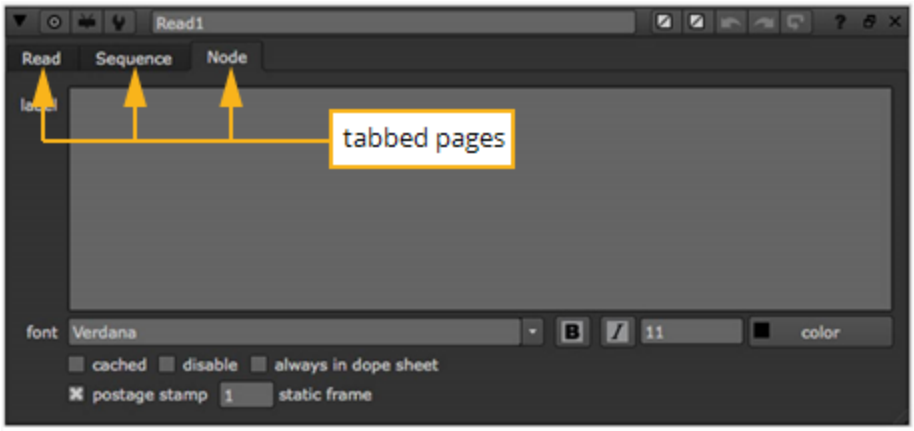
Control	Function
	Hide or show the node's tabbed pages.
	Centers the node in the Node Graph.
	Centers one of the node's inputs in the Node Graph. Select the input from the dropdown menu that appears.
	You can save, load, and manage node presets here.
name field (for example, **Blur1**)	You can enter a new name for the node here.
 (left)	Changes the color of the node. You can drag and drop this button on top of another color button to copy the color. To revert to the default color defined in your **Preferences**, right-click on the button and select **Set color to default**. An **X** on the button indicates the color is unset, and the color defined in the Preferences is used.
 (right)	Changes the color used for the node's controls in the Viewer. You can drag and drop this button on top of another color button to copy the color. To revert to the default color defined in your Preferences, right-click on the button and select **Set color to default**. An **X** on the button indicates the color is unset, and the color defined in the Preferences is used.
	Undoes the last change made to the node.
	Redoes the last change undone.

Control	Function
	Reverts any changes made after the current node panel was opened.
	Displays Help related to the node and its controls. **Note:** If the Online Help is unavailable, a tooltip displays information about the node.
	Floats the panel. Clicking this button again docks the panel back into the pane if the originating pane is open.
	Closes the node panel. **Alt**+click this to close all the panels in the pane. **Ctrl/Cmd**+click to close all panels except the one clicked on.

Floating control panels also include the following buttons:

Control	Function
	Reverts any changes made after the panel was opened.
	Reverts any changes made after the panel was opened and closes the panel. Hitting this button right after a node was created also deletes the node from the Node Graph.
	Closes the panel.

Many node panels also contain several tabbed pages.



On the **Node** tab, you can usually adjust the following controls:

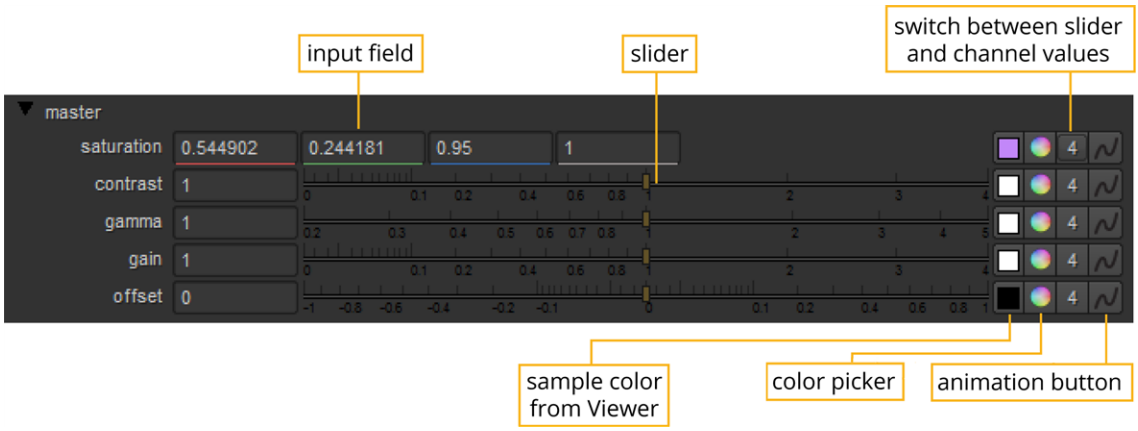
Control	Function
label	Lets you add comments to the node. The comments are displayed on the node's surface. If you like, you can use HTML in the label field. For example, to have your comments appear in bold, you can enter **My Comment**. To add an icon called *MyIcon.png* to the node, you can use ****. Save the icon in your plug-in path directory. (For more information on plug-in path directories, see Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts.) Most common image formats work, but we recommend using **.png**. Note that the HTML has been changed to a slightly non-standard form where newlines are significant. If there is a newline character in your data, a new line is displayed in the label.
font	Lets you change the font for any text displayed on the node.
	Bolds any text displayed on the node.
	Emphasizes any text displayed on the node.

Control	Function
	Lets you change the font size of any text displayed on the node.
	Lets you change the color of any text displayed on the node.
hide input	Check this to hide the node's incoming pipe. This control does not appear on all nodes.
cached	Check this to keep the data upstream from the node in memory, so that it can be read quickly. When this is checked, a yellow line displays under the node in the Node Graph.
disable	Check this to disable the node. Uncheck to re-enable. (You can also disable or re-enable a node by selecting it in the Node Graph and pressing D.)
dope sheet	Check this to force the node to always display in the Dope Sheet. You can also press **Alt**+**D** to toggle this on and off. By default, the Dope Sheet displays all your Read nodes that have their control panels open, and any nodes you've used to create keyframes.
bookmark	Check this to show the node in the bookmark list. This allows you to quickly navigate to the node. For more information, see Navigating Inside the Node Graph.
postage stamp	Check this to display a thumbnail render of the node's output on its surface. You can also press **Alt**+**P** on the Properties Bin to toggle postage stamps.
static frame	If you want the postage stamp to display a fixed frame (rather than update to match the current frame), enter the frame to display here. For this to work, you also need to press **Shift**+**S** to open the **Preferences** dialog, go to the **Node Graph** tab, and make sure **postage stamp mode** is set to **Static frame**. Note that if the frame number you use is outside the frame range for the node, it is clamped to the first or last frame in the range.
lifeteme range	Enter a start and end frame to control the frame range when this node is active. You can quickly toggle the lifetime on and off with **use lifetime**.

Displaying Controls

To display a node's controls, double-click the node. It's properties panel appears.

The image below shows the controls available for editing parameters. Note that the presence of each control varies according to the parameter's function.



Using Input Fields

You can key values directly into a field, press the arrow keys to increment and decrement values, or use the middle-mouse button to activate virtual sliders.

Changing Field Value

To key in field values:

1. Double-click in the field to select the whole value.
2. Type the value you want to replace the selection.

Tip: You can also enter expressions (programmatic instructions for generating values) into fields. You always start an expression by typing =. See Expressions for information about how to format expressions.

Tip: Nuke also allows you to enter formulas into fields, making it easy to do quick calculations. For example, if you wanted to halve a value of 378, you could simply type **378/2** into a field and press **Enter** to get 189.

You can increment or decrement field values by hundreds, tens, tenths, hundredths, and so on. The magnitude of change depends on the initial position of your cursor. For example if you wanted to increment the initial value of **20.51** by ones, you would insert your cursor before the **0**.

To increment or decrement a field value:

1. Click to insert the cursor just prior to the digit you want to increment or decrement.
2. Press the **up arrow** to increment by one unit, or the **down arrow** to decrement by one unit.

Tip: You can also increment and decrement values by scrolling the mouse wheel (if available) or by holding **Alt** while left-mouse dragging on the value. The latter method is particularly useful for tablet users.

Using Virtual Sliders

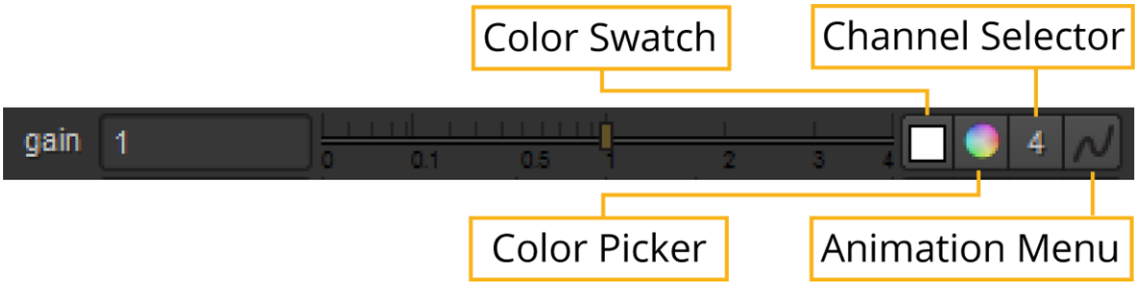
To use a virtual slider, you can hover the cursor over a numeric control, click the middle-mouse button, and drag the field to adjust the associated control.



Tip: Hold **Shift** to increase slider sensitivity and **Alt** to decrease slider sensitivity.

Using the Color Controls

In the **Properties** panels, there are usually four extra controls to the right of a control including **color swatch**, **color picker**, **channel selector** and the **animation menu**. The first three are color controls. For more information about using the animation menu, see Animating Parameters.



You can use the **color swatch** button to activate the eye dropper tool. You can copy a color from one swatch to another by dragging and dropping the color swatch with the color you want to use over the color swatch you want to replace.

The **color picker** button can be used to display and make color selections. Depending on the settings in the **Preferences** dialog, you may use an in-panel color picker, or a floating color-picker window. (See Appendix A: Preferences for more information.)

You can use the **channel selector** to toggle between the slider and manually enter values for each of the channels.

Using the Color Swatch

You can use the color swatch to sample a color from the Viewer, by doing the following:

1. Click the **color swatch** to activate the eye dropper tool.
2. Move the cursor over the Viewer, over the color you want to sample.

   You can zoom in and pan as necessary until the area you want to sample is clear.
3. **Ctrl/Cmd**+click to sample a color value, or **Ctrl/Cmd**+**Alt**+click to sample a color from the node's input while viewing its output.

   You can sample a region instead of just a single pixel by also pressing **Shift**+drag when you're sampling the color. When sampling a region, Nuke calculates the average color of the region.

   If you are not happy with the pixel or region you've sample, you can simply repeat the sample procedure until you are happy with the selected pixel or region.

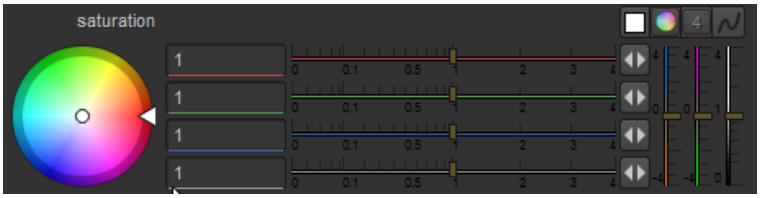


Tip: You can discard sampled pixels by **Ctrl/Cmd**+right-clicking in the Viewer.

4. When you are happy with the color sample, click the color swatch again. This closes the eye dropper tool and now displays your selected sample.

Using the In-Panel Color Picker

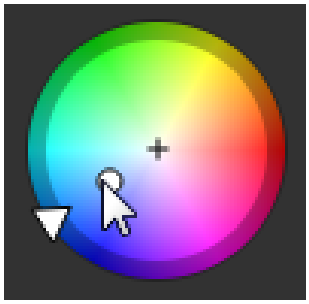
If your **Preferences** > **Panels** > **Control Panels** > **Color Panel** > **color picker button opens** dropdown is set to **in-panel color picker**, you can display the color sliders and wheel within the **Properties** panel by clicking the **color picker** button . You can have multiple in-pane color pickers open simultaneously.



Tip: Holding **Ctrl**/**Cmd** and clicking the color picker button opens the alternate color picker to the one specified in the **Preferences** dialog.

Using the In-Panel Color Wheel

You can adjust the hue by dragging the marker on the edge of the color wheel (or the marker inside the wheel) around to the required hue. You can adjust the saturation by dragging the marker inside the wheel in or out to the required saturation.



Tip: You can also **Ctrl**/**Cmd**+click on the color wheel to only affect the hue, and **Shift**+click to only affect the saturation.

To adjust the value (brightness) of the color, hold down **Ctrl**/**Cmd**+**Shift** and drag right or left.

By default, the color wheel uses relative cursor positioning. This means the cursor position can be offset from the marker position, which allows for very fine adjustments. You can disable this behavior and use absolute cursor positioning by holding **Alt** while moving the markers.

Note: If any red, green, or blue values are higher than 1, the marker inside the wheel turns red. If the red, green, and blue channels all have a value of less than 0.1, the marker turns blue. If at least one channel is negative and at least one is positive, the hue indicator turns red.

Using the In-Panel Color Sliders

To increment the value by 0.01, left-click on the arrow button . To decrement the value by 0.01, right-click on the arrow button. Use **Shift**+click for 0.1, and **Alt**+click for 0.001.

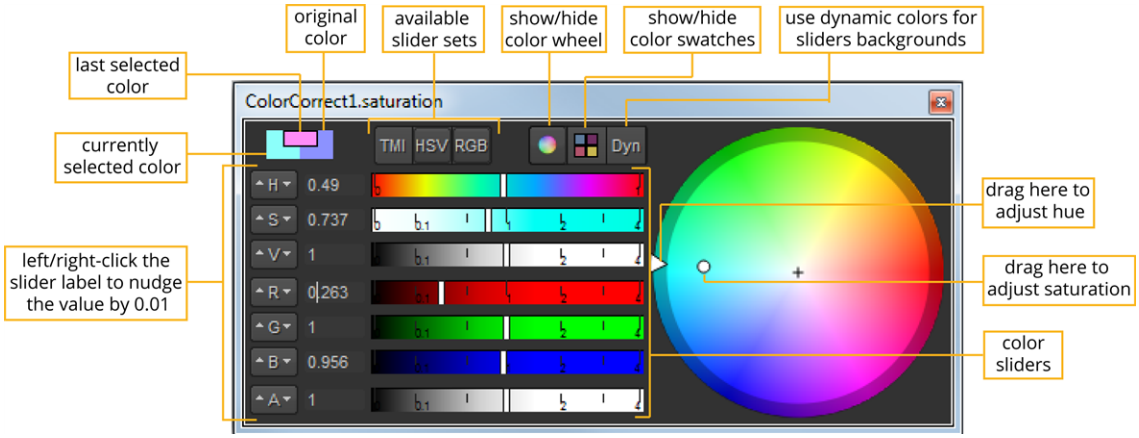
You can also click and drag right or left on the button to scrub the value up or down. Use **Shift**+drag to scrub quickly, or **Alt**+drag to scrub slowly.

Using the Floating Color Picker Window

If your **Preferences** > **Panels** > **Control Panels** > **Color Panel** > **color picker button opens** dropdown is set to **floating color picker**, you can display color sliders and wheel in a floating window by clicking the **color picker** button .

Tip: Holding **Ctrl**/**Cmd** and clicking the color picker button opens the alternate color picker to the one specified in the **Preferences** dialog.

You can change the floating window to be horizontal by dragging on one of the corners of the window to resize it. When it is wide enough, the sliders automatically become horizontal.



• From the **TMI**, **HSV**, and **RGB** buttons, you can select which slider you want to display.

- There are three states of the color wheel: hide color wheel, show color wheel, and show color wheel with square. You can cycle through these states by simply clicking on the **color wheel** button multiple times.
- You can choose to show the color swatches by clicking the **color swatch** button.
- If you want the background of the sliders to show what the value of the color would be if the sliders were set to the current position, click the **Dyn** button.

Using the Color Wheel in the Floating Window

Using the color wheel in the floating window is exactly the same as using the in-panel color wheel; drag the marker on the edge of the wheel to adjust the hue, and drag the marker inside the wheel to adjust the saturation.

However, you can also pan and zoom on the color wheel in the floating window:

- To pan, press **Alt** and drag the cursor over the color wheel.
- To zoom, press **Alt** and drag left or right with the middle-mouse button.
- You can reset the zoom and/or pan by simply middle-clicking on the color wheel.

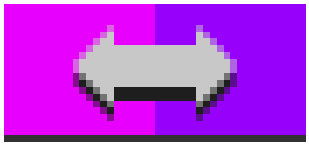
Using the Color Sliders in the Floating Window

Using the sliders in the floating color window is exactly the same as using the in-panel sliders.

Using the Color Swatches in the Floating Window

When you're happy with a color, you can save it by right-clicking on the swatch where you want to save it. This replaces the original swatch that was there. You can also drag and drop a color you want to save on any of the swatches to replace it.

The rectangle above the sliders shows the original color on the right, the currently selected color on the left. When you hover the cursor over this, an arrow appears allowing you to copy one to the other by clicking once on the rectangle.



Opening Another Floating Color Picker Window

You can open multiple color picker floating windows, by pressing **Ctrl**/**Cmd**+**Alt**+click on another parameter's color picker button.

Using the Channel Selector

By default, many parameters in Nuke automatically group channels for you. For example, if you drag on the **gain** slider in the ColorCorrect node, you simultaneously affect the R, G, and B channels (assuming you're processing the RGB layer.) You can use the **channel selector** button  to display and edit the individual channel values. The number displayed on the channel selector button represents the number of available channels for editing.

You can edit an individual channel's value by doing the following:

1. Click the **channel selector** button.

   The available channels appear with value fields for each. For example, when you press the **gain** - **channel selector** button, four channel value fields appear (R,G,B, and A) as the channel selector shows that there are four available channels for editing.



2. Enter the required values in the channel value fields.

   After you are happy with the values, simply click the channel selector button again to close and save your settings.

Color Slider Functions

RGBA Sliders

Slider	Function
The red slider (**R**)	This allows you to control the red channel's value (or the first channel in a layer if you are processing another layer besides RGBA).
The green slider (**G**)	This allows you to control the green channel's value (or the second channel in a layer if you are processing another layer besides RGBA).
The blue slider (**B**)	This allows you to control the blue channel's value (or the third channel in a layer if you are processing another layer besides RGBA).
The alpha slider (**A**)	This allows you to control the alpha channel's value (or the fourth channel in a layer if you are processing another layer besides RGBA).

TMI Sliders

Slider	Function
The temperature slider (**T**)	This allows you to control apparent color temperature by inversely affecting red and blue values (assuming you are processing the RGBA layer).
The magenta/green slider (**M**)	This allows you to control the mix of green and magenta hues. To add more magenta (increase the red and blue channels' values, while decreasing the green channel's), drag up. To add more green (increase the green channel's value, while decreasing the red and blue channels'), drag down.
The intensity slider (**I**)	This allows you to simultaneously control the red, green, and blue channel values. To increase the value of all channels by the same amount, drag up. To decrease the value of all channels by the same amount, drag down.

HSV Sliders

Slider	Function
The hue slider (**H**)	This allows you to control the color's location on the traditional color wheel (for example, whether the color is red, yellow, or violet).
The saturation slider (**S**)	This allows you to control the intensity or purity of the color.
The value slider (**V**)	This allows you to control the brightness of the color (the maximum of red, green, and blue values).



Note: The HSV sliders are only available in the floating color picker window. In the in-panel color picker, you can use the color wheel to adjust hue, saturation, and value.

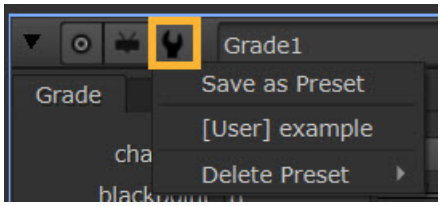
Load Custom Settings with Node Presets

Sometimes the default control values on a node aren't the ones you use most often. If you find yourself frequently adjusting many control defaults, you may want to save your customized default values as a Node Preset.

To Create a Node Preset:

1. Set the controls in the Node Properties to the values you want.
2. Click the **load node presets** button on the control panel title bar and select **Save as Preset**.
3. Give your preset a name and click **Create**.

   Your new preset now appears as an option when you click the **load node presets** button. In this instance, *[User] example*.



Loading and Deleting a Preset

1. To load a preset on a node, click the **load node presets** button and select your preset from the dropdown. User presets are your own personal presets, and shared presets can also be used by other users.
2. To delete a preset, click the **load node presets** button and select **Delete Preset**. In the dialog that displays, select the preset you want to delete from the dropdown menu, and click **Delete**.

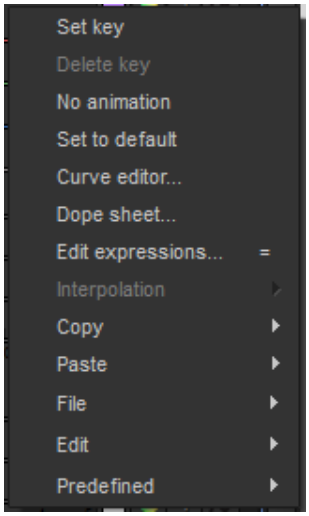
Tip: Set up a Node Preset in the Write node with your desired **file type** to avoid having to select this each time.

Animating Parameters

Animating a parameter refers to changing its value over time. You do so by setting keyframes (frames at which you explicitly define a value) and allowing Nuke to interpolate the values in between. You can animate most of Nuke's parameters in this manner.

Working with Animated Parameters

The **Animation** menu allows you to set key frames, delete keys, and perform other editing operations on the curves for animated parameters.



Setting Keyframes

To set keyframes, do the following:

1. Use a Viewer to navigate to a frame where you want to place a key.
2. Click the animation button  next to the parameter you want to animate.
3. Select **Set key** from the dropdown menu. The parameter's input field turns blue, indicating that a keyframe has been inserted. Nuke switches to autokey mode: when you change the parameters value at another frame, it automatically inserts another keyframe for you.

You can also set a key for all the controls in a node. To do so, right-click on a node's properties panel and select **Set key on all knobs**.

4. Navigate to the next frame where you want to place a key.
5. Edit the parameter's value using the input field, regular slider, or color slider. The moment you change the value, Nuke creates a keyframe.
6. Continue adding key frames as necessary.
7. Use the Viewer's scrubber to preview the result.

Deleting Keyframes

To delete a single keyframe:

1. Use the Viewer's next edit/keyframe and previous edit/keyframe buttons to navigate to the keyframe that you want to remove. Notice that the scrub bar indicates key frames with a blue mark.
2. Click the animation button .
3. Select **Delete key** from the dropdown menu.

   Nuke removes the keyframe.

To delete all key frames from a parameter:

1. Click the animation button.
2. Select **No animation** from the dropdown menu.

   A confirmation dialog appears.
3. Click **Yes**.

   Nuke removes all key frames from the parameter, and sets the static value to match that of the current frame.

Linking Animated Parameters with Tracker

You can link any controls with others by using expressions (see Expressions), but you can also quickly and easily link with the Tracker node using the **Link to** option in the Animation menu. For example, to link the **translate** control of the RotoPaint node with a Tracker node, do the following:

1. Create the Tracker node you want to link to.
2. On the **Transform** tab of the RotoPaint node's control panel, click on the **translate** animation menu.
3. Select **Link to** > **Tracker linking dialog...**
4. Select the Tracker node you want to use in the tracker node dropdown and in the **link to** dropdown, select whether you want to link to the position of the track or the translate values of it.
5. Select which tracks you want to use by checking the **track** boxes.

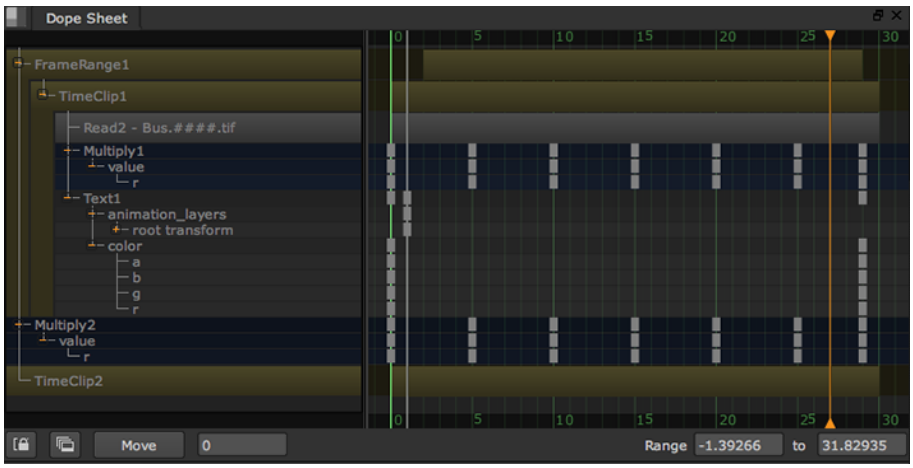
   The expression fields update with the appropriate expression syntax.

6. Then click **OK**, and your linking is done.

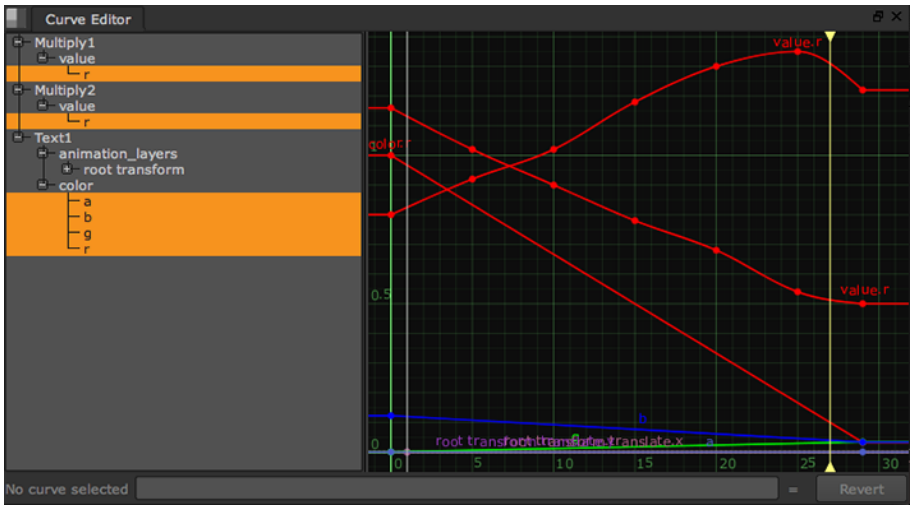
   Your Bezier shape's translate value now changes when the Tracker value is changed.

Animated Parameters in the Dope Sheet and the Curve Editor

As you add keyframes to a parameter, Nuke automatically both adds markers to the Dope Sheet for them and plots a curve on its Curve Editor panel.



The Dope Sheet provides an easy way to edit keyframes as well as reposition, trim, and slip clips.



The Curve Editor allows you to adjust keyframes and the interpolation between them.

You can add key frames, delete key frames, and even adjust the interpolation between key frames without ever looking at the Dope Sheet or Curve Editor. However, as the animation grows more complex, you may

find it easier to edit the animation by manipulating its keyframes directly. For more information on how to do so, see Editing Clips in the Dope Sheet and Using the Curve Editor.

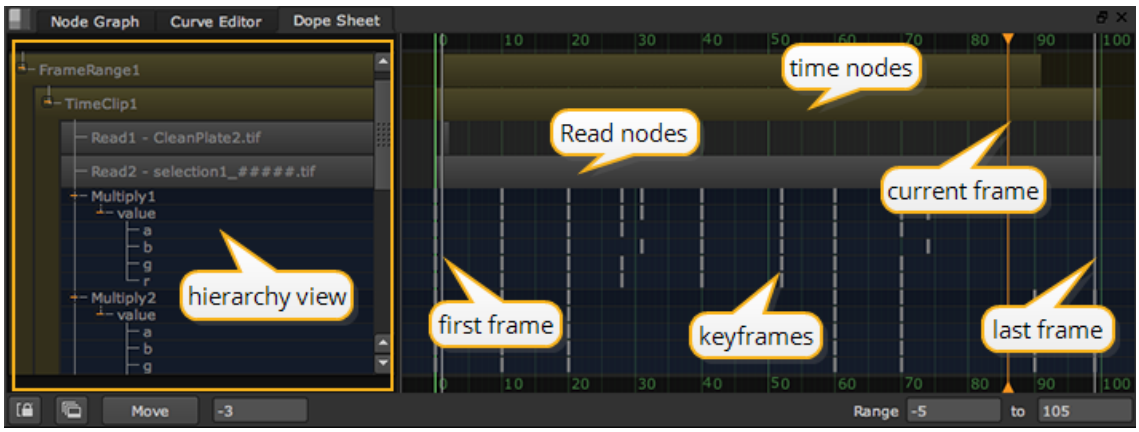
Using the Dope Sheet

The Dope Sheet gives you access to all the keyframes you've created and provides an easy way of editing them. You can also use it to reposition, trim, and slip Read nodes and certain **Time** nodes.

Also see Using the Curve Editor.

Using the Dope Sheet Interface

To show the Dope Sheet, click the **Dope Sheet** tab. If you can't see the tab, right-click the Node Graph title bar and select **Windows** > **Dope Sheet**.



In the Dope Sheet, you can see the following:

Hierarchy view	Click through the hierarchy view to see your nodes and a hierarchical list of animated controls. If you can't see a node in the list, try opening its properties panel. To make a node appear in the Dope Sheet without having its properties panel open, select the **dope sheet** checkbox in the bottom of the **Node** tab of the node's properties panel. You can display the Read and TimeClip nodes in the Dope Sheet when their properties are closed, by clicking the middle button in the bottom-left of the Dope Sheet.

Read nodes	Any Read nodes that have their properties panels open display as gray bars in the Dope Sheet.
Time nodes	Any AppendClip, FrameRange, Retime, TimeOffset, TimeWarp, and TimeClip nodes that have their properties panels open display as green bars.
Keyframes	Your keyframes display as gray markers in the keyframe grid. If you go down to the lowest level of hierarchy, you can also view the keyframes set for different views in stereoscopic or multi-view projects.
Current frame indicator	The current frame indicator displays as an orange line.
First and last frame in the project	The first and last frame in the project's frame range (as defined in the Project Settings) display as white/gray lines.

You can also use the following controls at the bottom of the Dope Sheet:

Control	Description
	Synchronize the frame range of your project between the Dope Sheet and the Curve Editor.
	Display all Read and TimeClip nodes in the Dope Sheet. By default, only the nodes that have their properties panels open show up on the Dope Sheet. If you click this button, all Read and TimeClip nodes in your script are displayed in the Dope Sheet. If you click the button again, the default view is restored.
	Move the selected keyframes by the number of frames specified. You can enter negative values to move the keyframes backwards. **Note:** You cannot move keyframes past other keyframes. If you don't enter a number in the **Move** field, it shows the amount of your previous move.
	Set the frame range to display in the Dope Sheet.

Viewing Keys in the Dope Sheet

To view keys in the Dope Sheet, you need to create some Read nodes, Time nodes, or keyframes first. The Dope Sheet displays:

- All the Read nodes that have their properties panels open.
- Any AppendClip, FrameRange, Retime, TimeOffset, TimeWarp, and TimeClip nodes that have their properties panels open.
- Any nodes that have keyframes set on them and have their properties panels open.

Tip: To force the above nodes to display in the Dope Sheet even when their properties aren't open, check the **dope sheet** box on the **Node** tab of the node's properties panel or select the node in the Node Graph and press **Alt**+**D**.

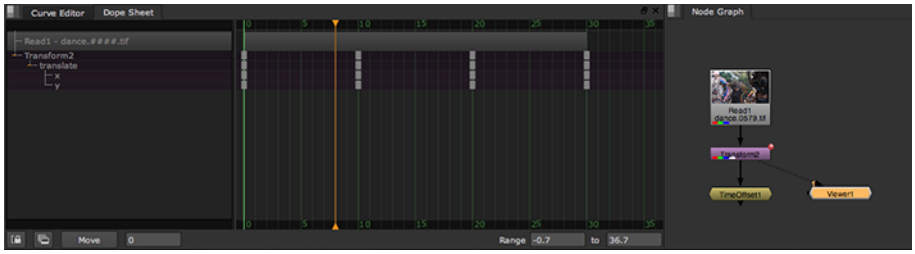
Tip: If you want to view an AudioRead node under your keyframes, you can display it by right-clicking and selecting **View** > **Audio** > **Draw Style** > **Below**. For more information on the audio options, see Audio in Nuke.

The Dope Sheet and Time Nodes

If you have an AppendClip, FrameRange, Retime, TimeOffset, TimeWarp, or TimeClip node downstream of the node being animated, the Dope Sheet shows your keyframes in the context of the currently active Viewer. Whenever you change the Viewer input, the keys in the Dope Sheet move accordingly to reflect where their effect is actually seen in the Viewer:

- If you view keys before a node that moves them in time, you see them at their original time.
- If you view keys after the time manipulation, you see them shifted by the time operation.

For example, in the below script, the Transform node has been animated on frames 0, 10, 20, and 30. When the Viewer is connected to the Transform node, the Dope Sheet shows keys on those frames.



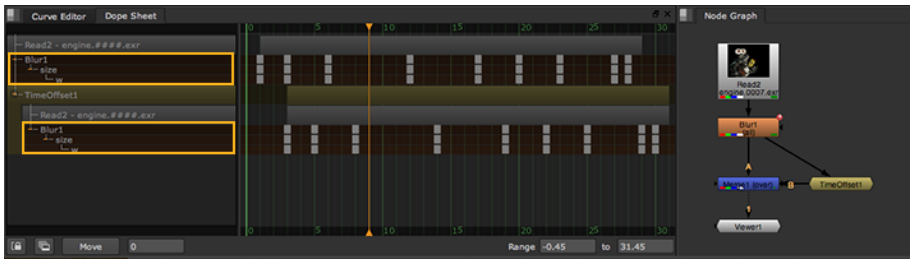
There is also a TimeOffset node in the script, set to move the input clip 5 frames forward in time. When the Viewer is connected to the TimeOffset node, the Dope Sheet shows the keys shifted by 5 frames - on frames 5, 15, 25, and 35.

Multiple Branches of Nodes in the Dope Sheet

If there are multiple paths from an animated node to the currently active Viewer, the keyframes for the animated node appear in the Dope Sheet multiple times. For example, in the below script, there are two paths from the animated Blur node to the Viewer:

- Blur > Merge > Viewer
- Blur > TimeOffset > Merge > Viewer

As a result, the keyframes for the Blur node appear in the Dope Sheet twice. Adjusting the keys in one location also updates them in the other.

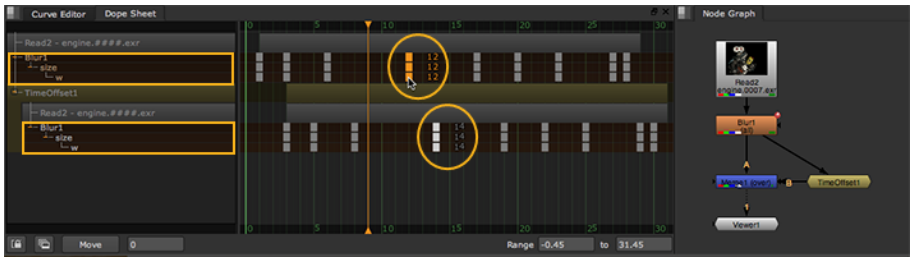


Selecting Keyframes

There are several ways to select and edit different keyframes:

Select multiple frames	Drag a marquee over your keyframes in the keyframe view.
Add or remove keyframes from a marquee selection	Hold **Shift** and drag another marquee around some of the selected keyframes.
Select a single keyframe	Click it in the keyframe view. Its parent keyframe and all other keyframes depending on it are selected too. The frame number of the keyframe is displayed next to it.
Select all keyframes in one node	Click the node name in the hierarchy view. All the child keyframes under that node are selected.

If there are multiple paths from an animated node to the currently active Viewer, the keyframes for the animated node appear in the Dope Sheet multiple times. When you select them in one location, they turn orange, while the corresponding keys in other locations turn white.



Note: When selecting keys that appear in the Dope Sheet multiple times (using multiple paths), marquee selection is constrained to one path only.

Copying and Pasting

You can copy and paste keyframes in the Dope Sheet. After selecting the keyframe(s) you want to copy, use the **Ctrl/Cmd**+**C** keyboard shortcut to copy the keyframe(s), and **Ctrl/Cmd**+**V** to paste them in another location of the Dope Sheet.

When using the **Ctrl/Cmd**+**V** keyboard shortcut, pasting keyframes is relative to the location of the current frame indicator. In other words, if you have keyframes at 1, 10, and 20 and you copy this selection before pasting the selection at frame 21, new keyframes are set at frames 21, 30, and 40.

To paste keyframes on the same frame as the source keys, select the control you want to receive the keys, right-click on the Dope Sheet, and select **Edit** > **Paste** > **Paste Absolute**.

Deleting and Adding Keyframes

You can remove keyframes in the Dope Sheet. To delete a keyframe, just select it and press **Delete**/**Backspace** button or right-click on the keyframe and select **Edit** > **Erase**.

You can also add keyframes in the Dope Sheet to add animation where there isn't any at the moment. Add keyframes by **Ctrl/Cmd**+**Alt**+clicking on an empty spot in the Dope sheet. A new keyframe is created and you can adjust it as necessary.

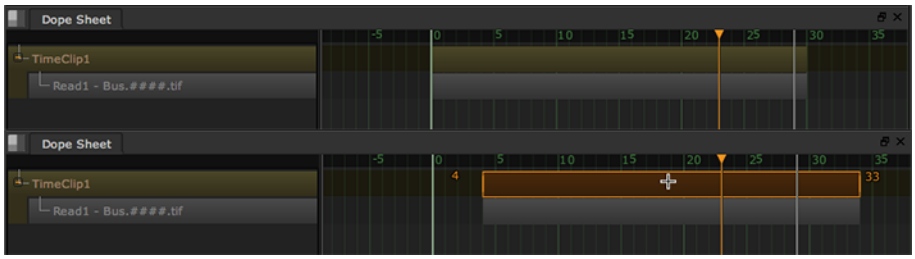
Editing Clips in the Dope Sheet

The Dope Sheet allows you to easily reposition, trim, and slip your clips as well as set a frame range for them:

- Repositioning clips refers to changing the position, but not the content or the duration, of the clip. In the Dope Sheet, you can reposition clips using the Read, TimeOffset, and TimeClip nodes.
- Trimming clips refers to removing unwanted frames from the head or tail of the clip. In the Dope Sheet, you can trim clips using the Read and TimeClip nodes.
- Slipping clips refers to changing the content of the clip that is seen, but not the position or duration. In the Dope Sheet, you can slip clips using the Read and TimeClip nodes.
- Setting a frame range for a clip controls which frames a downstream AppendClip node uses from the input, which frames are displayed in the Viewer when the timeline range dropdown menu is set to **Input**, and which frames are sent to the flipbook. In the Dope Sheet, you can set the frame range for a clip using the FrameRange node.

Repositioning, Trimming, and Slipping Clips Using Read and TimeClip Nodes

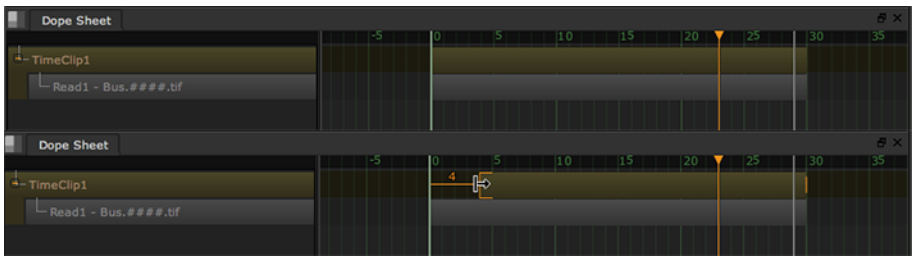
- To reposition a clip, place the cursor over the Read or TimeClip bar in the Dope Sheet and drag left or right.



The new start and end frame numbers are displayed on either side of the bar, and the **frameoffset** control in the Read or TimeClip properties panel is adjusted automatically.

Tip: You can also reposition a clip by selecting the bar, double-clicking on the first and last frame numbers next to it, and adjusting them in the **x** fields that pop up.

- To trim a clip, place the cursor in the beginning or end of the bar and drag left or right.



The new start or end frame is displayed next to the bar, and an orange line appears to represent the original frame range. The **framerange** control in the Read or TimeClip properties panel is adjusted automatically.

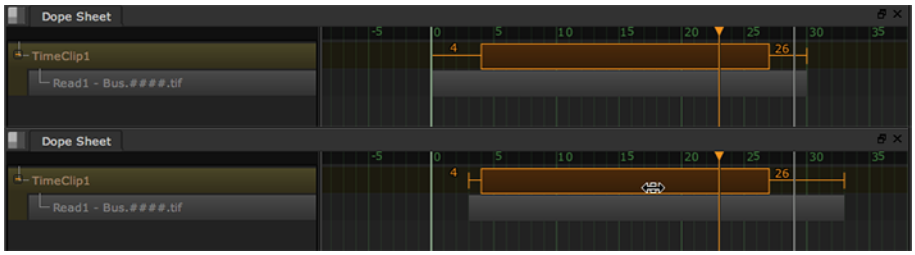
Tip: You can also trim a clip by selecting the bar, double-clicking on the first or last frame number next to it, and adjusting the number in the **x** field that pops up.

Tip: The number of frames you can trim from either end of a clip is constrained by the clip's original frame range. However, if necessary, you can trim beyond the original range by holding **Ctrl**/**Cmd** while trimming.

If necessary, you can change how the clip displays inside the clip handles (the unused frames beyond the start or end of the clip). Adjust the **before** and **after** dropdown menus next to the **frame range** control in the Read or TimeClip properties to change what happens before and after the frame range limits:

- **hold** - select to show a still picture of the first/last frame of the frame range.
- **loop** - select to start over and keep looping the span of the frame range outside the first/last frame of the frame range.
- **bounce** - select to play the span of the frame range backwards and forwards between the frame range limits.
- **black** - select to display a black frame outside of the first/last frame.

- To slip a clip, first trim the clip to create clip handles. Then, place the cursor over the bottom half of the bar and drag left or right.

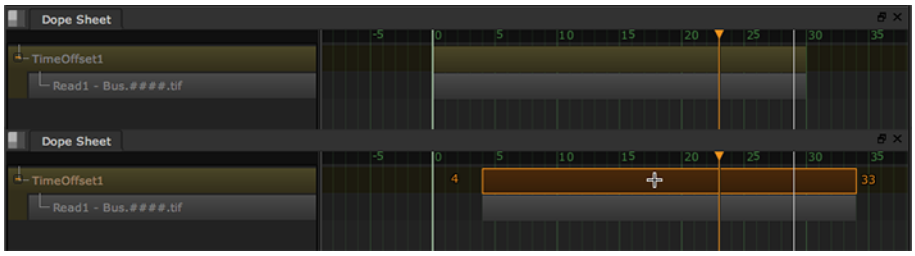


The clip handles move to indicate what portion of the clip is visible. The **frame range** and **frame offset** controls in the Read or TimeClip properties panel are adjusted automatically.

Tip: The number of frames you can slip a clip by is constrained by the clip's original frame range. However, if necessary, you can slip beyond the original range by holding **Ctrl**/**Cmd** while slipping.

Repositioning Clips Using TimeOffset Nodes

To reposition a clip, place the cursor over the TimeOffset bar in the Dope Sheet and drag left or right.

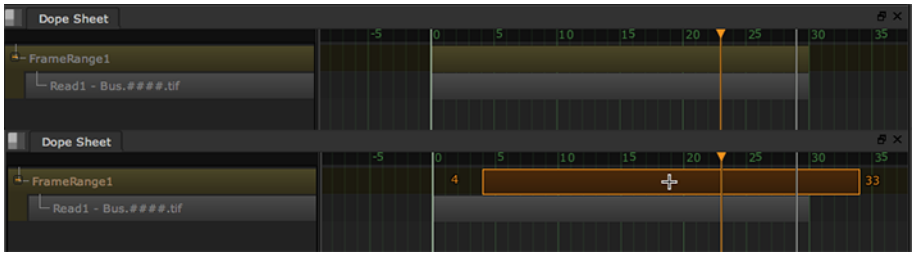


The new start and end frame numbers are displayed on either side of the bar, and the **time offset (frames)** control in the TimeOffset properties panel is adjusted automatically.

Tip: You can also reposition a clip by selecting the bar, double-clicking on the first and last frame numbers next to it, and adjusting them in the **x** fields that pop up.

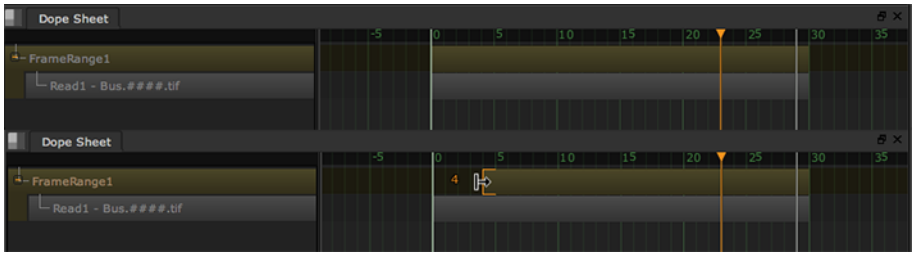
To Set the Frame Range Using FrameRange Nodes

- To set both the first and the last frame for a clip, place the cursor over the FrameRange bar in the Dope Sheet and drag left or right.



  The new start and end frame numbers are displayed on either side of the bar, and the **frame range** control in the FrameRange properties panel is adjusted automatically.

- To set only the first or the last frame for a clip, place the cursor in the beginning or end of the bar and drag left or right.



  The new start or end frame is displayed next to the bar, and the **frame range** control in the FrameRange properties panel is adjusted automatically.

Tip: You can also set the frame range for a clip by selecting the bar, double-clicking on the first and last frame numbers next to it, and adjusting them in the **x** fields that pop up.

Synchronizing the Frame Range

You can synchronize the frame range of your project between the Dope Sheet and the Curve Editor. This is useful when you're using both of them to edit your animation curves and keyframes at the same time. To synchronize the frame range:

1. Right-click anywhere on the Dope Sheet.
2. Select **View** > **Synchronize frame range**. Any changes you now make to the frame range in the Dope Sheet are applied in the Curve Editor too, and any changes you make in the Curve Editor are also reflected in the Dope Sheet. Zooming in and out in either also zooms the other view.

Tip: You can also do this by clicking the synchronize button at the bottom of the Dope Sheet.

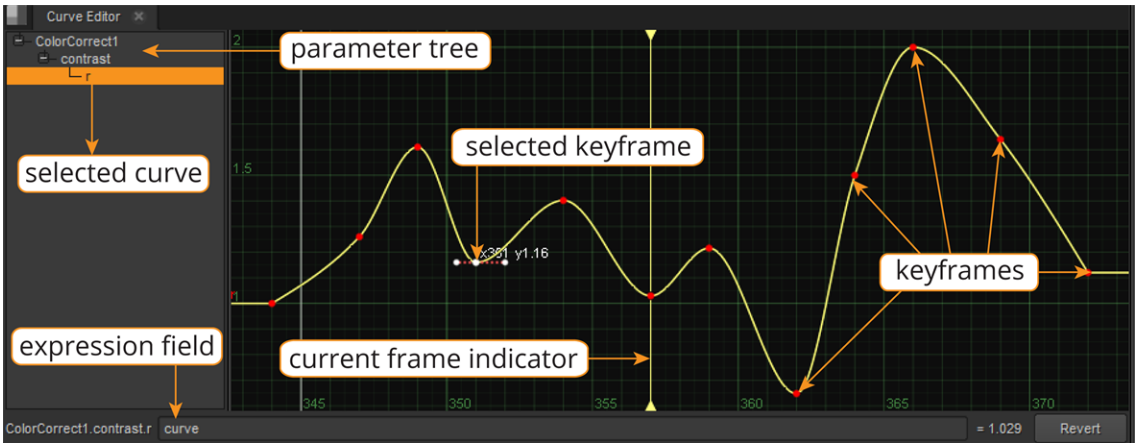
Using the Curve Editor

The Curve Editor enables you to edit curves without physically entering information in the **Properties** pane.

Displaying an Animation Curve

To reveal an animation curve, do the following:

1. Click the animation button next to the parameter whose curve you wish to view.
2. Select **Curve Editor**. The Curve Editor panel appears with a focus on the selected parameter's curve.
   The vertical, or y axis, denotes the value of the parameter.
   The horizontal, or x axis, denotes time (in frame units).



Displaying Curves in the Editor

To display curves in the editor, do the following:

1. In the parameter tree on the left, click the **+** and **-** signs to expand and collapse the hierarchy as necessary.
2. Click a parameter's name to make its curve the focus of the editor. To focus on multiple curves at the same time, either:
   - Hold **Shift** and click on curves to create a consecutive list,

     OR
   - Hold **Ctrl/Cmd** and click on individual curves to add them to the selection.
3. To display separate curves for each channel, separate the channels for the relevant control in the node's properties panel.

The parameter tree on the left allows you to focus on any curve in the script.

Tip: If you want to view an AudioRead node under your animation curves, you can display it by right-clicking anywhere in the curve editor and selecting **View** > **Audio** > **Draw Style** > **Below**. For more information on the audio options, see Audio in Nuke.

Enabling/Disable a Curve in the Editor

To enable or disable a curve in the Editor, in the parameter tree on the left, click the **+** and **-** signs to expand and collapse the hierarchy as necessary.

Zooming In or Out in the Editor

To zoom in or out in the Editor, do the following:

1. Click on the area you want to zoom in on or out of.
2. Press the **+** button to zoom in, or the **-** button to zoom out,

   OR

   Scroll up with the mouse wheel to zoom in, or down to zoom out.

Tip: Depending on your mouse preferences, you can zoom to a custom area in the Curve Editor, middle-click on the Editor and drag to select an area with a marquee. When you release the mouse button, the Editor zooms to fit the selected area in the Editor.

Panning in the Editor

To pan in the Editor, hold the middle-mouse button and drag over the Editor. You can also use **Alt**+drag.

Resetting Zoom and Panning

To reset the zoom or panning:

1. Right-click on the Curve Editor.
2. From the menu that opens, select **View** > **Frame All**.

   OR

   Press **A** on the Editor.

   OR

   Click the middle-mouse button (dependent on your mouse preferences).

   Nuke centers the curve in the Editor, resetting the zoom.

Centering a Portion of the Curve

To center a point of the curve in the Editor:

1. Select the points you want to center in the editor.
2. Right-click on the Editor, and select **View** > **Frame Selected** (or press **F** on the Editor).

   Nuke centers the selected portion of the curve in the editor. If no points are selected, Nuke centers the selected curve, or all curves.

Editing Curves

You edit curves by moving the points on the curve to new locations. If necessary, you can add more points to the curve. You can also sketch curves freely, use the usual editing functions such as copy and paste, smooth curves with filtering, interpolate curves, loop, reverse or negate curves, and use expressions to modify curves.

Adding Points to a Curve

To add points to a curve, do the following:

1. Click on the curve you want to edit. The curve turns yellow to indicate it's selected.
2. **Ctrl**/**Cmd**+**Alt**+click on the part of the Curve Editor you want to add a point to. You can add points both on the curve and outside the curve,

   OR

1. Right-click on the Editor and select **Edit** > **Generate**. The **Generate keys** dialog opens.
2. In the **Start at** field, enter the first frame you want to use as a keyframe.
3. In the **End at** field, enter the last frame you want to use as a keyframe.
4. In the **Increment** field, enter the frame increment you want to use between the first and the last keyframe. For example, if you want every tenth frame to be a keyframe, enter **10**.
5. In the last field, enter the value you want to use for y. If you do not enter a value here, the key frames are added to the current curve without modifying the curve shape.
6. Click **OK**.

Selecting Points on a Curve

You can select points in the following ways:

- To select individual points, click on the point you want to select.
- To select multiple points, **Shift**+click on the points, or drag a marquee around them.

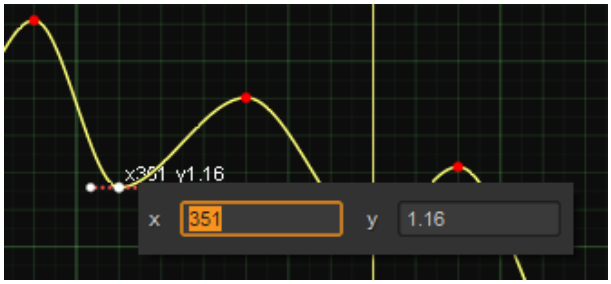
  A box is drawn around the points to indicate they have been selected.
- To select all points, press **Ctrl**+**A** (Mac users press **Cmd**+**A**).

  A box is drawn around the points to indicate they have been selected.

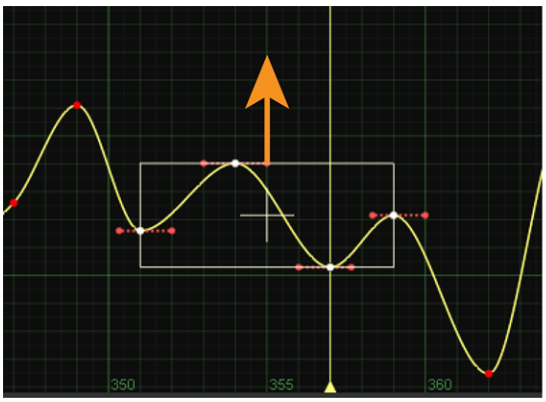
Moving Points on a Curve

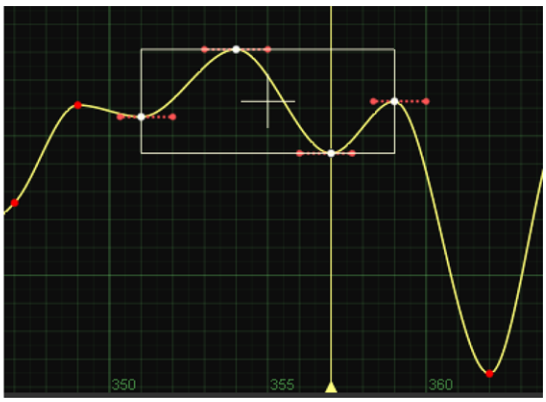
- To move a point along either the x or y axis only, drag the point to a new location.

- To move a point in any direction, **Ctrl**+drag (Mac users **Cmd**+drag) the point to a new location. You can also nudge points using the numeric keypad arrows.
- To adjust the values of a point numerically, select the point and double-click on the x or y value that appears next to it.



- By default, when you move a point, its position on the x axis is rounded to the nearest integer. To disable this, you can right-click on the Curve Editor and select **Edit** > **Frame Snap**. You can also momentarily disable the snapping by pressing **Shift** while moving a point.
- To move several points at the same time, select them and drag the selection box to a new location.





To add or remove points to or from the selection box, **Shift**+click on the points.

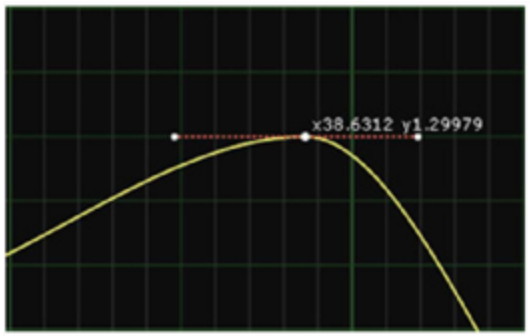
To resize and scale the selection box, drag its edges. If the selection box is very narrow, you can press **Ctrl**/**Cmd** when resizing it. This allows you to resize the box in one dimension only. For example, if you have a box that's wide on the x axis but flat on the y axis, you can resize it in this way along the x axis.

To avoid accidentally moving a point inside the selection box, press **Ctrl**/**Cmd**+**Shift** when dragging the box to hide the points inside the box.

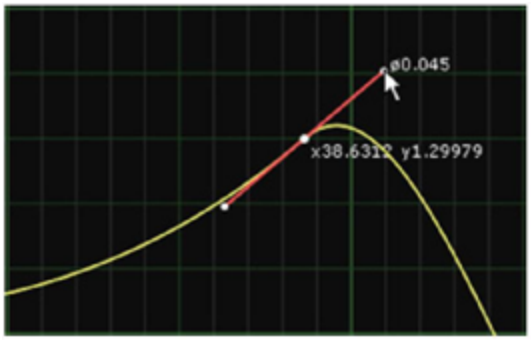
Adjusting the Slope Around the Points

To adjust the slope around the points, do the following:

1. Select a point on the curve. Red tangent handles appear on both sides of the point.



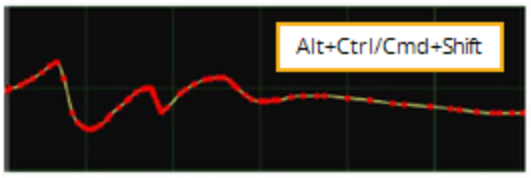
2. Drag the tangent handles to a new location. The curve follows the handles.



Sketching a Curve

You can sketch a curve freely by doing the following:

Press **Alt**+**Ctrl**+**Shift** (Mac users press **Alt**+**Cmd**+**Shift**) while drawing a curve on the editor. Nuke sketches a curve that follows your mouse movements.



Cutting, Copying, and Pasting

You can cut, copy, or paste any selected points, expressions, or curves, by doing the following:

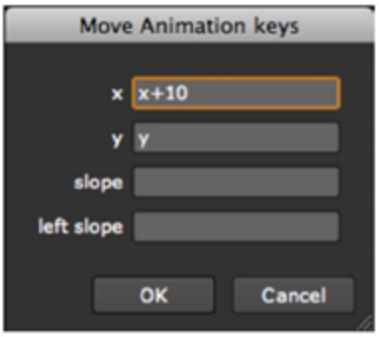
1. Right-click on the Curve Editor.
2. From the menu that opens, select **Edit** and the editing function you want to use on the entire curve, for example:
   - **Edit** > **Copy** > **Copy Selected Keys** to only copy the points you have currently selected.
   - **Edit** > **Copy** > **Copy Curves** to copy an entire curve.

- **Edit** > **Copy** > **Copy Expressions** to copy the expression that creates the curve.
- **Edit** > **Copy** > **Copy Links** to copy a curve and keep its values linked to the original curve, so that if you change the original, your changes also affect the copied curve.

Move Selected Points on the Curve

To move selected points on the curve by a fixed value:

1. Select all the points you want to move.
2. Right-click on the editor and select **Edit** > **Move**. The **Move Animation Keys** dialog opens.
3. In the **x** and **y** fields, define how you want to move the points along the x and y axes. For example, to shift the selected points to the right by a value of 10, enter **x+10** in the x field.

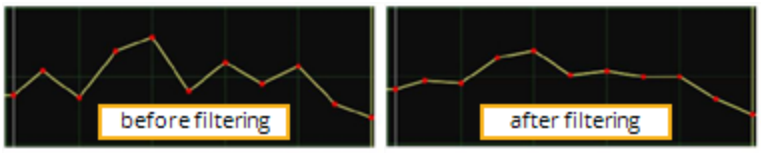


4. In the **slope** and **left slope** fields, define how you want to move the points' tangent handles.

Smoothing the Curve

To smooth the curve with filtering, do the following:

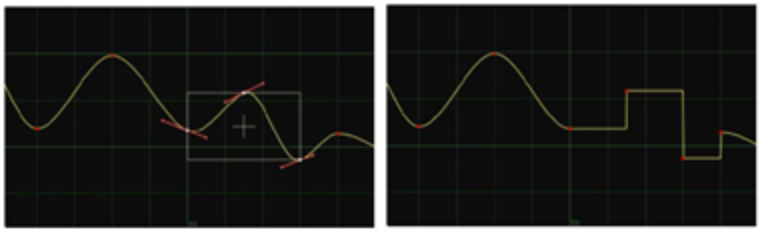
1. Select the portion of the curve that needs smoothing.
2. Right-click on the editor and select **Edit** > **Filter**. The **Filter Multiple** dialog opens.
3. In the **No. of times to filter** field, specify how many times you want to filter the curve. Filtering sets new values on each point based on the average values of their neighboring points. The more filtering, the smoother the curve.



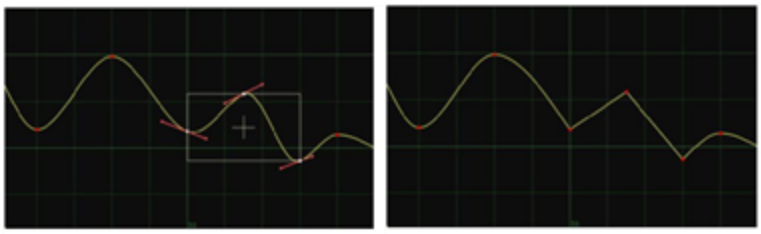
Interpolating Parts of a Curve

You can interpolate parts of a curve, by doing the following:

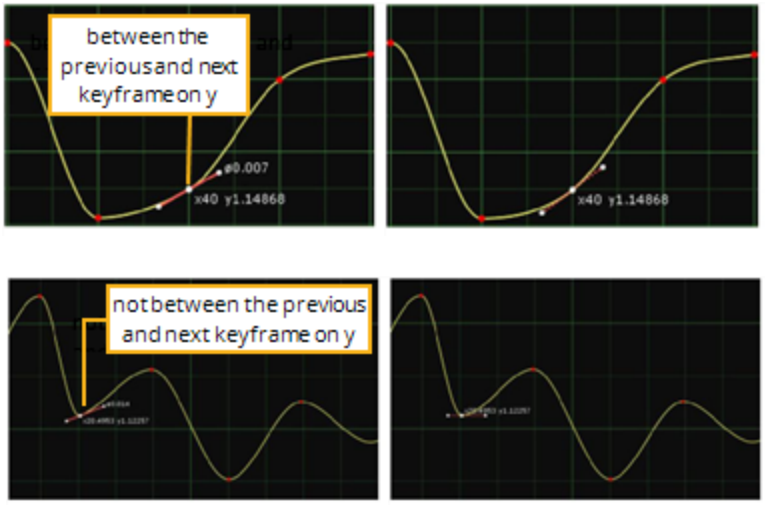
1. Select the point(s) between or around which you want to interpolate the curve.
2. Right-click on the Editor. Select **Interpolation** and the type of interpolation you want to use. Select:
   - **Constant** to force a constant value after each selected point.



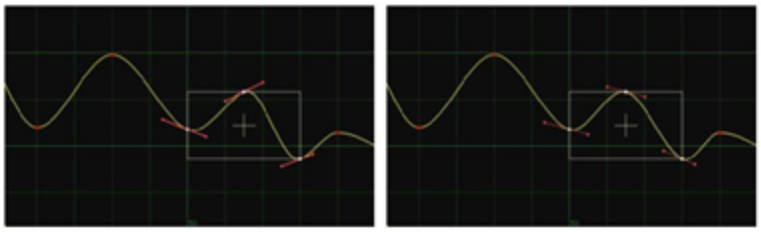
   - **Linear** to use linear interpolation. This produces sharp changes at key frames and straight lines between them.



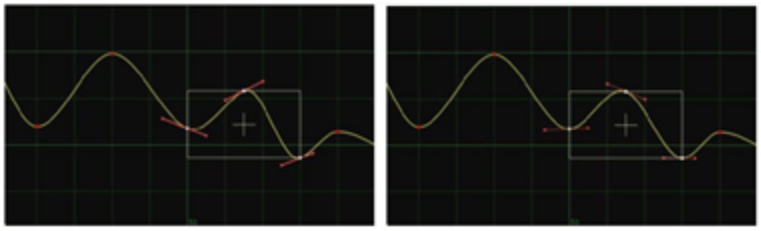
   - **Smooth** to set the tangents' slopes equal to the slope between the keyframe to the left and the keyframe to the right if the selected point is between these two key frames along the y axis. If the selected point is not between these key frames and has a larger or smaller value than both key frames, the tangents' slopes are made horizontal. This ensures the resulting curve never exceeds the keyframe value.



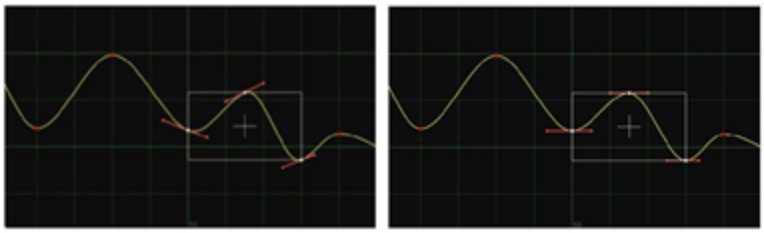
   - **Catmull-Rom** to set the tangents' slope equal to the slope between the keyframe to the left and the keyframe to the right regardless of where the selected point is located. The resulting curve can exceed the keyframe values.



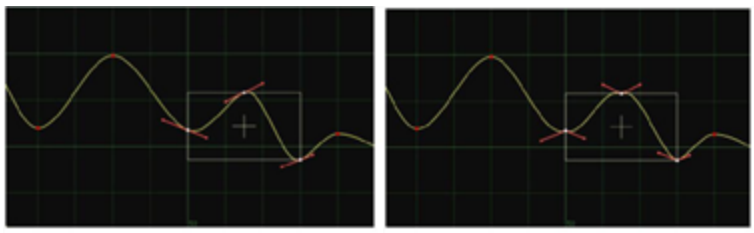
• **Cubic** to set the slope so that the second derivative is continuous. This smooths the curve.



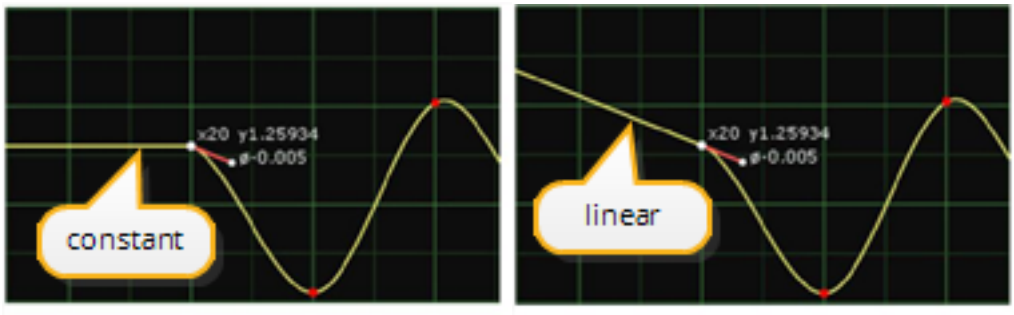
• **Horizontal** to make the tangents horizontal, setting the slope around the selected points to zero.



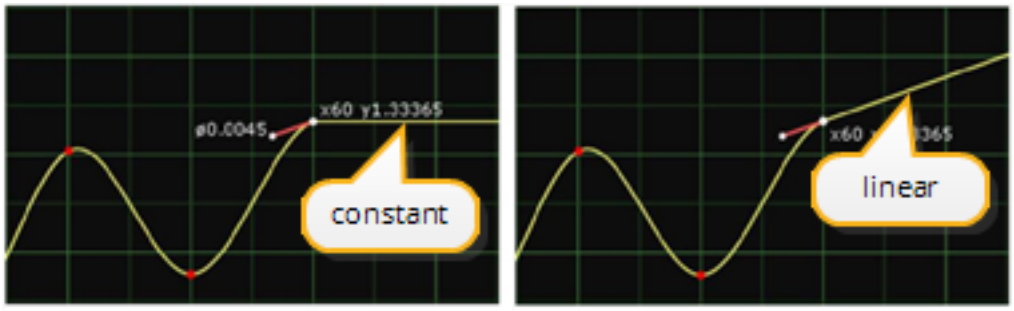
• **Break** to adjust the two tangents of a selected point independent of each other.



• **Before** > **Constant** or **Linear** to interpolate the parts of the curve that are on the left side of the first point. This option only works if you have selected the first point on the curve.



• **After** > **Constant** or **Linear** to only interpolate the parts of the curve that are on the right side of the last point. This option only works if you have selected the last point on the curve.

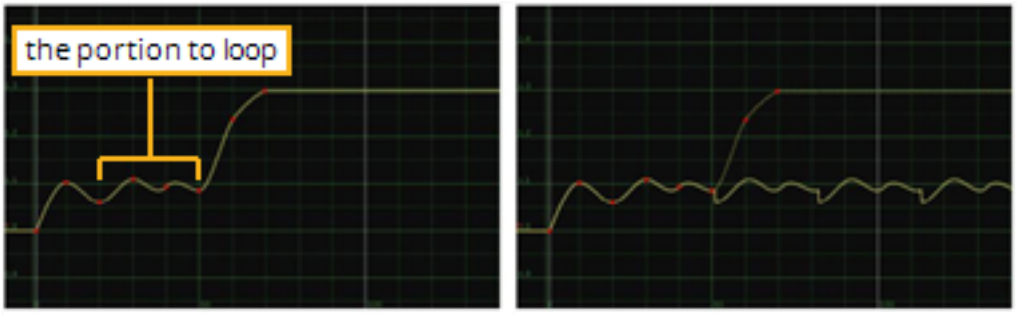


Repeating a Part of the Curve

To repeat a portion of the curve, throughout the curve, do the following:

1. Right-click on the editor and select **Predefined** > **Loop**. The **Loop** dialogue opens.
2. In the **First frame of loop** field, enter first frame of the portion you want to repeat throughout the curve.
3. In the **Last frame of loop** field, enter the last frame of the portion you want to repeat.
4. Click **OK**.

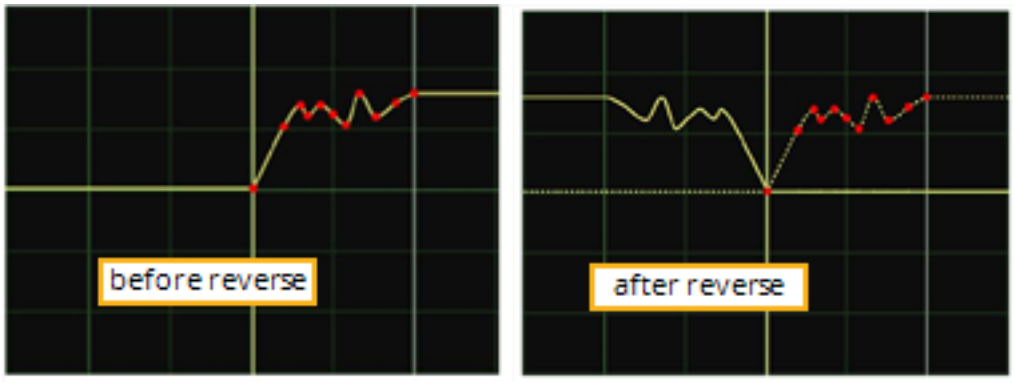
   The shape of the curve between these frames is repeated throughout the rest of the curve. The solid line represents the actual curve, and the dotted line the original curve with the key frames.



Reversing a Curve

You can reverse a curve by right-clicking on the editor and select **Predefined** > **Reverse**.

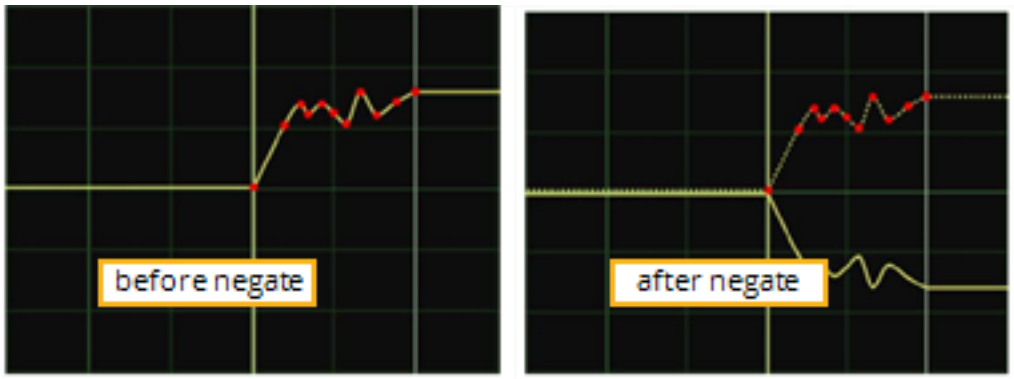
This makes the curve go backward in time. Both the new curve and the original curve are displayed. The solid line represents the actual curve, and the dotted line contains the key frames that you can modify.



Negating a Curve

You can negate a curve by right-clicking on the editor and select **Predefined** > **Negate**.

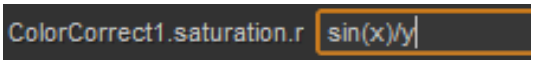
The curve becomes the negative of the key frames. For example, a value of 5 turns into -5. Both the new curve and the original curve are displayed. The solid line represents the actual curve, and the dotted line contains the key frames that you can modify.



Modifying a Curve with an Expression

You can modify a curve using an expression by doing the following:

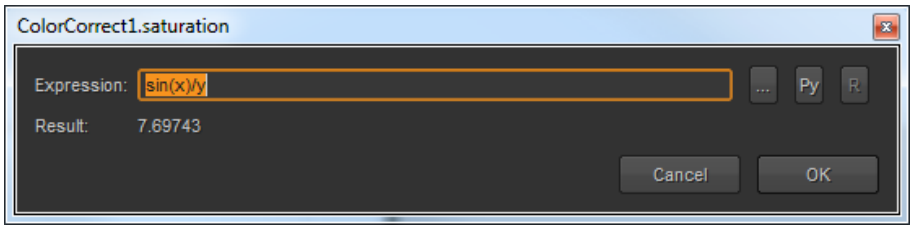
1. Enter the expression in the expression field at the bottom of the Curve Editor,



2. Click the **Revert** button to set the field back to previous valid expression.

OR

1. Right-click on the Editor, and select **Edit** > **Edit expression**.
2. In the dialog that opens, type the expression you want to use for the curve, for example, **sin(x)/x**.

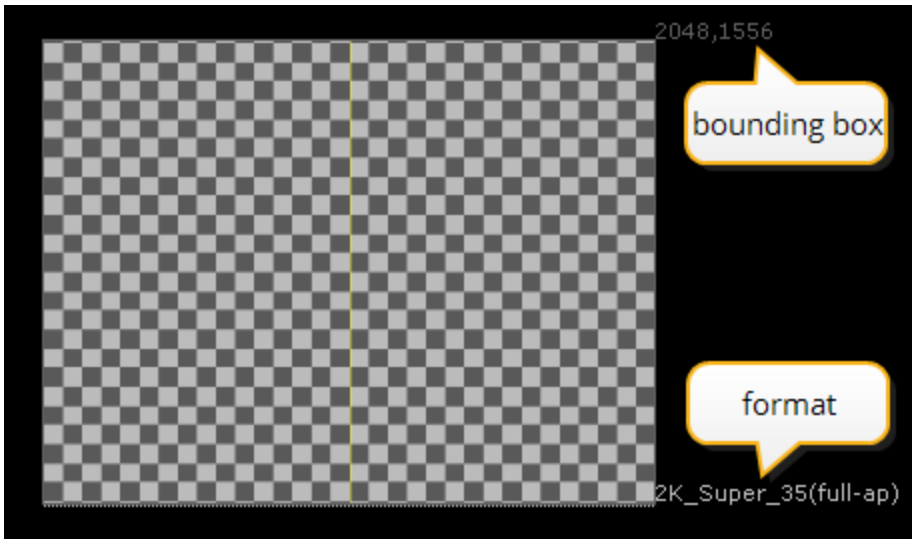

ColorCorrect1.saturation
Expression:
sin(x)/y
...
Py
R
Result:
7.69743
Cancel
OK

3. Click **OK**.

Compositing Viewers

A Viewer node displays the render output of any connected process nodes in the Viewer panel and does not edit any data. Viewer nodes allow you to quickly assign the right values to parameters as they allow you to edit in context - that is, edit a given node's parameters upstream in a script while viewing the effect of those changes downstream.

Nuke's Viewer output is labeled with the bounding box size and the format. The bounding box is the area the Nuke treats as containing image data and the format is the resolution of the input image. These two are the same by default so that the output is the same as the resolution.



See Reformatting Elements and Adjusting the Bounding Box for more information on format and bounding boxes.

You can place as many Viewer nodes in a script as you wish, which allows you to simultaneously view multiple outputs. You can also pipe the output from up to ten process nodes into a single Viewer node, and then cycle through the various displays. This allows you to easily compare an image before and after processing by a given effect.



Note: The maximum image size the Nuke Viewer can display is 2^32 = 4,294,967,296 pixels. This is the same as 64k x 64k, 128k x 32k, 256k x 16k, or 512k x 8k. If your image is larger than this, it is resized and you get the following warning:
"Viewer image is clipped to <size> x <size>!"
For example, if your image resolution is 60,000 x 4473, Nuke is able to display the image because the number of pixels is less than 2^32. However, if the resolution is 110,000 x 50,000 (more than 2^32 pixels), the image is resized to 85,899 x 50,000.
In addition to the Viewer, this limit is also applied to the bounding box of the images being passed between each node.

Adding Viewer Nodes

Viewers have corresponding nodes that appear in the Node Graph. These nodes do not produce output for rendering; they generate display data only. You can connect Viewer nodes as described in Working with Nodes. In practice, it is faster to use the Viewer keyboard shortcuts described below.

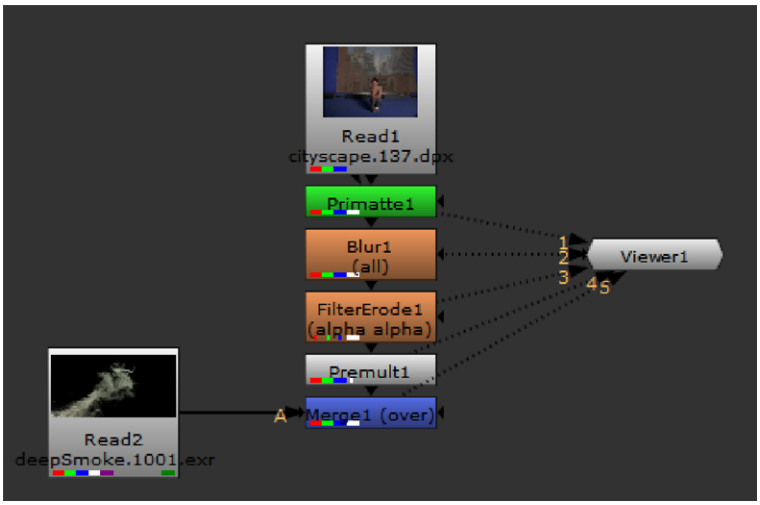
To add a Viewer node:

1. Select the node that you want to view the output of.
2. Do one of the following:
   - Using the menu bar, choose **Viewer** > **Create New Viewer**.
   - Using the Toolbar, choose **Image** > **Viewer**.
   - Using a keyboard shortcut, press **Ctrl**/**Cmd**+**I**.

   Nuke connects a Viewer node to the node you selected in step 1, and displays the output of the node in the Viewer panel. You can also insert a Viewer node and set up its first connection by simply pressing **1** over the Node Graph.

Connecting Viewer Nodes

After you add a Viewer node to the script, you can quickly pipe any process node's output to it simply by selecting the process node then pressing any number key. Doing so, pipes the output to one of the ten input ports available on every Viewer node (the **0** key represents the tenth slot).



Toggling Views

If a Viewer node has multiple inputs, like the one depicted above, you can press the up or down arrow keys to quickly cycle through the views (your cursor needs to be in the Viewer window). To view a particular node press the number key (**1, 2, 3... 0**) corresponding to the pipe number whose contents you wish to view.

Panning and Zooming the Viewer Window

Nuke offers several ways, including some useful keyboard shortcuts, to focus on the area of the Viewer you need to see.

Panning the Frame

To pan the frame, hold the middle-mouse button and drag on the display (you can also use **Alt**+drag). The frame follows the mouse pointer.

Note: In many Linux windows managers, the **Alt** key is used by default as a mouse modifier key. This can cause problems in 3D applications where **Alt** is used for camera navigation in 3D environments.
You can use key mapping to assign the mouse modifier to another key, such as the (Super or Windows logo) key, but the method changes depending on which flavor of Linux you're using. Please refer to the documentation on key mapping for your particular Linux distribution for more information.

Re-centering the Frame

You can re-center the frame by clicking the middle-mouse button or pressing **F**.

Zooming In and Out

To zoom in on the frame:

1. Move your pointer over the area of the display on which you want to zoom.
2. Drag the mouse while pressing the middle mouse button and the left mouse button.
   OR
   Press the plus button (**+**) repeatedly until the frame attains the desired scale.
   OR
   Select **zoom in** from the zoom dropdown menu in the top right corner.

To zoom out from the frame:

1. Move your pointer over the area of the display from which you want to zoom.
2. Drag the mouse while pressing the middle mouse button and the left mouse button.
   OR
   Press the minus button (**-**) repeatedly until the frame displays at the desired scale.
   OR
   Select **zoom out** from the zoom dropdown menu in the top right corner.

Restoring the Zoom to 100%

You can restore the zoom to 100% by pressing **Ctrl**/**Cmd**+**1**.

Showing / Hiding Floating Viewers

You can hide and show floating Viewers using a handy keyboard shortcut.

To hide or show a floating Viewer, press **`** (the backtick key).

Showing / Hiding Viewer Toolbars

To hide or show the top toolbar, press **{**.

To hide or show the bottom toolbar, press **}**.

Locking the Viewer Zoom Level

You can choose to lock the Viewer zoom level, so that it doesn't change when you switch between inputs of different sizes. Right-click on the Viewer, and select **Prevent Auto Zoom** to toggle between maintaining the same zoom level for all inputs (on) and changing it according to on-screen image dimensions (off). Alternatively, you can also press **Alt+Z** on Viewer, or toggle **prevent auto zoom** in the Viewer settings.

Viewing Overscan in the Viewer

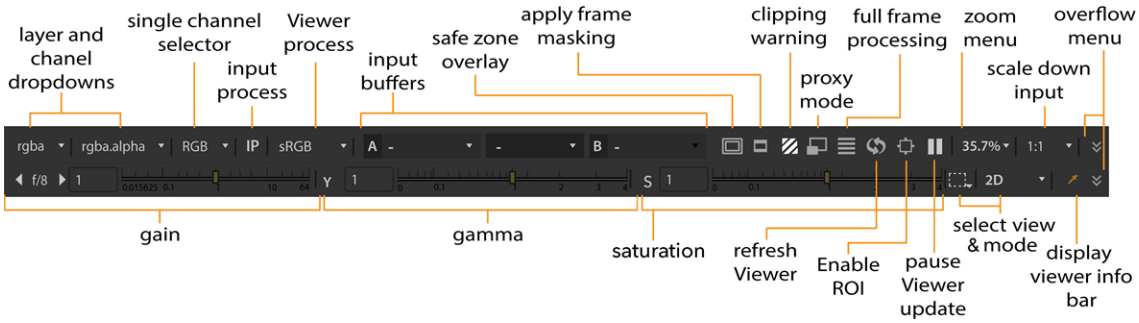
If you want the Viewer to show overscan, that is any pixels extending beyond the left/right or top/bottom of the frame, you can check the **showoverscan** box in the Viewer settings or check the option in the Viewer right-click menu. Viewing pixels beyond the edges of the frame can be useful if some of your nodes need to have access outside the frame. You can also control the amount of overscan shown by specifying the pixel amount in the **overscan** field.

Using the Viewer Controls

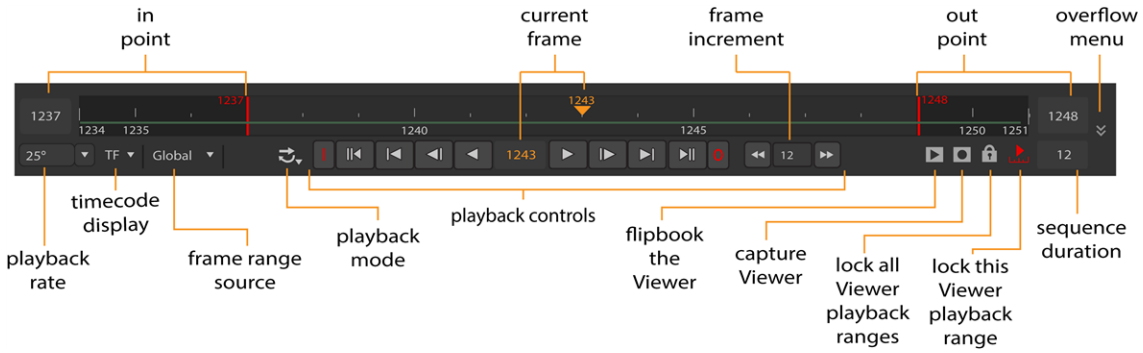
A Viewer's on-screen controls let you navigate the timeline, display channels, zoom, choose cameras (3D mode), and create display wipes and composites.

Compositing Viewer Controls

There are many useful tools at the top of the Viewer, some of which allow you to select layers and channels, adjust gain and gamma, and zoom and scale down the image in the Viewer.



The tools at the bottom of the Viewer allow you to adjust the playback settings, including setting the frame range, selecting the playback mode, and locking the Viewer playback range.



Drag the orange marker along the timeline to quickly cue to a specific frame. The number of the current frame appears below the center of the timeline. You can also cue to a frame by typing its number directly into this field.

Tip: The current frame and in an out point fields also accept simple mathematical functions, such as +/-20 to jump forward or backward 20 frames.

By default, Nuke automatically adjusts the timeline of every Viewer window to show the frame range defined in your Project Settings. If no frame range is defined, the frame range of the first image you read in is used as the global frame range.

Viewer timeline controls also have a frame range source dropdown menu that you can use to define where the timeline gets its frame range from. You can set this menu to **Global**, **Input**, or **Custom**. **Global** is the default setting described above.

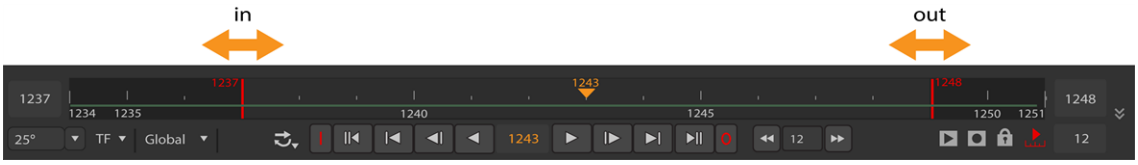
The **playback rate** field (frames-per-second) displays the project's playback speed. Nuke attempts to maintain this speed throughout playback, although this adjusts depending on the resolution of the imagery and your hardware configuration.

Note: The asterisk (*) denotes the playback speed selected using the **Project Settings** > **Root** > **fps** control.

To have the Viewer adjust the timeline to show the "in" and "out" frames of the current input clip, select **Input** from the frame range source dropdown menu. The number of the first frame in the clip is shown in the left end of the timeline and the number of the last frame in the right end. If you change the input of the Viewer, the frame range on the timeline is adjusted accordingly.

To manually adjust the frame range for the current Viewer window, pan and zoom on the timeline until you see the desired frame range and **Custom** becomes selected in the frame range source dropdown menu. **Alt**+drag to pan, and **MMB**+drag to zoom in. You can also zoom in on or out of the timeline using the mouse wheel. To reset the zoom, press the middle mouse button over the timeline.

To adjust the playback range for the current Viewer window, click+drag the red playback range markers on the timeline to a new "in" and "out" frames as shown below, or enter a new playback range in the playback range field and press the frame range lock button.



To toggle between the new playback range and the visible timeline range, click the lock frame button again.

The **fps** field (frames-per-second) initially displays the project's playback speed. Nuke attempts to maintain this speed throughout playback, although this adjusts depending on the resolution of the imagery and your hardware configuration.

The following table lists the functions of the playback buttons:

Buttons	Functions
	The **Play backward** and **Play forward** buttons play the sequence backward or forward at the script's frame rate. When you press a play buttons, it toggles to a stop a button.
	The **Back 1 Frame** and **Forward 1 Frame** buttons cue the sequence to the previous or next frame.
	The **Previous keyframe** and **Next keyframe** buttons cue the sequence to the script's previous or next keyframe.
	The **First frame** and **Last frame** buttons cue the sequence to the first and last frame.
	The **Frame Increment** field allow you to specify the number of frames by which the Previous increment/Next increment buttons cue the sequence. This is set to 10 frames by default.

The **Playback Mode** button lets you control how many times and in what direction the Viewer plays back the sequence. Click the button to toggle between the following modes:

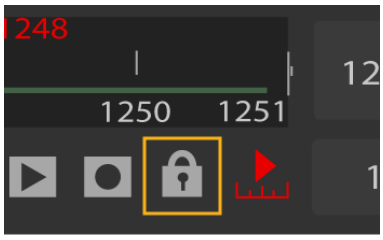
Button	Function
Repeat	Repeatedly plays the sequence in a loop.
Bounce	Repeatedly plays the image back and forth from head to tail.
Stop	Plays once through the section between in and out point and stops at the out point. If these are not marked, then it plays from the beginning to the end of the sequence.
Continue	Plays once from the beginning to the end of the sequence, ignoring any in and out points.

Jumping to a Specific Frame

You can move quickly to a specific frame on the timeline by choosing **Viewer** > **Go to frame** (or by pressing **Alt**+**G**), entering a frame number in the dialog that appears, and clicking **OK**.

Synchronizing Viewer Playback

The 'Lock All Viewer Playback Ranges' button allows you to toggle synchronized playback of Viewer windows. By default, all Viewers are locked - that is, if you cue to a frame in one Viewer, all other Viewers follow suit.



When the lock icon changes from a closed lock to an open lock, that Viewer's playback becomes independent of other Viewers, and not cued to the other Viewers.

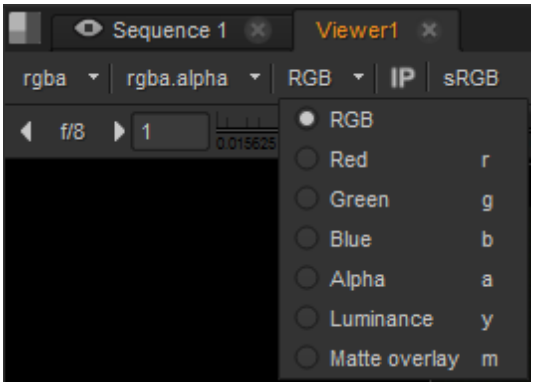
Pausing the Display

The 'Pause Viewer Update' button stops the Viewer from updating and holds the last frame rendered. To reactivate display rendering for all frames, press the button again.

You can click the 'Refresh Viewer' button (or press **U**) to manually update the display while keeping Viewer paused.

Displaying a Single Channel

You can press the **R**, **G**, **B**, and **A** keys on your keyboard to display the red, green, blue, and alpha channels respectively. Or, you can also select a channel from the **RGB** dropdown menu in the top-left corner.



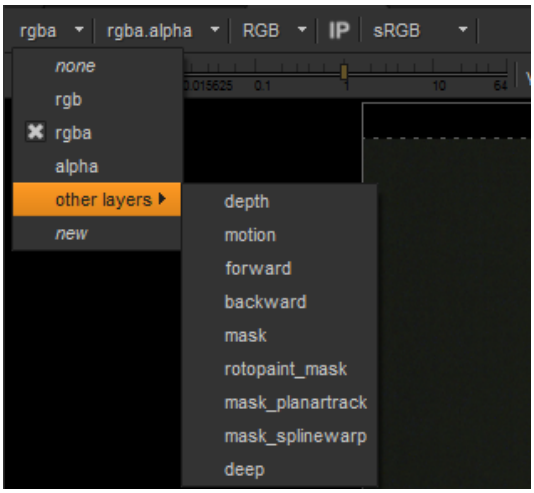
Press one of the channel keys again to toggle back and display all channels.

Tip: If you press **Shift** while selecting the channel, your selection only affects the currently active input of the Viewer node. This way, you can display different channels from the Viewer's different inputs. For example, when keying it can be useful to view the RBG channels from one input and the alpha channel from another, and toggle between the two. To achieve this, do the following:

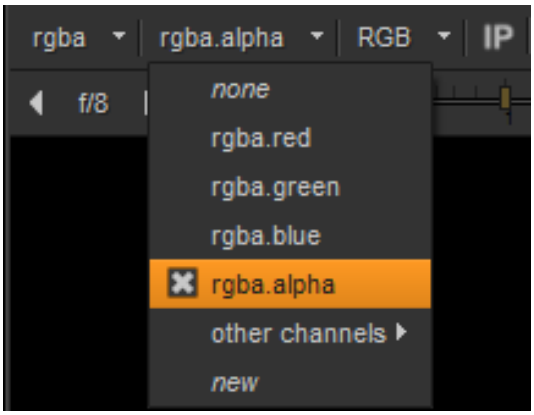
1. Create a Viewer with several inputs.
2. Activate one of the inputs by pressing its number (for example 1) on the Viewer.
3. Press **Shift** and select **RGB** from the channel dropdown menu.
4. Activate another input (for example, input 2) by pressing its number on the Viewer.
5. Press **Shift** and select **A** from the channel dropdown menu.
6. Toggle between the inputs by pressing their numbers or the up and down arrow keys.

Layer and Channel Dropdown Menus

The layer dropdown menu lets you choose a set of color channels to display in the Viewer. By default, this is set to display the rgba layer, but you can choose any layer in the data stream.



The channel dropdown controls which channel appears when you view the “alpha” channel. The default setting displays the alpha channel when you press the **A** key, but you can change this by selecting any channel in the data stream, as shown below.



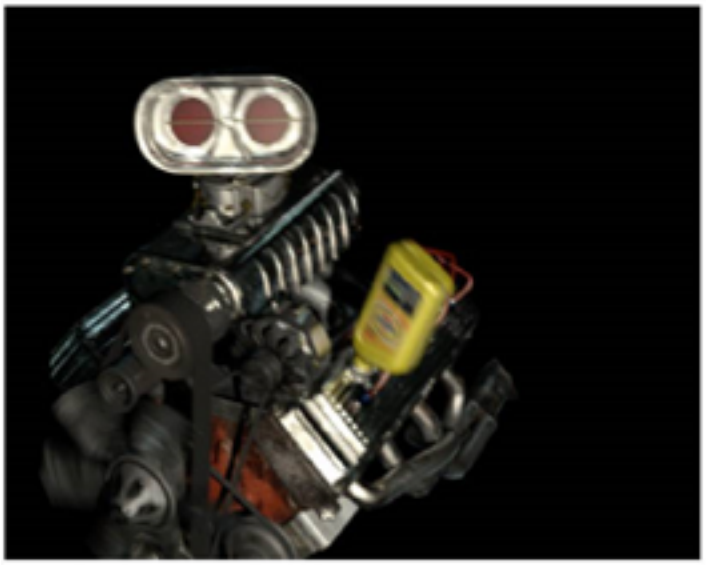
Displaying Alpha Mattes

When you’ve read in an image that has an alpha channel, you can display the alpha channel as a red overlay on top of the image’s red, green, and blue channels.

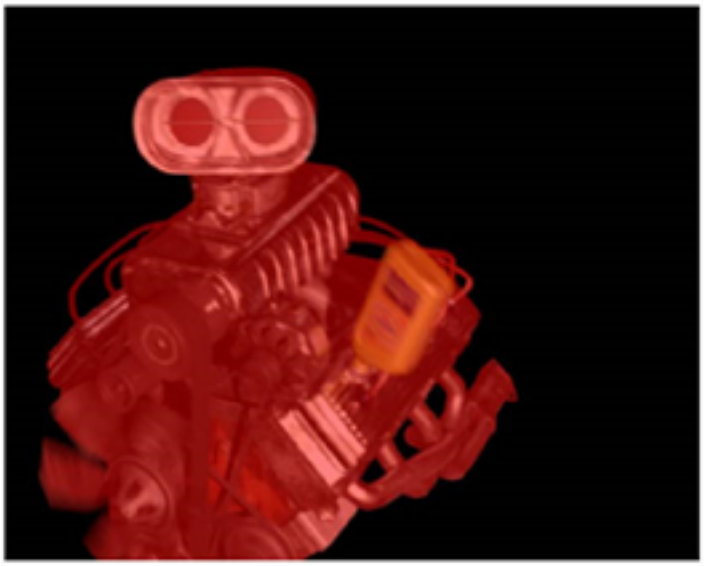
To Display an Image's Alpha Channel on its RGB Channels

1. Select **Image** > **Read** to read in an image.
2. Connect a Viewer node to the Read node.

   By default, Nuke displays the red, green, and blue channels in the Viewer.



3. Click on the Viewer to make sure it's the currently active panel.
4. Press **M**.
   Nuke displays the image's alpha channel as a red overlay on top of the RGB channels.



5. To return to the RGB display, press **M** again.

Viewer Info Bar

The info bar displays information about the image format, bounding box, and pixel underlying the pointer, a sampled pixel or region of pixels. From left to right, the indicator displays the following about the current image and pixel or sample: image resolution and the size of the bounding box, the pixel's x and y position

and its Red, Green, Blue, and Alpha values, and other values depending on the color type you have selected from the color type menu on the right.

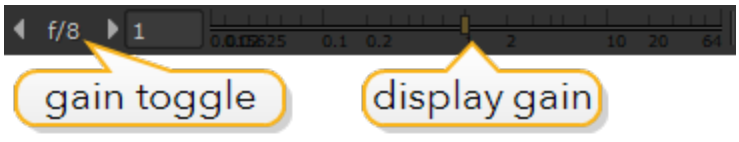
Color Type	Description	Values for RGB 0.1, 0.1, 0.1, 1.0 (Dark Gray)
Spotmeter	Colorspace values measured in f-stops at 1/48th of a second shutter speed.	f/5.9 @48th sec
8-bit	RGB colorspace values.	25, 25, 25, 255
8-bit Hex	The same colorspace values as 8-bit, but displayed in hexadecimal.	1A 1A 1A FF
log	Logarithmic colorspace values.	397, 397, 397
HSVL	The default setting, shows colorspace values as Hue, Saturation, Value, and Level.	0, 0.00, 0.10, 0.10000

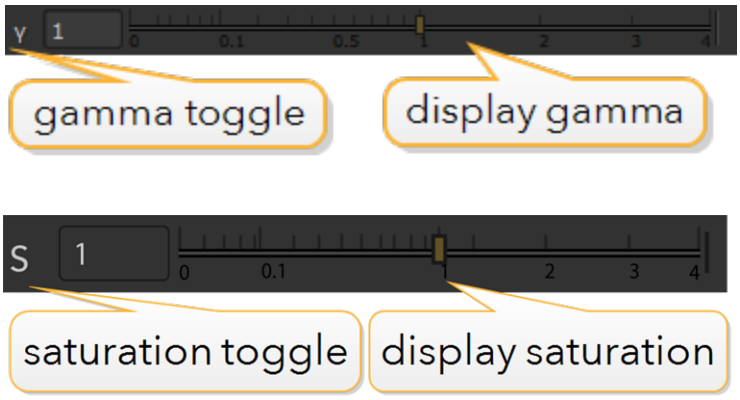
You can sample a single pixel from the Viewer by pressing **Ctrl/Cmd** while clicking, a region from the Viewer by pressing **Ctrl/Cmd**+**Shift** while dragging, a single pixel from the node's input by pressing **Ctrl**/**Cmd**+**Alt** while clicking, and a region from the node's input by pressing **Ctrl**/**Cmd**+**Alt**+**Shift** while dragging.

To cancel a pixel selection, hold **Ctrl**/**Cmd** and right-click in the Viewer.

Adjust Display Gain, Gamma and Saturation

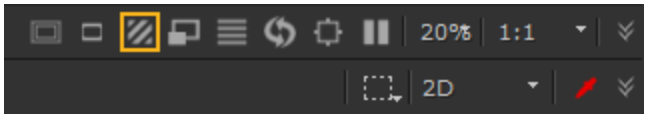
The gain, gamma and saturation sliders let you adjust the displayed image, without affecting your final output. These controls are useful for tasks like spotting holes in mattes. You can boost or reduce gain by entering a multiplier (exposure value), dragging on the slider, or using the F-Stop arrows. Boost or reduce gamma by entering a gamma level or dragging the gamma slider. Saturation correction is applied after the gain and Viewer processes, and any custom input process. Saturation uses Rec709 coefficients to compute luminance, and uses the same color space as that which is set in the Viewer.





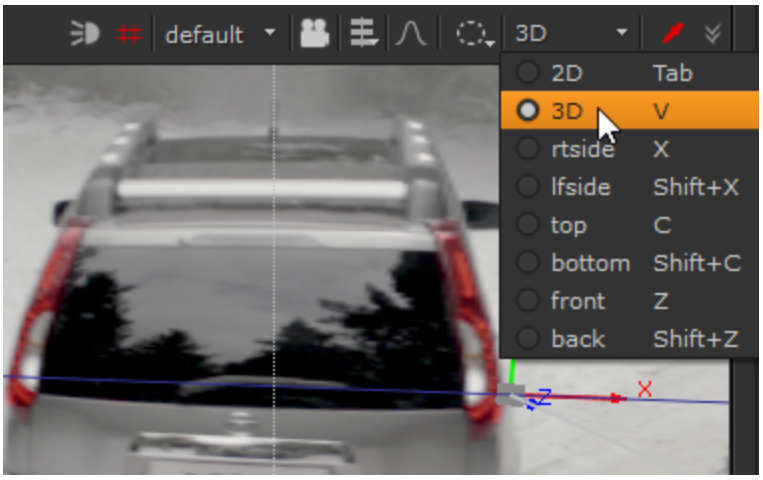
The gain, gamma and saturation toggle buttons let you switch between the default values of 1 (no change) and the last gain and gamma adjustments you made in the Viewer.

Press the 'Clipping Warning' button to apply stripes to all pixels outside the range 0.0 to 1.0.



2D / 3D Toggle and Camera Controls

The 2D / 3D dropdown menu lets you toggle between 2D and 3D display modes in the current Viewer. This menu also lets you choose between different orthographic (non-perspective) views when working in the 3D mode.

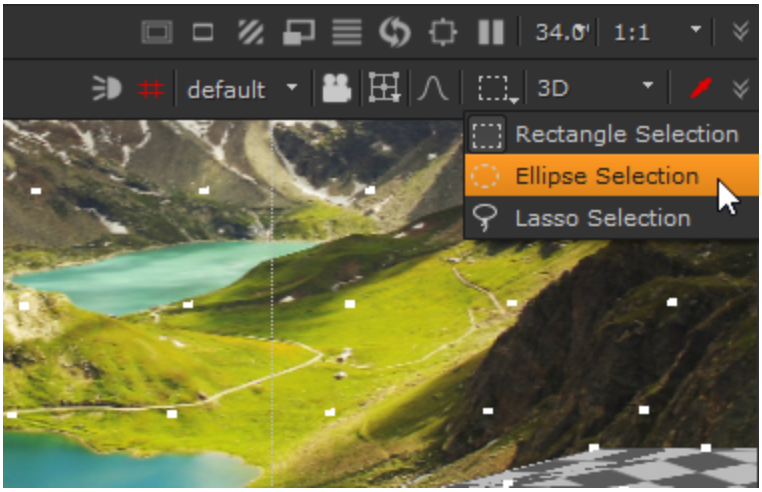


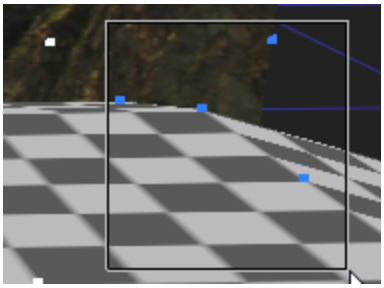
The camera dropdown menu on the right lets you choose which camera to look through when multiple cameras exist in your 3D scene. For more information on these controls, see Classic 3D Compositing.

Viewer Selection Modes

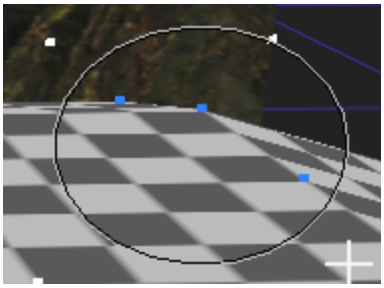
Nuke's Viewer features three selection modes, **Rectangle**, **Ellipse**, and **Lasso**. Selection modes work the same in all Viewer contexts, whether you're in 2D selecting Roto splines or 3D selecting vertices, faces, or objects. The selection mode dropdown is located above the Viewer at the top-right.

The 3D Viewer also includes a soft selection mode, which modifies selections based on a falloff curve to create softer, more organic transitions. See Soft Selection for more information.

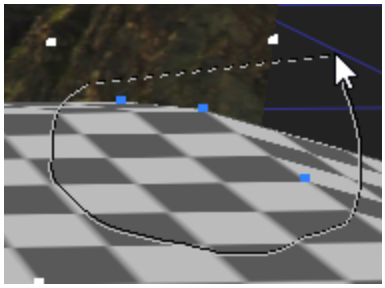




Rectangle



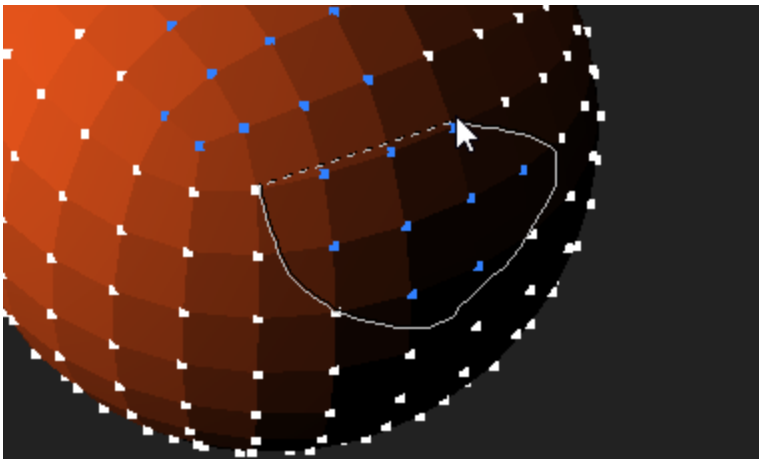
Ellipse



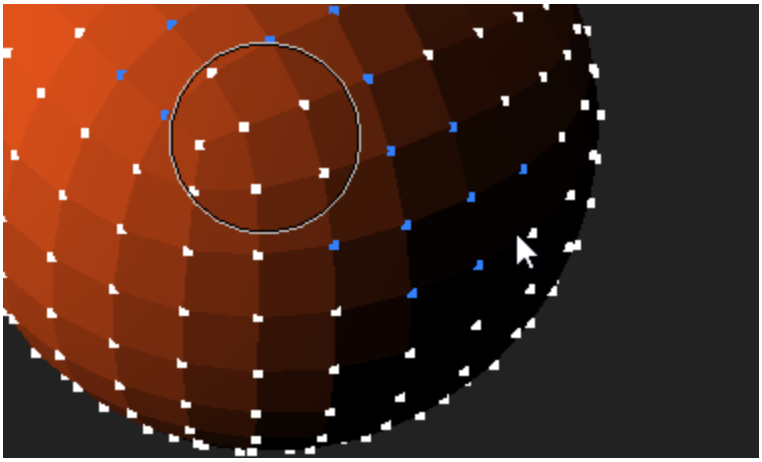
Lasso

Each mode uses the same selection modifiers:

- **Shift** - additive selection. Holding **Shift** and selecting vertices, faces, or objects adds the selection to any existing selection.



- **Alt** + **Shift** - subtractive selection. Holding **Alt + Shift** and selecting vertices, faces, or objects removes the selection from any existing selection.



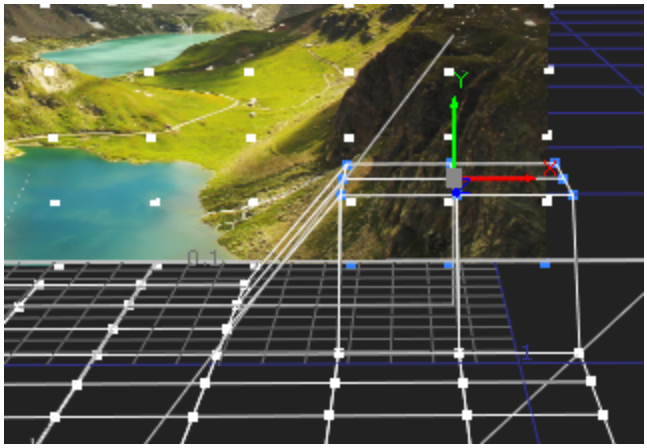
Soft Selection

Soft selection makes it easier to make subtle changes to geometry in the 3D Viewer, creating soft transitions in your geometry modifications using EditGeo and ModelBuilder. See Modifying Object Shapes and Using ModelBuilder for more information.

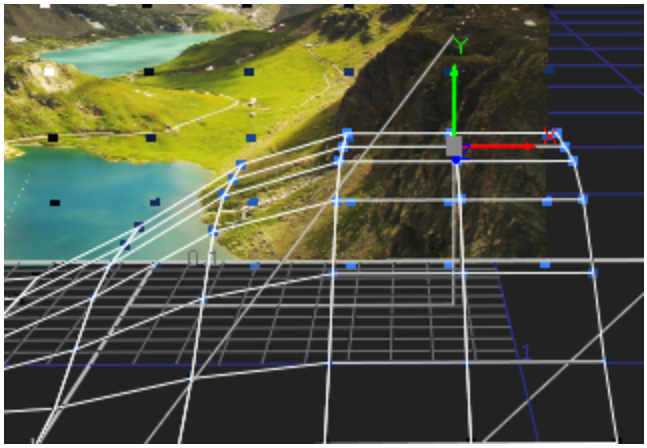
When enabled, selecting vertices or faces in the Viewer adds additional selections according to a **Falloff Radius** and **Falloff Curve** in the Viewer's node **Properties** > **3D** tab under **Soft Selection**. Soft selection includes preset falloff selection curves, such as **Linear** and **Steep In**, and allows you to create custom curves.

To enable soft selection, click  above the 3D Viewer and then select the target vertices or faces. Dragging the transform handle adjusts the selected vertices or faces along with the additional selections described by the falloff.

Tip: You can also soft selection on and off by pressing **N** in the Viewer.



Falloff disabled on a Card



S-shape falloff enabled on a Card

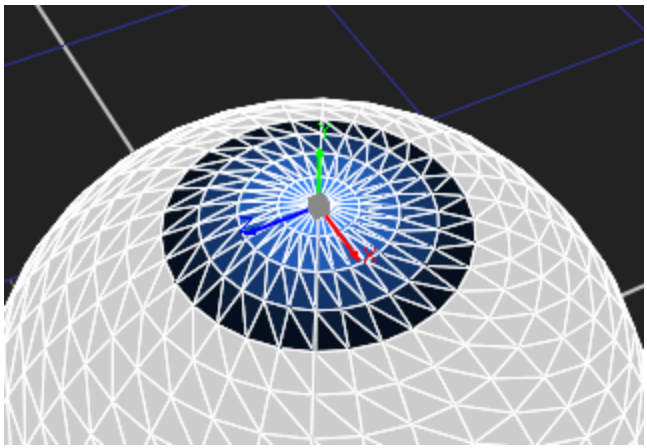
The Viewer **F** and **H** keyboard shortcuts also work with selections:

- Pressing **F** focuses the Viewer differently depending on selection:
  - If you have vertices or faces selected, pressing **F** focuses the Viewer on the selection.
  - With no selections in the Viewer, pressing **F** focuses on the whole scene.
- Pressing **H** focuses the Viewer differently depending on the view mode:
  - If you're using a custom camera, pressing **H** switches the view to the camera view.
  - In any view with the **default** camera, pressing **H** focuses on the whole scene, the same as pressing **F** with no selection.

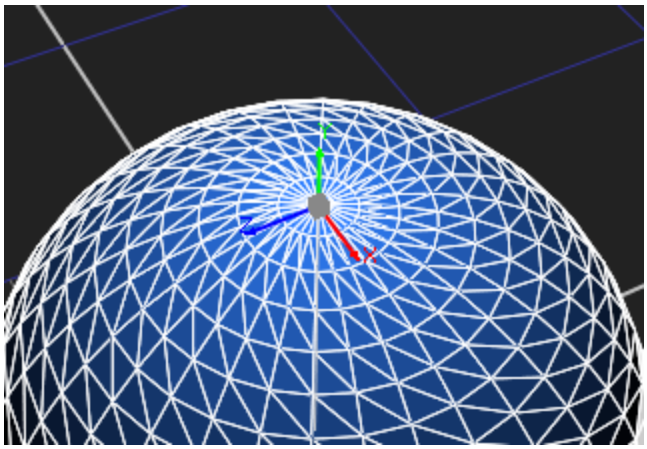
Controlling Selection Falloff

Selection falloff determines how vertices or faces farther away from your selection are affected by changes to the geometry. The greater the **Falloff Radius**, the more you affect remote vertices or faces. You can use one of the built-in **Falloff Curve** presets, such as **Linear** or **Steep In** to modify selection more accurately or create your own curve from scratch.

1. Add your selection to the Viewer using the EditGeo or ModelBuilder nodes.
2. Press **S** in the Viewer to display the Viewer node **Properties** panel.
3. Click the **3D** tab and then open the **Soft Selection** properties.
4. Select the required **Falloff Curve** and then adjust the **Falloff Radius** to update the selection falloff.



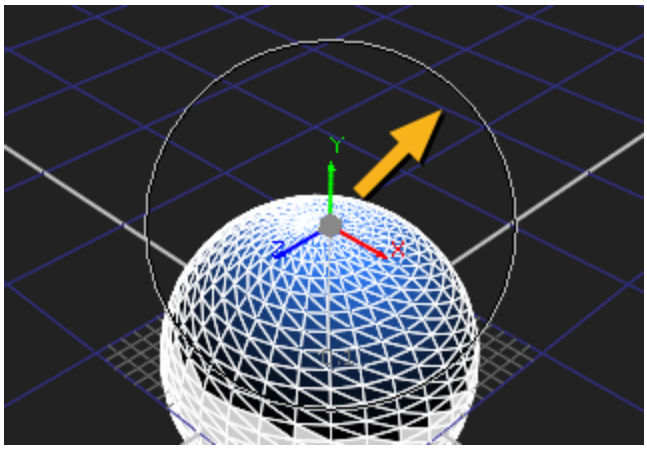
Low radius with a **Linear** curve

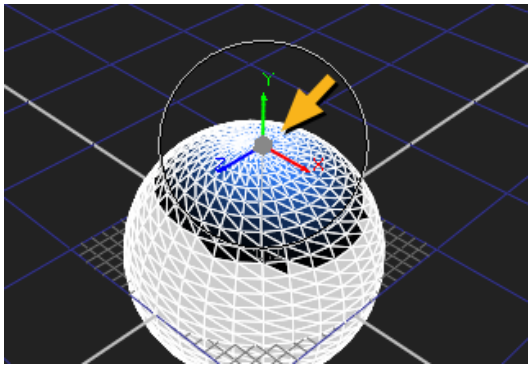


Default radius with a **Linear** curve

OR

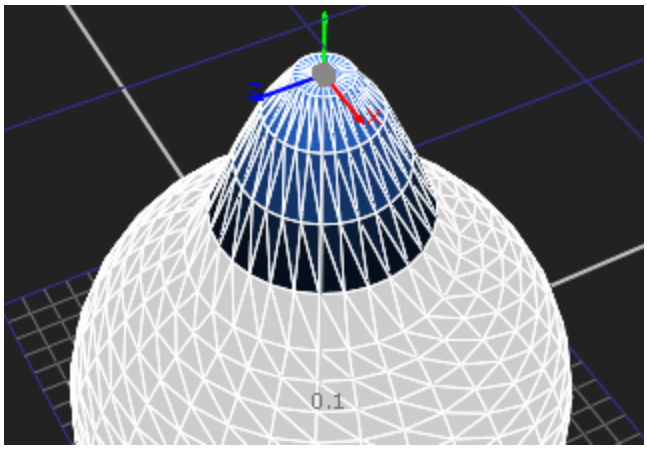
You can hold the **N** key over the Viewer and then left-click and drag or middle-click and drag to control the falloff.



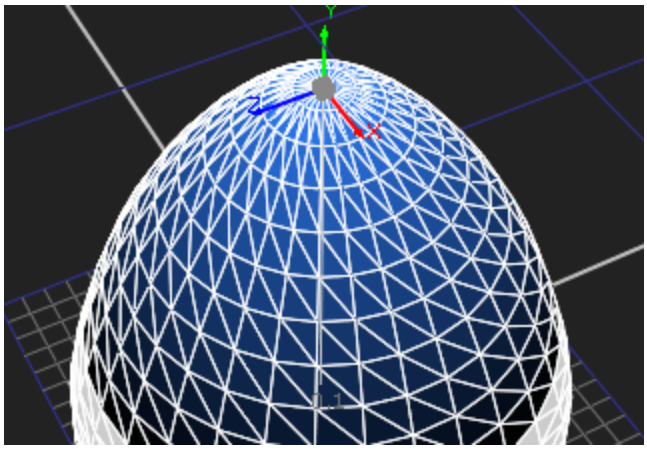


Note: Left-click and drag remembers the extent of the **Falloff Radius** between operations, but middle-click and drag always starts from a **Falloff Radius** of 0.

5. Adjust your geometry as required.



Low radius with a **Linear** curve



Default radius with a **Linear** curve

Controlling Zoom and Resolution

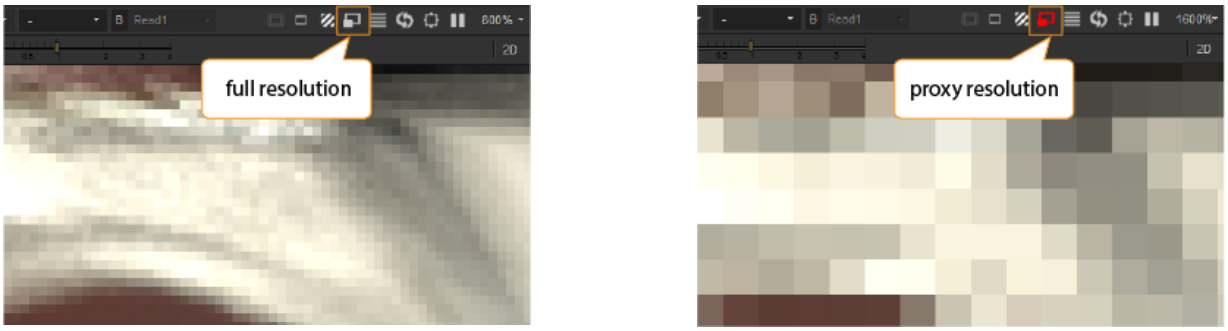
Nuke includes a dedicated set of controls to determine what is rendered in the Viewer and at what resolution, allowing you to work on specific areas of the image at the required quality level while keeping the interface responsive.

Using the Zoom Menu

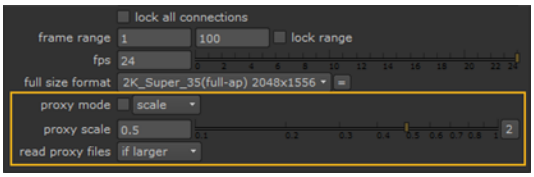
The Zoom dropdown menu lets you select the magnification factor by which the current image is displayed. This menu also shows the keyboard shortcuts to press to quickly switch between the different zoom settings.

Proxy Mode

Nuke can generate low-res proxies for displayed frames as needed. Press **Ctrl/Cmd+P** or click the proxy mode toggle button on the Viewer to activate the proxy display mode.



By default, the proxy scale is set to **0.5**. You can change the proxy scale in the **Projects Settings**, which you can open by selecting **Edit** > **Project Settings** (or pressing **S**).



Proxy display resolution defined on the Project Settings properties panel.

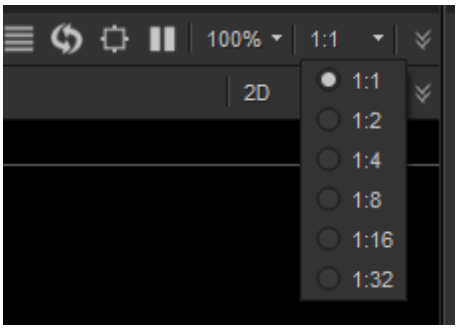
You can also read in rendered proxies using the Read nodes' controls. The proxy file does not need to match the proxy resolution in use. Depending on your settings, either the full-res or proxy file is scaled to the required proxy size. For more information, see Read Nodes and Proxy Files.

Lowering the Display Resolution of Individual Viewers

Viewers also have a dropdown menu that allows you to easily switch to lower display resolutions, regardless of whether you have activated proxy mode or not. Using this multiplier setting, you can, for example, change the display resolution of an individual Viewer to 50% of the current (be it full-size or proxy) resolution. This is useful if you want to have Nuke display your images more quickly without having to touch the project settings. It also comes in handy if you have just a few very large plates in your script, as you can choose to use lower resolutions when viewing just these plates.

To Lower the Display Resolution of Individual Viewers:

From the Viewer's down-rez dropdown menu, choose the factor by which you want to lower the display resolution.



For example, if you have a 4K plate and are using a proxy scale of 1:2, your plate is still 2K even in the proxy mode. Setting the down-rez factor to **1:2** in the Viewer scales the plate down further to 50% of the proxy resolution, that is to 1K. This gives you much faster (but less accurate) feedback.

Pixel Aspect Ratio

The pixel aspect ratio determines whether your images are displayed using square or rectangular pixels. By default, the Viewer uses the pixel aspect ratio defined in your project settings. To see the current setting, select **Edit** > **Project Settings** (or press **S**).

For example, a pixel aspect ratio of 2 accurately displays anamorphic footage the way it is projected, as shown in the image:



The Viewer uses the pixel aspect ratio defined for the script.

If you want to ignore the pixel aspect ratio, you can toggle it by pressing **Ctrl**/**Cmd**+**Shift**+**P** over the Viewer window.

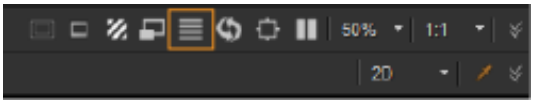


Press **Ctrl/Cmd**+**Shift**+**P** over the Viewer window to ignore the pixel aspect ratio.

Full-frame processing

By default, when you display or play through a sequence in the Viewer, Nuke only calculates and caches the scanlines for the visible area. It's not caching the full frame, but a subset of the scanlines you're viewing. For example, if you have set the zoom level to be **÷4**, Nuke is only caching 1/4 of the scanlines that make up the frame. In a lot of cases, this allows for faster playback and may be what you want.

However, if you play through a sequence and then pan around or change the zoom level, Nuke has to calculate the scanlines it didn't calculate before. This may not be what you want if you are performing tasks that require you to constantly pan around or zoom in and out of the plate (such as paint and roto). If this is the case, you can click on the **full frame processing** button in the controls at the top of the Viewer to force Nuke to process every scanline in the image. Compared to the default mode, this may take slightly longer initially and requires more space in the Viewer cache, but once Nuke has cached the frames you require, you can pan around and change the zoom level without breaking the cache or affecting playback.

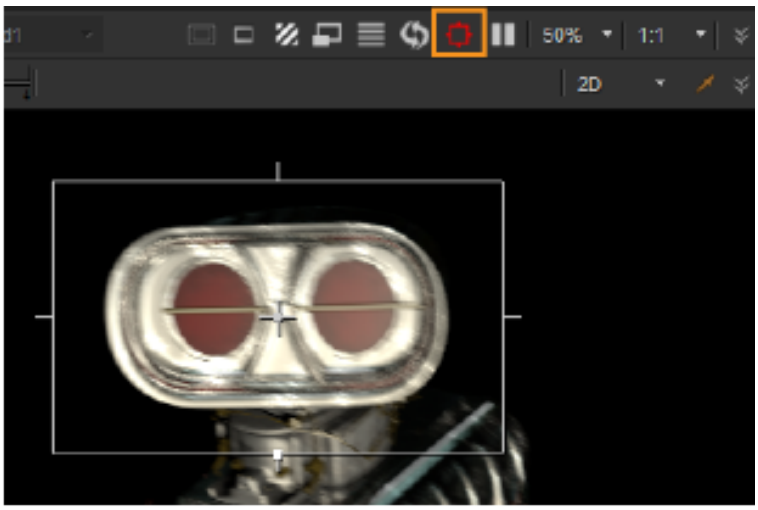


Region of Interest (ROI)

The **ROI** button lets you enable rendering only through a region of interest - a portion of the image you explicitly select. This is useful for quickly viewing render results in a process-heavy script.

To Define a Region of Interest:

1. Click on the **ROI** button in the Viewer controls. The ROI overlay appears.



2. Drag to resize and move the ROI overlay as necessary.

To Free-Draw a Region of Interest

1. Over the Viewer, press **Alt**+**W** once (do not hold the keys down). The ROI button turns red, but the ROI overlay does not appear. This allows you to freely draw your own ROI rather than adjust the default overlay.
2. Drag a marquee to draw the region of interest where you need it.

To Clear a Region of Interest:

1. After you've set a region of interest, you can clear it by pressing **Alt**+**W** over the Viewer. You can then drag a new marquee to define a new region of interest.
2. To turn off the feature and update the whole Viewer with the recent changes, click the **ROI** button again (or press **Shift**+**W**).

Viewer Overlays and Input Processes

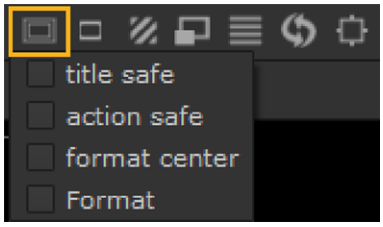
Nuke's Viewer overlays and render modes help you to position elements correctly and compare shots in the Node Graph. You can also use **Input Process** and **Viewer Process** to modify the image from the viewed node before it is displayed on your monitor.

Guides and Masks

The Viewer guides and masks assist with placing effects and text within the current format. For example, text placed with the **Title Safe** guide is visible to the audience. Use the **Guideline** and **Mask** dropdown menus to select the required overlays.

Note: Guides and masks are not applied at render, they are simply Viewer overlays.

Guidelines are intended to highlight areas of the current format that won't appear in the final render. By default, no guidelines are selected. You can select any of the following guideline options:



• **title safe** - any text intended for the audience should reside within this zone.



• **action safe** - any visual elements intended for the audience should reside within this zone.



• **format center** - overlays a crosshair in the center of the format area.



• **format** - any formatting changes must be applied to the area outlined in red.



Tip: The above guideline options also exist in the Viewer Settings. Press **S** on the Viewer to display its settings and adjust the **safe zone** and **format center** controls.

Masks can be used to simulate a particular format, for example, **4:3** or **16:9**. You can also choose the mask overlay type.

• **none** - no masking is applied to the Viewer. This is the default state.



• **lines** - any mask applied is highlighted using a pair of lines in the Viewer.



• **half** - any mask applied is highlighted using semi-transparent shading.



• **full** - any mask applied is highlighted using black shading.



• **blanking ratio** - select the masking ratio applied to the Viewer, for example, **4:3** or**16:9**.

Tip: The above mask options also exist in the Viewer Settings. Press **S** on the Viewer to display its settings and adjust the **mask region outside ratio** and **mask mode** controls.

Adding Custom Guides and Masks

You can add custom guides and masks to the Viewer dropdowns by creating a file called **custom_guides.py** and placing it in your **.nuke** folder.

Tip: For information on locating your **.nuke** folder by platform, see Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts.

The guides and masks that ship with the application are kept in the Nuke bundle, here:
`<installation_directory>/pythonextensions/site-packages/custom_guides.py`

Copy the **custom_guides.py** file to your **.nuke** folder and then add the guides and masks you require to the existing Python code inside the **.py** to append them to the Viewer dropdowns. For example:
`guides.Guide("myGuide", 0.75, 1, 0.8, 0.3, guides.kGuideMasked)`

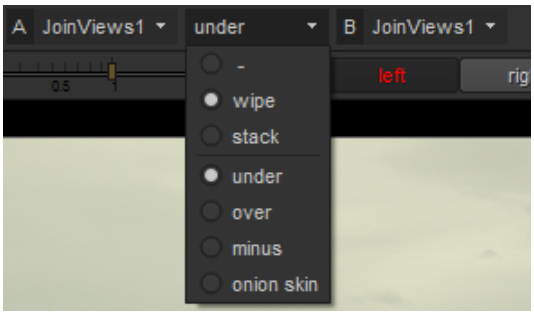
OR
`guides.MaskGuide("5:3",5.0/3.0)`

Using the Viewer Composite Display Modes

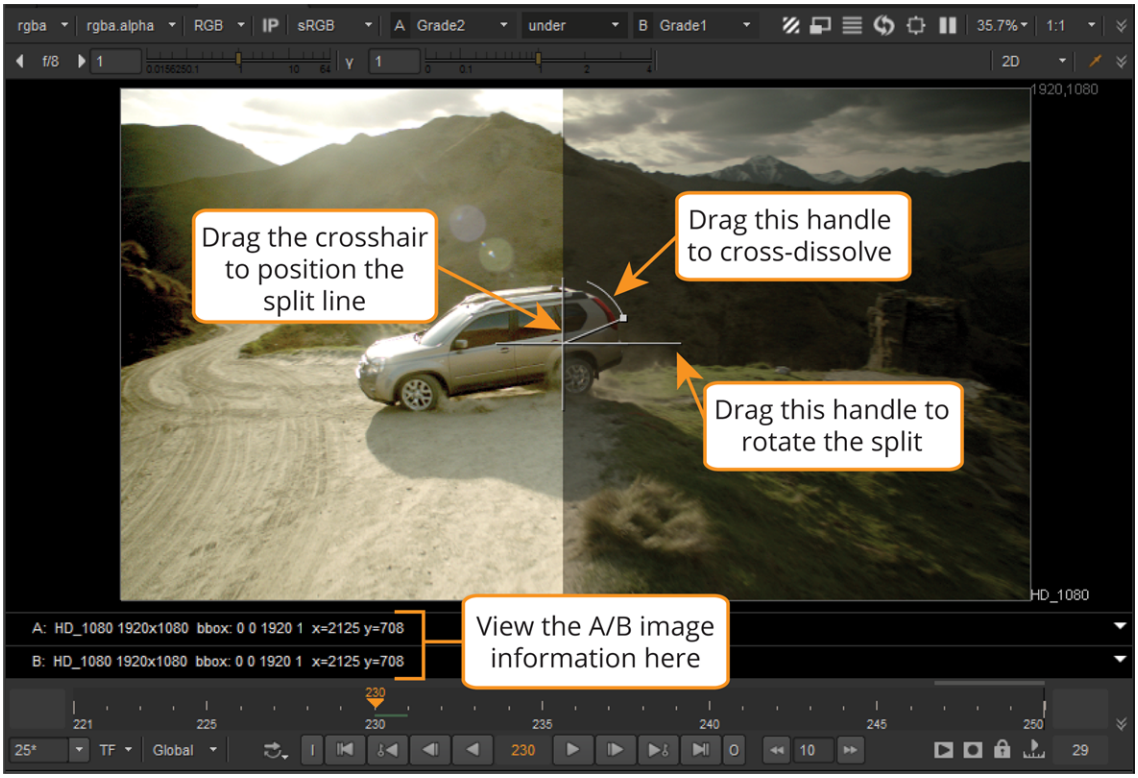
The wipe control provides an option for displaying a split-screen of two images, which can help you compare before and after versions for color correction, filtering, and other image manipulation. This control also includes display compositing options to overlay different images.

To Display a Comparison Wipe:

1. Select a node in your script and press **1** to display its output in the Viewer.
2. Select the node you want to compare and press **2**.
   The **2** keystroke connects the image to the Viewer (assigning the next available connection, number 2).
3. From the **A** and **B** dropdown menus on top of the Viewer, select the images you want to compare. The menus display a list of nodes most recently connected to the Viewer.
4. From the Viewer composite dropdown menu in the middle, select **wipe**.



The two images are displayed split-screen in the Viewer. You can view their details in the A and B information bars at the bottom of the Viewer.



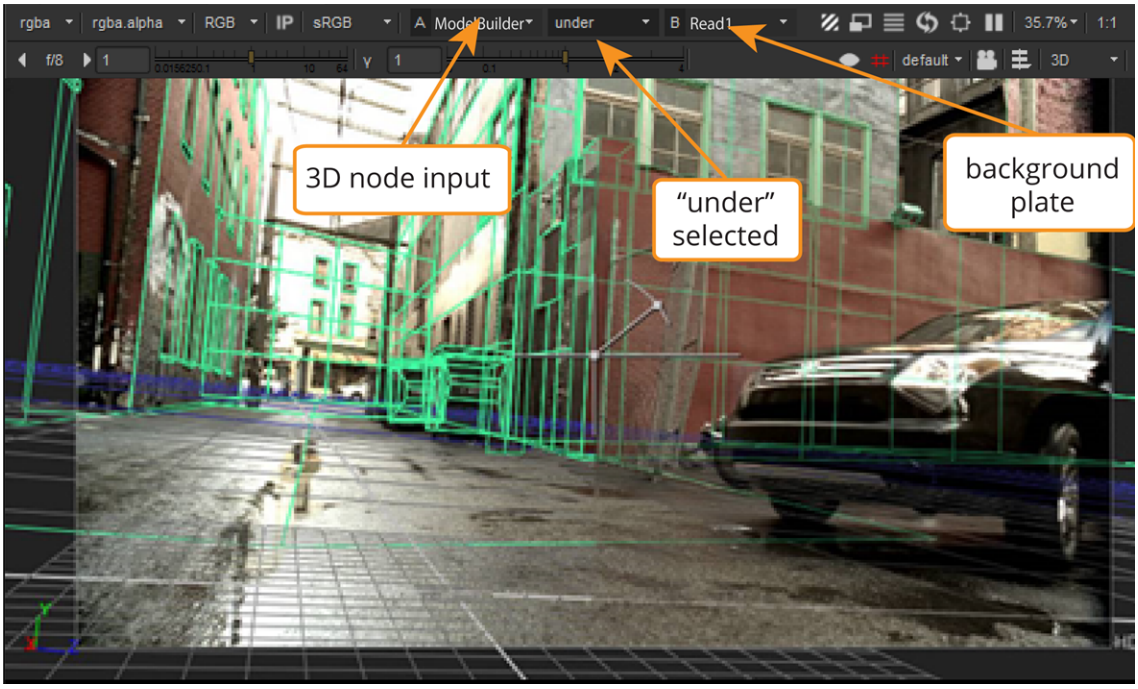
5. Drag the handles of the crosshair to adjust the wipe:
   - Drag the crosshair center to change its position.
   - Drag the long handle (on the right) to rotate the wipe.
   - Drag the “arc” handle to cross-dissolve the second image.
6. When finished with the split-screen, select none (**-**) from the Viewer composite dropdown menu.

Tip: If you press **Shift** while selecting a channel, your selection only affects the currently active input of the Viewer node. This way, you can display different channels from the Viewer's different inputs. For example, when keying it can be useful to view the RBG channels from one input and the alpha channel from another, and toggle between the two.

The display composite options - **over**, **under**, and **minus** - can also be selected to overlay two images. When the two images are 2D, this allows you to create a quick comp.

When one image is 2D and the other is a 3D node, you can use **under** to line up the wireframe preview with the 2D reference, and see how the 3D matches prior to a full render.

One example of this is when you want to preview a wireframe 3D scene with a background plate that you are trying to match, as shown below. For more information, see the Classic 3D Compositing chapter.



Special thanks to belowtheradar TV for use of the above footage.

Input Process and Viewer Process Controls

Input Process and **Viewer Process** operations can be used to modify the image from the viewed node before it is displayed on your monitor. Both only affect the Viewer in which they are activated and do not affect your rendered output. Input Process is a legacy system which uses a node instantiated in the Node Graph to process the image. This is handy for script-specific, temporary, or experimental use, but can be error prone due to the node accidentally being deleted or changed and is limited to a single node. The

Viewer Process system allows a gizmo (or compiled node) to be registered from the Python programming language at start-up. The registered item appears in a dropdown menu in the Viewer and the node is instantiated internally within the Viewer when the item is selected so there is no danger of accidental deletion or modification. This also enables multiple Viewer Processes to be registered at different points of start-up (as Nuke works through the NUKE_PATH **menu.py** files).

The Viewer settings contain an option for the Input Process to be applied before or after the Viewer Process, so the two may be used in conjunction, for instance, with the Input Process applying a projection mask after the Viewer Process applies a film look profile. While you could combine the two into a single Viewer Process node, it can be advantageous to keep operations separated. Having both the Viewer Process and Input Process available provides a great deal of flexibility.

You can create an Input Process by creating a node in the Node Graph and naming it as an Input Process using Nuke's **Edit** menu. Once an Input Process has been named, the IP button appears in the Viewer controls. When the IP button is activated, any image you view is passed through the Input Process.

Unlike Input Processes, Viewer Processes are registered using Python. They can be session independent and always appear in the Viewer's Viewer Process dropdown menu. There are two predefined Viewer Processes, **sRGB** and **rec709**, but you can also build and add your own. When a Viewer Process is selected from the Viewer Process dropdown menu, any image you view is passed through that Viewer Process.

Whenever possible, the Input Process and Viewer Process are executed on the GPU. 1D LUT and 3D LUT (Vectorfield) have GPU implementations, so the built-in Viewer Processes run on the GPU (unless **gl buffer depth** has been set to **byte** in the Viewer settings, in which case all processing is done on the CPU). To get the GPU'd versions of the nodes for use in a custom Viewer Process gizmo, press **x** over the Node Graph, enter **ViewerGain**, **ViewerGamma**, or **ViewerClipTest** in the command entry window, and press **Return**.

The following table lists the differences between an Input Process and a Viewer Process.

Input Process	Viewer Process
Set by selecting the node in the Node Graph and choosing **Edit** > **Node** > **Use as Input Process**.	Registered using Python.
Activated using the IP button in the Viewer controls.	Activated using the Viewer Process dropdown menu in the Viewer controls.
Requires that the node exists in the Node Graph. Can quickly and easily be modified by artists. Can also be accidentally deleted, disabling the effect.	Is defined in a text file called **menu.py** that is run at start-up. Accessible for artists, but not likely to be accidentally modified or deleted.
Script dependent. Unless your Input Process node is saved in the template.nk file that is loaded at start-	Session independent. The Viewer Processes registered in **menu.py** are always available in each

Input Process	Viewer Process
up, the Input Process is lost when you restart Nuke.	new session of Nuke.
There can only be one Input Process at a time. Setting a new Input Process overrides any previously used Input Process.	There can be an unlimited number of Viewer Processes available in the Viewer Process dropdown menu. For example, it is possible to register Viewer Processes in any **menu.py** file at start-up, so Viewer Processes can be added at any directory in your NUKE_PATH.
Useful for temporary or non-critical viewing options that you want in the current shot for convenience, or for testing Viewer Processes before registering them. Can also be used for other things, such as field charts or masks that may be switched on or off and changed around in the shot.	Useful for viewing options that you often need or that should not be modified by artists on a shot-by-shot basis.



Note: Note that Input Processes and Viewer Processes are part of a built-in, fixed pipeline of nodes that are applied to images before they are displayed in the Viewer. This pipeline is either:
gain > Input Process > Viewer Process > gamma > dither > channels > cliptest (if **input process order** has been set to **before viewer process** in the Viewer settings)
OR
gain > Viewer Process > Input Process > gamma > dither > channels > cliptest (if **input process order** has been set to **after viewer process** in the Viewer settings).
However, depending on what the Input Process and Viewer Process are doing, the order in the built-in pipeline may not be the correct order. Therefore, if your Input Process or Viewer Process have controls that also exist for the Viewer, such as float controls named **gain**, **gamma**, or **cliptest**, then the Viewer drives them from the corresponding Viewer controls and does not do that image processing itself. This allows you to implement these controls in your Input Process or Viewer Process node/gizmo using whatever nodes and order you want. If your Input Process and Viewer Process do not have these controls, then the Viewer applies the effects in its normal way according to the built-in pipeline.
In the built-in pipeline, dither is applied to diffuse round-off errors in conversion of floating point data to the actual display bit depth. Although the cliptest is drawn at the end, it is computed on the image as input to the Viewer.

Note: By default, the predefined Viewer Processes, **sRGB** and **rec709**, affect all channels. However, if you want them to only affect the red, green, and blue channels, you can activate **apply LUT to color channels only** in the individual Viewer Settings or on the **Viewers** tab of the Preferences.

Input Process Controls

To activate or deactivate the effect of an Input Process, click the IP button in the Viewer controls. Note that the IP button only appears if the **input process** field in the Viewer settings is not empty. The button is also only enabled when a node in the Node Graph is set as an Input Process.



To open the Viewer settings, press **S** on the Viewer, or select **Viewer Settings** from the Viewer's right-click menu. By default, **input process** is set to **VIEWER_INPUT**. If a node called VIEWER_INPUT exists in the Node Graph, it is automatically used as the input process for the Viewer. This ensures backwards compatibility with pre-5.2 scripts.

However, the Input Process node does not have to be named VIEWER_INPUT. You can use any node as an Input Process. Do the following:

1. Select the node in the Node Graph and choose **Edit** > **Node** > **Use as Input Process**.

   Alternatively, you can press **S** on the Viewer to open the Viewer settings and enter the name of the node in the **input process** field.
2. In the Viewer settings, you can also define whether the Input Process is applied before or after the Viewer Process currently in use. To do so, set **input process order** either to **before viewer process** or **after viewer process**.

The Input Process node should not be connected to other nodes in the Node Graph. If you attempt to connect it, an error is displayed in the Viewer. If you delete the Input Process node from the Node Graph, the effect of the Input Process is disabled.

Viewer Process Controls

To activate a Viewer Process, select it from the Viewer Process dropdown menu in the top right corner of the Viewer. Any images you now view using this Viewer are passed through the selected Viewer Process.

Nuke includes the following predefined Viewer Process gizmos: **sRGB**, **rec709**, and **rec1886**. By default, **sRGB** is used because it is a best-guess default for a typical computer monitor.

In addition to using the predefined Viewer Processes, you can also add your own by registering a node or gizmo as a Viewer Process. You can register as many Viewer Processes with custom Viewer LUTs as you like. For more information on creating and registering custom Viewer Processes, see Creating Custom Viewer Processes.

All available Viewer Processes (both predefined and custom ones) appear in the Viewer Process dropdown menu in the Viewer controls. To disable the use of a Viewer Process, select **None** from the Viewer Process dropdown menu.

To open the properties panel of the currently active Viewer Process, select **show panel** from the Viewer Process dropdown menu.

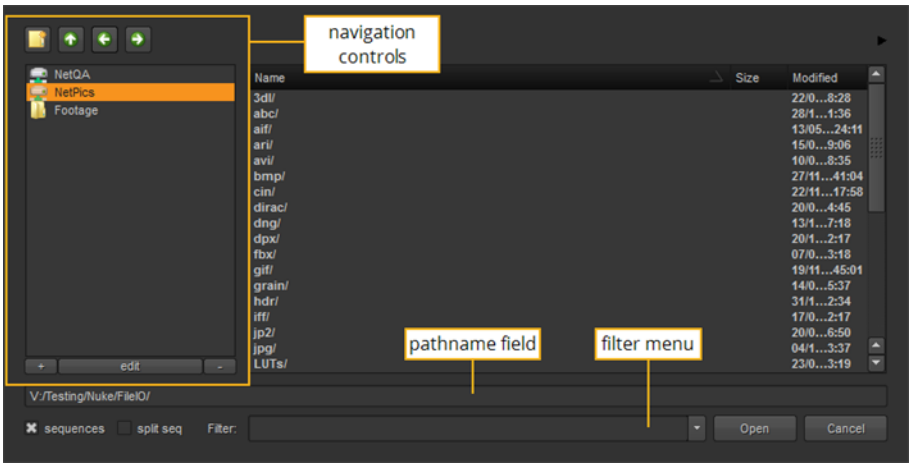
Note that if the control you want to adjust has the same name as any of the Viewer controls (for example, **gain** or **gamma**), you should adjust the control on the Viewer. This drives the control in the following:

- the Input Process' controls if an Input Process is in use
- in the Viewer Process' controls if no Input Process is in use.

Tip: If you want to render out a file with the Viewer Process effect baked in, you can select **Edit** > **Node** > **Copy Viewer Process to Node Graph** to create an instance of the Viewer Process node in the Node Graph.

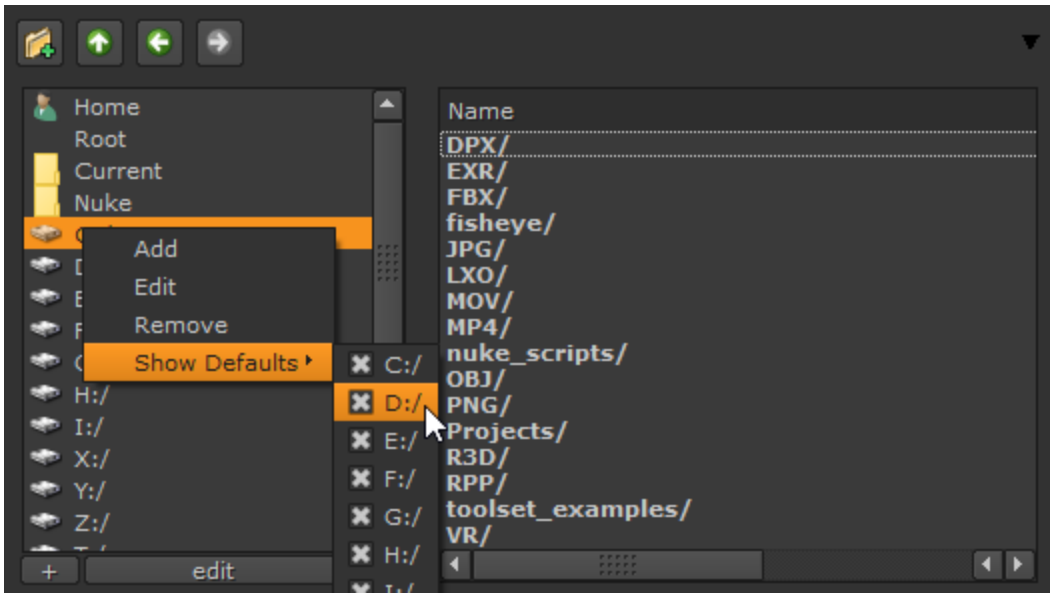
Using the File Browser

Whenever you load or save files in Nuke, you'll see a browser similar to the one shown in the image below. The directory navigation buttons let you create or access the directory from which you wish to read or write data.



The navigation controls let you move through the directory structure, bookmark favorite directories, and create new directory folders.

Windows only: You can show/hide the drives that Windows auto creates by right-clicking the required drive, selecting **Show Defaults**, and checking or unchecking the drive.



To Use the Navigation Controls

- Click the **Create New Directory** button to create a new directory at your current position in the file hierarchy.
- Click **Up one directory** to go up one directory closer to the root.
- Click **Previous directory** to go back one directory.

- Click **Next directory**  to go forward one directory.
- Click **Home** to access the directory defined as your local working directory.
- Click **Root** to ascend to the very top of your local drive or server's file hierarchy.
- Click **Nuke** to access the directory you (or your system administrator) defined as your network working directory.
- Click the **+** button to add a directory bookmark.
- Click the **edit** button to edit the name or path name to a bookmark.
- Click the **-** button to remove a directory bookmark.

Path Name Field

The path name field displays the current directory path, lets you navigate to a new path, and also enter a file name for scripts and rendered images.



To Use the Path Name Field:

1. To navigate to a directory, type the path name in the field.
2. To enter a script name, browse to a directory path and enter the file name after the displayed path.

To limit the file list to specific file types, use the **filter** dropdown menu and **Sequences** checkbox.

To Use the Filter Dropdown Menu and Sequences Checkbox

- Select ***.nk** to display only Nuke script files.
- Select ***** to display all files (except hidden files), regardless of whether they're associated with Nuke.
- Select .* * to display all files, including hidden files.
- Select ***/** to display directory names, but not their contents.
- Check **Sequences** to display image sequences as single titles, as in fgelement.####.cin 1-50 rather than fgelement.0001.cin, fgelement.0002.cin, fgelement. 0003.cin, and so on.

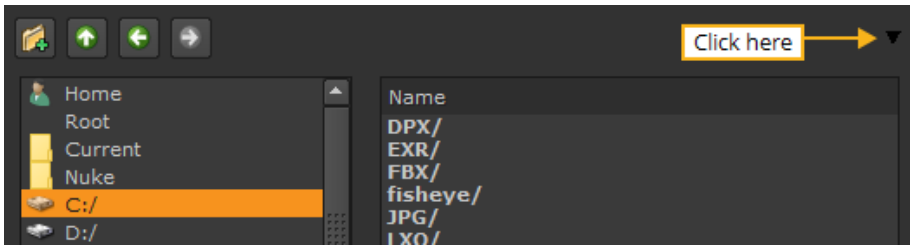
Note: File sequences with no file extension (for example, fgelement.0001, fgelement.0002, fgelement.0003, and so on) are not displayed as single titles the first time you view the directory in the File Browser. However, they are displayed as single titles once you have navigated to another directory and back again.

Note: By default, Nuke may not be able to display custom file extensions (for example, .cext) as single titles. To fix this, you can register your custom file extension as a sequence type using Python:
1. Create a file called **init.py** in your plug-in path directory if one doesn't already exist. For more information on plug-in path directories, see Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts.
2. Open the **init.py** file in a text editor and add an entry in the following format (replacing **cext** with your custom file extension):
nuke.addSequenceFileExtension("cext")

• You can also split incomplete sequences into separate Read nodes using the **split seq** checkbox.

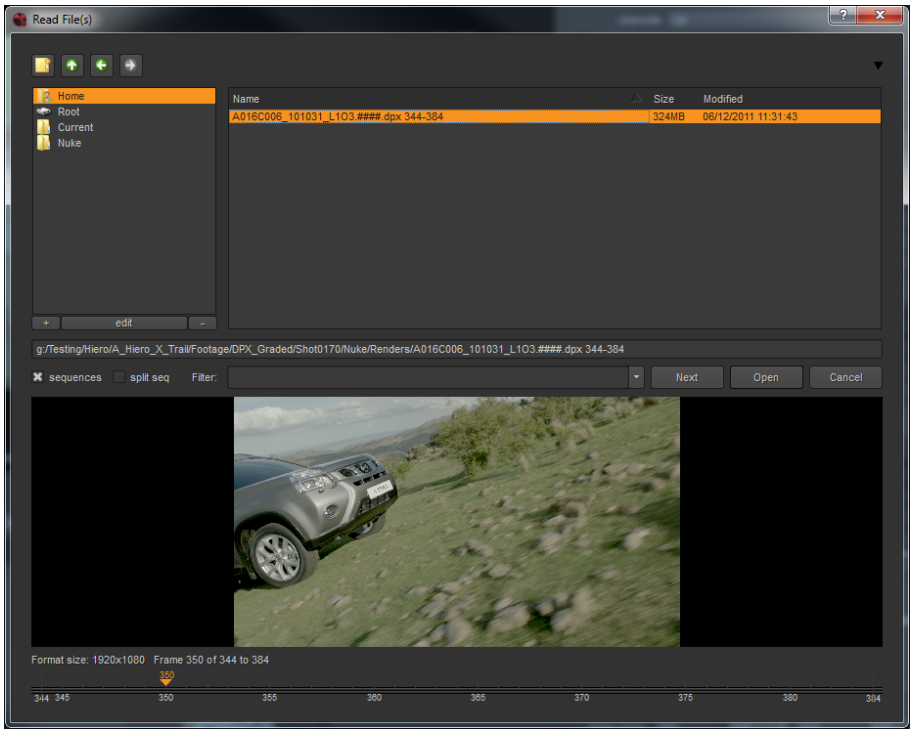
To Preview Files in the File Browser

1. Click the black arrow in the top right corner of the file browser.



The file browser expands to include a small viewer.

2. Select the file you want to preview. Nuke displays the file in the file browser.



To Select Multiple Files with the File Browser

1. Browse to the folder where the files are located.
2. **Ctrl**+click on all the files you want to open to select them (Mac users **Cmd**+click).
3. You can open files from multiple directories by clicking **Next** and browsing to the next file location.

   As you browse around, files that you have previously selected appear highlighted in the browser.
4. Click **Open**.

   Nuke opens all the files you selected.

To Open Incomplete Sequences in Separate Read Nodes

1. Locate an incomplete sequence.
2. Check **split seq** and select all the resultant files.
3. Click **Open** to open the sequence in separate Read nodes.

Undoing and Redoing

Nuke generally gives you an undo history that extends back to the first action of the application's current session.

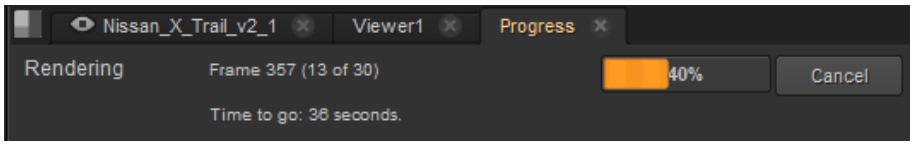
- To undo an action in the workspace, select **Edit** > **Undo** (or press **Ctrl/Cmd**+**Z**). Repeat as necessary.
- To redo an action in the workspace, select **Edit** > **Redo** (or press **Ctrl/Cmd**+**Y**). Repeat as necessary.
- To undo a change in a properties panel, click the **Undo** arrow button in the properties panel.
- To redo a change in a properties panel, click the **Redo** arrow button in the properties panel.
- To undo all changes made after the properties panel was opened, click the **Revert** button.

  OR

  Right-click on the properties panel and select **Revert knobs**.
- To set all controls back to their default values, right-click on the properties panel and select **Set knobs to default**.

Progress Bars

Nuke displays a progress bar for each active task it performs. By default, progress bars appear in a pop-up dialog, but you can also display them in the Progress panel. To do so, click on a content menu button and select **Windows > Progress**. This opens a Progress panel. The next time you get a progress bar, it appears in the panel. If you delete the panel, progress bars appear in a pop-up dialog again.



If you want to have the Progress panel appear in the same position in the future, you can save it as part of a workspace. For more information, see Customizing Workspaces.

Handling Errors

Sometimes things may not go as you planned and you might face an error message in Nuke. When this happens, an error alert displays in the Viewer and on the node that has a problem in the Node Graph.

You can choose to view the error message itself in the **Error Console** tab next to the Properties pane. If you can't see the **Error Console**, click the content menu  button and select **Windows > Error Console** to display it. If you have an error in a Read or a Write node, or you are missing a plug-in, the error message also displays in a pop-up window.

If you see an error alert on your node or in the Viewer, you can click the **Error Console** tab to open it and view the error message.

In the **Error Console** error list, you can double-click on a message and, if possible, Nuke takes you to the control panel of the node that's in error. This isn't always possible because of the nature of the error. You can also click the **clear output**  button on the **Error Console** to clear all error messages on the tab.



Note: If you have a node in your Node Graph that is producing an error, but it's not connected to the node tree, Nuke won't show a pop-up error message for the node and you can still view the resulting image in the Viewer, if you have one connected. This enables working without having to stop to close error messages in the event that you have erroring nodes in the script that aren't connected to your node tree.

Using Nuke Studio's Timeline Environment

This section is designed to help you learn how to use Nuke Studio's Timeline environment, including ingesting media and using the Timeline Viewer.

Achieving Real-time Playback

The following is a list of recommended hardware configurations for Windows and Linux that Foundry have certified for real-time playback in the Nuke Studio and Hiero timeline Viewer (see the note below). Nuke's compositing Viewer is not designed for real-time playback. Please note that playback may also work on other machine configurations, but those listed below have been tested.



Note: For realtime playback, clips and sequences should be 16-bit RGBA DPX files, localized to fast storage. See Localizing Media for more information.

In the case of complex or multi-format timelines, files must be cached to the fast storage using the Timeline Disk Caching. See Caching Frames in the Disk Cache for more information.

Windows and Linux Certified Hardware

Nuke Studio is currently tested on the HP Z workstation series. The following are the tested components for an HP Z840.

Processor	2 x Intel Xeon E5-2687Wv3 CPU
RAM	128 GB DDR4-2133 (8x16 GB) 2 CPU Registered RAM
Internal system drives	1 x HP Z Turbo Drive G2 256 GB PCIe
GPU	NVIDIA Quadro M6000 12 GB
Video monitor output	AJA Kona 4K or BMD DeckLink 4K Extreme 12G

Real-time playback is currently tested with the following internal storage accessories:

Drives	8 x Samsung SM863 960 GB
Dock	2 x HP B8K60AA - 4 Bay SSD carrier
Raid card	LSI Megaraid ROC 9270-8i
Raid cable	2 x Avago 05-26113-00 0.6 Meter internal cable SFF8643 To X4 SATA HDD (MINI SAS HD)
Raid type	RAID 5, Always write-through, with a 256k Stripe size

The table below shows guidelines for the data rates required for real-time playback of single track, 16-bit RGBA DPX sequences, with no soft effects in Nuke Studio's timeline Viewer. The **Data rate** is the required disk speed needed to achieve real-time streaming playback, for a given format, at a given playback rate.



Note: The sample rates are taken from the AJA System Test application, with the **Settings** > **Disk Test** set to **File per frame**, and should be used as guidelines only.

The AJA System Test application is available for download for Mac and Windows from https://www.aja.com/products/aja-system-test

Frames per second (fps)	Format	Data rate MB/s
25 fps	4K Film (4096x3112)	2431.25
	4K UHD (3840x2160)	1582.03
	2K Film (2048x1556)	607.81
	HD 1080p (1920x1080)	395.51
30 fps	4K Film (4096x3112)	2917.50
	4K UHD (3840x2160)	1898.44
	2K Film (2048x1556)	729.38
	HD 1080p (1920x1080)	474.61

Frames per second (fps)	Format	Data rate MB/s
60 fps	4K Film (4096x3112)	5835.00
	4K UHD (3840x2160)	3796.88
	2K Film (2048x1556)	1458.75
	HD 1080p (1920x1080)	949.22

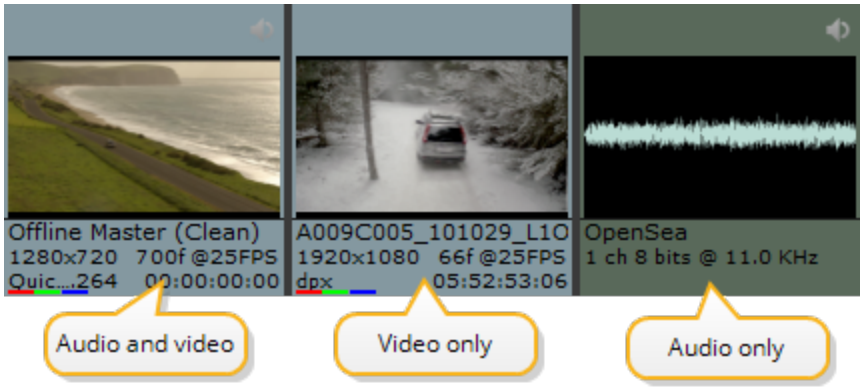
About Clips and Shots

The interface sorts your clips into three broad categories: Audio and Video, Audio Only, and Video Only. Clips are displayed differently depending on their content, location, and in the case of the Viewer, the current mode.

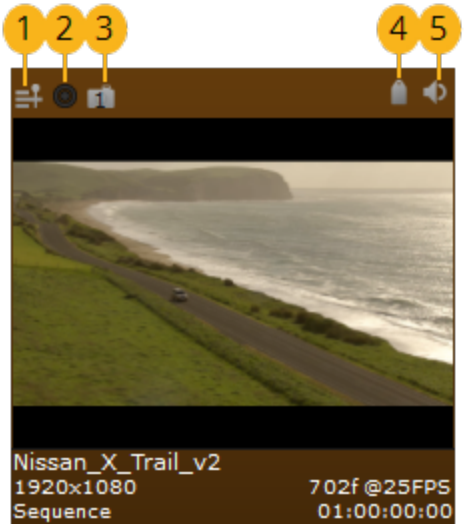
Note: The timeline Viewer currently treats all alpha channels as premultiplied, which can result in the Viewer background being "added" to the image. If you're working with un-premultiplied images, set the Viewer background to **Black**.

Source Clips

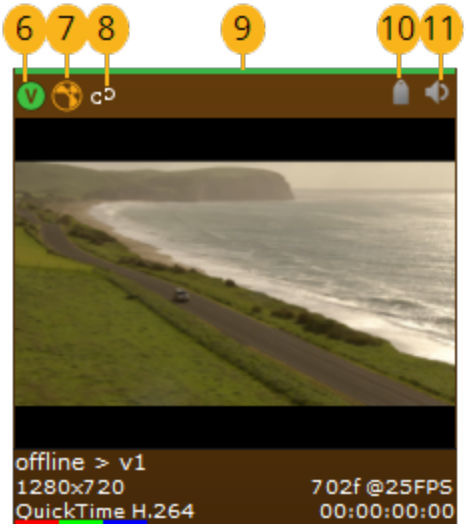
Source clips are representations of files on disk. Shots on the timeline refer to source clips, so making changes to a clip in a bin affects all shots that refer to that source clip.



Source sequences and clips display useful indicators above the thumbnail:



Sequences in the **Project** bin.

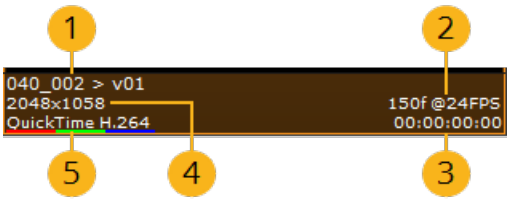


Source clips in the **Project** bin.

Sequence Icons		
1		Indicates that this bin item is a sequence, rather than a clip. Sequences contain shots, audio, comps, soft effects, and other timeline items. See Managing Timelines for more information.
2		Portions of the sequence are cached. If the icon is orange, the entire sequence is cached. Cached sequences provide reliable playback for more complex timelines by rendering frames to disk using the GPU. See Caching Frames in the Disk Cache for more information.
3		You can revert the sequence to a previous state using a snapshot. The number indicates how many snapshots exist. See Using Snapshots for more information.
4		Tags have been applied at the sequence level. Tags are useful organizational tools, particularly if you're collaborating with other artists. See Using Tags for more information.
5		The sequence includes audio. See Audio and the Timeline for more information.
Source Clip Icons		
6		The clip has multiple views, such as the left and right eyes in stereoscopic footage. The source footage can be one file per view or one file containing multiple views, for example **.exr** files. See Stereoscopic and Multi-View Projects for more information.

7		The clip represents a **.nk** Nuke script containing a valid Write node. You can import Nuke scripts using the **File** > **Import File(s)** menu or create a comp directly in the timeline. See Create Comp for more information.
8		The state of the version links between the source clip and shots, which can be linked, unlinked, or mixed. See Version Linking for more information.
9		The clip's localization state, whether or not a local copy is saved to disk to improve playback performance. See Managing Localization for more information.
10		Tags have been applied at the clip level. Tags are useful organizational tools, particularly if you're collaborating with other artists. See Using Tags for more information.
11		The clip includes audio. See Audio and the Timeline for more information.

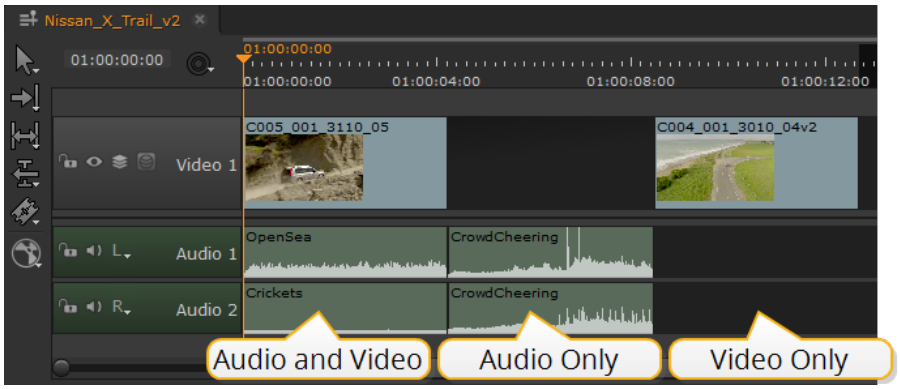
Source sequences and clips display other useful information under the thumbnail:



1	The name of the file on disk referenced by the clip, without the file extension. This name also includes the version of the clip currently in use, **v01** in the example.
2	The number of frames referenced by the clip and the framerate encoded in the file(s), 150 frames at 24 fps in the example.
3	The starting timecode of the file referenced by the clip.
4	The resolution of the file referenced by the clip, 2048 x 1058 in the example.
5	The file type referenced by the clip, **.mov** encoded using H.264 in the example. **Note:** The colored chips under the thumbnail represent the layers available in the clip, such as **color** in the example image. Other layers include **alpha**, **depth**, and **motion**, similar to Nuke.

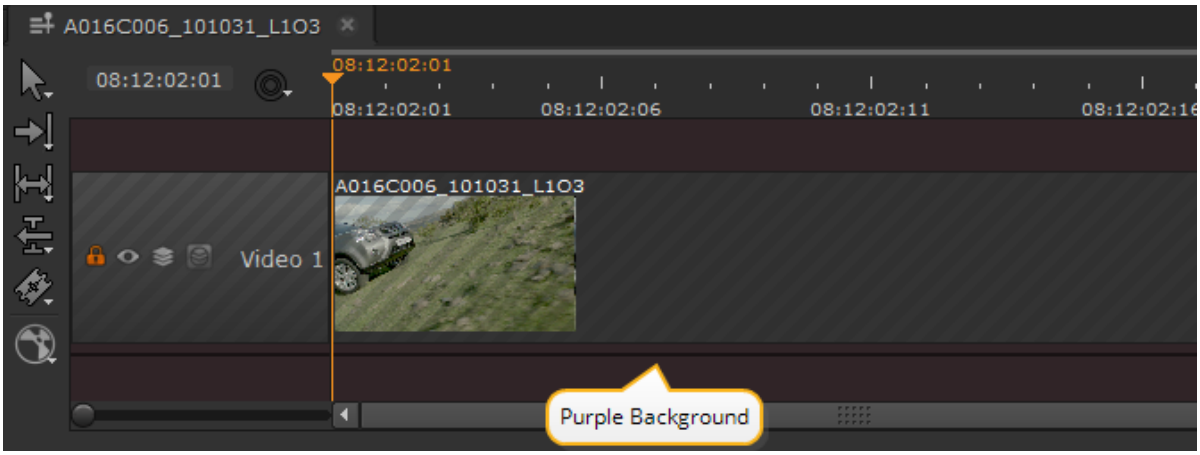
Shots

Shots are representations of source clips, they are not saved on disk. Making changes to a shot only affects that instance. See Managing Timelines for more information.



A Source Clip Opened in the Timeline View

Opening a source clip in the timeline view allows you to control its output and add soft effects to the source clip. Clips opened as timelines always contain all available frames, but you can adjust their output using the **original range** controls in the clip's properties or in and out points. See Using In and Out Markers and Adding Effects on the Timeline for more information.



Clip and Shot Properties

The **Properties** contain standard controls similar to Nuke's Read node **Properties** and format-specific controls, depending on the selected source clip or shot. You can also control localization for individual clips and shots using the **Localization Policy** dropdown.

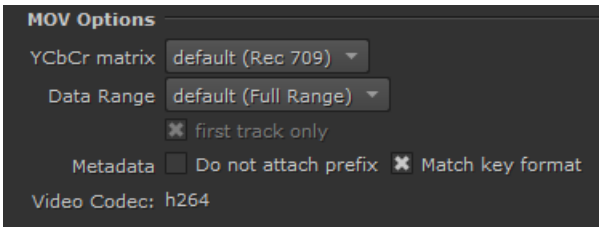
See Localizing Media for more information on localization.

If the clip **Properties** are not displayed in your workspace, double-click a source clip or shot or navigate to **Window** > **Properties** to open them in a floating window.

Note: The **Properties** panel allows you to override the **Preferences** and **Project Settings** on a per file basis. See Appendix A: Preferences and Timeline Environment Project Settings for more information.

As an example, some **.mov** files allow you to control the **YCbCr matrix** and **Data Range** for clips, as well as the standard controls available for all clips. R3D and ARRIRAW media use their own software development kits (SDKs) to control the extensive settings usually seen on RED and ARRIRAW Cameras.

Note: The RED **Decode Resolution** and ARRIRAW **Resolution** and **Proxy** dropdowns control the maximum allowed Viewer resolution, overriding the Viewer **Image Quality** setting.



In addition, source clip properties are accessible through the same Python API as Nuke, improving scripting capabilities and integration into existing pipelines.

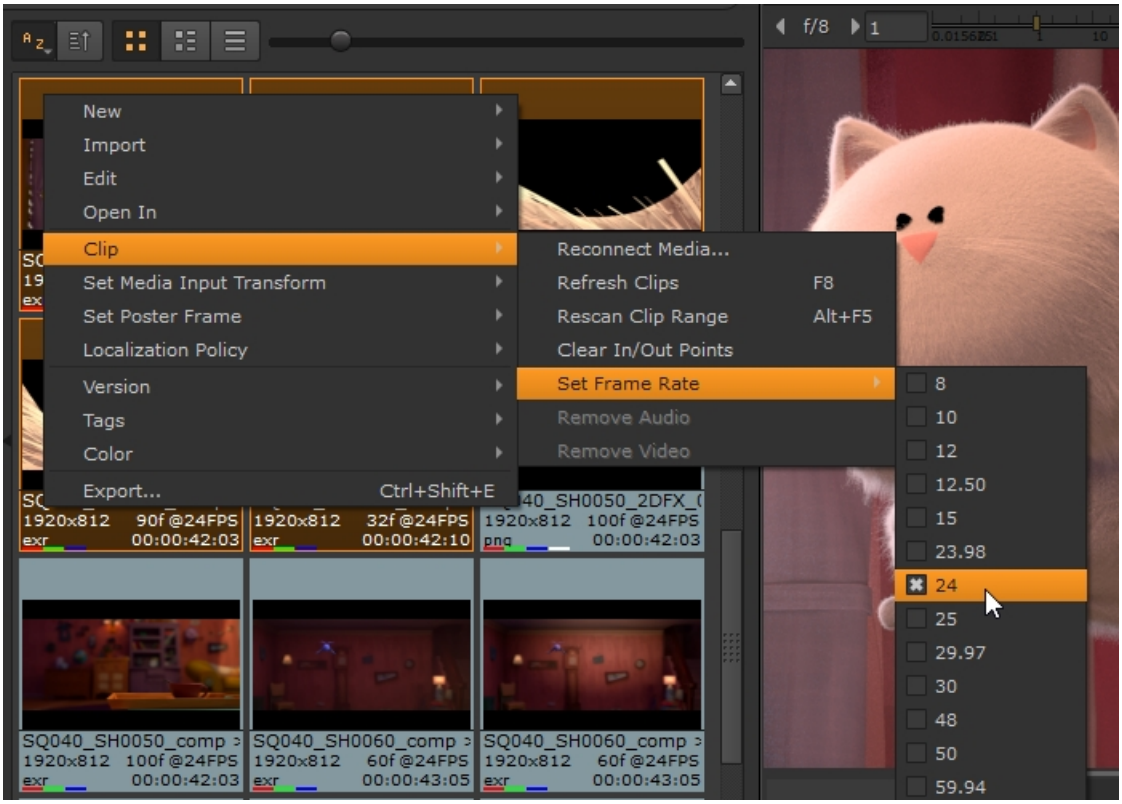
Setting Clip Frame Rates

You can change a clip's frame rate, or frames per second (FPS), from the **Project** bin. This allows you to set or change the FPS of clips before adding them to a sequence in case the clip's FPS is set incorrectly or badly defined in the file's metadata. You can set a single clip's FPS or multiple selections simultaneously.

To set a clip's frame rate:

1. Select the target clip or clips in the **Project** bin.
2. Right-click a selected clip and navigate to **Clip** > **Set Frame Rate**.

3. Select the required frame rate from the menu.



The selected clip or clips are updated to playback at the chosen FPS.

Note: Changing a clip's frame rate does not change the FPS of related shots in a sequence. This is because a shot's FPS is set by the sequence **Frame Rate** control and does not rely on the source clip's frame rate.

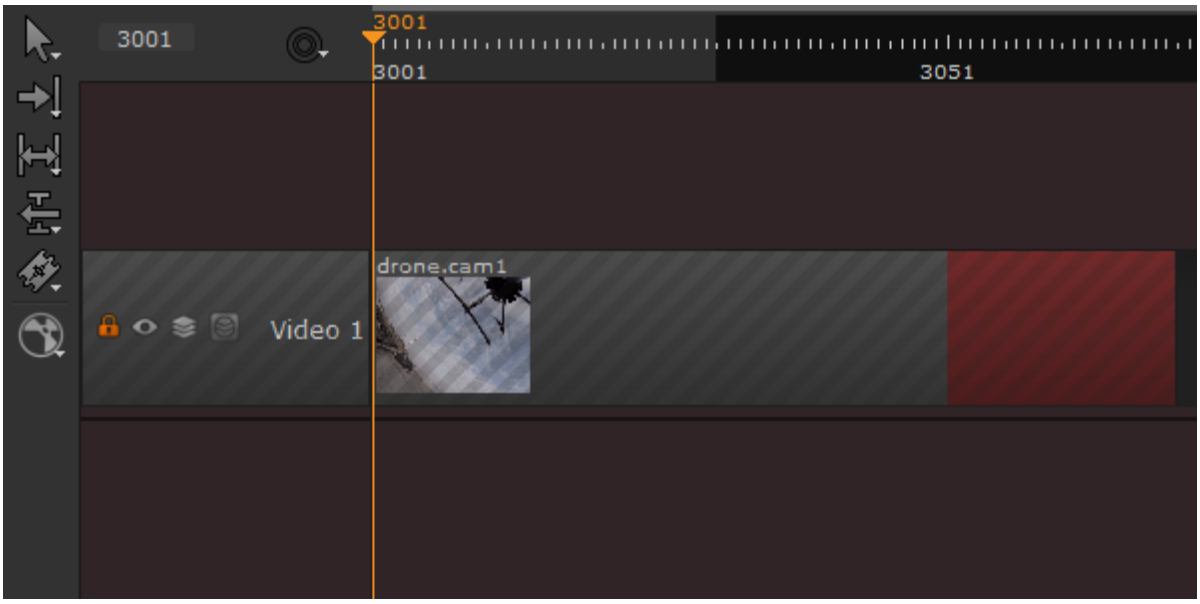
Setting Source Clip Ranges

The **original range** controls in the source clip/shot **Properties** panel allows you to set the output range for source clips when you add them to a sequence as shots. The entire frame range of the clip is read on import, but altering the **original range** changes the output of the shot on the timeline.

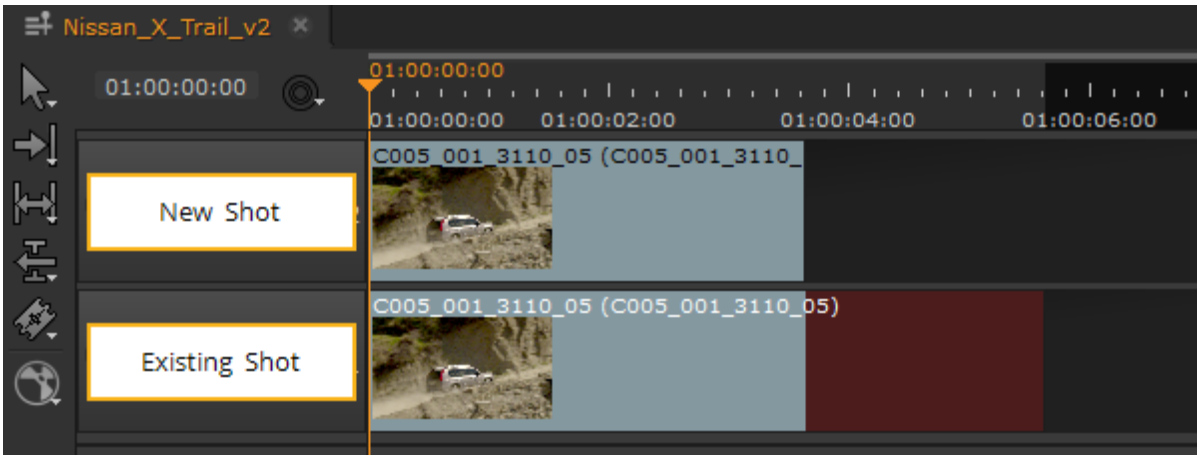
Note: You can reset source clip to its original frame range at any time by clicking **rescan**. Rescanning also appends new frames as they become available.

Changing the **original range** affects source clips and shots differently. Opening a source clip in the Viewer, or in the right-click Timeline View, and adjusting the **original range** limits or extends the clip to that range of frames.

Extending the range past the original range adds handles, shown in red.



Adjusting the **original range** changes the output of any existing shots, but still allows you to trim and slip the shot range using the available handles. New shots pick up the new **original range** specified in the **Properties**.



Using Relative Paths to Media

Nuke Studio's timeline Read nodes support relative paths in the **file** control, allowing you to import media relative to a specific project directory. The **Project Settings** control this behavior for the current project.

Tip: If you want to set the project path for all new projects, navigate to the **Preferences** under **Project Defaults** > **General** and set the **project directory** there.

To set a Read **file** path to a relative location:

1. Open the **Project Settings** by clicking **Project** > **Edit Settings**.
2. Enter the required file path in the **Project Directory** field.

   Tip: Click **Hrox Directory** to automatically enter an expression that evaluates to the **.hrox** location.

3. Double-click the required source clip or shot to display its **Properties**.
4. In the **file** field, adjust the path to the relative location of the **.hrox** project file. For example, if the project directory and absolute path to the source clip are:

   ```
   C:/projects/timeline_read/scripts/
   C:/projects/timeline_read/footage/renders/shot110/A009C005_101029_L1O3.%04d.dpx
   ```

   Then the relative path to the clip is:

   ```
   ./../footage/renders/shot110/A009C005_101029_L1O3.%04d.dpx
   ```

Ingesting Media

Adding media is as simple as drag-and-drop from a file browser or selecting **File** > **Import File(s)** or **Import Folder(s)**. The application imports your media into the bin view providing you with a thumbnail of all of your clips and preserving the original folder and file hierarchy.

The media is soft imported, creating symbolic links to locations on disk. See Using the Copy Exporter for information on how to quickly consolidate your media and projects, or Localizing Media to help stabilize playback.

Note: Projects containing large amounts of movie files (for example **.r3d** and **.mov**) may exceed the number of available file handles per process, causing problems opening new files or projects and exporting.

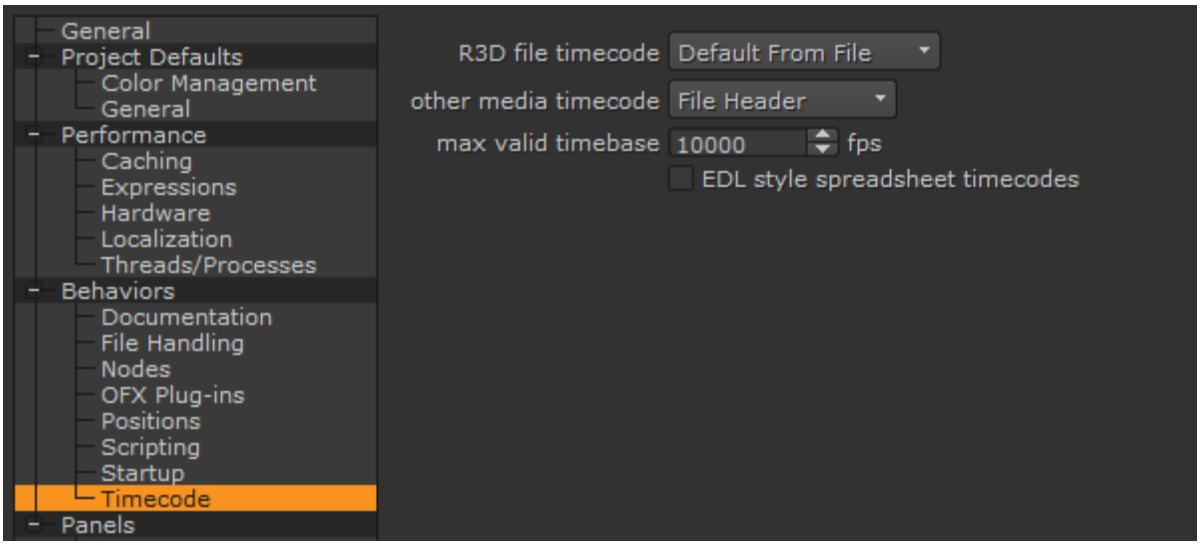
You can increase the default limit of 1024 by entering the following command from the terminal, then running the application from the same session:

```
ulimit -Sn 2048
```

Clips with no inherent frame rate information are assigned a frame rate at ingest as specified in the **Preferences**.

Article: See Knowledge Base article Q100155 for more information on changing the frame rate of clips in Nuke Studio.

1. Open the **Preferences** dialog by pressing **Shift**+**S**.
2. Select **Behaviors** > **Timecode** from the sub-menu on the left.



3. Use the **RED file timecode** dropdown to determine R3D clip behavior:
   - **Default from File** - use the default set by the R3D file in question.
   - **Absolute Timecode** - force the use of the Absolute Timecode as specified in the clip metadata.
   - **Edge Timecode** - force the use of the Edge Timecode as specified in the clip metadata.
4. Use the **other media timecode** dropdown to determine clip behavior for all other clips:
   - **File Header** - the file metadata header is used to derive the timecode, if it exists. This option defaults to **Frame Number** if the header is missing.

- **Frame Number** - ignores the metadata header, even when present, and derives the timecode from the frames in the clip.

5. Set the **max valid timebase** allowed from the image header, above which the value is clamped.

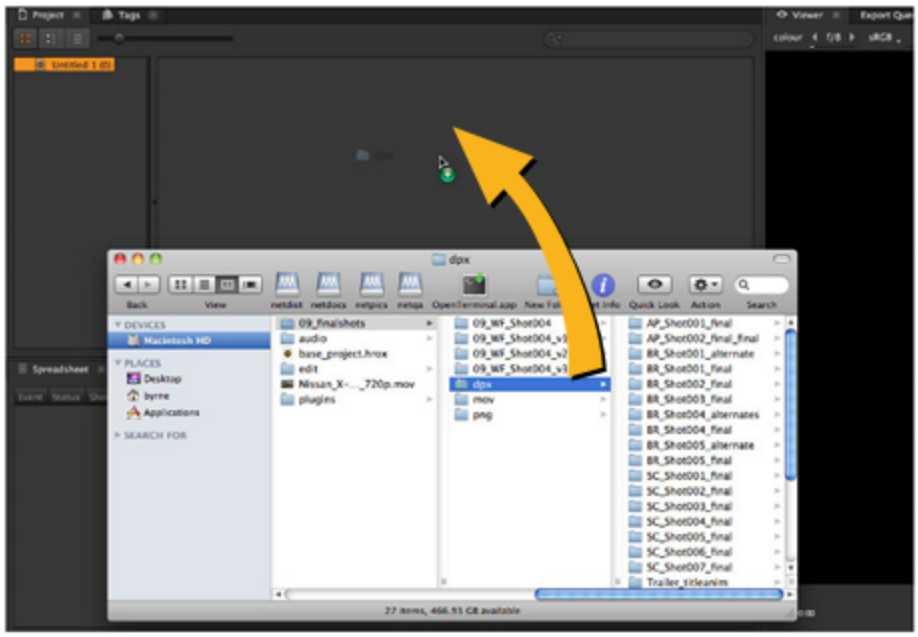
   Image files are often created with application specific timebase values in the header description. This can lead to reading in spuriously high frame rates, and the clamp aims to prevent this from happening.

   If your clips do have extremely high frame rates, increase this value as necessary to avoid clamping.

6. Enable or disable **EDL style spreadsheet timecodes**:
   - When disabled, the **srcOut** and **dstOut** values use the film convention, representing the **last** frame of the cut.
   - When enabled, the **srcOut** and **dstOut** values use the video convention, representing the frame directly **after** the cut.
7. Click **OK** to save your settings.

Using Drag-and-Drop

Locate your media in a file browser and drag the frame range, clip, folder, or folders into the **Project** tab.



Ingest behavior depends on the target:

- Dragging a folder into the **Project** tab automatically ingests all the contents of the folder, including other folders and their contents.
- Dragging a movie file, such as a **.mov** or **.r3d**, automatically ingests the entire clip.

- Dragging a single file or file range, that is part of an image sequence, is controlled by the **Preferences** > **Behaviors** > **File Handling** > **scan for file sequence range on drop into Bin view** checkbox:
  - **Enabled** - the default setting, dragging a single file or file range, that is part of an image sequence, creates a clip in the bin view containing all available frames.

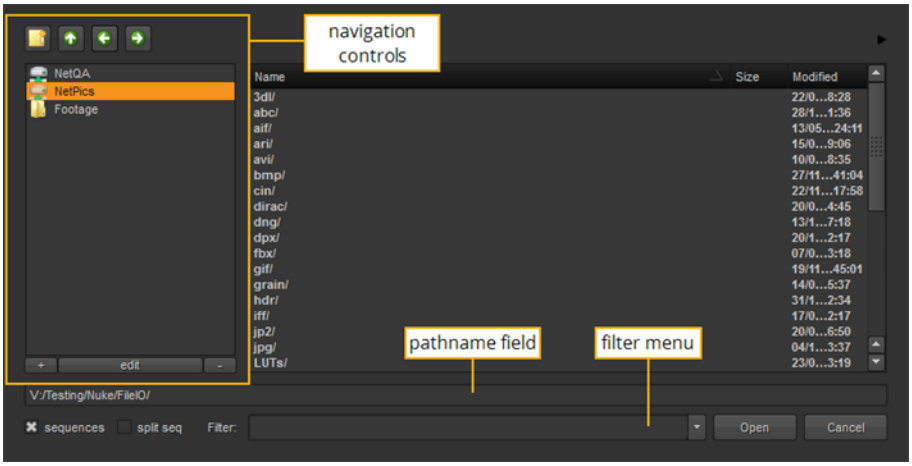
    For example, dragging frames 1-5 and 11-20 ingests the entire frame range.
  - **Disabled** - only the dragged frame or range is imported into the bin.

    For example, dragging frames 1-5 and 11-20 ingests two distinct clips, one containing 5 frames and one containing 10 frames.

Using the File Browser

If you prefer to work with menus, you can also import clips using the file browser. You can import individual clip files, ranges, or entire folders, depending on the amount of media you intend to use.

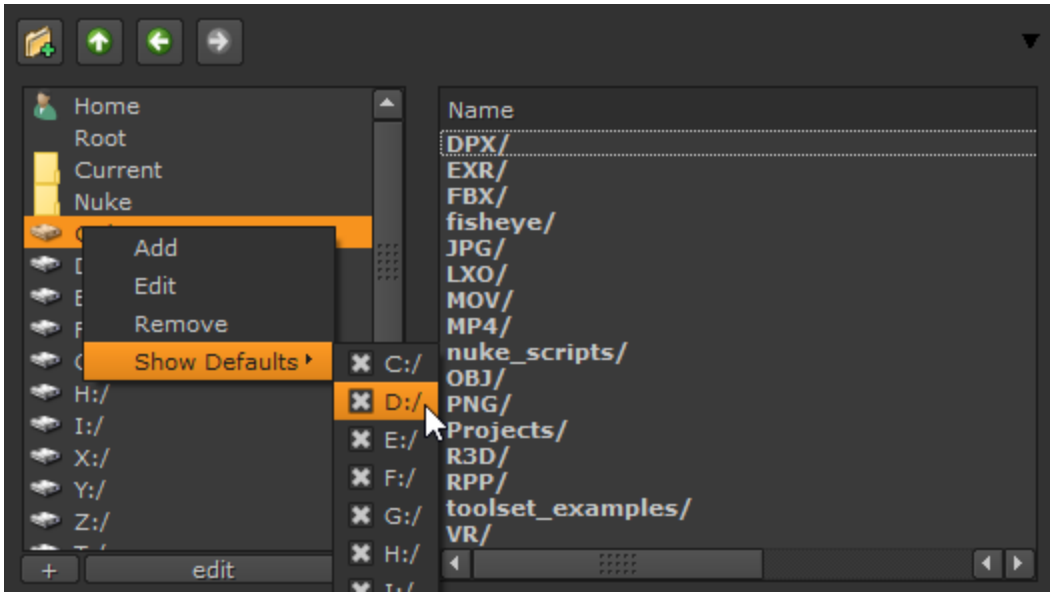
Whenever you load or save files, a browser similar to the one shown below is displayed. The directory navigation buttons let you create or access the directory from which you wish to read or write data.



The navigation controls let you move through the directory structure, bookmark favorite directories, and create new directory folders.

Note: If you import folders, use the **Import Options** dialog to filter your ingest using inclusion and exclusion parameters, separated by spaces. The dialog's **Include patterns** field defaults to **{supportedfiles}**, which resolves to a list of all known supported file extensions. To add your own custom extensions to this, you can use **{supportedfiles} *.ext** (replacing .**ext** with your custom file extension).

Windows only: You can show/hide the drives that Windows auto creates by right-clicking the required drive, selecting **ShowDefaults**, and checking or unchecking the drive.

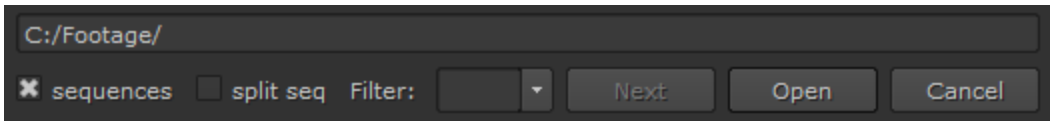


To Use the Navigation Controls

- Click the **Create New Directory** button to create a new directory at your current position in the file hierarchy.
- Click **Up one directory** to go up one directory closer to the root.
- Click **Previous directory** to go back one directory.
- Click **Next directory** to go forward one directory.
- Click the **+** button to add a directory bookmark.
- Click the **edit** button to edit the name or path name to a bookmark.
- Click the **-** button to remove a directory bookmark.

Path Name Field

The path name field displays the current directory path, lets you navigate to a new path, and also enter a file name for scripts and rendered images.



- To navigate to a directory, type the path name in the field.
- To enter a script name, browse to a directory path and enter the file name after the displayed path.
- To limit the file list to specific file types, use the **filter** dropdown menu and **Sequences** checkbox.

To Use the Filter Dropdown Menu and Sequences Checkbox

- Select ***.nk** to display only Nuke script files.
- Select * to display all files (except hidden files), regardless of what they're associated with.
- Select .* * to display all files, including hidden files.
- Select ***/** to display directory names, but not their contents.
- Check **sequences** to display image sequences as single titles, as in fgelement.####.cin 1-50 rather than fgelement.0001.cin, fgelement.0002.cin, fgelement. 0003.cin, and so on.



Note: File sequences with no file extension (for example, fgelement.0001, fgelement.0002, fgelement.0003, and so on) are not displayed as single titles the first time you view the directory in the File Browser. However, they are displayed as single titles once you have navigated to another directory and back again.

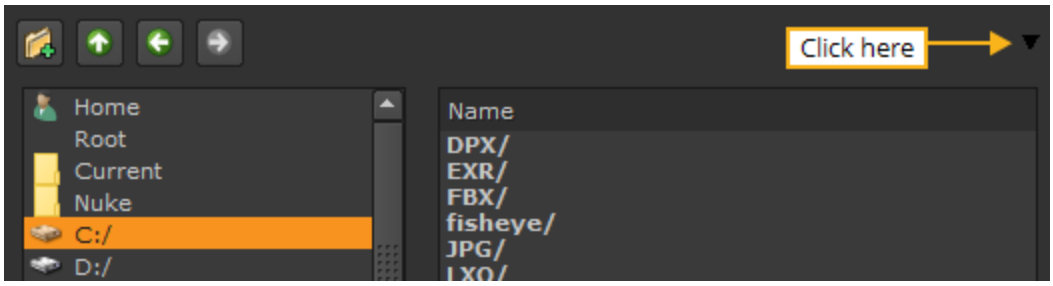
Note: By default, the application may not be able to display custom file extensions (for example, **.cext**) as single titles. To fix this, you can register your custom file extension as a sequence type using Python:
1. Create a file called **init.py** in your plug-in path directory if one doesn't already exist. For more information on plug-in path directories, see Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts.
2. Open the **init.py** file in a text editor and add an entry in the following format (replacing **cext** with your custom file extension):

```
nuke.addSequenceFileExtension("cext")
```

• You can also split incomplete sequences into separate Read nodes using the **split seq** checkbox.

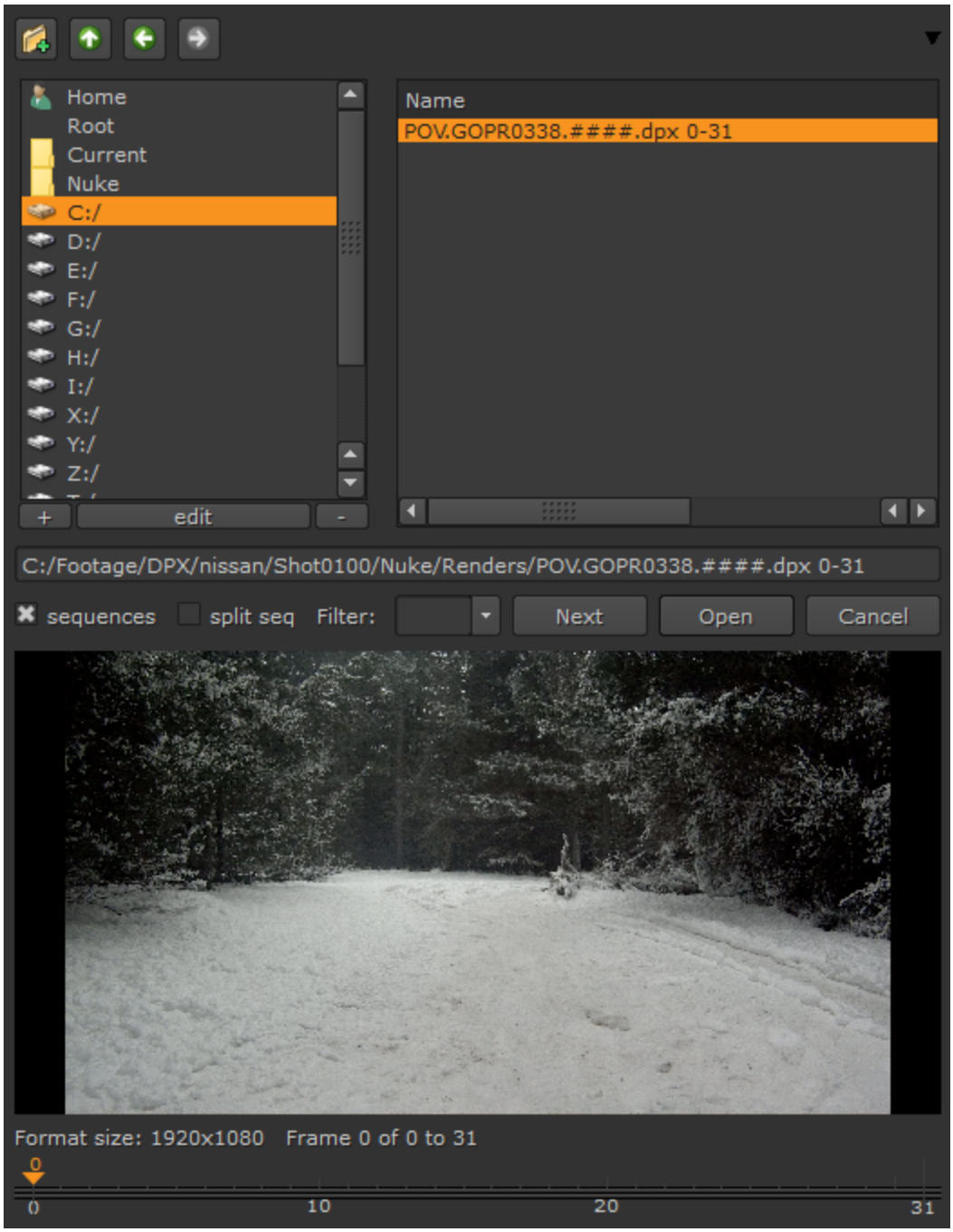
To Preview Files in the File Browser

1. Click the black arrow in the top-right corner of the file browser.



The file browser expands to include a small viewer.

2. Select the file you want to preview in the file browser to view it.



To Select Multiple Files with the File Browser

1. Browse to the folder where the files are located.
2. **Ctrl**+click on all the files you want to open to select them (Mac users **Cmd**+click).
3. You can open files from multiple directories by clicking **Next** and browsing to the next file location.
4. Click **Open**.

   All the selected files open.

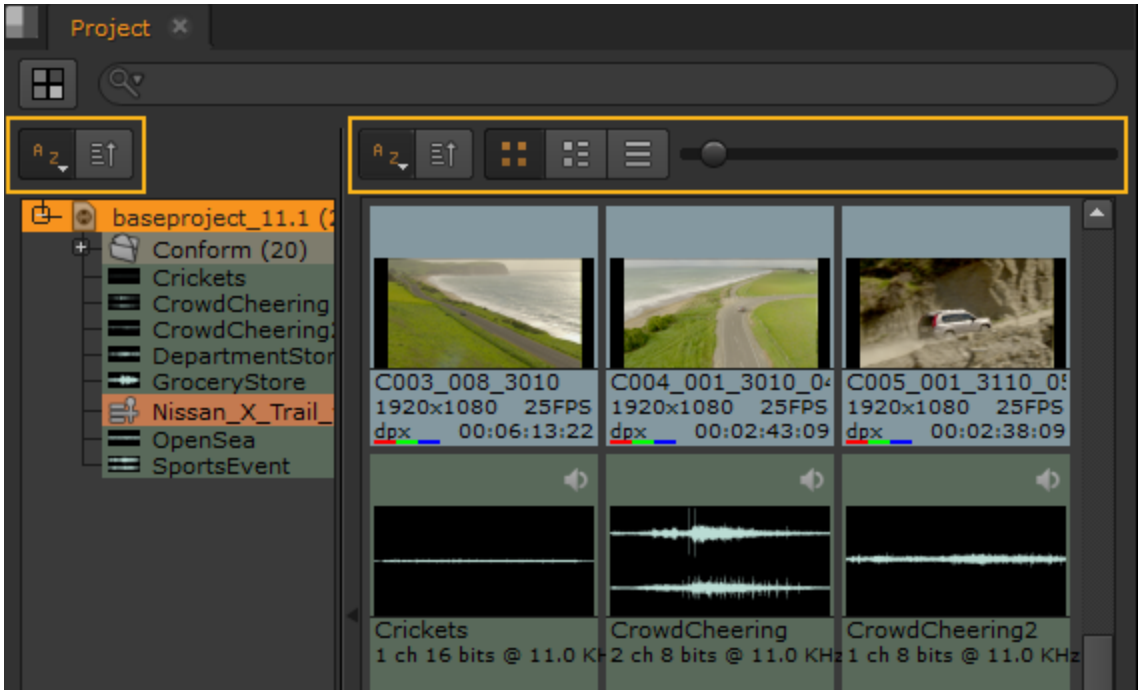
Sorting and Searching Media

Nuke Studio's **Project** panel includes several sorting and searching methods so you can organize, manage, and navigate through media more easily. You can sort items alphabetically, by type, or manually and select the appearance of media within bins. The search functionality allows you to enter strings and apply searches on all or partial matches. You can control the poster frame displayed by items in the project, selecting either an absolute or relative frame.

You can also color-code items in the **Project** panel and timeline to quickly sort and locate media. See Color-coding Source Clips and Shots for more information.

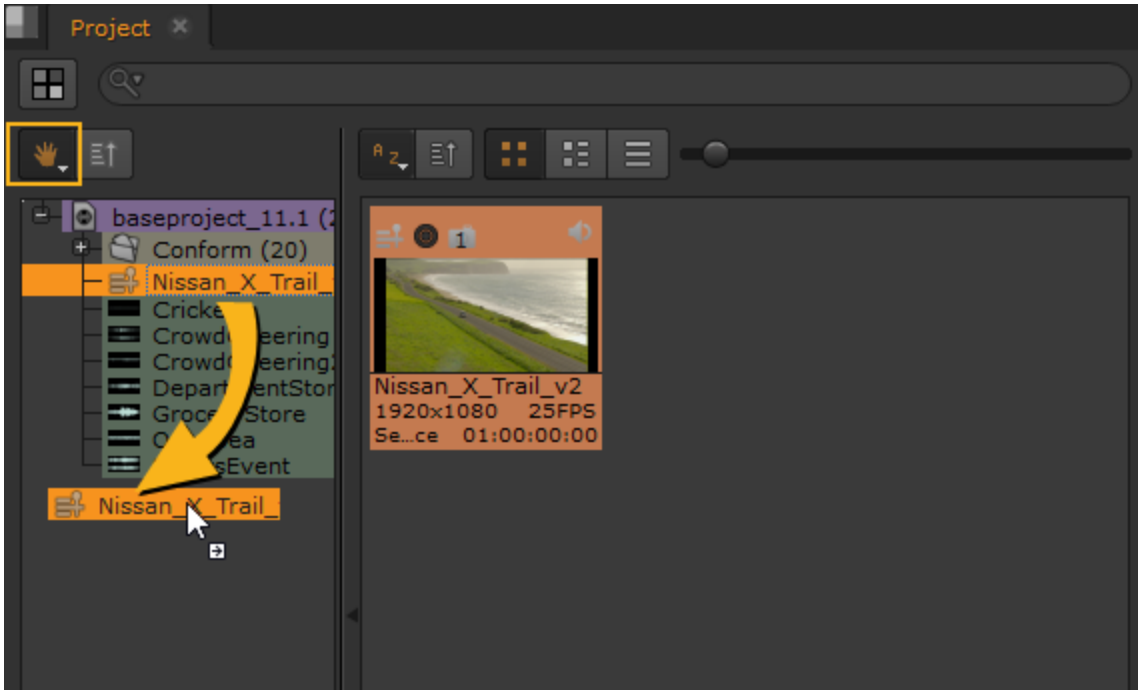
Sorting the Project Panel

The **Project** panel can be sorted in the left-hand directory side and right-hand file side independently, though files have more sorting and display options.

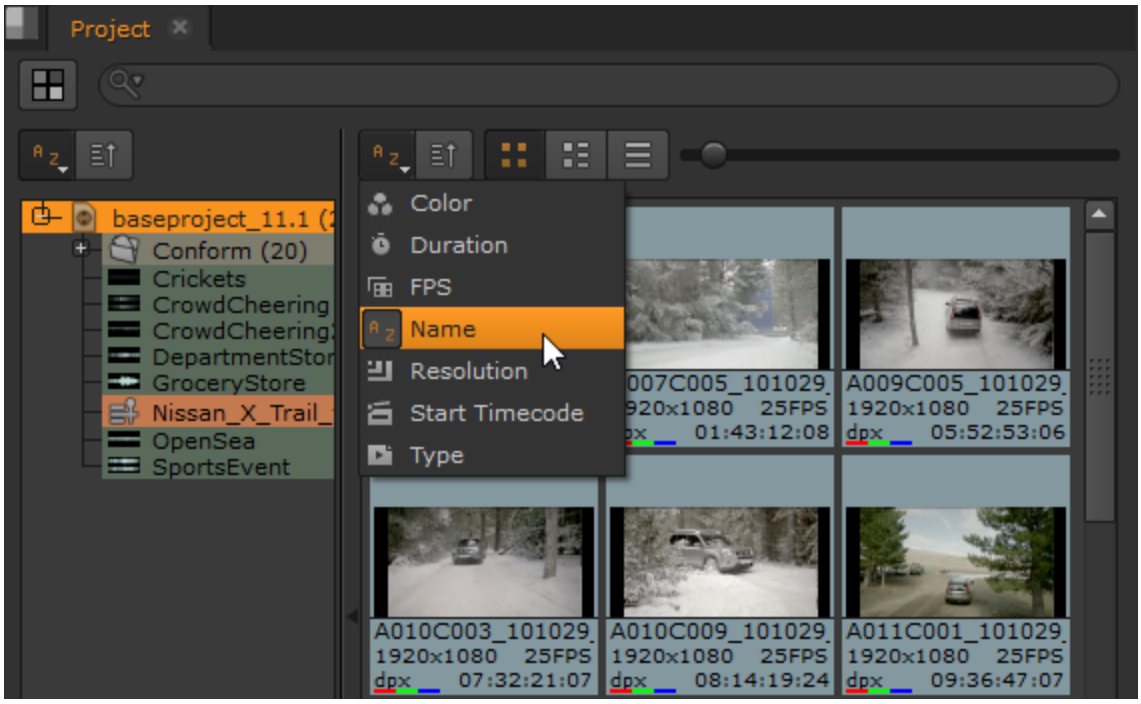


In the left-hand directory pane, items are sorted alphabetically in ascending order by default. To sort project items manually:

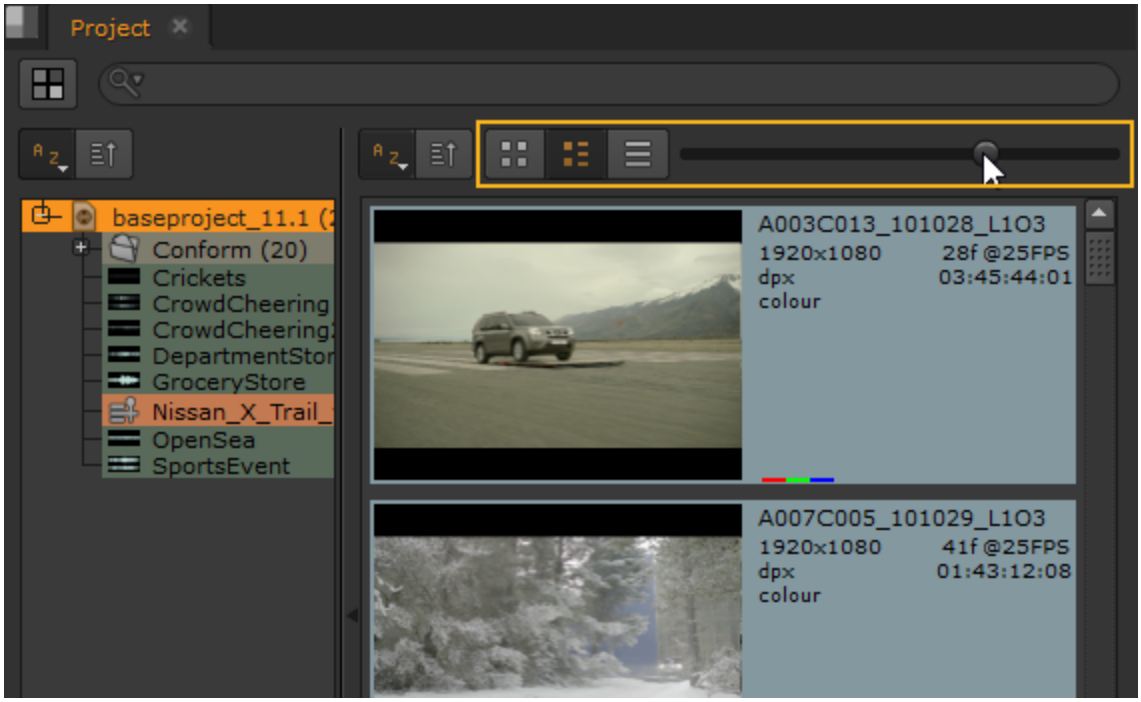
1. Click and hold, or right-click, the sorting dropdown and select **Manual**.
2. Drag-and-drop items into your preferred order.



The right-hand bin pane is also sorted alphabetically in ascending order by default, and there are more options available in the sorting dropdown. Click and hold the dropdown to display the various options, including **Duration**, **Resolution**, and **Type**.



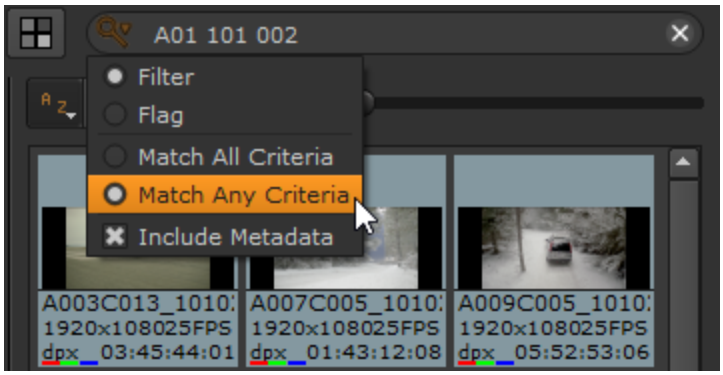
You can also change the layout and size of items in the right-hand bin pane. Choose a layout from thumbnails, list, and details and use the slider to determine the size of the items. The slider is disabled in details mode.



Searching for Media

Nuke Studio's search functionality allows you to enter strings and apply searches on all or partial matches with the option to include metadata searches. Nuke Studio searches for items that match any of the input string and displays only those items by default.

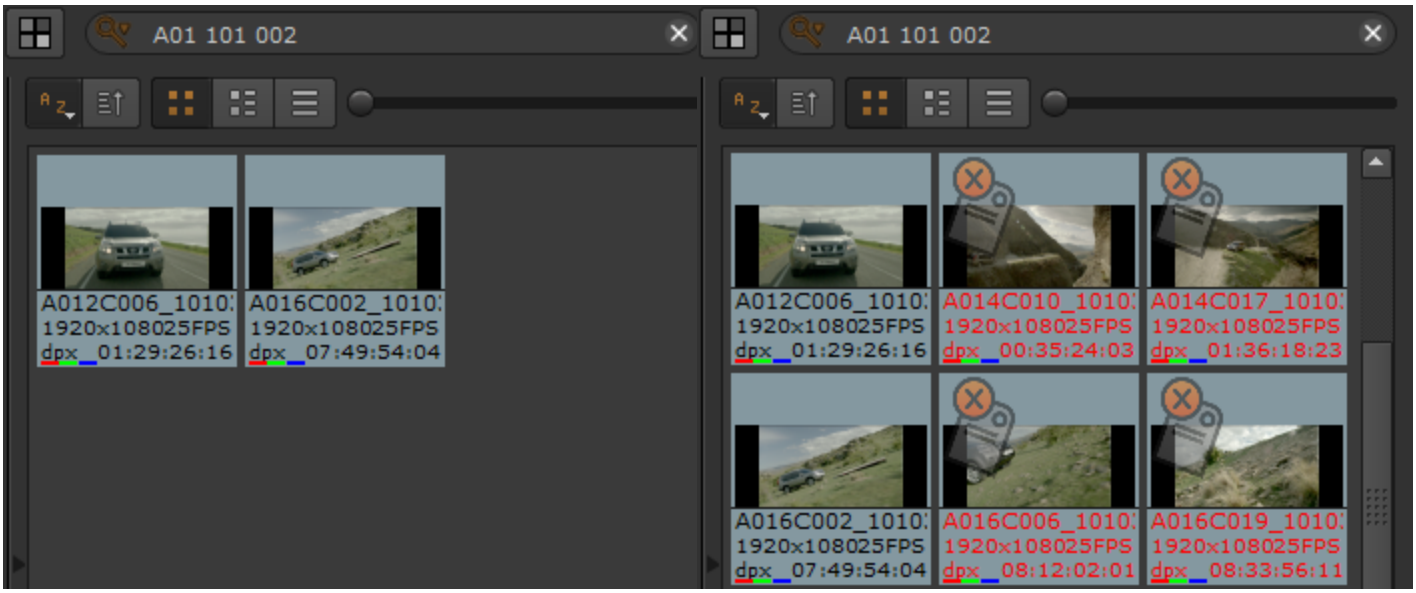
Tip: Nuke Studio also examines file metadata for the search string by default, but you can disable this behavior by disabling **Include Metadata** in the search dropdown.



- **Match All Criteria** - media that matches **all** the entries in the search string are matched. For example, entering A01 DPX only matches media containing both A01 and DPX.
- **Match Any Criteria** - media that matches **any** of the entries in the search string are matched. Using the same example, A01 DPX matches any media containing A01 or DPX.

Tip: This functionality extends to the media spreadsheet if you prefer to search for items there. See About the Media Spreadsheet for more information.

Filtering and flagging produce the same search results, but they are presented differently. Filtering only displays files that match some or all of the search string, whereas flagging displays all content and flags files that **don't** match some or all of the search string.



Filtering ... ... and Flagging

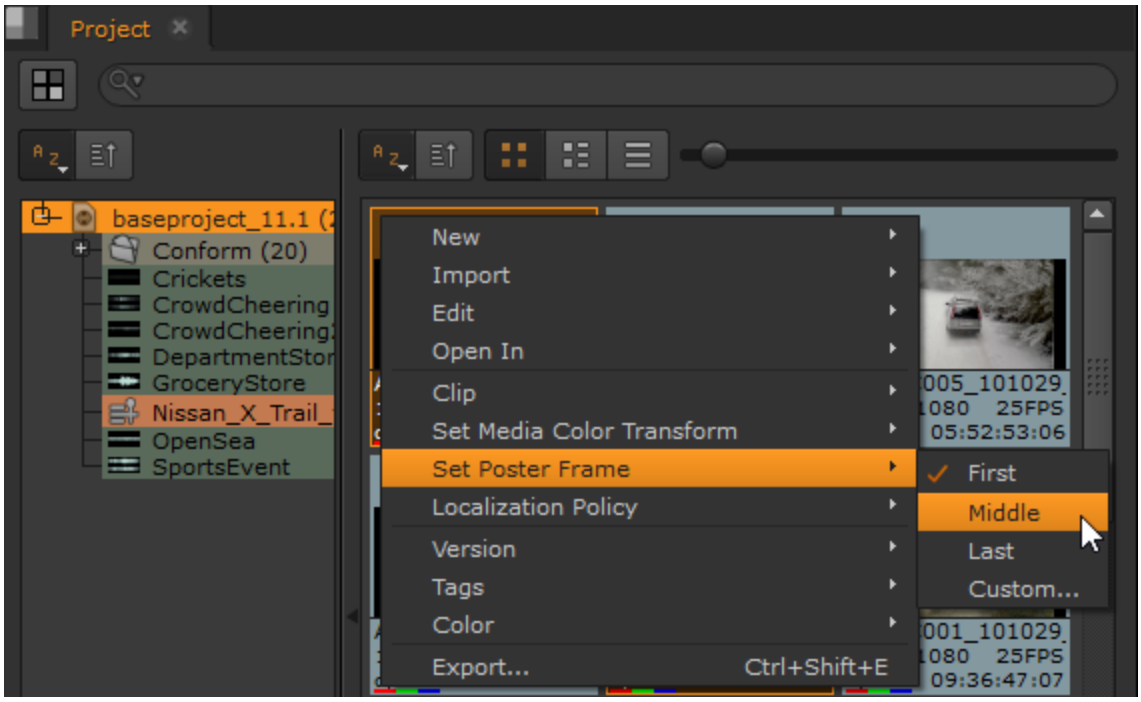
You can also filter and flag media by tag. See Filtering and Flagging Media Using Tags for more information.

Setting Poster Frames

You can assign poster frames to image items in the Project panel so that they represent the frames you intend to use if your sequences. This is particularly useful if your source clips contain handles of blank frames at the beginning of the footage.

To set a poster frame for source clips:

1. Select the clip or clips in the **Project** panel.
2. Right-click the selection, and select **Set Poster Frame**.



3. Select a preset frame or click **Custom** to select an absolute or relative frame as the poster frame for the source clip:
   - **Absolute** - the poster frame number is derived from the file(s) on disk. For example, a **.dpx** sequence **myClip.####.dpx 1001-1500** can have an absolute poster frame between **1001** and **1500**.
   - **Relative** - the poster frame number is derived from the number of frames in the clip starting at 0. For example, a **.dpx** sequence **myClip.####.dpx 1001-1500** can have a relative poster frame between **0** and **499**.

Note: If you select more than one source clip, you can't set an **Absolute** frame as the poster frame.

Color-coding Source Clips and Shots

In large projects, the **Project** bin and timeline can quickly become busy and difficult to manage. Adding colors to items or file types can help you find what you're looking for in bins and sequences. You can quickly

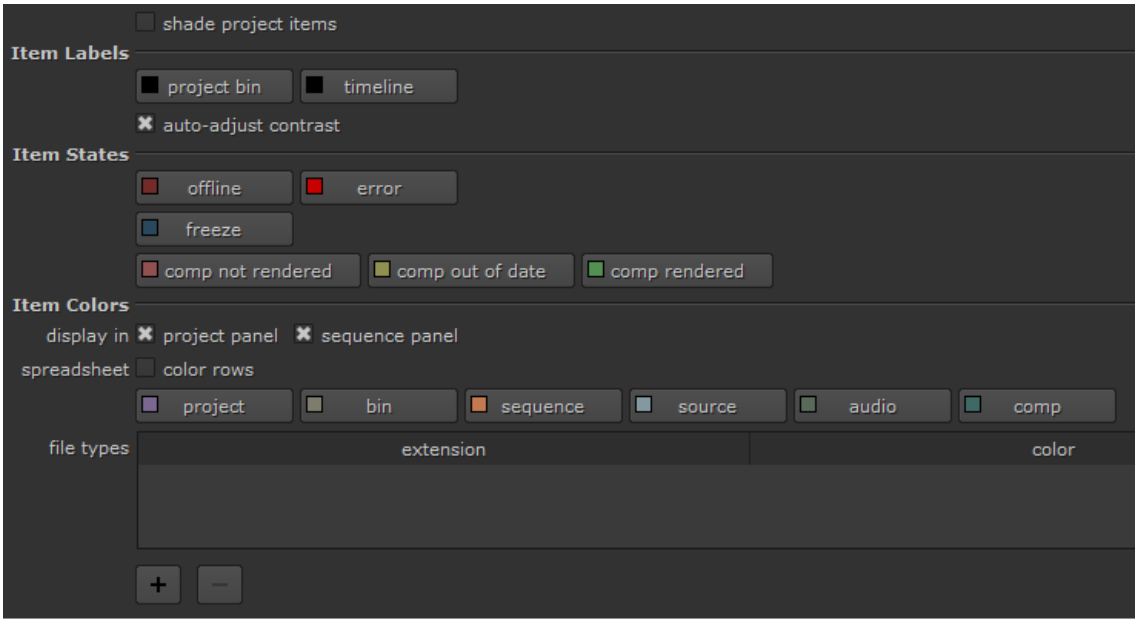
enable and disable colors in the **Project** panel and timeline in the **Preferences** under **Panels** > **Project Items**. See the Appendix A: Preferences page under Panels for more information.

You can set general defaults for items such as bins, sequences, and source clips as well as for specific file types such as **.exr** or **.mov** files. You can also set custom colors for individual selections or groups using the right-click menu or color picker button at the top left of the **Project** panel.

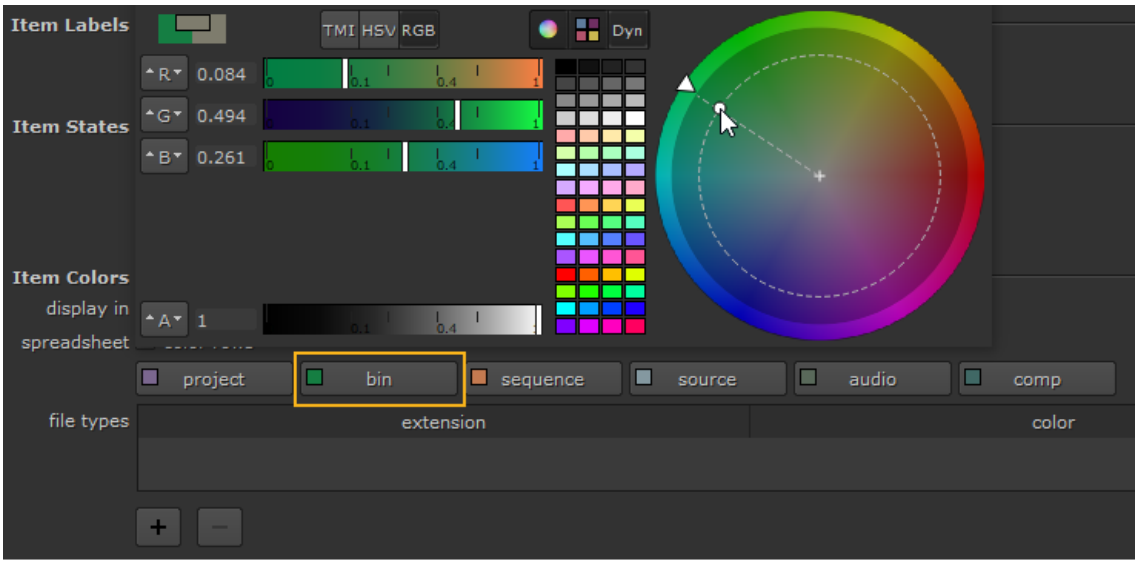
Setting Default Colors

Project items have a default color assigned to them by type, such as orange for sequences and gray for source clips. You can change these assignments using the **Preferences** > **Panels** > **Project Items** dialog.

Note: You can disable the color scheme in the **Project** panel, timeline, and spreadsheet at any time using the **display in** controls in the **Preferences**.



To change a default color, click a button in the **Project Items** preferences and use the color wheel to change the color.



You can also set the label color for the **Project** panel and timeline or change the color used to indicate the state of a shot on the timeline, such as **offline**.

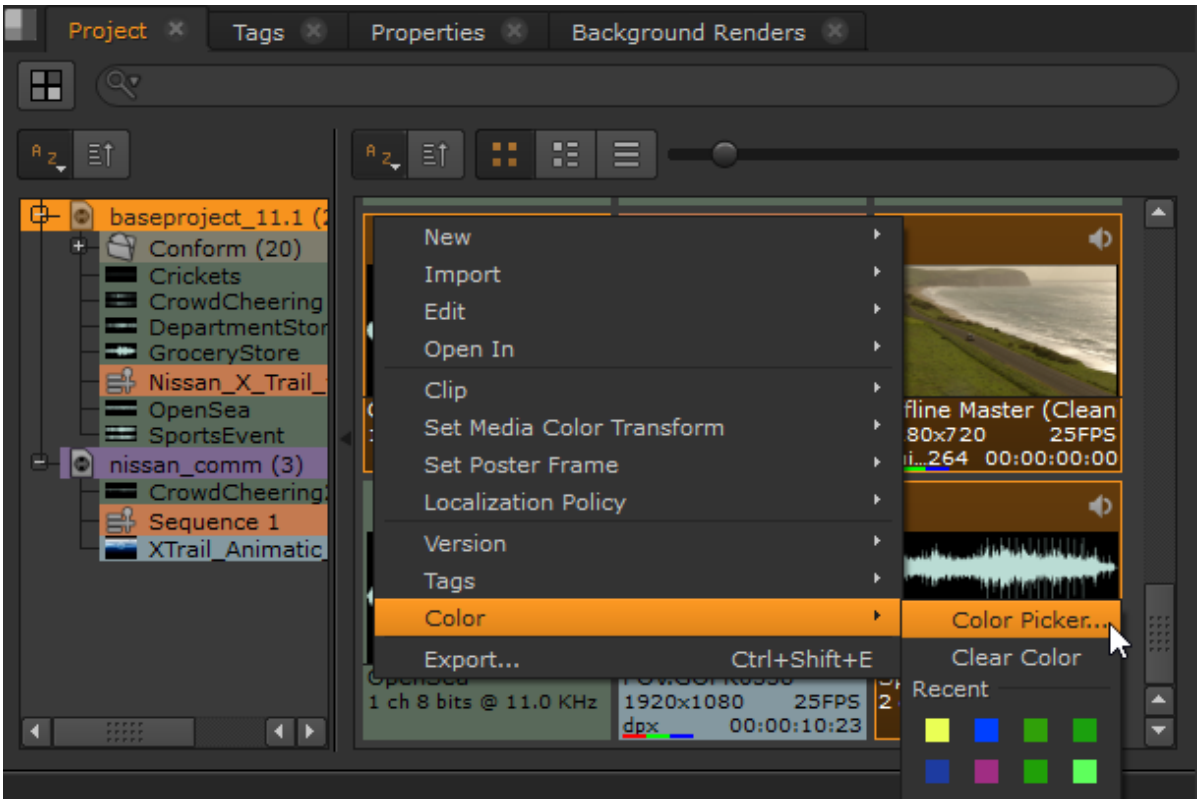
Setting Colors by Selection

You can keep track of particular clips and shots by manually assigning them a color of their own. This custom color overrides any color you set by type. You can set colors on items individually or by selecting multiple instances in the **Project** panel or timeline.

1. In the **Project** panel or timeline, select the items you want to color.

> **Tip:** The selection tools can help you make multiple selections quickly in the timeline. See Using the Selection Tools for more information.

2. Right-click the selection and choose **Color** > **Color Picker** or click a color under **Recent**.



Tip: You can also color selections by clicking the button and picking the color.

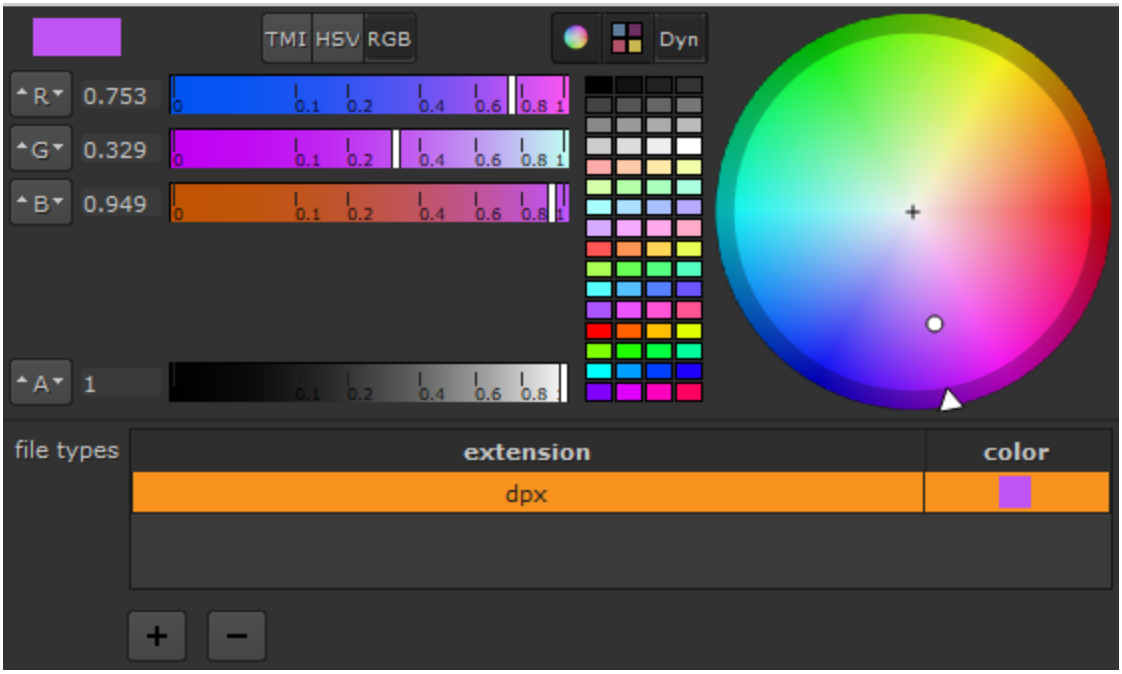
3. Click **Clear Color** to remove any custom colors applied to the selection. The selection reverts to the default item color set in the **Preferences**.

Setting Colors by File Type

Another way to organize your project is to assign colors to items by file extension. This custom color overrides any color you set by item type.

1. Open the **Preferences** by pressing **Shift**+**S**.
2. Navigate to the **Panels** > **Project Items** sub-menu.
3. Under file types, click the + button.

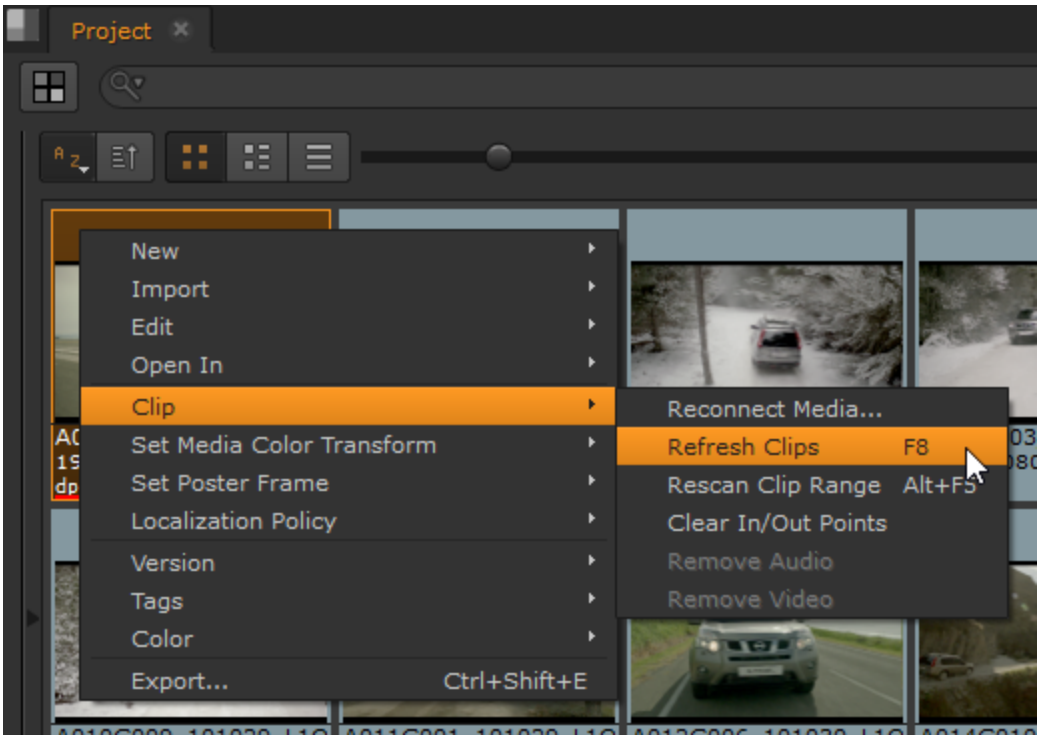
   A new entry is added to the file types table.
4. Click the **extension** dropdown and select the required file type.
5. Double-click the **color** swatch and select the new color using the color wheel.



6. Click **OK** to apply your changes.

Reconnecting and Refreshing Clips

During the post process, media inevitably changes location or form. You can reload or replace your media using the reconnect, refresh, and rescan functions.



Though all three options deal with reloading source clips, each has a particular use dependent on context:

- **Reconnect Media** - allows you to redirect the filepath when the source file location changes.
- **Refresh Clips (F8)** - allows you to reload the clip when the source file location has not changed, such as when work has been done on the clip offline. Selecting refresh only refreshes the clip's current frame range.
- **Rescan Clip Range (Alt+F5)** - similar to **Refresh Clips**, above, but rescan also checks for additional frames that may have been added to the source file and adds them to the source clip's frame range.

Localizing Media

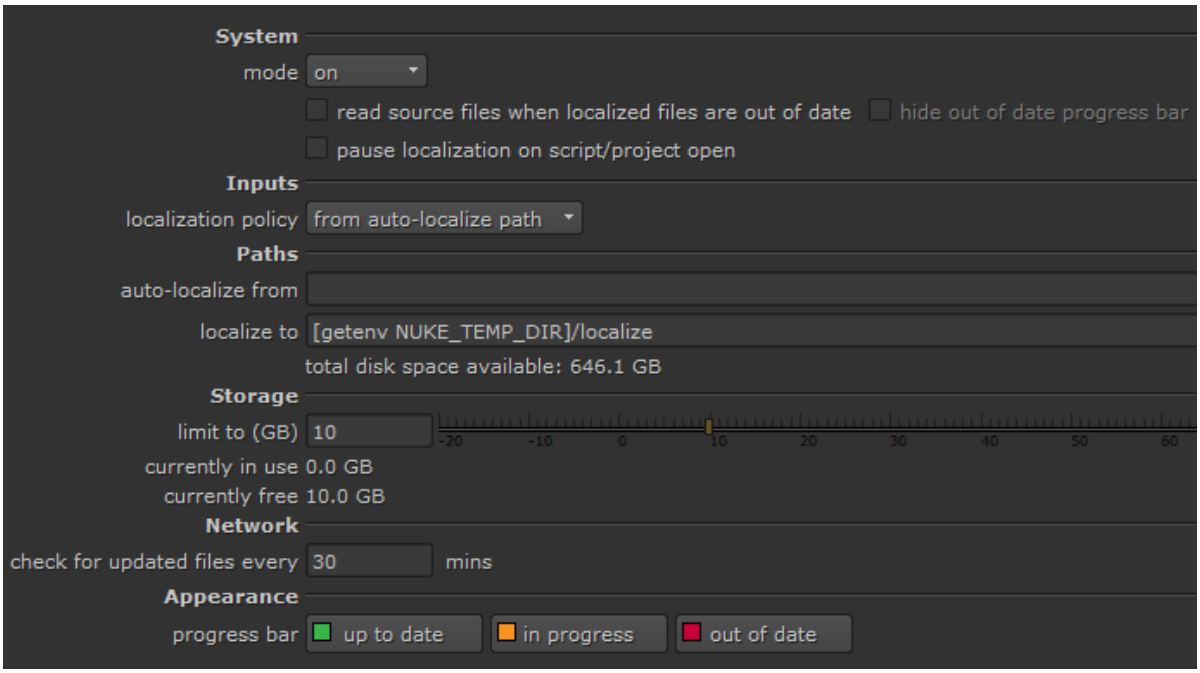
Nuke Studio has the facility to cache source clips locally, either individually or by setting an automatically localized folder (NUKE_TEMP_DIR/localize, by default), to help guarantee playback stability. Local caching is controlled initially in the **Preferences** dialog, then on a clip-by-clip basis.

Setting Localization Preferences

The **Preferences** control how Nuke Studio deals with new source clips as they are ingested, but does not affect existing source clips in the project. If you plan to localize source clips in your project, it's a good idea to

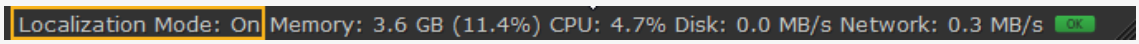
set the **Preferences** before ingesting your source clips.

1. Press **Shift**+**S** to open the **Preferences** dialog and navigate to **Performance** > **Localization**.



2. Set the required localization **mode:**
   - **on** - checks for updates to source clips for all localization policies, and localizes those files set to **On** or **From auto-localize path** automatically. **On demand** source clips must always be localized manually.
   - **manual** - checks for updates to source clips and prompts you to update them manually. Only those with the policy set to **off** are not updated.
   - **off** - no source clips are localized, regardless of the their localization policy.

Note: The current localization **mode** is displayed in the status bar at the bottom-right of the interface.



3. Set the default **localization policy** for new source clips using the dropdown:

Note: The **localization policy** for existing source clips in the project must be set individually. See Managing Localization for more information.

   - **on** - always localize source clips with this policy.
   - **from auto-localize path** - localize these source clips automatically if they reside in the **auto-localize from** directory.

- **on demand** - only localize these source clips when you manually update them. See Updating On Demand Clips for more information.
- **off** - never localize these source clips.

4. Enter a file path for **auto-localize from**, if required.

   Any files that reside in this directory are automatically cached when ingested into Nuke Studio, providing that the **Localization Policy** is set to **From auto-localize path**.



Note: Localization in both the timeline and Node Graph is paused during compositing Viewer playback so that performance is not affected.

5. Enter a file path for **localize to**. Leaving this field as the default creates a sub-directory in the **Temp Directory** as the local cache.



Note: On Windows, files saved to the **localize to** directory replace **** (double back slashes) and **:** (colon drive signifiers) with underscores so that the file path works as expected between operating systems. For example:

\\windowspath\to\my\network\file.dpx is saved as **__windowspath\to\my\network\file.dpx**
t:\my\network\path\file.dpx is saved as **t_\my\network\path\file.dpx**

6. Enter a value for **limit to (GB)** to control how much disk space is available in the cache directory.



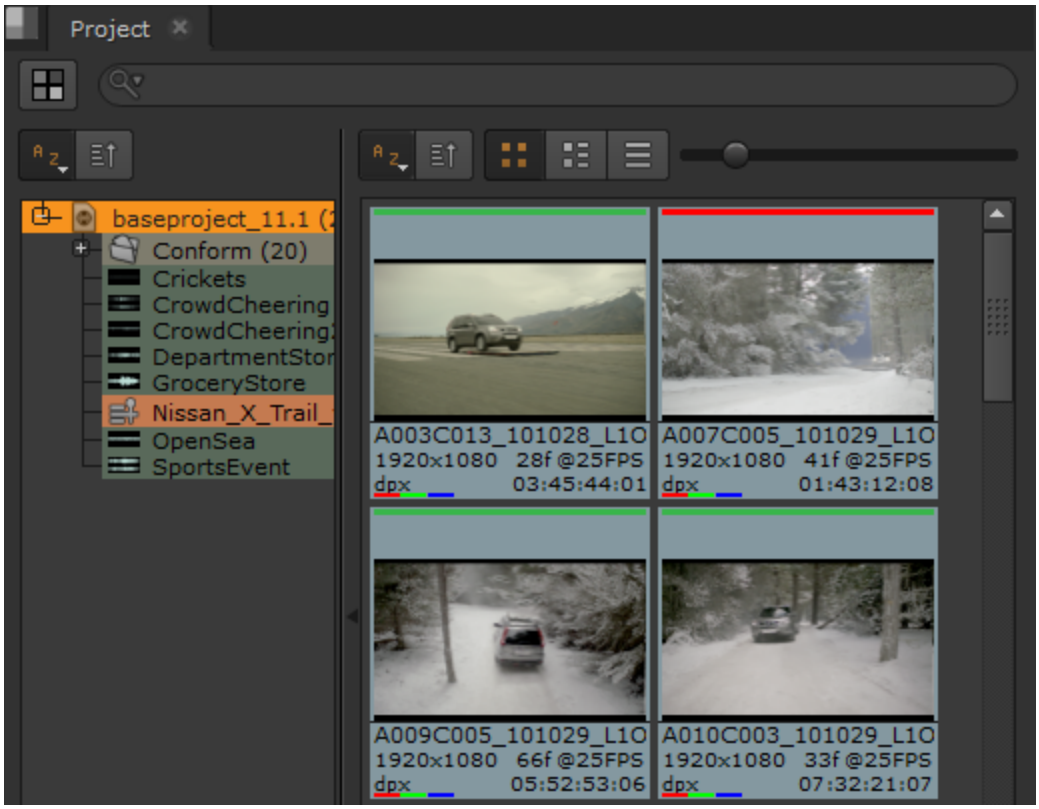
Note: Negative values in this field reserve the specified amount of space at all times. For example, -2 stops 2 GB of memory being used for caching.

7. You can specify the time interval (in minutes) before localized files are checked for updates using the **check for updated files every ## mins** control.

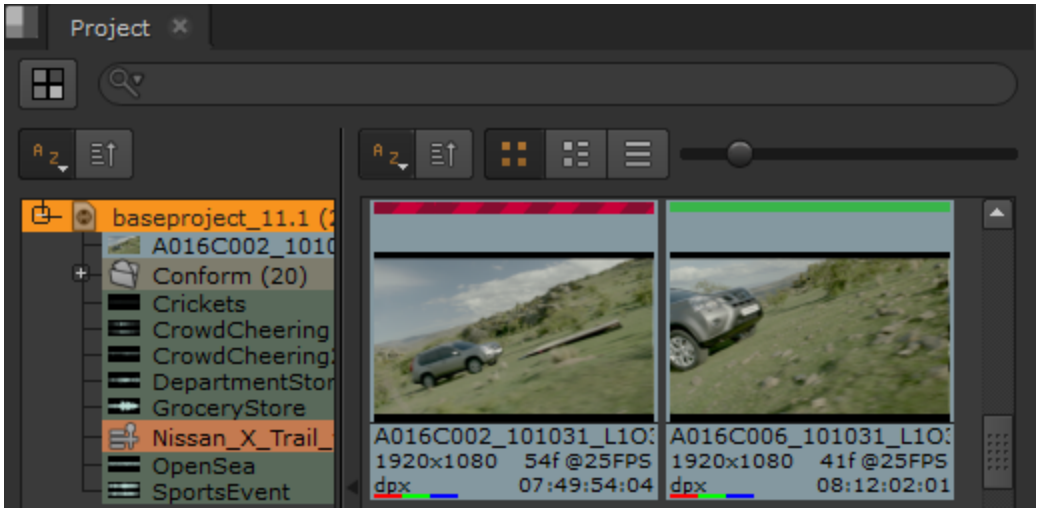
   The default setting checks for new versions of files every 30 minutes, but you can set this control to any value. Any files that have changed since the last update check are flagged red in the **Project** bin.



Tip: You can change the default localization indicators using the **Appearance** controls.



If **read source files when localized files are out of date** is enabled, source clips referencing cached files that have changed since they were localized revert to reading the source files. Source clips that are reading from the source files are marked with a striped red bar on the clip.



Enabling **hide out of date progress bar** hides the localization state of out of date files so that the source clip appears the same as regular clips.



Note: The out of date localized files are not discarded, disabling **read source files when localized files are out of date** picks up the out of date files instead of reading the source files.

Managing Localization

As well as the overall **Preferences** for when new source clips should be localized, you can set localization on a file-by-file basis. The **localization policy** for any existing source clips in your project must be managed individually.

Tip: If you find that localization is slow to copy files, you can increase the number of threads that Nuke uses to process jobs. Set the NUKE_LOCALIZATION_NUMWATCHERS environment variable to the number of threads you want to use. See Environment Variables for more information.

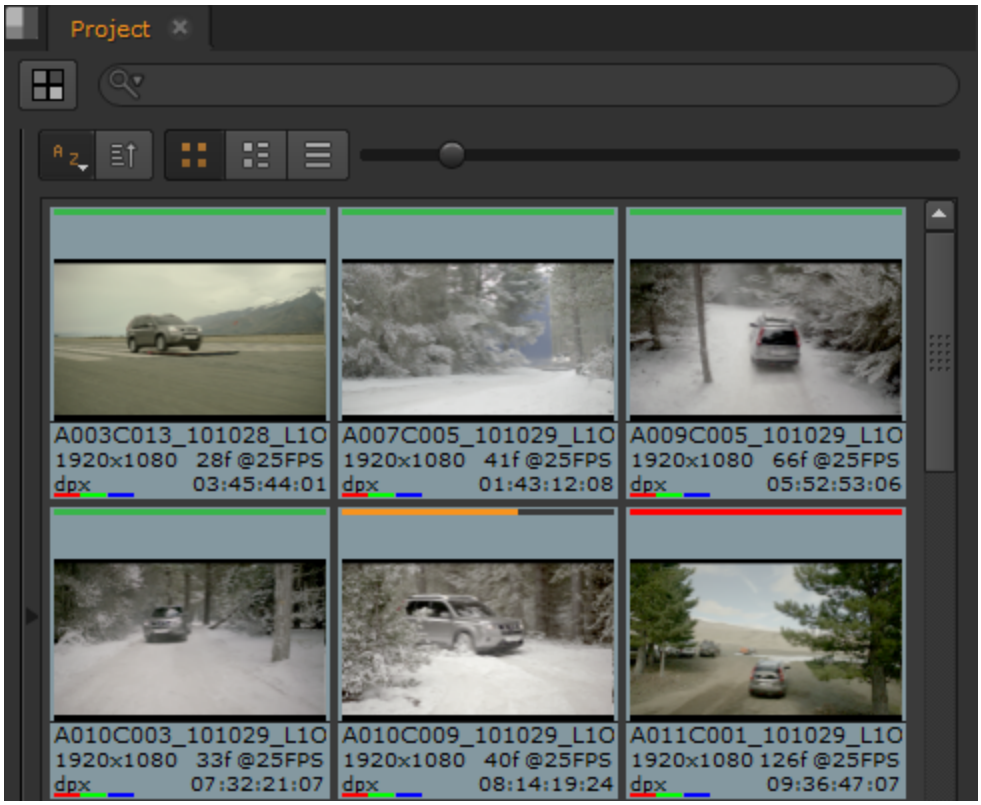
To set the localization policy for source clips:

1. Select the clip(s) in the bin view.
2. Right-click and select **Localization Policy** to display the available options:
   - **On** - the files are localized, regardless of location, as long as the **limit to (GB)** limit is not breached.
   - **From auto-localize path** - the files are localized if they reside in the **auto-localize from** directory, as long as the **limit to (GB)** limit is not breached.
   - **On Demand** - the files are only localized when you update them manually. See Updating On Demand Clips for more information.
   - **Off** - the files are never localized, regardless of location.

   As clips are localized, an amber progress bar displays in the thumbnail. Fully cached clips are marked with a green bar at the top of the thumbnail and out-of-date clips are marked in red.

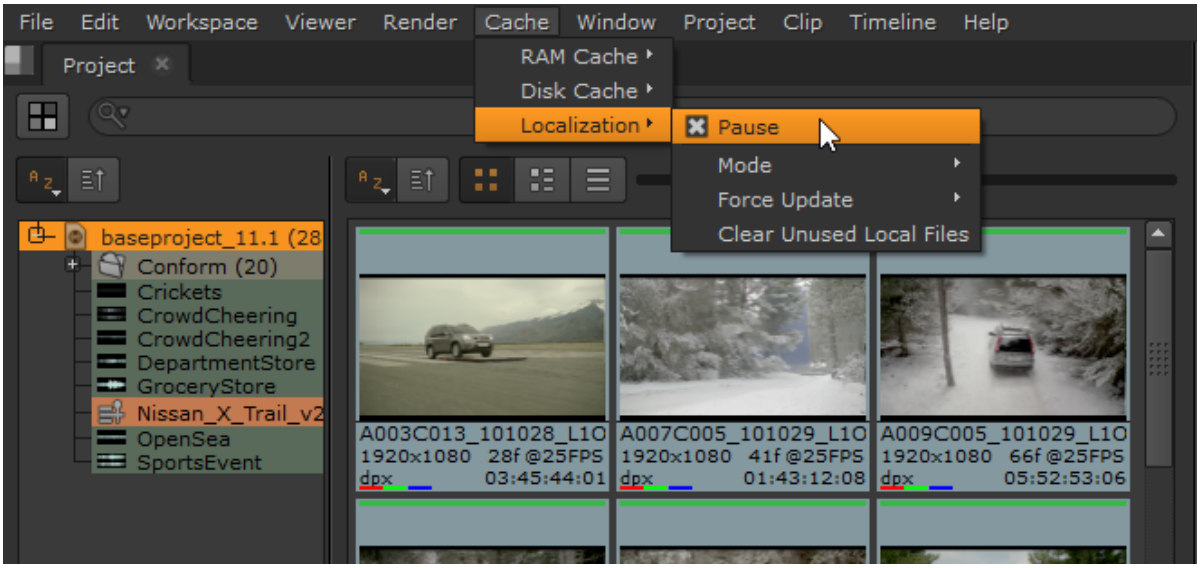


Note: Container formats, such as **.mov** and **.r3d**, do not display progress bars during localization. The source clip only shows the green indicator once the entire file is localized successfully.



Note: Nuke Studio also features a playback cache and timeline disk cache, allowing frames to be cached in RAM or disk. See Caching Frames in the Playback Cache and Caching Frames in the Disk Cache for more information.

3. If you need to pause localization temporarily, navigate to **Cache** > **Localization** and select **Pause**.



4. If you find that your cache is filling up regularly, you can:

- Increase the amount of available space for localization by raising the **limit to (GB)** preference,
- Navigate to **Cache** > **Localization** > **Clear Unused Local Files** (see Clearing Localized Files), or
- Manually clear files from the cache directory in NUKE_TEMP_DIR/localize, by default.

5. You can force Nuke Studio to check for updated source files, rather than waiting for the **check for updated files every** interval to expire, by selecting **Force Update** for **All** files, just **Selected** source clips, or **On Demand only**.



Note: Each file has its own update time, which is reset whenever the source files are checked.

The following table is a quick reference guide to when and how source clips are localized, where:

- **green** - clips are localized automatically.
- **amber** - clips are localized when updated manually.
- **red** - clips are not localized.

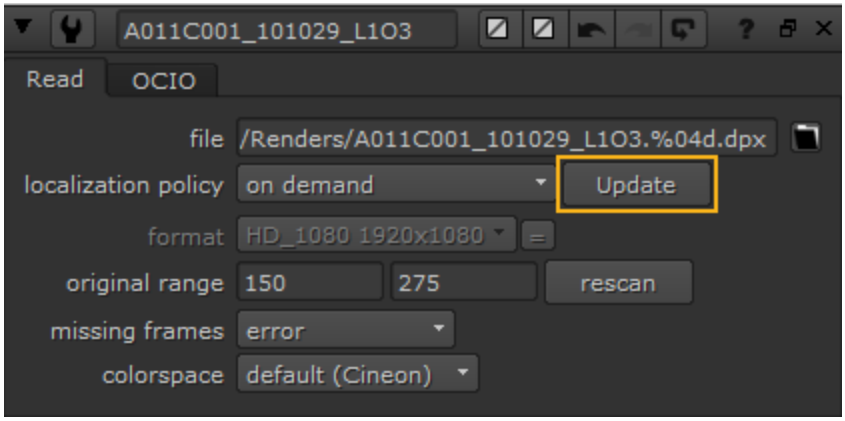
System Preference	Source Clip Preference			
	on	auto-path	on demand	off
on				
manual				
off				

Updating On Demand Clips

Source clips with their **localization policy** set to **on demand** are polled to check for updates at the interval set in the **Preferences**. If the file has changed, the source clip is marked with a red bar to show it is out of date.

To update a single **on demand** source clip:

1. Double-click the source clip in the **Project** bin to display its **Properties**.
2. Click **Update**.

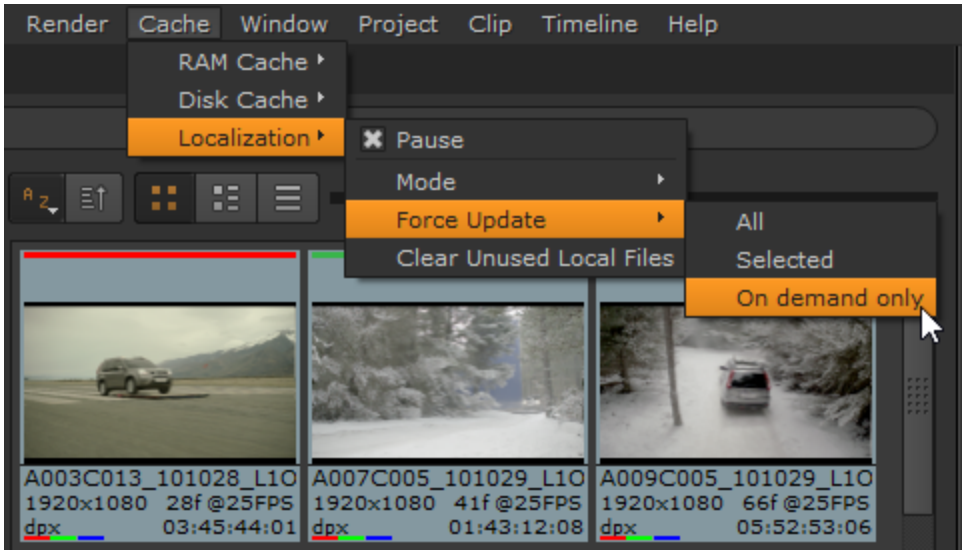


Note: If a Read node's localization policy is set to **on demand** and it hasn't been localized previously, clicking the **Update** button localizes the file.

The local copy of the source clip is updated from the remote source clip.

To update all **on demand** source clips:

1. Navigate to **Localization** > **Force Update**.
2. Click **On demand only**.

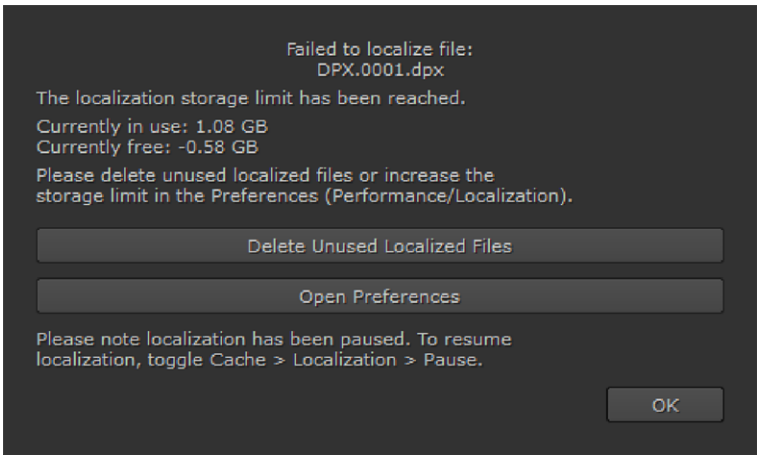


The local copy of all **on demand** source clips is updated from the remote source clip.

Clearing Localized Files

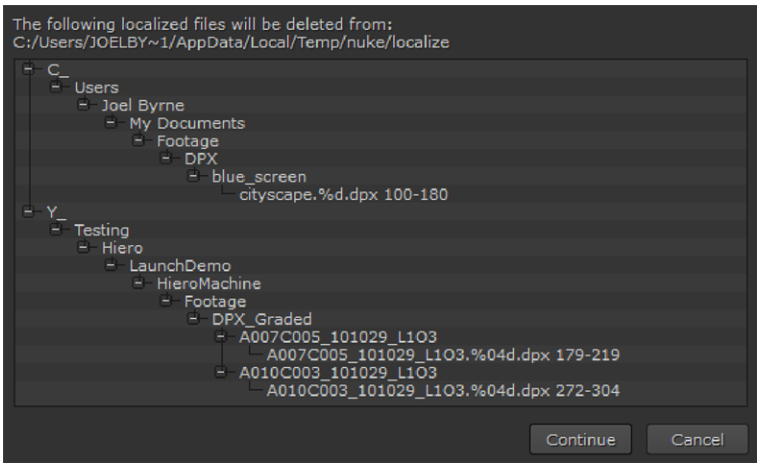
Localizing a large amount of files can fill up the localization cache quite quickly if you leave the **limit to (GB)** preference at the default 10 GB. When the cache runs out of space, a **Failed to Localize File** dialog displays

and localization pauses.



You can delete localized files by clicking **Delete Unused Local Files** (or by navigating to **Cache** > **Localization** > **Clear Unused Local Files**). Nuke Studio displays a dialog containing all the files that are marked for delete.

Tip: You can also open the **Preferences** from this dialog to adjust the localization behavior, such as increasing the **limit to (GB)** preference.



Click **Continue** to delete the localized files or **Cancel** to keep the cached files.

Using the Timeline Viewer

Nuke Studio's Timeline environment supports two distinct Viewer types: clip and sequence. This chapter describes the difference between the two and how to use them.

Clip Viewers, sometimes referred to as source Viewers, are marked with the  icon and deal exclusively with source clips. You can set In and Out points and apply tags to the Viewer, but the source clips are unaffected.

Sequence Viewers, also known as record Viewers, are marked with the  icon and deal with sequences and shots on the timeline. You can set In and Out points and apply tags here too, but you can also edit the shots on the timeline by trimming, retiming, and so on. See Timeline Editing Tools for more information.

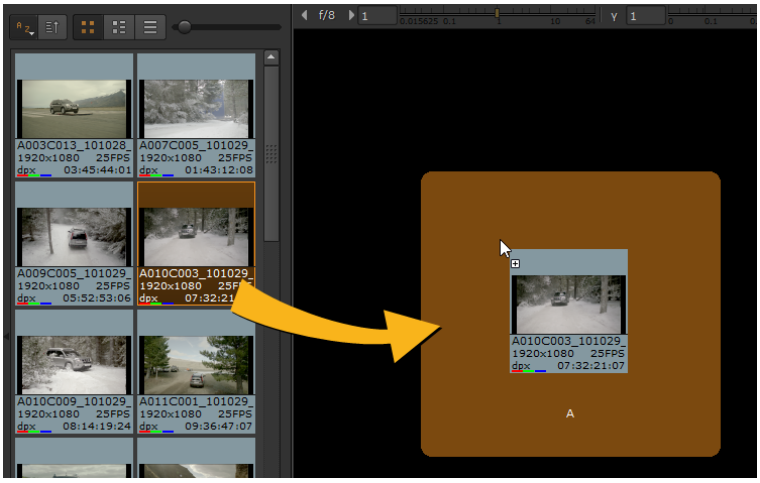
The **Editing** workspace combines both clip and sequence Viewers by default, enabling you to add source clips to the timeline using insert and overwrite functions. See Insert, Overwrite, and 3-Point Editing for more information.



To view your media in a Viewer, simply drag-and-drop a clip or sequence from the **Project** tab on to a Viewer input, or double-click the item to send it to the appropriate Viewer.

Note: The Viewer currently treats all alpha channels as premultiplied, which can result in the Viewer background being "added" to the image. If you're working with un-premultiplied images, set the Viewer background to **Black**.

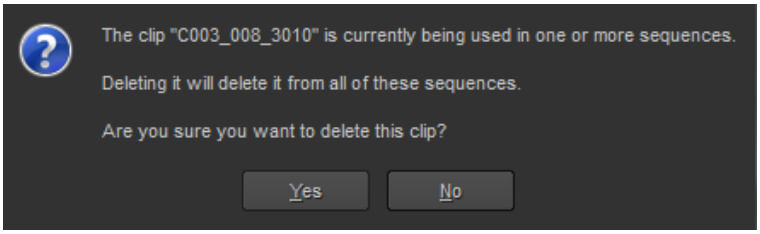
See Appendix A: Preferences for more information.



Deleting Media

To remove media from the bin view, select the clip(s) or bin and press **Backspace** or **Delete**.

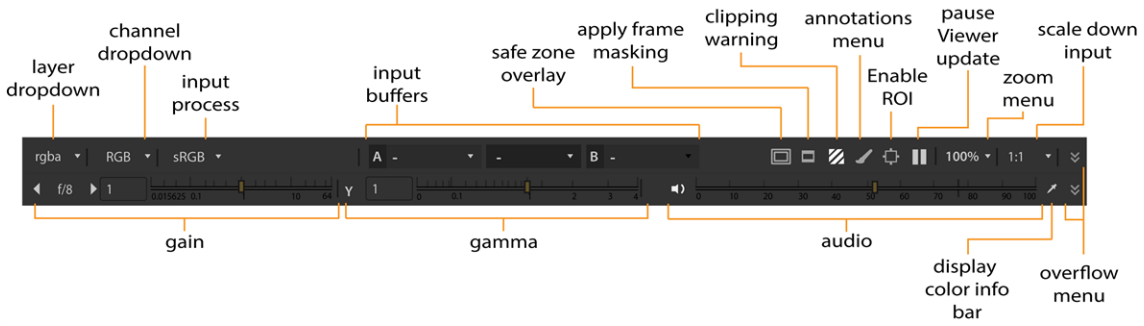
If any of the media is in use in a sequence, the following warning displays:



Click **Yes** to delete the media from the bin view, but bear in mind that all instances of the deleted media are removed from your current sequences.

Timeline Playback Tools

There are many useful tools at the top of the Viewer, some of which allow you to select channels, adjust saturation, gain and gamma, and zoom and scale down the image in the Viewer.

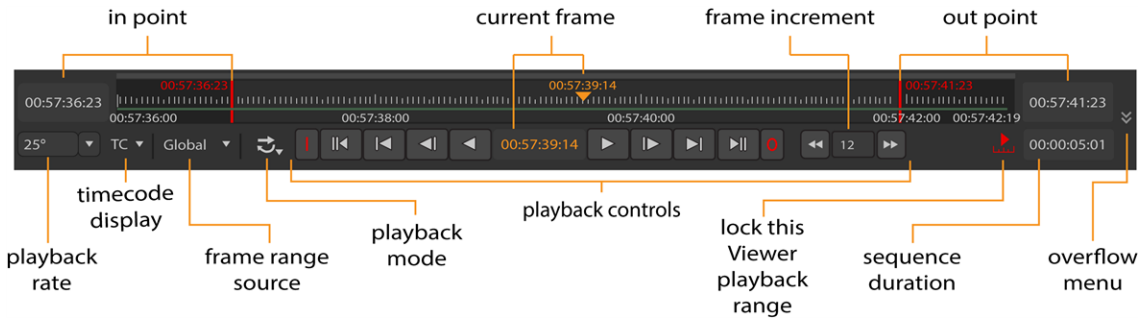


For more information about the tools above the Viewer, see Timeline Viewer Tools.

The tools at the bottom of the Viewer allow you to adjust the playback settings, including setting the frame range, selecting the playback mode, and locking the Viewer playback range.

Drag the orange marker along the timeline to quickly cue to a specific frame or timecode. The number of the current frame or timecode appears below the center of the timeline. You can also cue to a frame or timecode by typing its number directly into this field.

Tip: The current frame and in an out point fields accept simple mathematical functions, such as +/-20 to jump forward or backward 20 frames or +/-00002000 to jump forward or backward 20 seconds.



By default, Nuke Studio automatically adjusts the timeline of every Viewer window to show the frame range defined in your Project Settings. If no frame range is defined, the frame range of the first image you read in is used as the global frame range.

Viewer timeline controls also have a frame range source dropdown menu that you can use to define where the timeline gets its frame range from. You can set this menu to **Global**, **Input**, or **Custom**. **Global** is the default setting described above.

The **playback rate** field (frames-per-second) initially displays the project's playback speed. Nuke Studio attempts to maintain this speed throughout playback, although this adjusts depending on the resolution of the imagery and your hardware configuration.



Note: The asterisk (*) denotes the **Sequence** playback speed selected using the **Frame Rate** dropdown or, for new projects, the **Project Settings** > **Sequence** > **Frame Rate** dropdown.

Playback Controls

The **playback rate** field (frames-per-second) initially displays the project's playback speed. The Viewer attempts to maintain this speed throughout playback, although this adjusts depending on the resolution of the imagery and your hardware configuration.

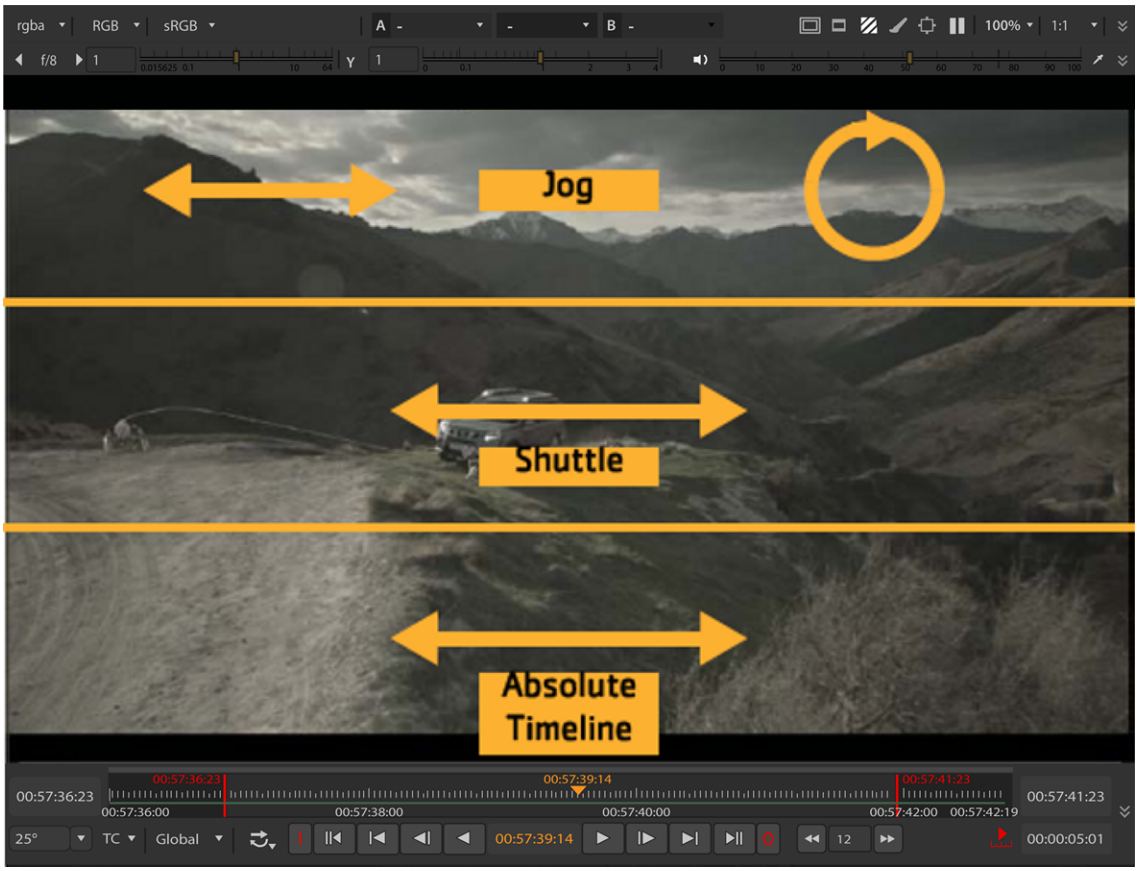
The following table lists the functions of the playback buttons:

Buttons	Functions
	The **Play backward** and **Play forward** buttons play the sequence backward or forward at the script's frame rate. When you press a play button, it toggles to a stop a button.
	The **Back 1 Frame** and **Forward 1 Frame** buttons cue the sequence to the previous or next frame.
	The **Previous keyframe** and **Next keyframe** buttons cue the sequence to the script's previous or next keyframe.
	The **First frame** and **Last frame** buttons cue the sequence to the first and last frame.
	The **Frame Increment** field allow you to specify the number of frames by which the Previous increment/Next increment buttons cue the sequence. This is set to 10 frames by default.

The **J**, **K**, and **L** keyboard shortcuts also control playback. The **K** keyboard shortcut is mapped to Pause/Play. **J** and **L** are mapped to backward and forward. Combinations are also supported:

- **K**+**J** - frame backward.
- **K**+**L** - frame forward.

- **K**+drag in the top third of the Viewer - standard jog controls. Dragging the cursor left and right moves the playhead backward and forward, frame-by-frame.
- The jog controls also detect rotary motion to jog through frames. Clockwise motion in the top third of the Viewer, while holding **K**, advances the playhead and anti-clockwise reverses the playhead.
- **K**+drag in the middle third of the Viewer - standard shuttle controls. Dragging the cursor left and right plays backward and forward, with increased frame rate toward the edges of the Viewer.
- **K**+drag in the bottom third of the Viewer - skips the playhead to absolute timeline position.



The **Playback Mode** button lets you control how many times and in what direction the Viewer plays back the sequence. Click the button to toggle between the following modes:

Button	Function
Repeat	Repeatedly plays the sequence in a loop.
Bounce	Repeatedly plays the image back and forth from head to tail.

Button	Function
Stop	Plays once through the section between in and out point and stops at the out point. If these are not marked, then it plays from the beginning to the end of the sequence.
Continue	Plays once from the beginning to the end of the sequence, ignoring any in and out points.

Timeline Viewer Tools

The Viewer has two sets of tools for manipulating your media: the Viewer tools and the playback tools. The Viewer tools, located at the top of the Timeline Viewer, are used to affect the mouse pointer as you move over the Viewer, and to select Viewer preferences:

Icon	Function	Description
rgba	Layers	Select the layer to output to the Viewer, for example **forward** motion vectors or **disparityL**. Only layers available in the clip are displayed - check the clip's thumbnail to see at a glance which layers are present:  - red color layer. - green color layer. - blue color layer. - alpha layer. - depth layer. - forward motion vector layers. - backward motion vector layers.

Icon	Function	Description
		- all other custom layers, such as disparity. **Note:** You can scroll through available layers using **PgUp** or **PgDn**.
	Channels	Select the channel(s) to output to the Viewer, for example **RGB**, single channel, **Alpha**, or **Luma**.
	Viewer color transform	Set the view transform used to display images in the Viewer, for example **sRGB** and **rec709**. **Note:** If you have specified an OpenColorIO configuration file in the **Preferences**, you may have more colorspace choices available.
A/B	Viewer Output	Click the A or B dropdown and select what you want to view. This can be selected tracks or tracks with selected tags. When both Viewer buffers contain an image, enable **wipe** to compare the two images. You can also use the center drop down to set the blend mode between images in the Viewer, for example **Onion Skin** or **Difference**, and the A/B buffer configuration.
	Guides	Enable or disable Viewer overlays: • **title safe** - any text intended for the audience should reside within this zone. • **action safe** - any visual elements intended for the audience should reside within this zone. • **format center** - adds a crosshair in the center of the format currently in the Viewer. • **Format** - adds a red, format-dependent box for the clip or sequence in the Viewer. Sequences support multi-format clips, see Viewing Multi-Format Timelines for more information.
	Mask	Enable or disable a range of Viewer masks, for example **16:9** or **1.85:1**.

Icon	Function	Description
	Clipping	Enable or disable Viewer warnings: • **No Warnings** - all clipping warnings are disabled. • **Exposure** - alerts you when the image is under (blue) or over (red) exposed.
	Annotations	Click to enable the Annotations tool bar. Annotations allow you draw and add text to clips in the Viewer. See Annotations for more information. **Note:** The Annotations button also controls existing annotation visibility.
	ROI	Click and drag to define a Region of Interest (ROI) in the Viewer. The scopes only display information within the ROI, when active.
	Pause playback caching	Pause or release Viewer playback caching, indicated by the green bar under the Viewer.
Scale		Set the scale applied to the clip in the Viewer, for example **25%**, **75%**, or **Fit**.
Image Quality		Set the Viewer image quality, for example **1:1**, **1:4**, or **1:16**. The default setting, **Auto**, resizes the image dependent on the Viewer zoom level, which may re-cache the image at a higher resolution. **Note:** Image quality, or proxy, for RED clips is dependent on the clip's **Decode Resolution** in the **Media** panel. For example, if you're viewing a 4K file and the **Decode Resolution** is set to **Half Premium**, a 1:1 proxy value is equal to 2K, 1:2 is equal to 1K, and so on.
	Non RT Playback	Sets the Viewer playback mode: • **Play All Frames** - the default setting, plays all frames in real-time (dependent on hardware). • **Skip Frames** - plays frames in real-time skipping where

Icon	Function	Description
		necessary to maintain the frame rate. • **Play All Frames, Buffering** - plays all frames by buffering and playing frames back as they become available.
	See through missing media	When disabled, any offline media on a timeline is treated as a blank clip so the Viewer cannot display the track underneath. This setting also applies to missing frames within a clip. **Note:** This control only affects the current Viewer.
	View	Select the Viewer display mode, for example **Audio and Video** or **Video Only**.
	Obey Alpha	Allows you to control the alpha channel independent of the Viewer **Blend Mode**. • **Enabled** - any alpha channel present in the image is treated as premultiplied transparency. • **Disabled** - the alpha channel is ignored.
	Audio latency	Sets the audio latency, in milliseconds, for the current Viewer only. Audio latency allows you to correct audio and video synchronization by changing the point at which audio playback starts. Positive values cause the audio track to start earlier in relation to the video track, and vice versa.
	Gain	Adjusts the gain applied to the linear input image before viewing, but doesn't affect your exported image.
	Gamma	Adjusts the gamma applied to the image after the viewing transform, but doesn't affect your exported image.
	Saturation	Adjusts the saturation applied to the linear input image before viewing, but doesn't affect your exported image.
	Mute / Audio	Click to mute audio output for the current Viewer or use the slider to control the audio output level.

Icon	Function	Description
		Tip: You can also control volume on a per track and per shot basis. See Audio and the Timeline for more information.
	Color Sample	Enable or disable the RGBA color information bar in the Viewer. **Note:** The **Color Sample** tool displays color information from the source file, not the colorspace selected in the Viewer color transform dropdown. See Working with Colorspaces for more information.

Using In and Out Markers

In and Out markers enable you to alter the duration of a shot to just the portions of the source or sequence that you require.

Tip: You can use the source clip/shot properties **original range** controls in similar way to In and Out points. See Setting Source Clip Ranges for more information.

When a clip containing In and Out points is added to a timeline, you can slip the clip around the markers to adjust the clip's output. See Timeline Editing Tools for more information. You can also use In and Out points to export certain portions of a clip or sequence. See Transcoding for more information.

To set In and Out markers:

1. Right-click on the required clip or sequence and select **Open In** > **Timeline View**.

   Tip: Source clips opened in the timeline view have a purple background in the timeline.

2. Move the playhead to the location of the In point and press **I** on your keyboard.

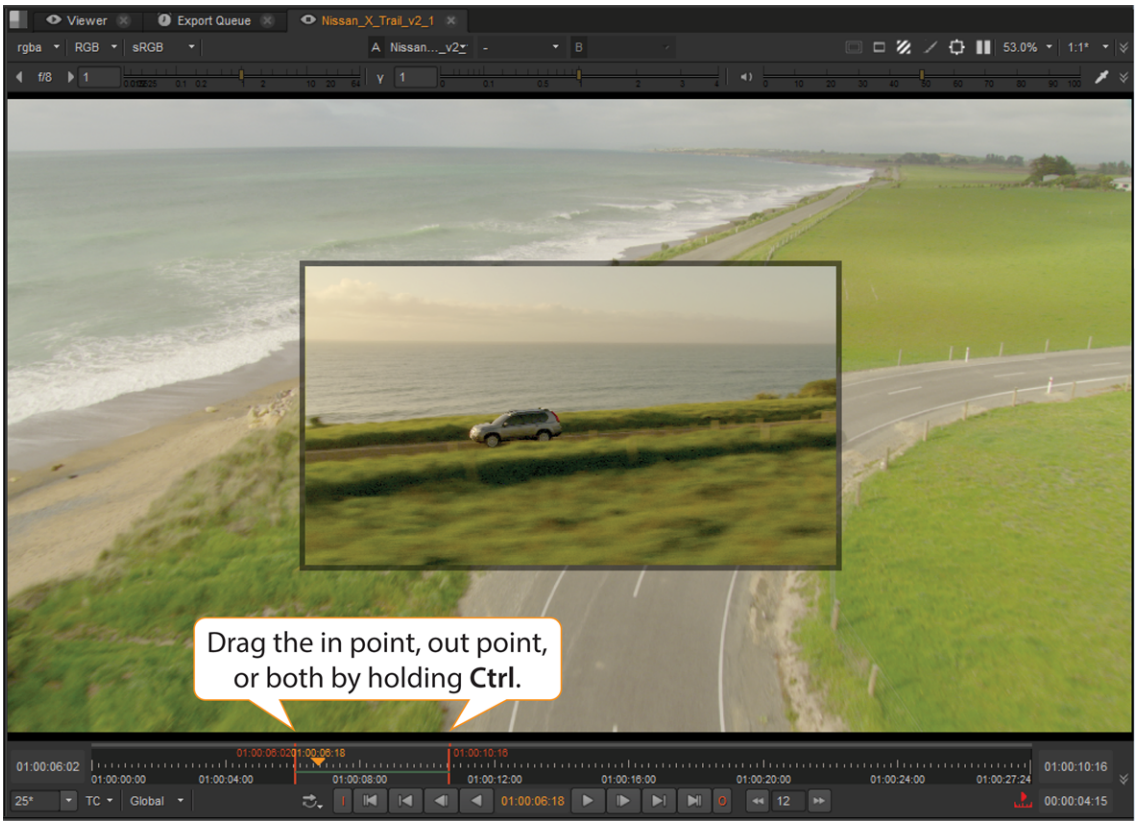
   The In point is marked by the In tab and the time is recorded in the playback controls.

3. Move the playhead to the location of the Out point and press **O** on your keyboard.

   The Out point is marked by the Out tab and the time is recorded in the playback controls.

Note: You can also set markers by navigating to **View** > **Mark In** or **Mark Out**, by using the **Timeline** menu to **Mark Selection** or **Mark Clip** dependent on clip selections on the timeline, or by right-clicking a shots and selecting **Open In** > **Viewer**.

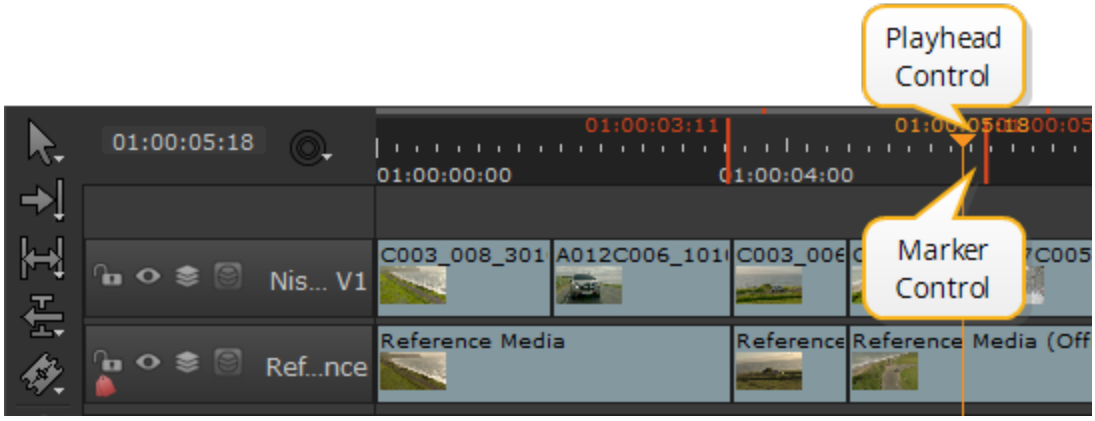
Click and drag the markers to adjust their position, or hold **Ctrl**/**Cmd** to move both markers at once, retaining their relative positions. A Viewer preview shows the current frame for the selected marker(s) and a timecode/frame popup helps to set the new position.



Clear the markers from your clip by navigating to **View** > **ClearIn Point** (Alt+I) and **Clear OutPoint** (Alt+O). The markers are removed completely, but you can reapply them by repositioning the playhead and pressing **I** or **O**.

Tip: You can also press **Alt**+**U** to remove both markers at once.

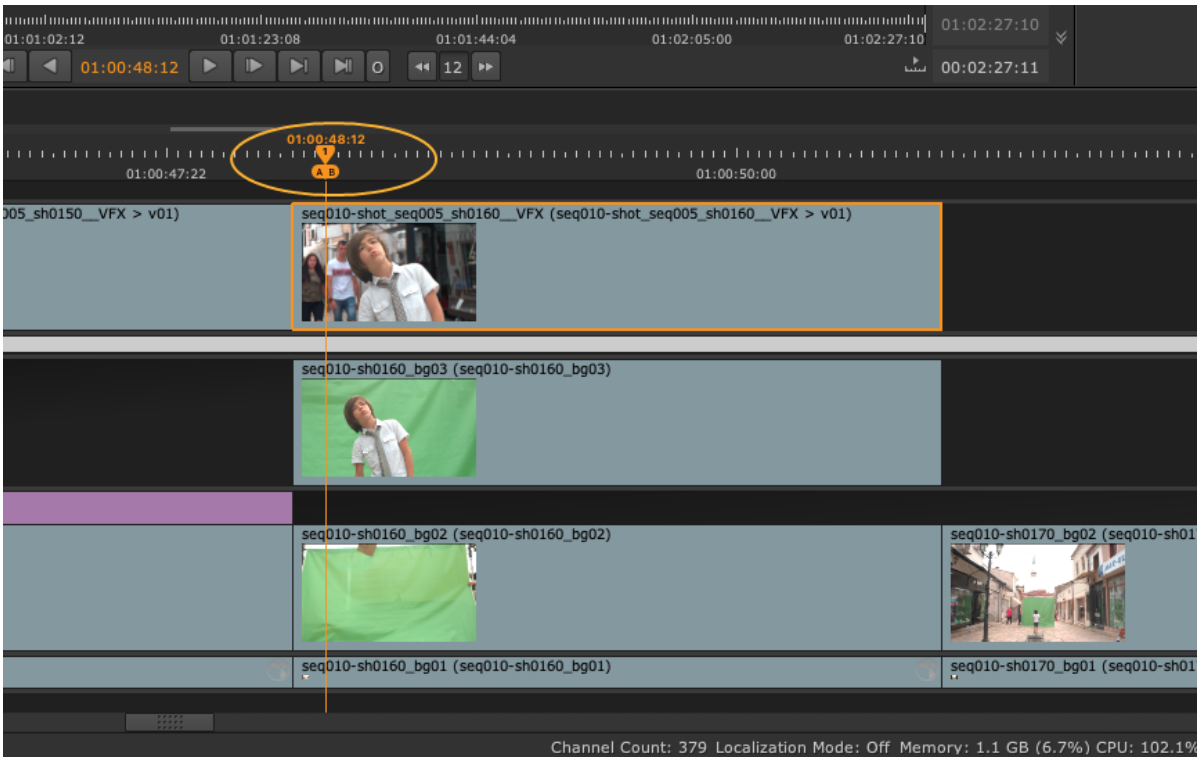
When the playhead is positioned near In and Out markers, the top half of the timecode scale controls the playhead and bottom half controls the markers.



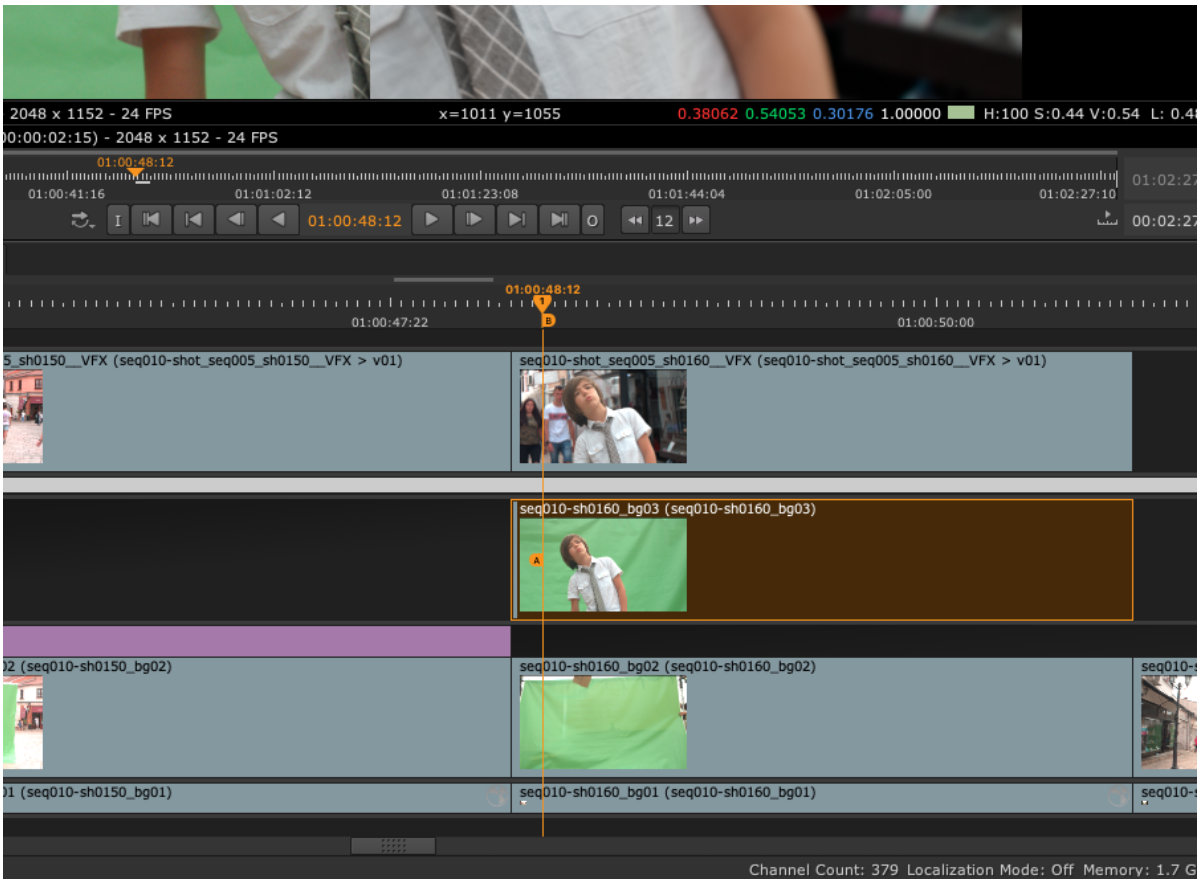
Playhead A/B Indicators

The timeline playhead indicates what is currently displayed in the Viewer using a number of different display markers. Swapping tracks between buffers automatically updates the playhead so that you always know which tracks you are looking at in the Viewer.

- Selecting **all tracks** in the current Viewer buffer displays an 'AB' at the top of the playhead to indicate that you're viewing all tracks in the sequence from top to bottom.

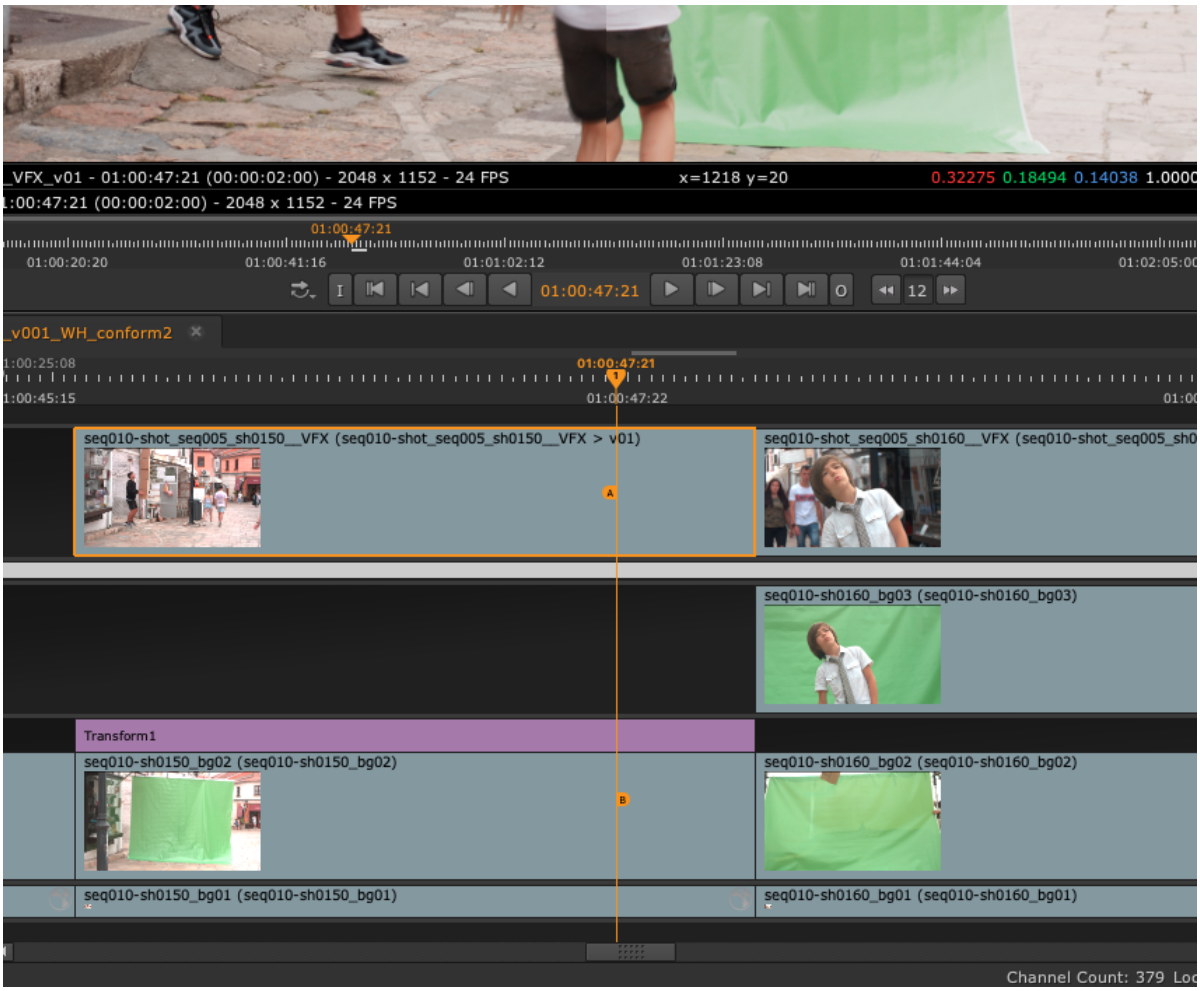


- Selecting a track for the current compare Viewer buffer displays your A or B tab at the playhead in the selected track.



- Selecting different tracks to the Viewer's A/B buffers draws A and B tabs on the playhead in the selected tracks.

> **Tip:** Use **A** or **B** hotkeys whilst selecting shots to quickly assign Compare viewers.



Using In and Out Markers

In and Out markers enable you to alter the duration of a clip to just the portions of the source or sequence that you require.

Tip: You can use the source clip/shot properties **original range** controls in similar way to In and Out points. See Setting Source Clip Ranges for more information.

When a clip containing In and Out points is added to a timeline, you can slip the clip around the markers to adjust the clip's output. See Timeline Editing Tools for more information. You can also use In and Out points to export certain portions of a clip or sequence. See Transcoding for more information.

To set In and Out markers:

1. Right-click on the required clip or sequence and select **Open In** > **Timeline View**.

Tip: Source clips opened in the timeline view have a purple background in the timeline.

2. Move the playhead to the location of the In point and press **I** on your keyboard.
   The In point is marked by the In tab and the time is recorded in the playback controls.
3. Move the playhead to the location of the Out point and press **O** on your keyboard.
   The Out point is marked by the Out tab and the time is recorded in the playback controls.

Note: You can also set markers by navigating to **View** > **Mark In** or **Mark Out**, by using the **Timeline** menu to **Mark Selection** or **Mark Clip** dependent on clip selections on the timeline, or by right-clicking a shots and selecting **Open In** > **Viewer**.

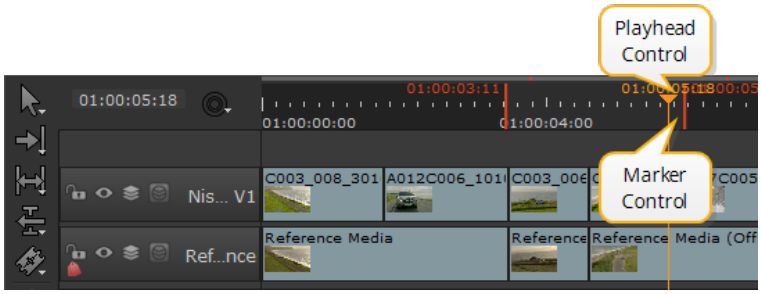
Click and drag the markers to adjust their position, or hold **Ctrl**/**Cmd** to move both markers at once, retaining their relative positions. A Viewer preview shows the current frame for the selected marker(s) and a timecode/frame popup helps to set the new position.



Clear the markers from your clip by navigating to **View** > **Clear In Point** (Alt+I) and **Clear Out Point** (Alt+O). The markers are removed completely, but you can reapply them by repositioning the playhead and pressing **I** or **O**.

Tip: You can also press **Alt**+**U** to remove both markers at once.

When the playhead is positioned near In and Out markers, the top half of the timecode scale controls the playhead and bottom half controls the markers.

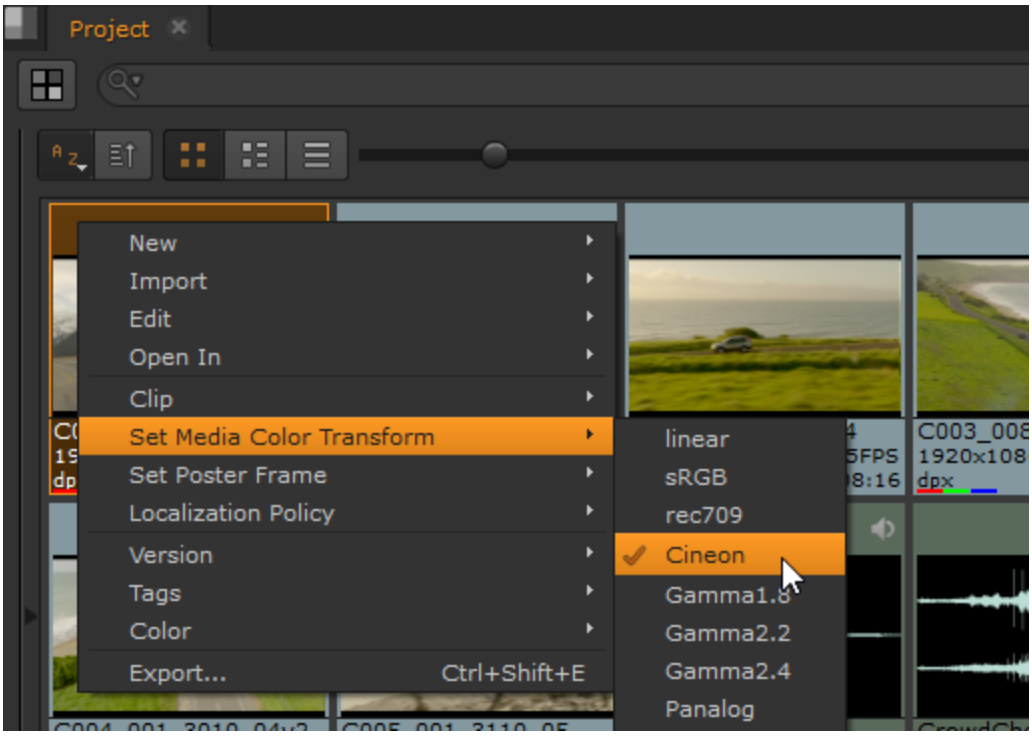


Working with Colorspaces

Colorspace changes are applicable to clips in bins and shots, as well as in the Viewer using the **Media** tab.

To apply colorspace changes to clips in bins:

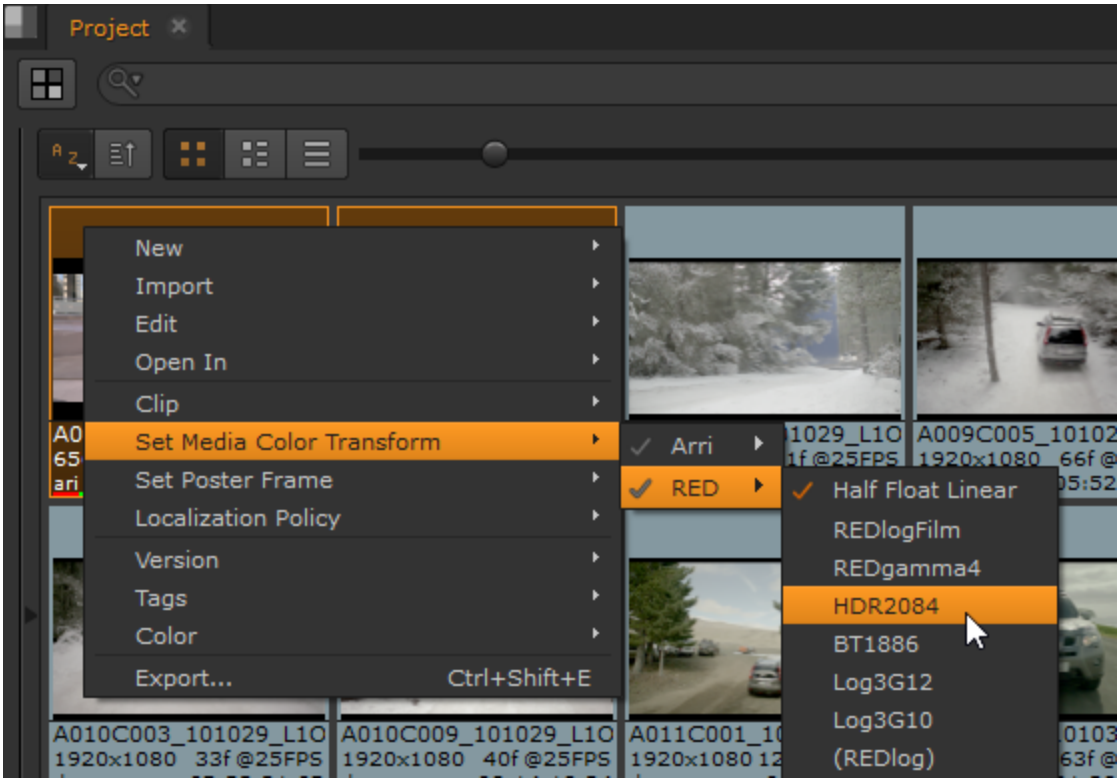
1. Select the clip or clips in the bin view.
2. Right-click a selected clip and navigate to **Set Media Color Transform**.



   The current colorspace is highlighted with a tick mark.
3. Select the colorspace to apply to the clip selection.

Note: Only colorspaces applicable to the selection are displayed. For example, **REDspace** and **LogC - CameraNative** are only available for R3D and ARRIRAW clips, respectively.

4. Selecting multiple formats supporting different colorspaces, for example R3Ds and ARRIRAW, breaks the available LUTs into sub-menus:



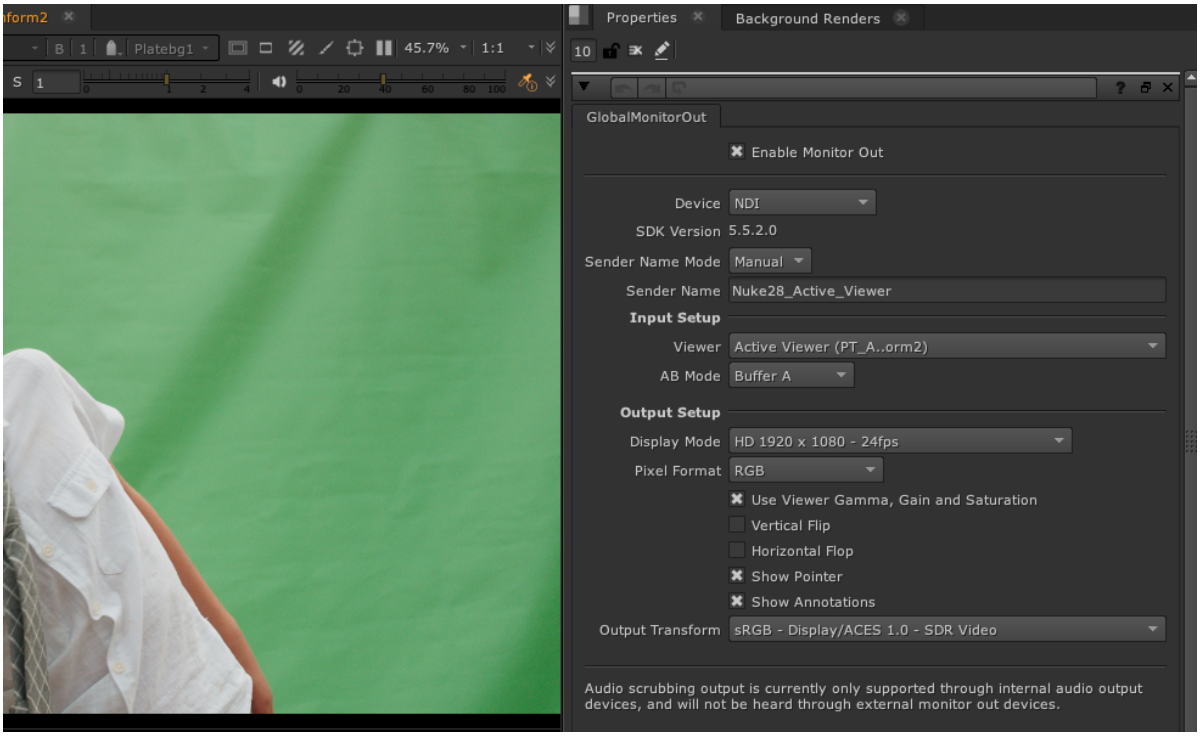
To apply colorspace changes to shots:

1. Select the item(s) on the timeline.
2. Right-click a selected item and navigate to **Set Media Color Transform**.
3. Select the colorspace to apply to the selection.

Monitor Output in Nuke Studio (NDI - SDI/HDMI)

The Monitor Out feature allows you to preview a timeline Viewer image in a floating window, NDI session, or on an external projector/video monitor to check the final result, including the correct output transform and aspect ratio.

A dedicated **Viewer Monitor Out** workspace provides easy access to the monitor out strip and its controls, but you can also access the controls through the ViewerMonitorOut in the node **Properties** panel.



Nuke Studio supports internal monitor out using a Floating Window, NDI sessions, and dedicated AJA or Blackmagic cards, selected from the **Device** dropdown in the ViewerMonitorOut's **Properties**:

- **Floating Window** - Nuke Studio's native monitor out uses a floating monitor and does not require additional hardware. Mac users can also take advantage of additional HDR technology including an extended sRGBf LUT. See Working with HDR Images for more information.
- **NDI** - A network protocol that enables video and metadata signals to be sent over standard IP networks in real-time. This option allows you to easily share a stream of the Viewer output to anyone else on the same network.

Note: To view the NDI stream, you need the NDI Studio Monitor or NDI Video Monitor application installed. See Streaming Viewer Output Over the Internet with NDI® for more information.

- **Dedicated AJA or Blackmagic Design cards** - Nuke Studio supports various AJA and Blackmagic devices that can be selected from the **Device** dropdown after installation. See AJA and Blackmagic Design Monitor Out Devices for more information on supported hardware and drivers.

Note: Audio scrubbing is not currently available through monitor out cards. Audio scrubbing is only supported through internal audio output devices.

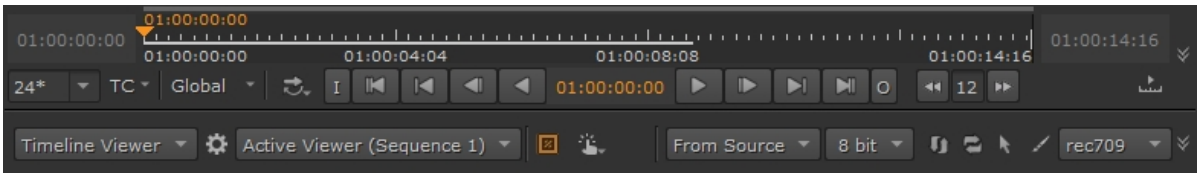
Nuke Studio also supports monitor out through VR headsets. See Reviewing on VR Headsets (HMD) for information on supported headsets.

Enabling Nuke Studio Monitor Out

Nuke Studio's monitor out uses a floating monitor, NDI session, or additional hardware from AJA or Blackmagic to display monitor out. The floating monitor allows Mac users to take advantage of additional HDR technology including an extended sRGBf LUT. See Working with HDR Images for more information.

1. Navigate to **Workspace** > **Viewer Monitor Out**.

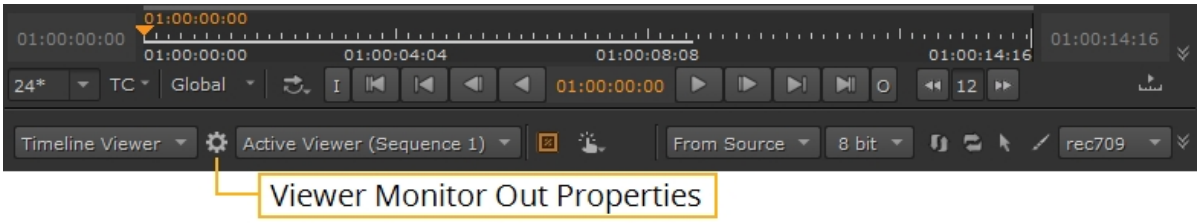
   The **Viewer Monitor Out** workspace is displayed with the **Monitor Out** toolbar under the Viewer by default.



   If you're working with multi-view footage in **Interactive** buffer mode, the monitor shows the output from the selected Viewer including the A/B player mode currently in use, such as **wipe** or **difference**. In **Active**, **Buffer A**, or **Buffer B** mode, additional controls display to determine the stereo mode and view to output. See Stereoscopic Scripts or Stereoscopic and Multi-View Projects for more information.



2. Click the cog icon in the **Viewer Monitor Out** panel to display the **Properties** panel.



3. Select the device you want to use from the output **Device** dropdown. All available devices are automatically detected and listed in this menu, along with the default options:

   • **Floating Window** - opens a floating monitor window, without the need for a monitor device and card. This is intended for full-screen use without displaying any of the interface.

Tip: Disable **Minimize with Application** to allow the floating window to remain visible when Nuke is minimized.

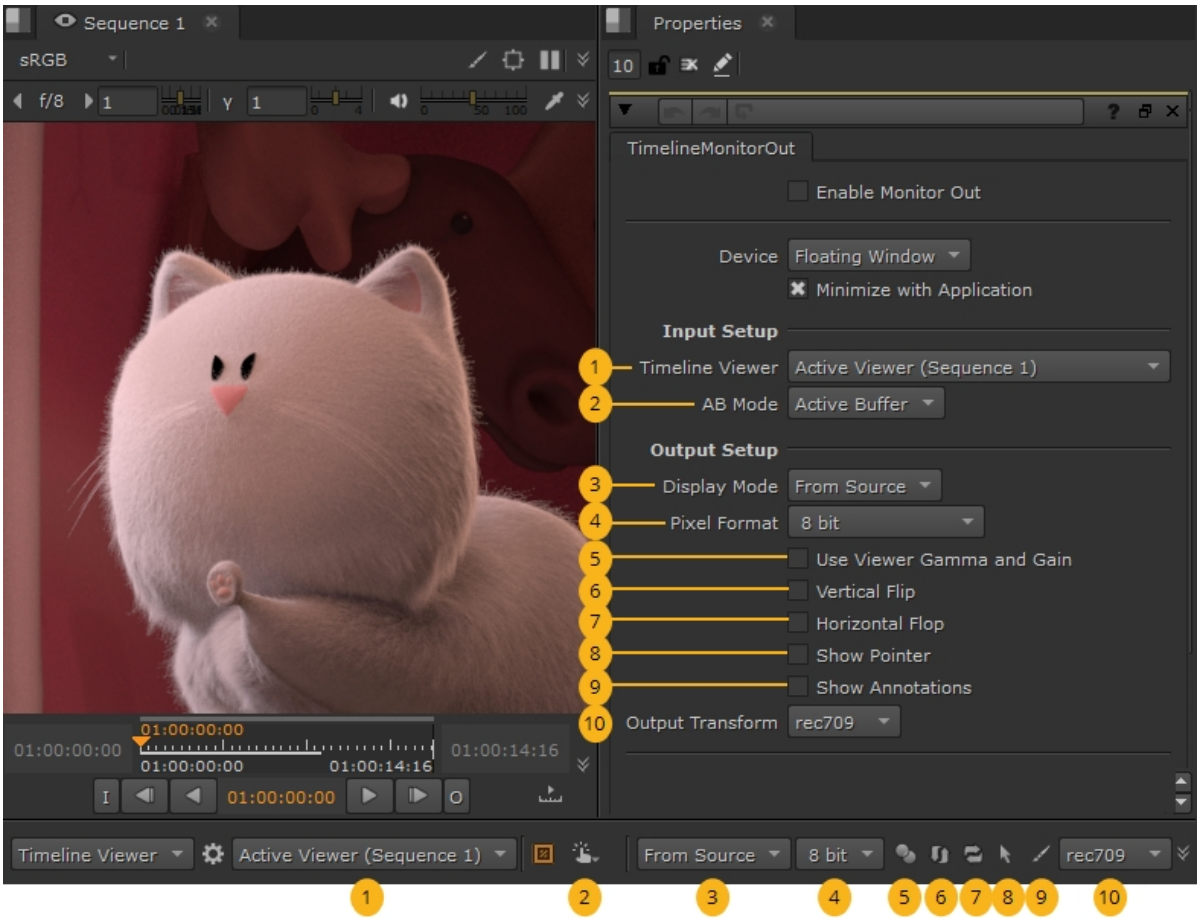
   • **NDI** - A network protocol that enables video and metadata signals to be sent over standard IP networks in real-time. This option allows you to easily share a stream of the Viewer output to anyone else on the network.



Note: To view the NDI stream, you need the NDI Studio Monitor or NDI Video Monitor application installed. See Streaming Viewer Output Over the Internet with NDI® for more information.

   • **Installed Cards** - transfers the Viewer contents to the selected card and displays SDK and driver information associated with the card.

   The **Viewer Monitor Out** and **Properties** panel duplicate some controls for convenience.



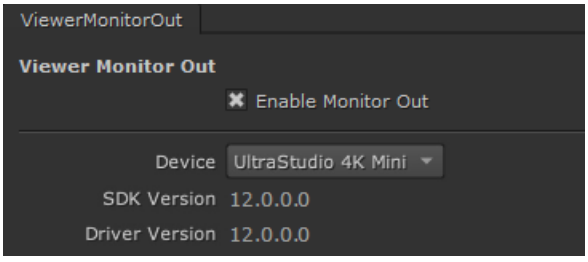
1	Viewer	Select the **Viewer** to feed to the output monitor using the source viewer dropdown. Selecting **Active Viewer** always displays the Viewer that is currently selected in the workspace.
2		Select the view mode using the **AB Mode** dropdown: • **Active Buffer** - the monitor always shows the output from the buffer currently in use by the selected Viewer. • **Buffer A** - the monitor always shows the output from the selected Viewer buffer A. • **Buffer B** - the monitor always shows the output from the selected Viewer buffer B. • **Interactive** - the monitor shows the output from the selected Viewer including the A/B player mode currently in use, such as **wipe** or **difference**.
3	Display Mode	Select the resolution output from the monitor card currently in use. The resolutions available depend on the card manufacturer.

		Note: Floating monitor out always uses the resolution of the Viewer in use.
4	Pixel Format	Select the pixel format output from the monitor card currently in use. The formats available depend on the card manufacturer.
5		Click the button or enable **Use Viewer Gamma, Gain and Saturation** to apply the active Viewer's filtering, gamma, gain and saturation to the monitor out.
6		Click the button or enable **Vertical Flip** to flip the output vertically. This control is handy if you're using a ceiling mounted projector.
7		Click the button or enable **Horizontal Flip** to flop the output horizontally.
8		Click to enable the mouse pointer position on the monitor. You can control the size of the cursor on the monitor in the **Preferences** under **Panels** > **Viewer (Monitor Out)** > **Mouse Pointer Scale**.
9		Click to display annotations active in the selected timeline Viewer on the monitor for review purposes. See Annotations for more information.
10	Output Transform	Select the color transform to apply to the image. If you've specified an OCIO configuration file in the preferences, these custom LUTs are also applicable.

AJA and Blackmagic Design Monitor Out Devices

Our monitor out architecture interfaces directly with the AJA and BlackMagic Design device drivers, which are unified across their respective hardware lines, meaning all current supported cards for the versions detailed in Third-Party Libraries and Fonts should work.

You can select installed cards from the MonitorOut's **Properties** panel using the **Device** control. The SDK and driver versions for the selected card are displayed for convenience.



We've tested the following AJA and Blackmagic Design hardware:

AJA Card:	KONA 4	KONA 5	KONA iO4k+	T-Tap Pro
See https://www.aja.com/family/desktop-io and https://www.aja.com/products/t-tap-pro#techspecs for more details on supported formats.				
Connection				
BNC	●	●	●	●
HDMI	●	●	●	●
Stereoscopic Support				
	Yes	Yes	No	No
Platforms				
	Win, Mac, Linux	Win, Mac, Linux	Win, Mac, Linux	Win, Mac
Drivers				
	Driver 17.5 or later, available here: • **Windows**: https://www.aja.com/support/item-download/10199/AJA-Software-Installer_Windows_v17.6_Release.zip • **macOS**: https://www.aja.com/support/item-download/10183/AJA-Software-Installer_macOS_v17.6_Release.zip • **Linux**: https://www.aja.com/support/item-download/10191/AJA-Software-Installer_Linux-Ubuntu_v17.6_Release.zip			

Blackmagic Card:	UltraStudio 4K Mini	4K Extreme	4K Extreme 12G	8K Pro
See https://www.blackmagicdesign.com/uk/products/ for more details on supported formats per card.				
Connection				

Blackmagic Card:	UltraStudio 4K Mini	4K Extreme	4K Extreme 12G	8K Pro
BNC	●	●	●	
HDMI			●	●
Stereoscopic Support				
	No	No	Yes	No
Platforms				
	Win, Mac, Linux	Win, Mac, Linux	Win, Mac, Linux	Win, Mac, Linux
Drivers				
	Driver 14.4 or later, available here: • **Windows**: https://www.blackmagicdesign.com/support/download/fd605bc94cd4424da637483c41165995/Windows • **macOS**: https://www.blackmagicdesign.com/support/download/fd605bc94cd4424da637483c41165995/Mac%20OS%20X • **Linux**: https://www.blackmagicdesign.com/support/download/fd605bc94cd4424da637483c41165995/Linux			

Blackmagic Card:	DeckLink Studio 4K	DeckLink 4K Extreme	DeckLink 4K Extreme 12G
See https://www.blackmagicdesign.com/uk/products/ for more details on supported formats per card.			
Interface			
BNC	●	●	●
HDMI	●	●	●
Stereoscopic Support			
	Yes	Yes	Yes

Blackmagic Card:	DeckLink Studio 4K	DeckLink 4K Extreme	DeckLink 4K Extreme 12G
	(Both views through one output, so the **Full Resolution** option is not available.)		
Platforms			
	Win, Mac, Linux	Win, Mac, Linux	Win, Mac, Linux
Drivers			
	Driver 14.4 or later, available here: • **Windows**: https://www.blackmagicdesign.com/support/download/fd605bc94cd4424da637483c41165995/Windows • **macOS**: https://www.blackmagicdesign.com/support/download/fd605bc94cd4424da637483c41165995/Mac%20OS%20X • **Linux**: https://www.blackmagicdesign.com/support/download/fd605bc94cd4424da637483c41165995/Linux		

Some monitor out cards allow you to extend or mirror your desktop so that the user interface is visible on the monitor. Please refer to your card's documentation for more information.

Working with HDR Images

Nuke handles HDR (high dynamic range) images in the same way as any other image, and so manipulating HDR footage in Nuke does not compromise the range of data available. Nuke can read in and out HDR with associated metadata and color primaries.

In the Read and Write nodes **MOV Options**, for the **YCbCr Matrix**, you can choose between **Rec 601**, **Rec 709**, **Rec 2020**, or **default** (which will be selected based on the metadata). **Rec 2020** can be used for HDR images, for more accurate color conversion.

Quicktime Color Metadata Control

If **default** is selected, movReader reads the nclc values, including the color matrix, from the container metadata (the colr atom), and uses this information to suggest a default value for the **YCbCr Matrix** knob.

When using the Write node, under **Advanced**, there are **Metadata Export** knobs for **Color Primaries** and **Transfer Function** , allowing for additional options.

- **Color Primaries** specifies the color primaries metadata to be written to the MOV file. This defines the XYZ coordinates of the RGB primaries and white point. Auto - Uses legacy colorspace assignment based on content resolution.
- **Transfer Function** specifies the transfer function (gamma) metadata to be written to the MOV file. This describes how RGB values are converted to and from linear light. Auto - Uses legacy colorspace assignment based on content resolution.

Nuke, Nuke Studio, and Hiero support viewing HDR output through approved monitors and Monitor Output devices such as AJA, Blackmagic Design, and Bluefish SDI/HDMI cards. See AJA and Blackmagic Design Monitor Out Devices for more details on supported cards. The following table contains details of the output devices we've tested and approved.

Device	Operating System	Connection
Dell UP3221Q	Linux and Windows	HDMI
	macOS (Catalina 10.15.x, or later)	Thunderbolt
LG EP950 and BP95E	Linux and Windows	HDMI
	macOS (Catalina 10.15.x, or later)	USB-C DisplayPort Alt Mode
Apple XDR	macOS (Catalina 10.15.x, or later)	Thunderbolt
EIZO CG3146, CG319X, CG279X, CG2700S, CG2700X, CS2740	Linux and Windows	HDMI, Display Port
	macOS (Catalina 10.15.x, or later)	USB-C Alt Mode

EDR Displays - Mac Only (Beta)

If you're running Nuke on macOS Catalina or later and have a compatible screen capable of displaying values above 1, you can view your projects with HDR luminance ranges using **sRGBf** in OCIO Color Management. You can also display images in the P3 gamut, giving more accurate color on wide gamut displays.



Note: This feature requires a 2019 Mac Pro running macOS 10.15, or later, and a suitable display or a current generation iMac Pro, iMac or MacBook Pro with an HDR screen.

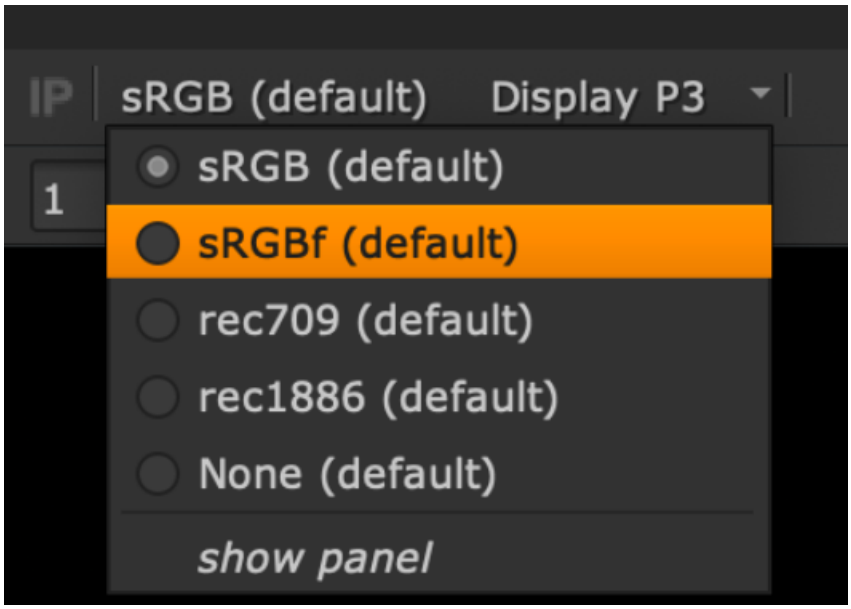
1. On an Apple XDR, set the profile of your display to one of the HDR options in the OS System Display Preferences:
   - Pro Display XDR (P3-1600 nits)
   - Apple Display (P3-500 nits)
   - HDR Video (P3-ST2084) - 1000 nits

   On other monitors, enable the **High Dynamic Range** checkbox.
2. In the **Preferences**, open the **Color Management** > **HDR** section and check **Enable macOS HDR Color Profile (Display P3) (Beta)**.

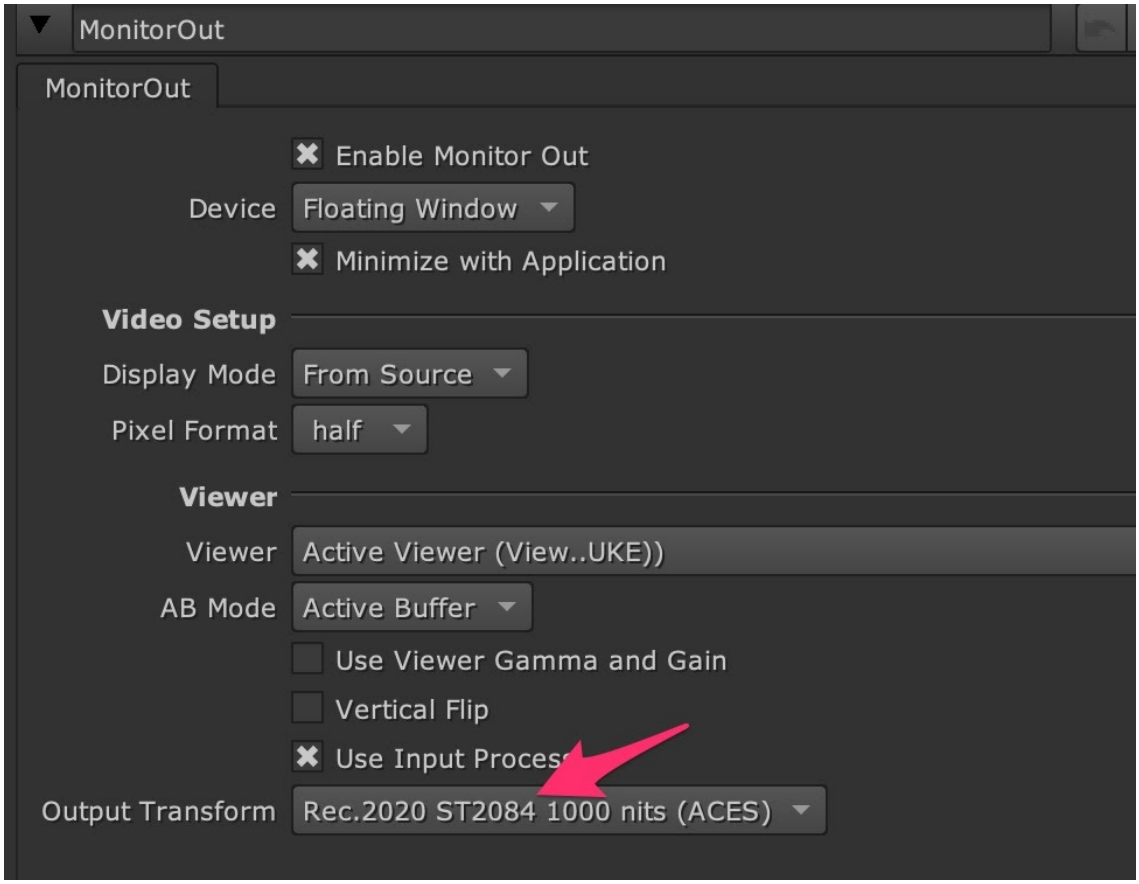
   

   Note: Enabling this control requires you to restart the application.

   This control sets the Viewer colorspace to **Display P3**, so that the screen can display values above 1. This setting also applies the P3 colorspace to the following;
   - All Viewers
   - Node Graph
   - Dope Sheet
   - Curve editors
   - Scopes
3. In the Viewer **Properties**, set the **gl buffer depth** to **half-float** or **float** to allow HDR values to display.
4. Set the **Viewer Process** to **sRGBf** using OCIO Color Management or provide your own extended color space. This ensures that the frame buffer is not clamped at 1.



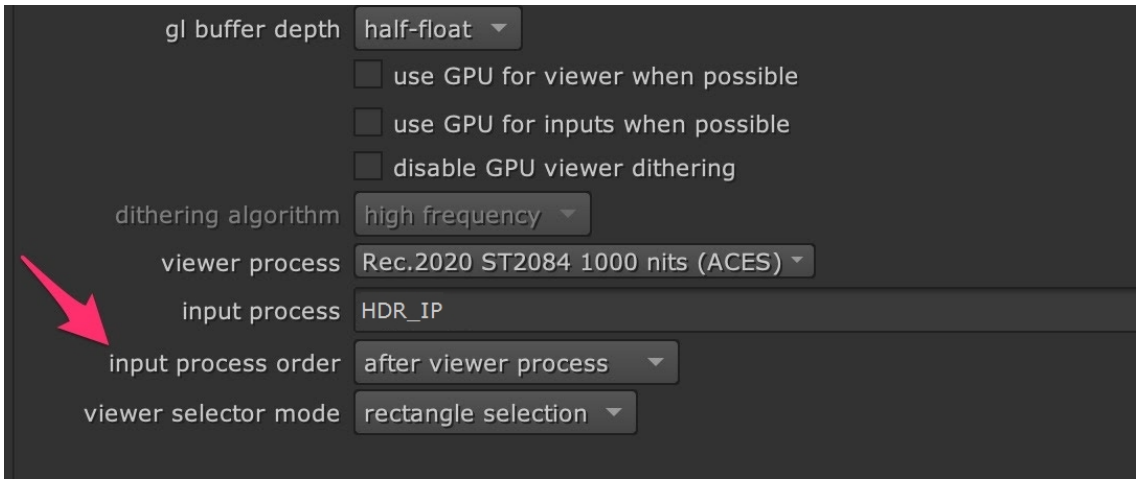
5. To use the **HDR Video (P3-ST2084)** setting on the monitor, select an appropriate ST2084 **Output Transform** in the MonitorOut **Properties** or in the Viewer.



A further color space conversion is required to allow HDR images to display correctly. An example of how this can be achieved in Nuke can be seen using a gizmo created by a customer, Nick Shaw - the gizmo is available for download here: PQ-EDR_v101.gizmo

For more information on gizmos and how to use them, see Accessing Gizmos in Nuke.

Using this gizmo as an **Input Process** and setting the **input process order** to **after viewer process** renders the correct image in the Viewer.



The gizmo also contains a slider called **Normalization (nits)**. This allows you to set the PQ nit value which is normalized to 1.0. This is useful, for example, to prevent clipping of a 1000 nit PQ signal on the 16" MacBook Pro's 500 nit display.

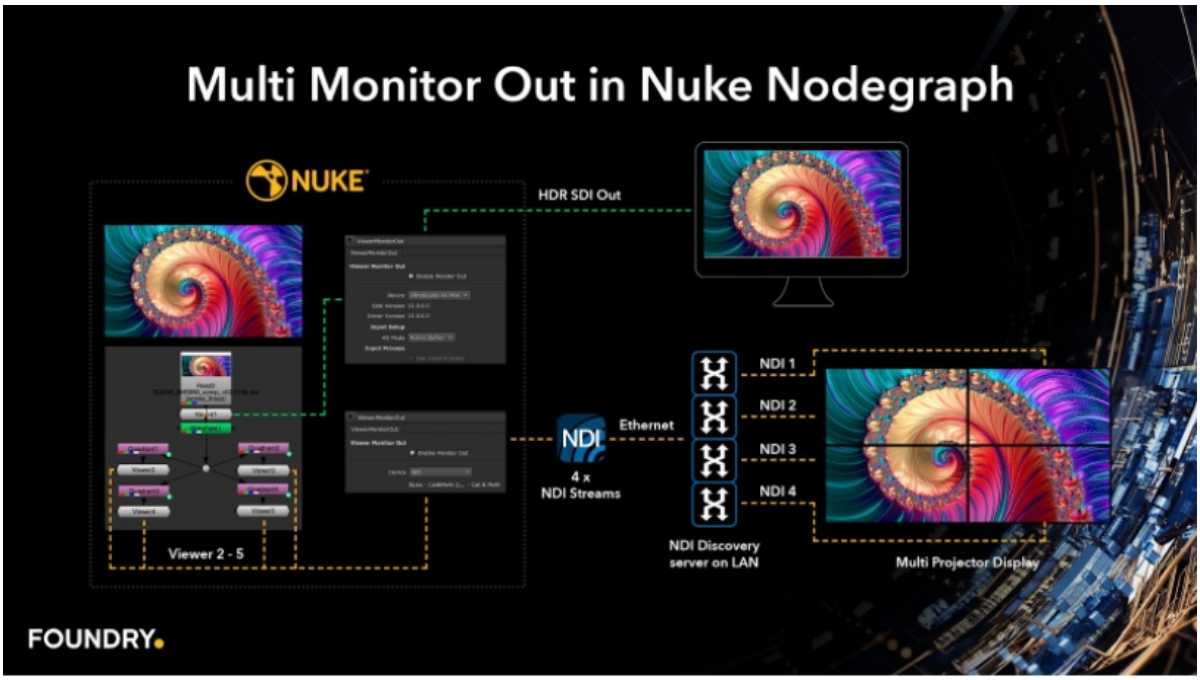
Streaming Viewer Output Over the Internet with NDI®

Network Device Interface, or NDI, is a network protocol from NewTek that enables you to send video, audio, and metadata signals over standard IP networks in real-time. NDI allows you to easily share a Nuke Viewer's output with anyone else on the same network.

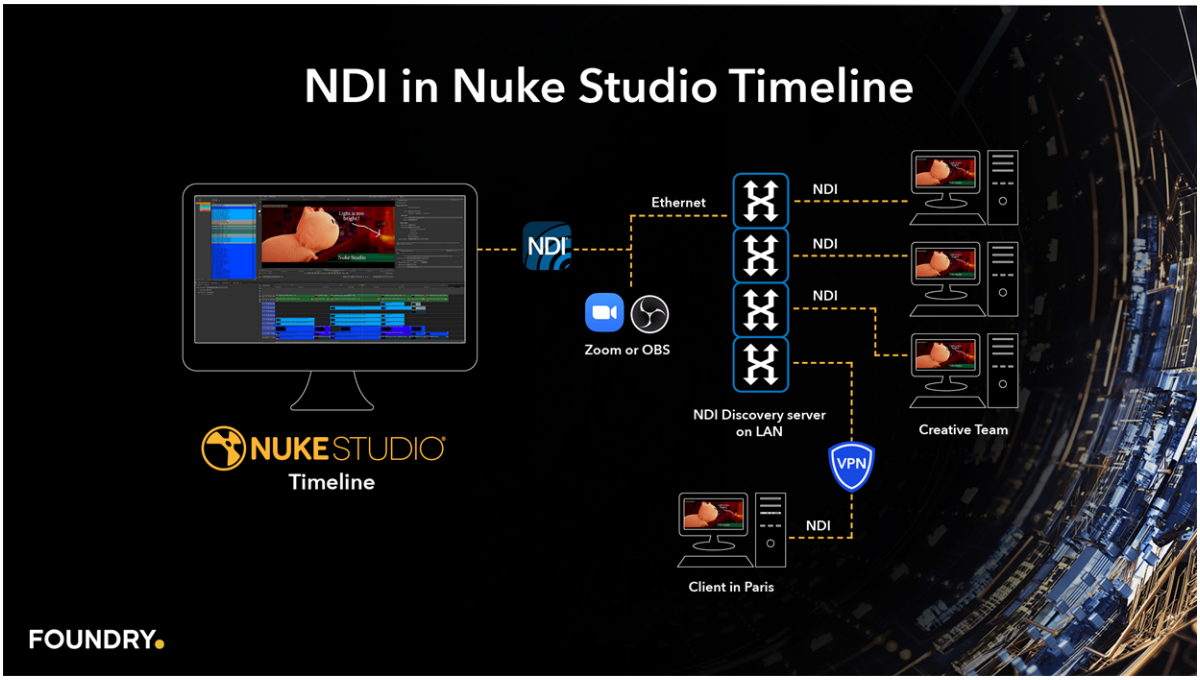
Nuke's monitor out functionality is controlled by Viewer nodes, so you can compare or combine multiple streams from as many Viewers as you like. Each Viewer can be enabled separately to output your shots to multiple output devices, whether they're dedicated monitor out devices, NDI streams, or Nuke's native Floating Window. The example shows five Nuke Viewers in total. The first Viewer at the top of the image is

feeding an HDR SDI monitor using a dedicated AJA or Blackmagic Design card and the other four are combined over NDI connections to a remote multi-projector display.

See Comparing and Combining Multi-Viewer Output with NDI for more details.



Nuke Studio's monitor out only allows you to stream one timeline Viewer to a dedicated monitor out device, NDI stream, or Nuke's native Floating Window. Sharing your Viewer over an NDI connection is a great way to collaborate in a review session with your creative team on the same local network or over VPN to remote customers.



Installing the NDI Tools

To view an NDI stream, you need the NDI Studio Monitor (Windows) or NDI Video Monitor (macOS) application installed on your machine. The monitor searches your network for NDI signals from Nuke or Nuke Studio and allows you to select the stream you'd like to view.

NDI Studio Monitor and NDI Video Monitor are available for download as part of the NDI Tools on Windows and macOS here:

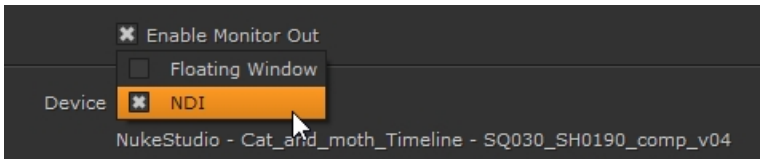
- NDI Tools

Note: NDI streams require you to bypass the operating system's firewall software on the remote machine to work correctly. Installing the NDI Tools on the remote machine should set exceptions in the firewall automatically, but you can add exceptions manually as described under Setting Up an NDI Stream from Nuke and Nuke Studio and Adding Firewall Exceptions on macOS.

Setting Up an NDI Stream from Nuke and Nuke Studio

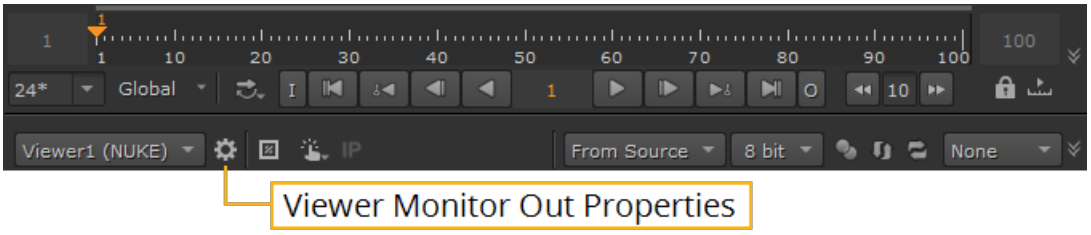
1. On the local machine in Nuke or Nuke Studio, go to **Edit** > **Preferences**, or press **Shift**+**S**, and navigate to **Panels** > **Viewer Monitor Out**.
2. Ensure that **Enable NDI** is checked.

3. In the monitor out **Properties**, set the monitor out **Device** to **NDI** to start broadcasting the stream over the network.

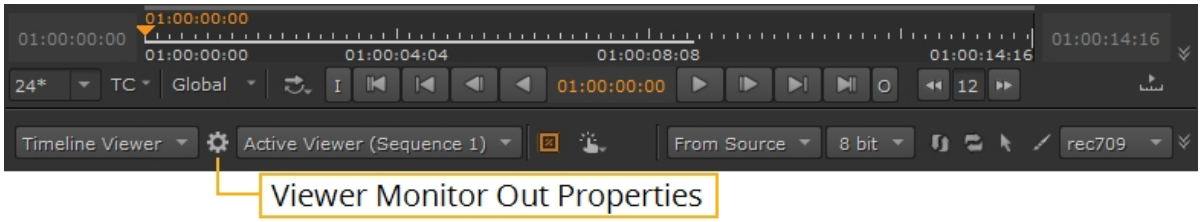


To display the **Properties** panel:

- **Nuke** - click the cog icon in the bottom-left of the **Viewer Monitor Out** panel or double-click the Viewer node.



- **Nuke Studio** - click the cog icon in the bottom-left of the **Viewer Monitor Out** panel.



4. Set the **Display Mode** for the receiver device of the NDI stream. If you don't change this, Nuke broadcasts the **source resolution** of the project over NDI. However, this format may not be supported on the receiving device, so you can set the NDI stream to scale to a specific resolution and frame rate here.

> **Note:** If the **Display Mode** is set from a timeline sequence, the NDI resolution is auto-scaled. Whereas the node graph viewer outputs an absolute pixel-for-pixel view, center cropping or padding, based on the resolution of the NDI signal. For example, a 3K (2880x1620) resolution comp set to output from the node graph viewer to UHD (3840x2160) would appear padded on a UHD device because the image is smaller than the frame size being broadcast, whereas the same 3K (2880x1620) comp set to output to HD (1920x1080) would appear cropped on the receiver side.

5. On the remote machine, download and install the NDI Tools for your operating system from the links under Installing the NDI Tools.

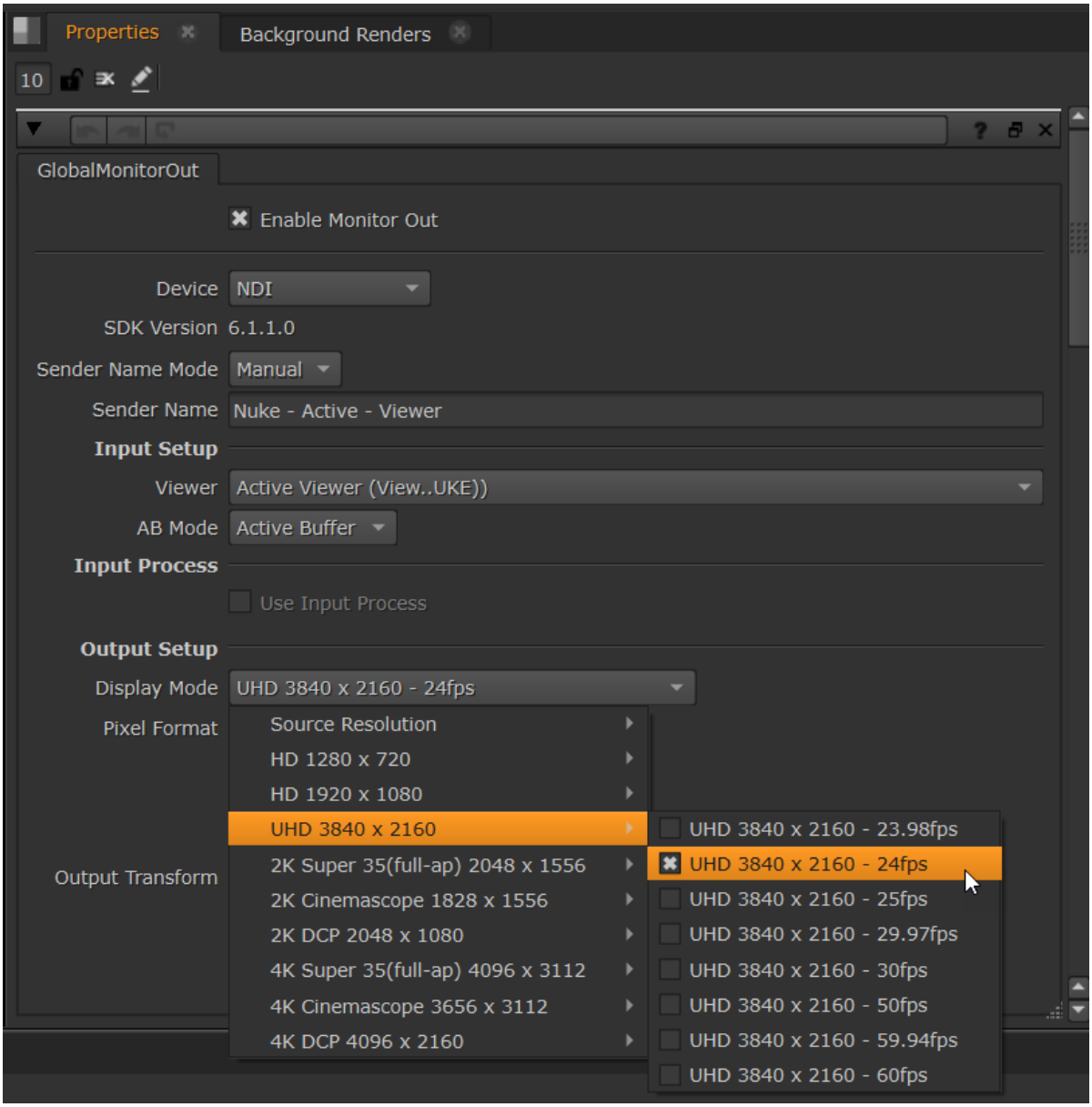
6. Launch the NDI Studio Monitor (Windows) or NDI Video Monitor (macOS), depending on your operating system:

   - **Windows** - In the NDI Studio Monitor, click the options  icon at the top-left and then select the required stream.
   - **macOS** - In the NDI Video Monitor, click the **NDI** menu at the top of the screen and then select the required stream.

   The stream name is shown in the Viewer **Properties** and on the NDI stream for identification and is constructed as follows: [Application] - [Project or Script] - [Viewer]

   For example: **Nuke - cat_and_moth - Viewer2**

   The Viewer is output through the NDI Studio Monitor or NDI Video Monitor.



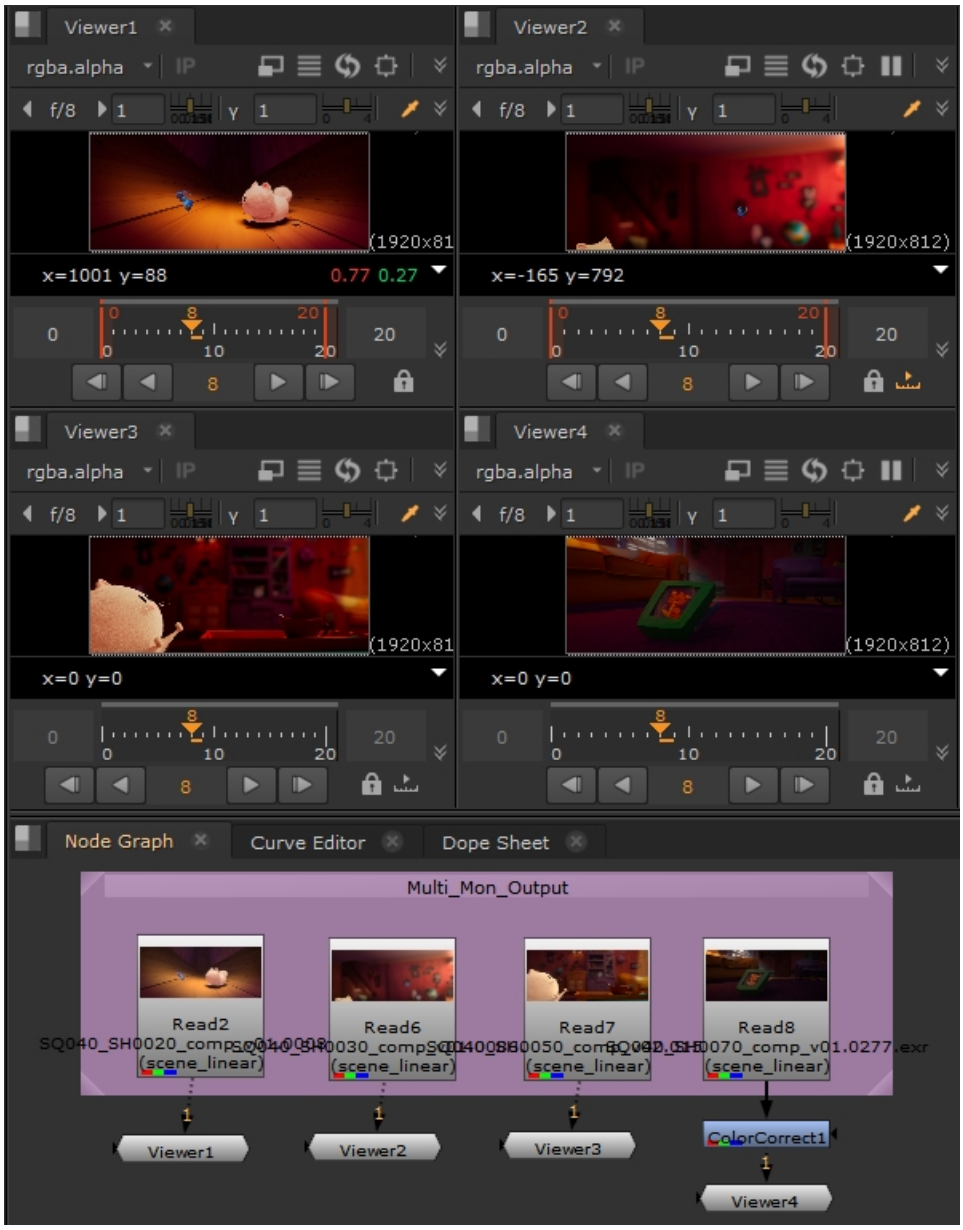
NDI output showing a custom named stream set in **Sender Name** and the **Display Mode** resolution set to UHD 3840 x 2160 24fps.

Note: When NDI is enabled in Monitor Out, audio will only play through the NDI stream.

Comparing and Combining Multi-Viewer Output with NDI

Nuke's monitor out functionality is controlled by Viewer nodes, so you can compare or combine multiple streams from as many Viewers as you like. Each Viewer can be enabled separately to output your shots to

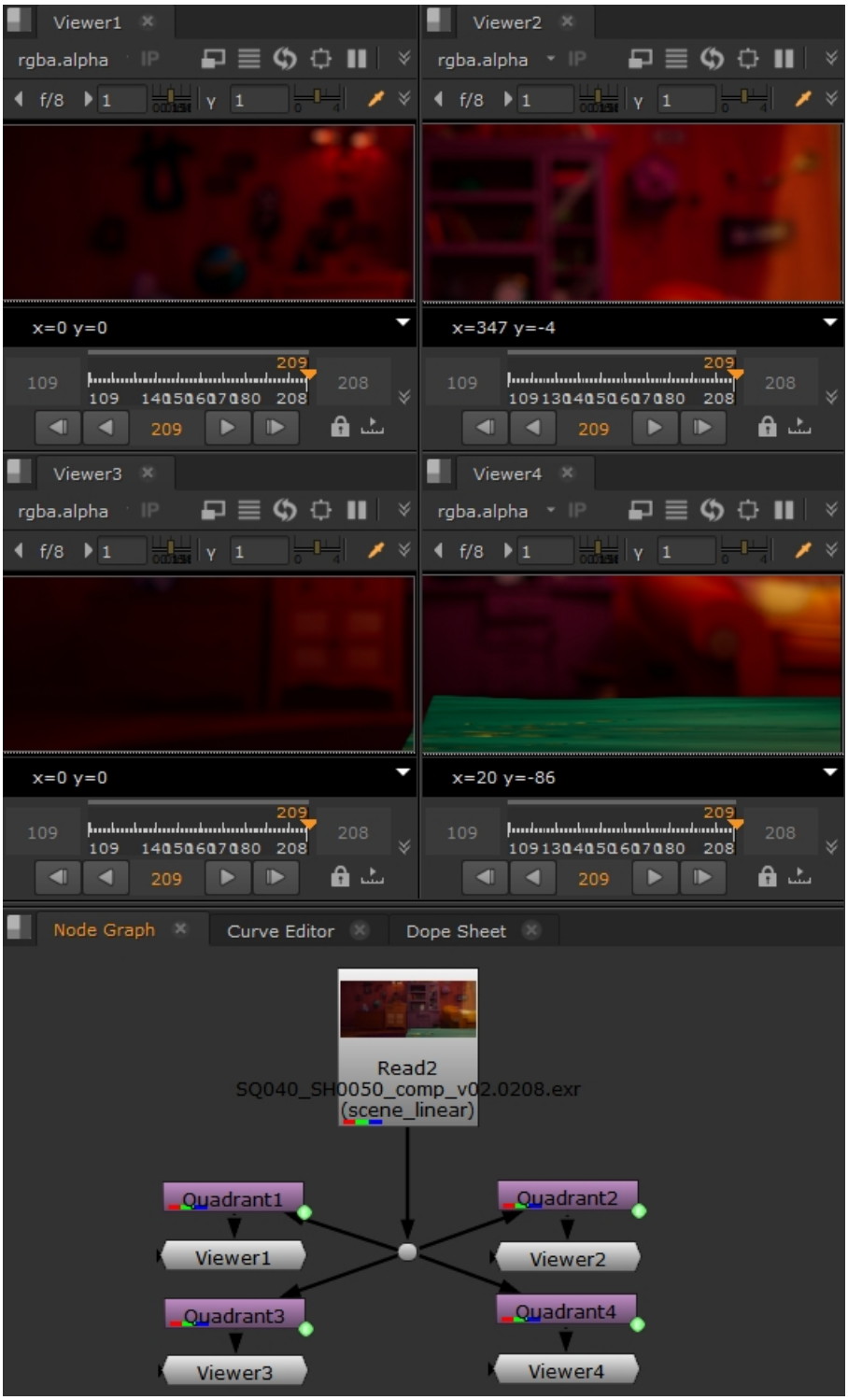
multiple NDI streams. For example, using four Viewers and four NDI streams, you could compare the format of multiple shots.



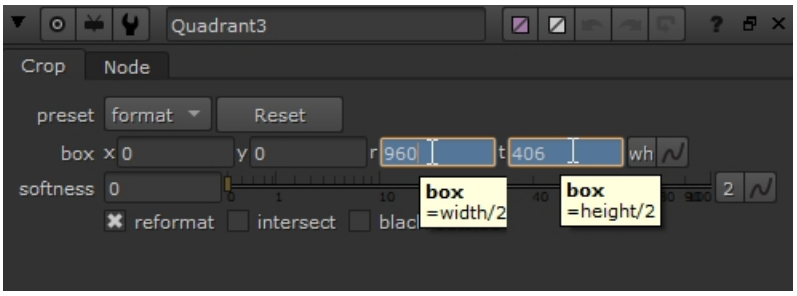
You can switch the **Output Transform** on each Viewer independently to check color consistency between the shots.

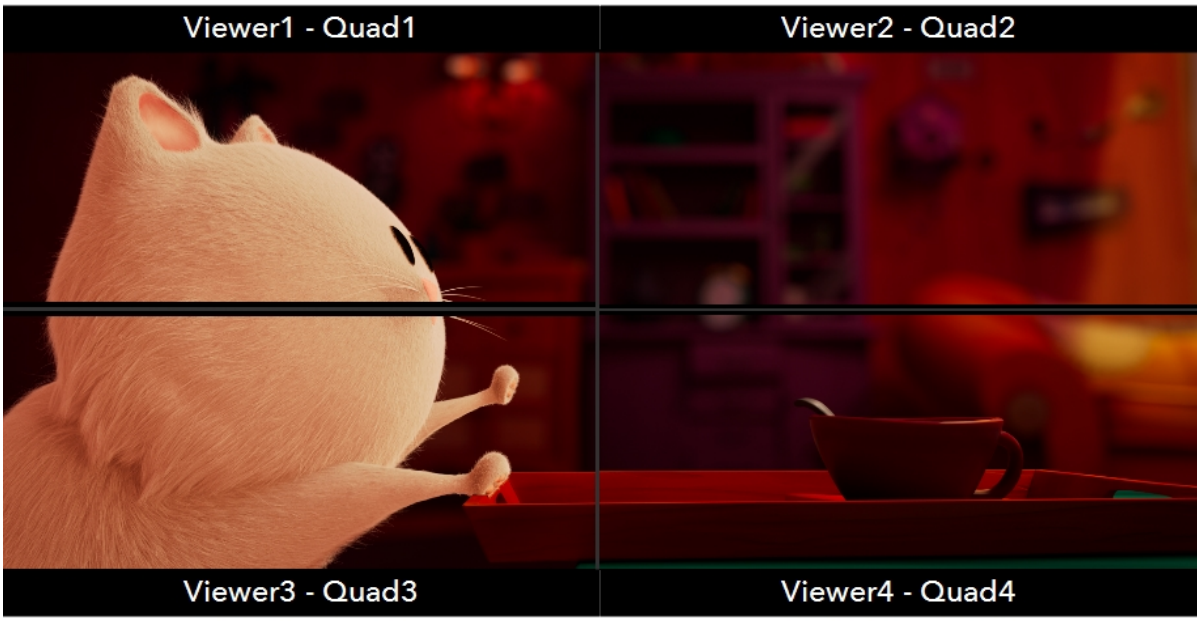


Using a similar four-Viewer setup, you could send four quadrants over NDI to four projectors to produce a larger image somewhere else on the same network.



Each quadrant is a Crop node that defines which part of the overall image is handled by each Viewer. For example, Quadrant3 is set up as shown here, with simple expressions to handle different formats.

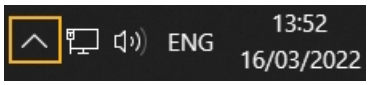




Adding Firewall Exceptions on Windows

Installing the NDI Tools on the remote machine should set exceptions in the firewall automatically, but you can add exceptions manually as follows:

1. Click the show hidden icons button on the right-hand side of the taskbar and then click **Windows Security**,

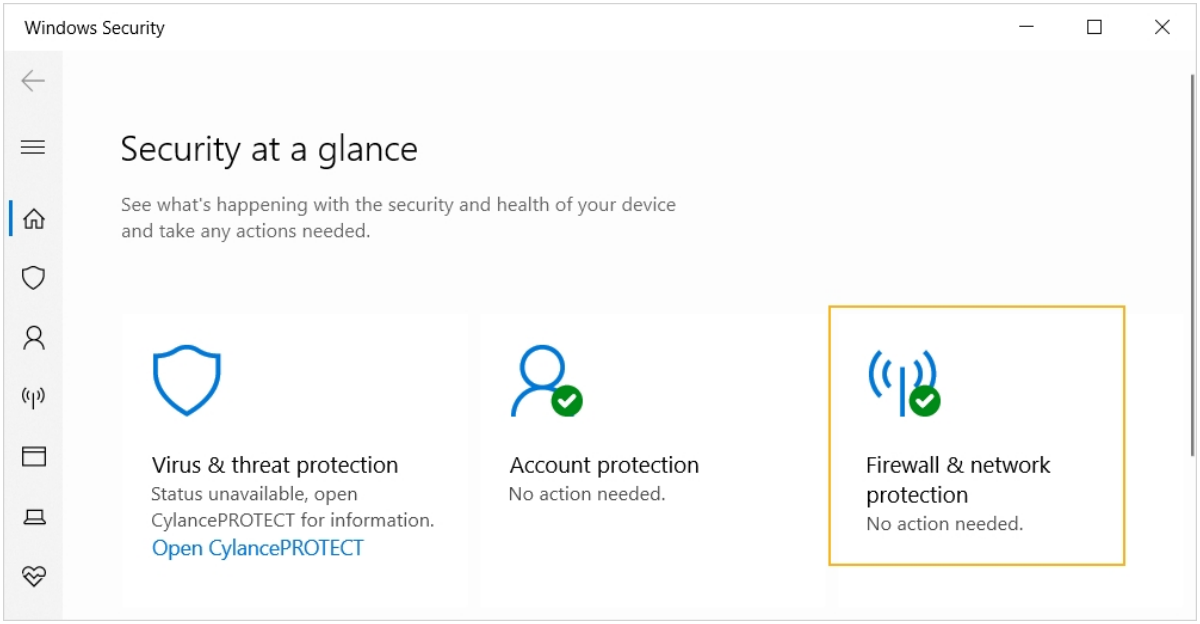
   

   OR

   Click the Windows Start button and type **Windows Security** and then click **Windows Security**.

   The **Windows Security** dialog is displayed.

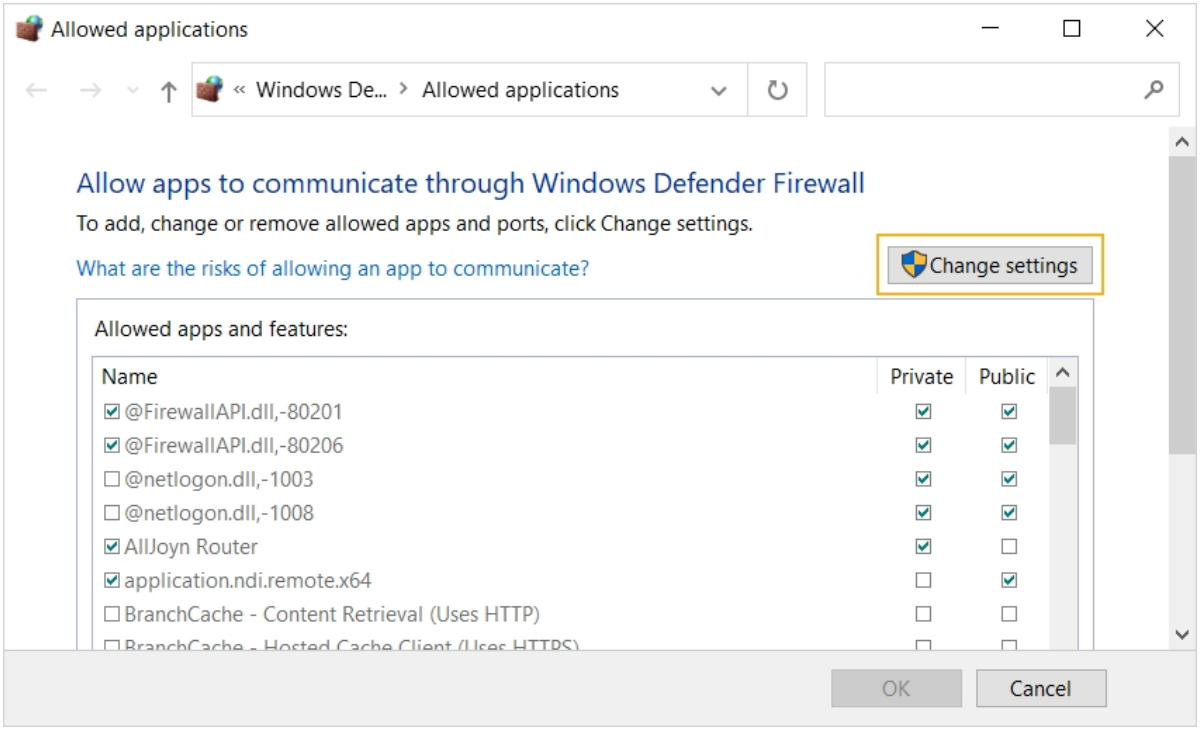
2. Click **Firewall & network protection**.

3. 

4. Click **Allow an app through the firewall**.

   The **Allowed applications** dialog is displayed.

5. Click **Change settings** to enable editing.

6. 

7. Scroll to the bottom of the window and click **Allow another app**.

8. Enter the file path to the NDI Studio Monitor executable or click **Browse** to locate the **.exe** using the explorer. NDI Studio Monitor is installed here by default:
   ```
   C:\Program Files\NDI\NDI 5 Tools\Studio
   Monitor\Application.Network.StudioMonitor.x64.exe
   ```

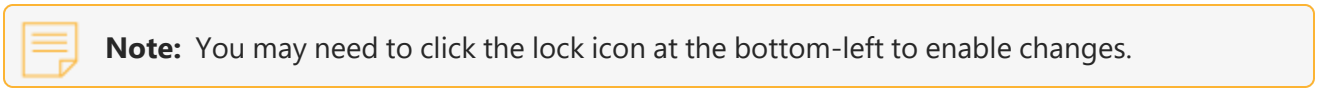
9. Click **Add** and then click **OK** to complete the firewall exception.

Refer to your operating system's documentation or speak to your network administrator for more details. For more information on the NDI Tools and network configuration best practice, see: https://ndi.video/tech/

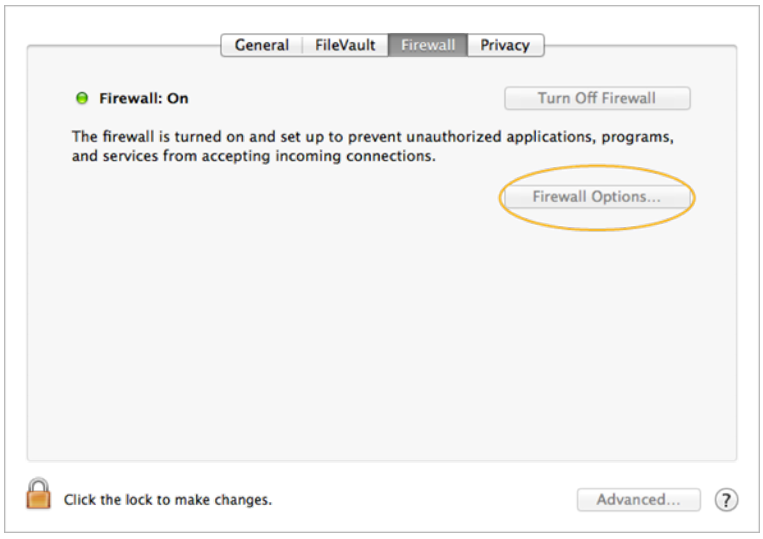
Adding Firewall Exceptions on macOS

Installing the NDI Tools on the remote machine should set exceptions in the firewall automatically, but you can add exceptions manually as follows:

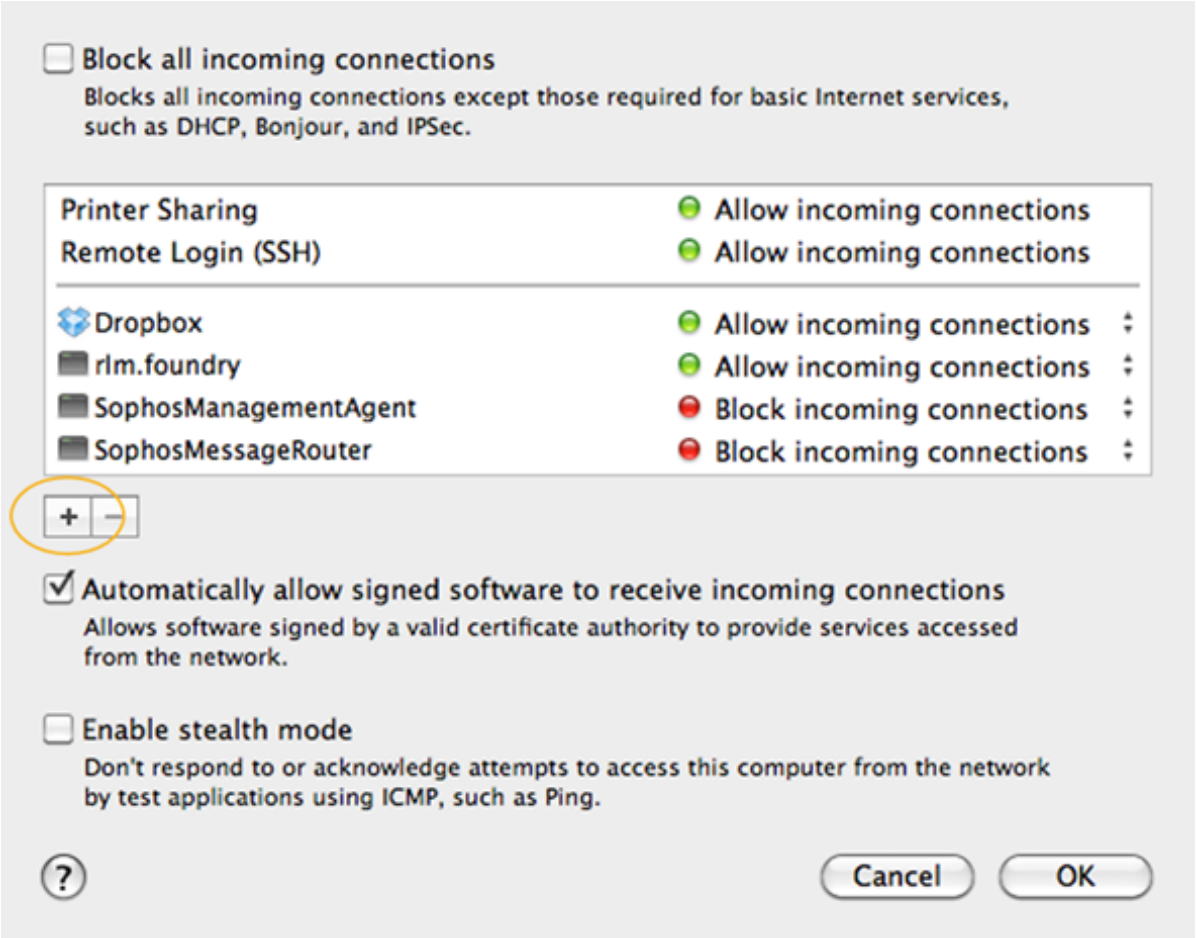
1. Launch **System Preferences**.
2. Click **Security & Privacy** and go to the **Firewall** tab.
3. Click the **Firewall Options** button.



Note: You may need to click the lock icon at the bottom-left to enable changes.



4. Under the list of incoming connections, click on the **+** button.



5. Enter the file path to the NDI Video Monitor application or click **Browse** to locate the **.app** using the Finder. NDI Video Monitor is installed here by default:
   `/Applications/NDI Video Monitor.app`

6. Select **NDI Video Monitor.app** and click **Add** and then **OK** to complete the firewall exception.

Refer to your operating system's documentation or speak to your network administrator for more details. For more information on the NDI Tools and network configuration best practice, see:
https://ndi.video/tech/

Using Scopes

Nuke provides scopes to help you evaluate your media. There are a number of global controls (**Preferences** > **Panels** > **Scopes**) that affect how the Scopes display information:

• **black point** - sets the black out of range warning level.

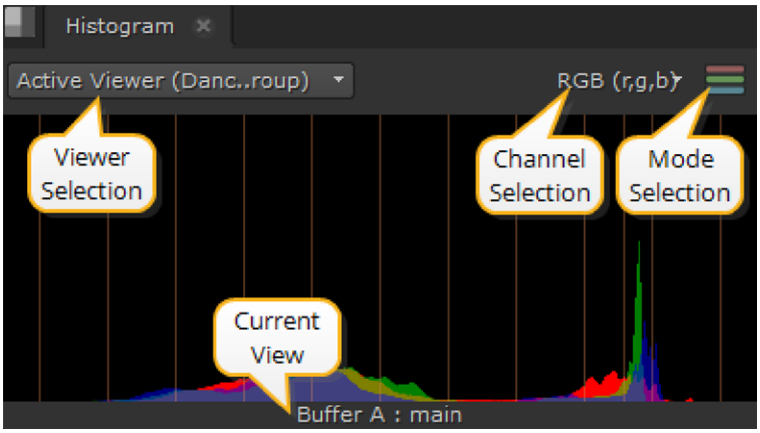
- **white point** - sets the white out of range warning level.
- **luma/chroma encoding** - sets the video standard to use when converting RGB to luma or chroma values in the scope displays, either **REC601** or **REC709**.
- **Include viewer color transforms** - when enabled, scope data includes the applied Viewer color transforms (gain, gamma, and LUT). When disabled, scope data does not include the applied Viewer color transforms. This may slow down rendering, as it may require image calculation.
- **Force full frame** - When enabled, scopes display data for the full frame, regardless of what portion of that frame is displayed in the Viewer. When disabled, scopes only display data for the current area requested by the Viewer rather than the full frame.

To open a scope, navigate to **Window** > **New Scope** and select the required scope from the list.

Histogram

The **Histogram** provides three color channel and luma channel information that describes the distribution of red, green, blue, and luma pixels throughout the current frame.

The Histogram graphs the number of pixels at each brightness level, and from left to right, the areas of the Histogram represent shadow, mid tones, and highlights.



Tip: You can pan the view area by holding **Alt**, or the middle mouse button, and dragging in the panel.

There are also **Viewer** and **Channel** selection controls on the **Histogram** tab:

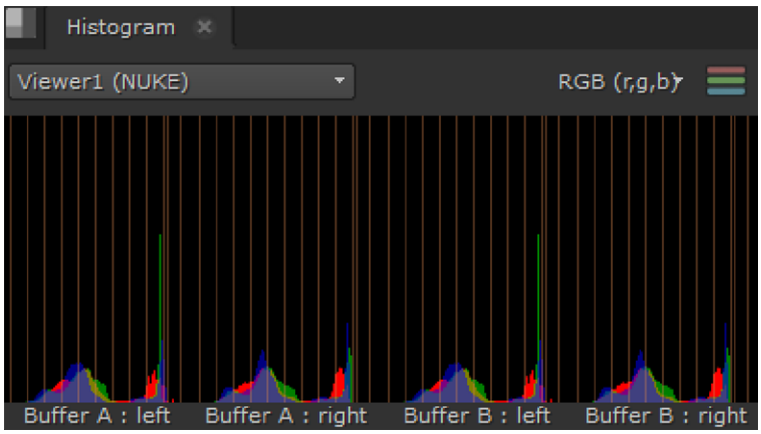
- **Viewer selection** - if you have multiple Viewers open, use the dropdown menu to associate Histogram output to the required clip.

  The default value, **Active Viewer**, automatically displays details on the last Viewer you selected.

- **Channel selection** - select the channels to output. The default setting displays RGB, but you can also view channels separately.
- **Mode selection** - select the mode to output. The default setting displays ganged RGB, but you can also view the channels separately.
- **Current View** - describes the view currently displayed in the scope, whether it's the A or B buffer and the view. The view defaults to **main**, unless **main** has been replaced in multi-view scripts or projects.

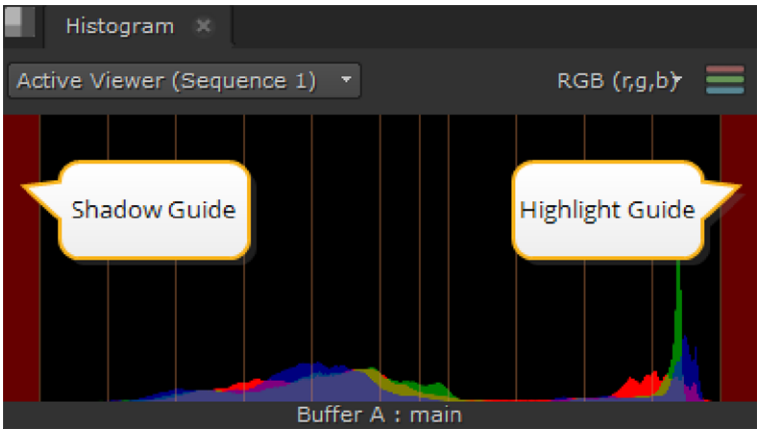
  Depending on which Viewer tools and views you have active, you can have up to four scopes displayed at once.

  For example, with two stereo Read nodes, one in each input buffer, and **wipe** and **Side by Side** active, the scopes display something like this:



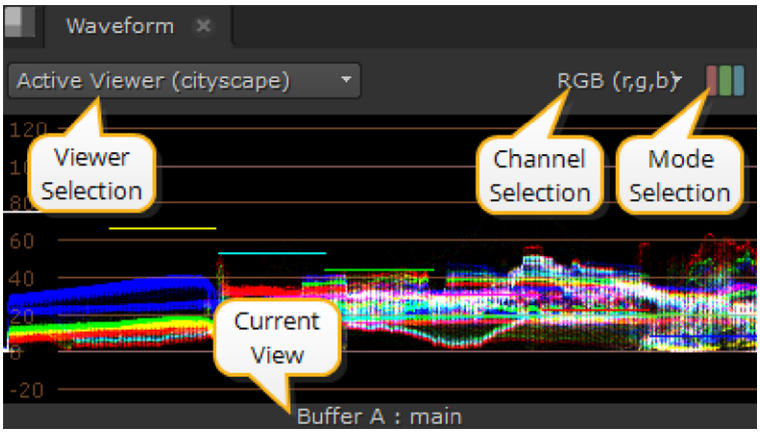
The scopes feature global customizable guides to help you view your clips. Navigate to **Preferences** > **Panels** > **Scopes** and enter values between **0** and **1** for the **Black** and **White** points. Note that this also sets the values for the Waveform display.

The guides at the edges of the Histogram turn red to warn you when the distribution is out of range:



Waveform

The **Waveform** scope provides information on clip luminance, or brightness, which you can use to decide whether the clip is over or under exposed. The white traces represent luminance values from 0 - 100% (black through the spectrum to white). The higher the waveform, the brighter the image in the Viewer.



Tip: You can pan the view area by holding **Alt**, or the middle mouse button, and dragging in the panel.

The upper white marker is used to measure when over exposure could be a problem. If your waveform has a lot of traces over the white marker, you should consider reducing the brightness of the clip. The opposite is true of the lower black marker.

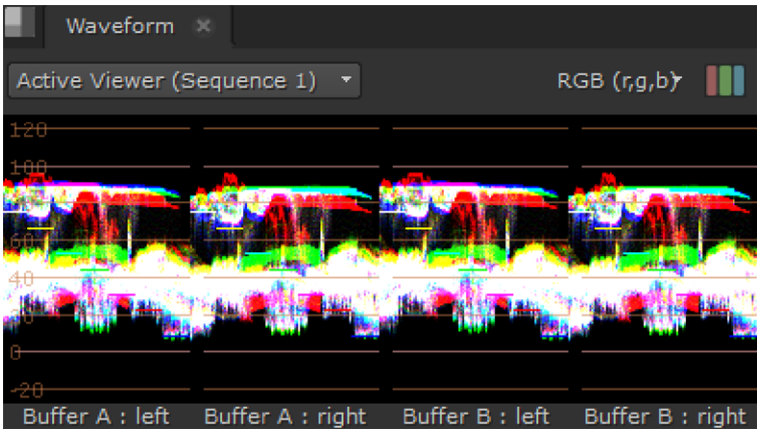
There are also **Viewer** and **Mode** selection controls on the **Waveform** tab:

- **Viewer selection** - if you have multiple Viewers open, use the dropdown menu to associate Waveform output to the required clip.

  The default value, **Active Viewer**, automatically displays details on the last Viewer you selected.
- **Channel selection** - select the channels to output. The default setting displays RGB, but you can also view channels separately.
- **Mode selection** - select the mode to output. The default setting displays ganged RGB, but you can also view the channels separately.
- **Current View** - describes the view currently displayed in the scope, whether it's the A or B buffer and the view. The view defaults to **main**, unless **main** has been replaced in multi-view scripts or projects.

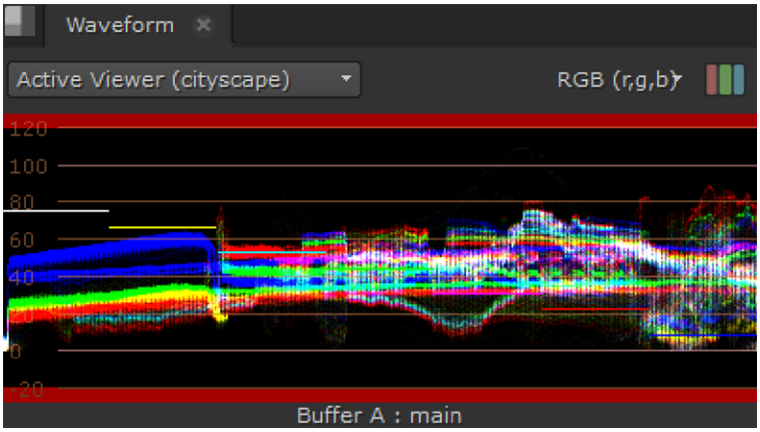
  Depending on which Viewer tools and views you have active, you can have up to four scopes displayed at once.

For example, with two stereo Read nodes, one in each input buffer, and **wipe** and **Side by Side** active, the scopes display something like this:



The scopes feature global customizable guides to help you view your clips. Navigate to **Preferences** > **Panels** > **Scopes** and enter values between **0** and **1** for the **Black** and **White** points. Note that this also sets the values for the Histogram display.

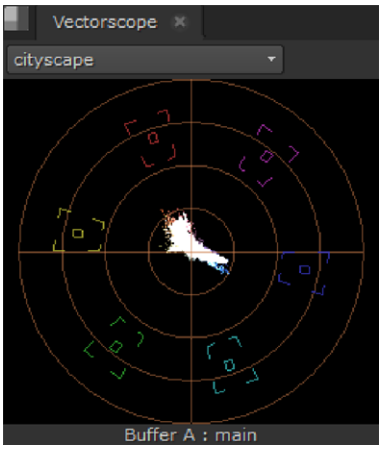
The guides at the top and bottom of the Waveform turn red to warn you when the distribution is out of range:



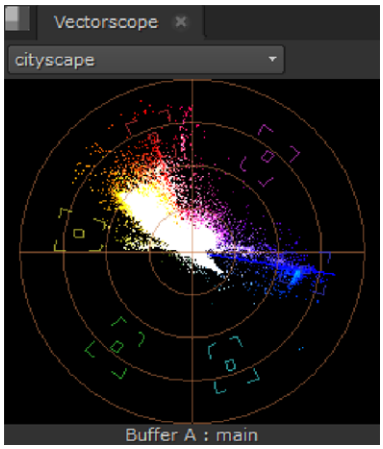
Vector

The **Vector** scope displays color, saturation, and hue information for the current frame. Similar to color wheels, Vector scopes display information radially, from the center outward. The farther from the center the data spans, the more saturation is represented.

In the image on the left, you can see that the frame represented contains mostly yellows and reds, but the values are not oversaturated. The image on the right represents a badly saturated frame. Notice the spill of red traces distributed toward the edge of the scope pass the target (the highlighted square).



Normal saturation.



High Saturation.

Tip: You can pan the view area by holding **Alt**, or the middle mouse button, and dragging in the panel.

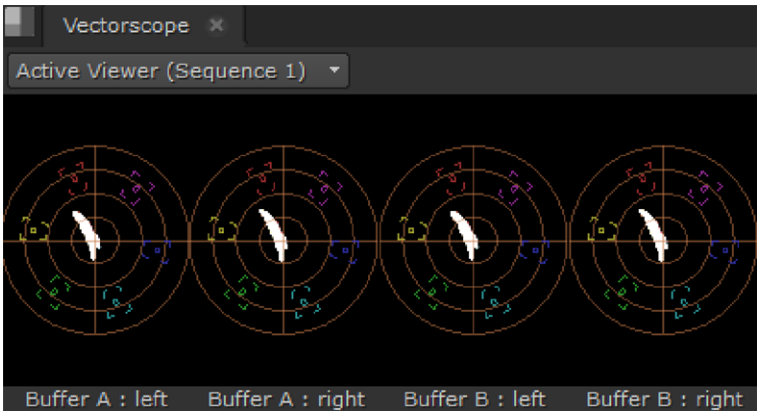
There is also a **Viewer** selection control and Current View label on the **Vectorscope** tab:

- **Viewer selection** - if you have multiple Viewers open, use the dropdown menu to associate Vector scope output to the required clip.

  The default value, **Active Viewer**, automatically displays details on the last Viewer you selected.
- **Current View** - describes the view currently displayed in the scope, whether it's the A or B buffer and the view. The view defaults to **main**, unless **main** has been replaced in multi-view scripts or projects.

  Depending on which Viewer tools and views you have active, you can have up to four scopes displayed at once.

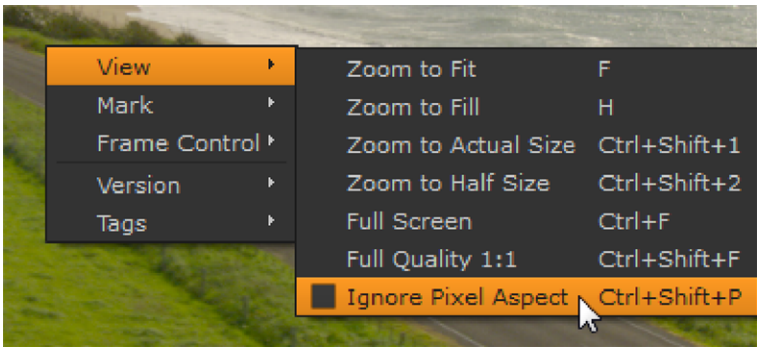
  For example, with two stereo Read nodes, one in each input buffer, and **wipe** and **Side by Side** active, the scopes display something like this:



About Anamorphic Media

The Viewer automatically recognizes anamorphic clips and displays them with the correct aspect ratio.

If for any reason you want to display an anamorphic clip with a 1:1 aspect ratio, right-click in the Viewer displaying the clip and enable the **Ignore Pixel Aspect** checkbox, or use the **Ctrl/Cmd**+**Shift**+**P** keyboard shortcut.

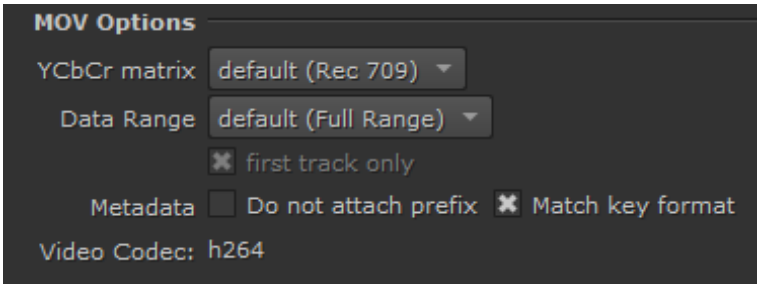


About QuickTime Media

Working with **.mov** files can be unpredictable when compared to other formats, so Nuke gives you a few QuickTime options when reading and writing **.mov** files.

Nuke attempts to select the 'best fit' combination by reading an extended list of metadata key/value pairs from the QuickTime header, including **nclc atom**, **gama atom**, and **ProRes** codec headers.

If you place a clip in the Viewer, or open a shot in the Viewer, and open the **Media** tab, you'll see that Nuke has a number of media-specific controls that you can manually override if the 'best fit' is not what you're looking for:



- **YCbCr Matrix** - sets the way Y'CbCr is converted to RGB. You can choose to use the new **Rec 601** and **Rec 709** or the **Legacy** encoding methods, which are the methods used previously in Nuke.
- **Codec** - sets the codec used to read (write already had a similar control) the QuickTime file.

  The codec dropdown defaults to a codec appropriate for the QuickTime in question, where available, and only lists those that declare themselves able to read the file.
- **Pixel Format** - sets the read and write pixel format, which includes bit depth, colorspace, pixel packing, and ranges.

  This setting defaults to the best format accepted by the codec, allowing Nuke to perform the conversion to RGB without the use of an unknown QuickTime transform, where possible. RGB pixel types rely on QuickTime to do the conversion from Y'CbCr when dealing with a non-RGB codec.

In addition to the **nclc**, **gama**, and **ProRes** data Nuke, and by extension Nuke, also write additional metadata into the file headers during export, retaining your QuickTime preferences. This combined metadata represents myriad potential QuickTimes preferences, so Nuke reads the available metadata in the following order, reverting down each level as the level above is unavailable or set to a reserved or unknown value:

- Foundry-specific metadata
- ProRes header data
- nclc atom data
- gama atom data
- The defaults associated with the chosen codec

In this way, the worst case scenario is that you end up with the chosen codec class' default values.

Compositing with Nuke

Each chapter in this section explains in detail a key feature of Nuke. You can use the section to familiarize yourself with the features you are particularly interested in, or to get answers to specific problems that arise during compositing. For information on the features in NukeX and Nuke Studio, see Advanced Compositing with NukeX and Nuke Studio or Timeline Editing in Nuke Studio and Hiero.

Organization of the Section

These are the topics covered by this section:

- Managing Scripts describes the basics for creating, saving, and loading scripts or comps.
- Reformatting Elements describes how you can reformat images through scaling, cropping, and pixel aspect adjustments. This chapter also covers working with bounding boxes.
- Channels shows you how to manage image data using Nuke's 1023-channel workflow.
- Merging Images teaches you how to layer background and foreground elements together, create contact sheets, and copy rectangles from one image to another.
- Removing Noise with Denoise teaches you to use Denoise node to remove noise from your footage. It uses spatial filtering to remove noise without losing image quality.
- Keying with ChromaKeyer teaches you to use the blue/greenscreen keyer ChromaKeyer in Nuke.
- Keying with Primatte teaches you to use the blue/greenscreen keyer Primatte in Nuke.
- Keying with Keylight teaches you to use the keyer tool Keylight in Nuke.
- Keying with Ultimatte shows you to use the Ultimatte keyer in Nuke.
- Using RotoPaint shows how to use Nuke's RotoPaint node.
- Tracking and Stabilizing shows how to generate and edit 2D tracking data for purposes of removing unwanted motion or applying it to other elements.
- Transforming Elements covers the tools for changing the size, location, and orientation of an image, including how to translate, scale, rotate, and skew elements in 2D and 3D space. This chapter also describes adding motion blur.
- Warping Images teaches you to use the GridWarp and SplineWarp nodes to warp and morph images.
- Temporal Operations explains how to apply time-based effects like clip retiming and motion blur. This chapter also explains how to perform editorial tasks, such as trimming and slipping.
- Working with Color explains a broad sampling of Nuke's many color correction tools.
- Filtering and Spatial Effects deals with applying filters, such as convolves and blurs.
- Creating Effects describes how you can create effects, such as star filter effects, on your images.

- Analyzing and Matching Clips explains how to use the CurveTool node to analyze and match image sequences.
- 3D Compositing with the New 3D System allows you to efficiently work with USD-based 3D scenes.
- Working with Gaussian Splats explains how to import, manipulate, and render Gaussian Splats in Nuke.
- Classic 3D Compositing teaches you how to create and manipulate 3D scenes composed of objects, materials, lights, and cameras.
- Stereoscopic Scripts describes how to composite stereoscopic material in Nuke.
- Working with Graph Scope Variables explains how to use variables and other tools in workflows managing multiple shots at once.
- Deep Compositing goes through using the deep compositing node set in Nuke.
- Working with File Metadata describes how to use Nuke's MetaData nodes to work with information embedded in images.
- Audio in Nuke covers using audio clips in Nuke.
- Previews and Rendering teaches you how to write out image sequences from scripts in order to preview results or create final elements.
- Organizing Scripts is designed to help you organize your Nuke scripts in a clear and meaningful way.
- Configuring Nuke explains how to set up Nuke for multiple artists working on the same project.
- Expressions explains how to apply expressions or scripting commands to Nuke parameters.
- The Script Editor and Python takes you through using Nuke's Script Editor for executing Python commands.

Managing Scripts

In this topic, you learn about Nuke's project files called **scripts** or **comps**. The topics covered include setting up, saving, and loading scripts. You'll also learn about managing your node tree in the Node Graph, using Precomp nodes, and working with file metadata.

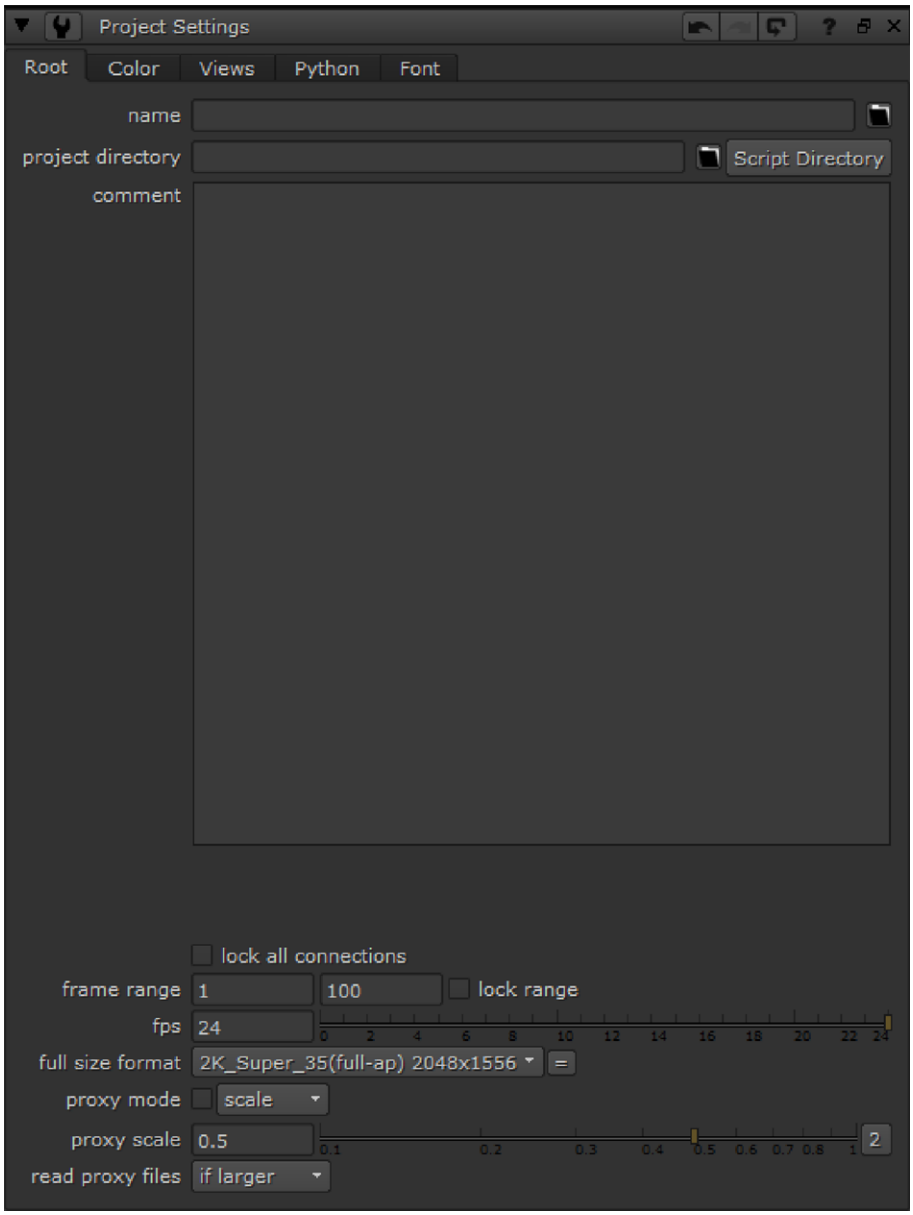
Setting Up Your Script

When you start working on a script, you should first define the settings for it. This involves assigning the script a name, frame range, frame rate, and default full and proxy resolution format.

Name, Time Span, and Frame Rate

To set the script name, frame range, and frame rate:

1. Select **Edit** > **Project Settings**, or simply press **S** over a blank portion of the workspace.
   The **Project Settings** panel appears.



2. On the **Root** tab, type a name for the script (say, **firstcomp.nk**) in the **name** field. Nuke's scripts always have the extension **.nk**.
3. Type the numbers of the first and last frames in the **frame range** fields to define length of time for your "shot".
4. In the **fps** field, enter the rate in frames per second (fps) at which you want your script's Viewers to play back footage. For film-based elements, 24 fps is appropriate.

Setting the Default Project Directory

In the Project Settings (**Edit** > **Project Settings**) you can define your default project directory. You can then refer to it with '**./**' in your file paths, for example **./test.jpg** for an image called **test** in your default project

directory.

You can also click the **Script Directory** button to create an expression that sets your **project directory** to be the directory where your current script is saved.

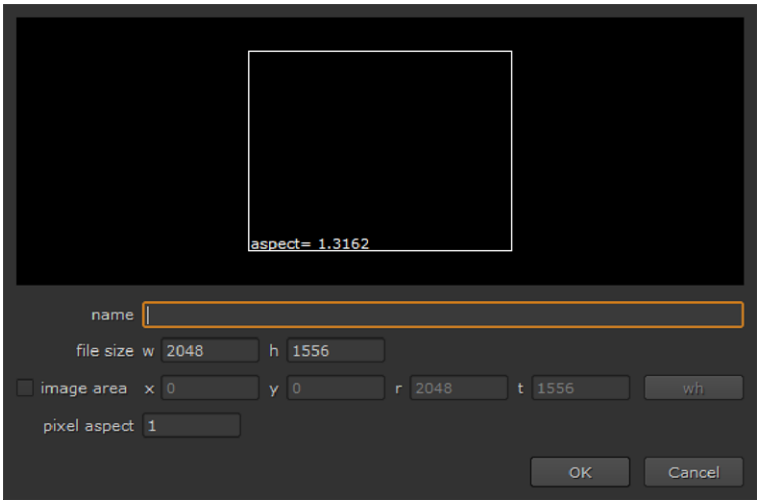
Full-Size Formats

When you start a new script in Nuke, you need to set up a full-size format. The full-size format determines the size of the image that you get from any disconnected node inputs. It also sets the default size of any script-generated elements, such as Constants and ColorBars.

Note: The full-size format does not affect the format of the elements you read into your script. Nuke is resolution-independent, which means it respects and keeps the resolution of your images. It won't automatically crop or pad elements to match the project settings.
If you want elements you read in to conform to the project settings, you can do this manually using the Reformat node. For more information, see Reformatting Image Sequences.
The full-size format is also used to calculate proxy scaling if a proxy format is used. For more information on the proxy mode and proxy formats, see Proxy Mode.

To Set Up a Full-Size Format

1. If it's not already open, select **Edit** > **ProjectSettings** (or press **S**) to display the **Project Settings** panel.
2. From the **full size format** dropdown menu, select the resolution for the final output of rendered images. If the format you want to use is not in the menu, select **new**. The **New format** dialog displays.



In the **name** field, enter a name for the new format.

In the **file size** fields, define the width and height of the format.

If you like, you can also define additional information, such as offsets and pixel aspect ratio.

Click **OK** to save the format. It now appears in the dropdown menu where you can select it.

Proxy Mode

When compositing with Nuke, you can work in two different modes: the **full-size mode** or **proxy mode**. In the full-size mode, images are read in exactly as they are on the disk, and all positions are in actual pixels in these images. This is the mode you want to use for accurate feedback and when rendering the final output.

In proxy mode, instead, a proxy scale factor is used. All images and all x/y positions are scaled by this factor. This produces the same (or at least very similar) composite at a different scale. For example, you can use a fraction of the full output resolution to speed up rendering and display calculation.

In addition to the above, a separate proxy file can also be read in place of a full-size image, provided you have specified one in the relevant Read node. This can further speed up the preview, by using a smaller image that reads faster and also saves time by not needing to be scaled. For more information, see Read Nodes and Proxy Files.

The proxy settings you define in the project settings affect both the proxies Nuke generates using the proxy scale factor and proxies read from files. Below, we discuss setting a proxy format and/or a proxy scale and defining how Read nodes use proxy files.



Note: Note that proxy versions of images are only used if you have activated the proxy mode. When the proxy mode is off, Nuke always uses the full-res files.

Proxy Format and Proxy Scale

In the **Project Settings** panel, you have the option of defining a **proxy format** and/or a **proxy scale** that you use in the proxy mode.

For the proxy **format**, you can define the image resolution as well as additional information about offsets and pixel aspect ratio. When using the proxy format, the scaling is proportionate to the full-size/proxy format relationship (not scaled to the proxy format).

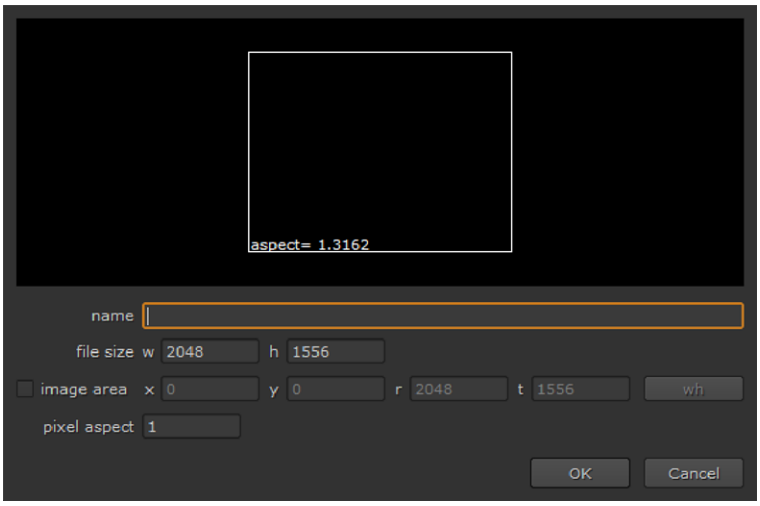
For the proxy **scale**, you only define a simple scale factor by which your images are scaled down whenever the proxy mode is activated. For example, you can use the scale factor of 0.5 to scale your images down to half the size.

If you like, you can define both a proxy format and a proxy scale, and then choose which one to use in proxy mode. A proxy scale is easier to set up, but a proxy format gives you more control over the low-res versions of your images. Below, we first describe how to set up proxy formats and then how to define a proxy scale.

To Set Up Proxy Formats

1. If it's not already open, select **Edit** > **Project Settings** (or press **S**) to display the **Project Settings** panel.
2. If you want to use a proxy format (rather than a proxy scale) whenever the proxy mode is activated, select **proxy mode** > **format** from the dropdown.
3. From the **proxy format** dropdown menu, select the resolution to use while working to speed things up. Notice that your images are not scaled to this resolution, but the scaling is proportionate to the full-size/proxy format relationship. Nuke divides the proxy format width by the full-size width and uses the result as the scale factor.

   If the proxy format you want to use is not in the dropdown menu, select **new**. The **New format** dialog displays.



   - In the **name** field, enter a name for the new format.
   - In the **file size** fields, define the width and height of the format.

Tip: You can type formulas in numeric fields to do quick calculations. For example, if your full-size format width is 4096 and you want your proxy format width to be 1/2 of that, you can enter **4096/2** in the **New format** dialog's **file size w** field and press **Enter**. Nuke then calculates the new width for you.

   Click **OK** to save the format. It now appears in the dropdown menu where you can select it.
4. To activate the proxy mode and use the low-res format for calculations and display, check **proxy mode**.



Alternatively, you can use the proxy toggle in the Viewer controls, or press **Ctrl**+**P** (**Cmd**+**P** on a Mac). For more information, see Using the Viewer Controls.

To Set Up a Proxy Scale

1. If it's not already open, select **Edit** > **Project Settings** (or press **S**) to display the **Project Settings** panel.
2. Select **scale** from the dropdown in the **Project Settings**.
3. Using the **proxy scale** input field or slider, specify the factor by which you want to scale the width and height of your images. For example, if you want to scale them down by 50%, use the value of **0.5**.
4. To activate the proxy mode and use the low-res format for calculations and display, check **proxy mode**.



Alternatively, you can use the proxy toggle in the Viewer controls, or press **Ctrl**+**P** (**Cmd**+**P** on a Mac).

Read Nodes and Proxy Files

As an alternative to letting Nuke generate proxies on the fly, proxy files can be specified using a second file name in the Read nodes (for how to do this, see Loading Image Sequences). If you don't have a proxy file, you can create one by activating the proxy mode and rendering your full-size images using a Write node (see Rendering Output in the Previews and Rendering chapter).

The proxy file does not need to match the proxy resolution in use. Depending on your project settings, either the full-res or proxy file is scaled to the required proxy size (that is, the size calculated by taking the full-size format and scaling it by the current proxy settings). However, if your proxy images match your typical proxy settings, you can save this time.

To Define Which File (Full-Res or Proxy) is Used in the Proxy Mode:

1. If it's not already open, select **Edit** > **Project Settings** (or press **S**) to display the **Project Settings** panel.
2. From **read proxy files**, select when to use the proxy file (rather than the full-res file) in a Read node:

- **never** - Never use the proxy file in the proxy mode. Instead, scale the full-size file as necessary.
- **if larger** - Use the smaller of the two images if it is larger or equal to the desired size, scaling down as needed. Otherwise, use the larger one, scaling down or up as needed. This is the default option.
- **if nearest** - Use the image that is closest to the desired size, scaling up or down as needed.
- **always** - Always use the proxy image in the proxy mode, scaling it up or down as necessary.

The option you choose affects all Read nodes in your script, provided that a proxy file is named and the proxy mode is on.

Write Nodes and Proxy Files

It is worth mentioning here that when a script is rendered in proxy mode, processing is done at the proxy scale and image output goes to the file name in the Write node's **proxy** field. If you do not specify a proxy file name, the render fails with an error. It never resizes the proxy image, and it does not write the proxy image over the full-size one.

For more information, see Rendering Output in the Previews and Rendering chapter.

Using Proxy Mode to Enlarge a Script

If you find it necessary to render a comp at higher than the original resolution it was intended for, you can also use the proxy resolution to scale up from the root full size format. For example, you could use a 2K composite to produce 4K or higher images in the proxy mode.

The reason you'd probably want to do this is because the comp you did needs to be re-run at higher resolution for a different output target and possibly with some new higher resolution elements. For example, you may need to re-render with new CG elements at print resolution rather than for film out, but you don't want to go through your script and modify everything that has an x/y position.

When scaling up the output of a script in the proxy mode, image generator nodes render at the larger size, larger images can be specified in the proxy field, and as with scaling down, all x/y positions are scaled to the proxy output resolution.

Using Small Proxy Files

If you have previously set up your script to use small proxy files, you do not have to remove these. Make sure **read proxy files** in **Project Settings** is set to anything other than **always**, and Nuke reads the larger original files and scales them up.

Using Large Proxy Files

If you actually have larger proxy files, you should enter them into the Read nodes' **proxy** field, and set **read proxy files** to anything other than **never**. Nuke then uses these larger files in the proxy mode. For maximum quality, these should be at exactly the desired proxy size so that no scaling is done to them. For example, if the required proxy size (defined by the project settings) is 4K, then the proxy image should be exactly 4K. Otherwise, Nuke scales it to match the project settings, which reduces the quality.

Rendering the Scaled-up Output

To render the larger output of a scaled-up script, you need to activate the proxy mode and enter a file name in the **proxy** field of the Write nodes. The larger images are then written to these files.

To Toggle Between Full Resolution and Proxy Mode

It's usually smart to work in proxy mode because most operations work quickly and more efficiently under the low-res display. You can switch between low- and high-resolution when you need greater precision (for example, when pulling a key or tracking), or when you're ready for final rendering.

1. Click on an empty area of the Nuke window.
2. Press **Ctrl**+**P** to toggle the display mode (**Cmd**+**P** on a Mac).

Nuke automatically scales script elements - Bezier shapes, B-spline shapes, paint curves, garbage masks, tracking curves, and so on - to keep the original placement on the image.

Loading Image Sequences

When you are ready to start compositing, you may want to begin by importing a background or foreground image sequence. Typically, you would read in both full- and proxy-resolution versions of the sequence. You can read in several image sequences in one go.

Importing Image Sequences

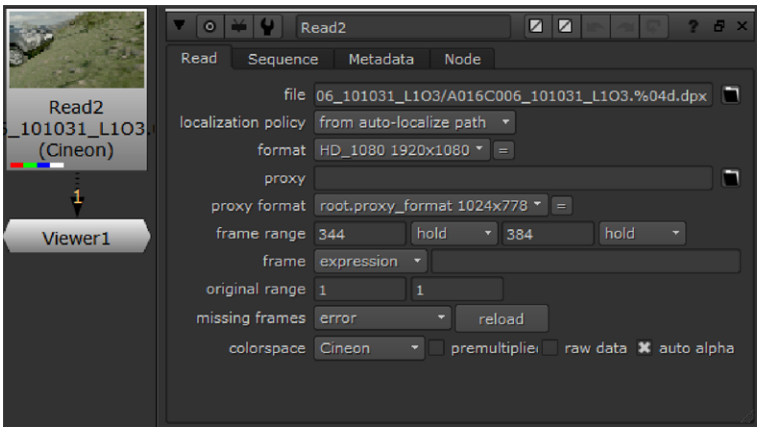
To import an image sequence using Nuke's file browser:

1. Select **Image** > **Read** (or press **R** over the Nuke Node Graph).

Tip: Pressing **R** with an existing Read node selected, opens the file browser at the location specified by that node.

2. Browse to the image sequence you want to import. For instructions on using the file browser, see Using the File Browser. Select the file you want to open. If you want to open several files at the same time, **Ctrl**/**Cmd**+click on the files, and then click **Open**.

   A Read node is inserted in the Node Graph.



   Nuke imports the image sequence and displays it as a thumbnail on the Read node. Generally, the Read node does not reformat or resize the sequence in any way, and the node's properties panel is updated to display the native resolution and the frame range for the sequence. Note that the **format** and **proxy format** fields in the controls indicate the format of the images, they do not cause the images read from files to be resized to this format.

3. You can cycle through the available versions of a file using the **Alt**+**Up**/**Down** arrow keys. If you want to skip ahead to the highest version available, press **Alt**+**Shift**+**Up** arrow key.

   Versions must written in the following format in order for Nuke recognize them:

   `../myFiles/grades/A003_C007_071105_`**`v01`**`_001.r3d`
   `../myFiles/grades/A003_C007_071105_`**`v02`**`_001.r3d`
   `../myFiles/grades/A003_C007_071105_`**`v03`**`_001.r3d`

4. If your sequence has a red, green and blue channel but no alpha channel, check the **auto alpha** box in the Read node control panel to set the alpha channel to 1. This prevents possible problems from occurring if Nuke tries to read the alpha channel and one doesn't exist. The **auto alpha** box is unchecked by default.

Note: Nuke reads images from their native format, but the Read node outputs the result using a linear colorspace. If necessary, you can change the Colorspace option in the Read node's properties panel, or insert a **Color** > **Colorspace** node to select the color scheme you want to output or calculate.

Note: The maximum image size the Nuke Viewer can display is 2^32 = 4,294,967,296 pixels. This is the same as 64k x 64k, 128k x 32k, 256k x 16k, or 512k x 8k. If your image is larger than this, it is resized and you get the following warning:
"Viewer image is clipped to <size> x <size>!"
For example, if your image resolution is 60,000 x 4473, Nuke is able to display the image because the number of pixels is less than 2^32. However, if the resolution is 110,000 x 50,000 (more than 2^32 pixels), the image is resized to 85,899 x 50,000.
In addition to the Viewer, this limit is also applied to the bounding box of the images being passed between each node.

5. If you have a proxy version of the image sequence, click the **proxy** field's folder icon  and navigate to the proxy version. Select **Open**. If you don't have a proxy version, don't worry: Nuke creates one on the fly according to the proxy scale or proxy format settings you specified in the project settings.

   The proxy file does not need to match the proxy resolution in use. Depending on your settings, either the full-res or proxy file is scaled to the required proxy size. For more information, see Proxy Mode.
6. You can access the metadata contained within the read file by clicking the **Metadata** tab. Once you know which keys exist in the file, you can reference them in expressions. See Expressions for more information.

   The metadata displayed depends on the file type. For example, a **.jpg** might only contain **input/** keys, whereas QuickTimes contain **input/** and **quicktime/** keys. See Working with File Metadata for more information.

To Import an Image Sequence from an External File Browser

To load an image, you can also drag and drop the image into the Node Graph from an external file browser (such as Windows Explorer or Mac Finder). To load an entire image sequence this way, drag and drop the directory that contains the images into the Node Graph.

Notes on Importing MOV Files

When reading **.mov** files, Nuke attempts to select the "best fit" combination by reading an extended list of metadata key/value pairs from the QuickTime header, including nclc atom, gama atom, and ProRes codec headers. You can manually override the following **MOV Options** in the Read node **Properties** panel:

Control	Description
YCbCr matrix	Sets the way Y'CbCr is converted to RGB. **Rec 601** and **Rec 709** follow the ITU.BC specifications.
Data Range	Sets the minimum and maximum color range values for decoded DNxHD files. If your source file was encoded with a reduced range (0-240 for 8-bit or 0-960 for 10-bit), set source range to **Video Range** to decode the full range (0-255 for 8-bit or 0-1023 for 10-bit).
first track only	When enabled, only the first view present in the file is displayed, regardless of how many views the file contains. When disabled, and there is more than one view set up in the project settings, you can switch between views using the buttons above the Viewer.
[Metadata] Do not attach prefix	When enabled, the metadata accompanying the file is saved without the **quicktime** prefix. **Note:** QuickTime container-specific metadata always carries the prefix. When disabled, all metadata is prefixed with **quicktime** for easy identification.
[Metadata] Match key format	When enabled, Nuke attempts to clean up meta keys into the standard Nuke metadata format. When disabled, the metadata is passed down the Node Graph unchanged.
Video Codec	Displays the codec used to encode the clip.



Note: When reading **.mov** files, Nuke looks for metadata in the following order and uses it to govern the settings on the Read node, falling down to each level when the level above is unavailable or set to a reserved or unknown value:

1. Foundry -specific user data
2. prores header
3. nclc atom
4. gama atom
5. defaults based on codec type

Certain **.mov** files may appear different in Nuke relative to Apple's Final Cut Pro, because Final Cut Pro introduces a gamma compensation based on assumptions about the content of the files and the viewing environment.

To limit the number of background processes that Nuke can run when reading **.mov** files, go to **Preferences** and set the number for **QuickTime decoders to use** in the **Performance** > **Threads/Processes** tab.

Notes on Importing AVI Files

.avi files can be supported by default or only via Nuke's reader that is based on the FFmpeg open source library. If you get an error when using **.avi** files in Read nodes, you may need to use the prefix **mov64:** before the file path and file name, for example, **mov64:\z:\job\FILM\IMG\final_comp_v01.####.avi**.

Notes on Importing OpenEXR Files

Nuke supports multi-part OpenEXR 3.3.6 images, which allow you to store your channels, layers, and views in separate parts of the file in order to speed up processing. You can load multi-part OpenEXR files in exactly the same way as single-part OpenEXR files.

The OpenEXR file format allows the display window to have the lower left corner in any position. Unfortunately, Nuke needs all formats to have a lower left corner coordinate of 0,0. In the Read node control panel, under **exrOptions**, you can check the **offset negative display window** box to tell Nuke to offset the image so that the display window left side x coordinate is 0. If you uncheck the box, Nuke shrinks the format from both sides the amount that it's negative in the x coordinate, in other words treat the area as overscan.

By default, the **exr** prefix is attached to metadata keys to make them distinct from other metadata in the tree. If you'd rather read metadata in "as is" without attaching a prefix, enable **do not attach prefix**.

When reading in **.exr** files, you can determine how pixels at the edge of the data window, or bounding box in Nuke terms, are treated using the **edge pixels** dropdown:

- **plate detect** - if the bounding box and format match exactly, then repeat all edges. Otherwise, add black at the edges.
- **edge detect** - for each matching edge, repeat the edge pixels. Add black at mismatched edges.
- **repeat** - always repeat the edge pixels outside the bounding box.
- **black** - always add black pixels outside the bounding box.

On Linux, the Nuke OpenEXR reader uses a memory-mapping function to improve performance reading PIZ-compressed **.exr** files. However, some customers have experienced hanging when reading large (frame size and number of channels) PIZ-compressed **.exr** files across an NFS network. If you experience this problem, you can tell Nuke not to use the mmap function by enabling this option. You can set it on a case-by-case basis or use a knobDefault in your **init.py** to always have it disabled. For more information on knobDefault, see the Nuke Python documentation (**Help** > **Documentation**).

Notes on Importing PSD Files

When loading a layered **.psd** file, a button called **Breakout Layers** appears under **psd Options** in the Read node control panel. Clicking this "breaks out" the **.psd** file into separate layers using some Shuffle nodes, and recombines these layers with a number of PSDMerge nodes. Each layer is grouped and labeled using a Backdrop node for clarity. In addition, a Crop node is inserted between the Shuffle and PSDMerge nodes, allowing you to adjust the bounding box for each layer individually.

The PSDMerge node is a type of merge node exclusive to this feature. In the control panel, there is an **operation** dropdown menu, a **mask** field (with an **invert** checkbox) and a **mix** slider. If the blend mode for a layer was set in Photoshop®, Nuke automatically sets this in the **operation** dropdown.



Note: The blend modes in PSDmerge are approximated and do not match Photoshop® exactly.

Naming Conventions

The file names of image sequences generally end in a number before the extension, for example image0001.rgb, image0002.rgb, image0003.rgb, and so on. When browsing for files like this, you may notice that the sequence appears as **image####.rgb**. Here, **####** is Nuke's way of indicating that the number is in a 4-digit incremental format. For a 3-digit format, such as image001**.rgb**, the frame number variable would be **###**.

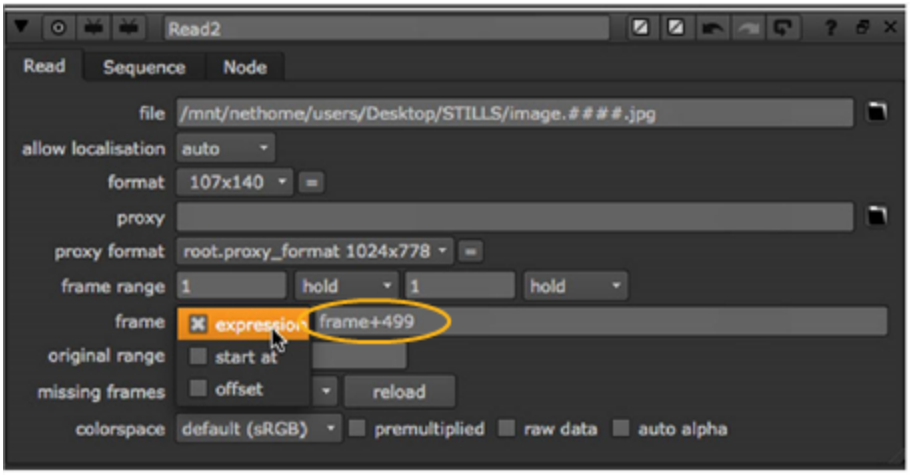
Nuke's file browser also understands unpadded file names, such as **image1.rgb**, **image2.rgb**, **image3.rgb**, and so on. They appear as **image#.rgb**.

Changing the Relation Between the Current Frame and the Frame Read In

By default, Nuke assumes an exact relation between the current frame processed, and the frame read in. For example, at frame 15, Nuke reads in image.0015.rgb. However, you can change this behavior using the **frame** parameter on the Read node. For instance, if you have a sequence that runs from image.0500.rgb to image.1000.rgb, you may want to read in image.0500.rgb at frame 1. Nuke lets you do this using expressions, specified start frames, and constant offsets. Each method is described below.

Using Expressions

1. Select **Image** > **Read** to import an image sequence.
2. In the Read node controls, set the **frame** dropdown menu to **expression**. Enter an expression in the field on the right. The expression changes the relation between the current frame and the frame read in.



For example, if your clip begins from image.0500.rgb and you want to place this first frame at frame 1 rather than frame 500, you can use the expression **frame+499**. This way, 499 frames are added to the current frame to get the number of the frame that's read in. At frame 1, image.0500.rgb is read in; at frame 2, image.0501.rgb is read in; and so on.

Another example of an expression is **frame*2**. This expression multiplies the current frame by two to get the number of the frame that's read in. This way, only every other frame in the clip is used. At frame 1, image.0002.rgb is read in; at frame 2, image.0004.rgb is read in; at frame 3, image.0006.rgb is read in; and so on.

Specifying a Start Frame for a Clip

1. Select **Image** > **Read** to import an image sequence.

2. In the Read node controls, set the **frame** dropdown menu to **start at**. Enter a start frame number in the field on the right. This specifies the frame where the first frame in the sequence is read in. In other words, all frames are offset so that the clip starts at the specified frame.

   For example, if your sequence begins from image.0500.rgb and you enter **1** in the field, image0500.rgb is read in at frame 1. Similarly, if you enter **100** in the field, image0500.rgb is read in at frame 100.

Offsetting All Frames by a Constant Value

1. Select **Image** > **Read** to import an image sequence.
2. In the Read node controls, set the **frame** dropdown menu to **offset**. Enter a constant offset in the field on the right. This constant value is added to the current frame to get the number of the frame that's read in.

   For example, if your clip begins from image.0500.rgb and you want to place this first frame at frame 1 rather than frame 500, you can use **499** as the constant offset. This way, 499 is added to the current frame to get the frame that's read in. At frame 1, image.0500.rgb is read in; at frame 2, image.0501 is read in, and so on.

   You can also use negative values as the constant offset. For example, if you use the value -10, Nuke subtracts ten from the current frame to get the frame that's read in. At frame 20, image.0010.rgb is read in; at frame 21, image.0011.rgb is read in; and so on.

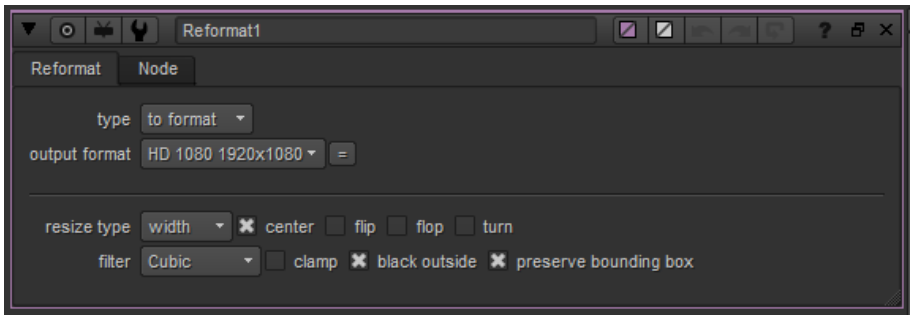
Reformatting Image Sequences

When you import image sequences, Nuke stores their format settings and makes them available to the Reformat node. You can then use the Reformat node to resize and reposition your image sequences to a different format. Reformat nodes also allow you to use plates of varying image resolution on a single script without running into issues when combining them.

To Insert a Reformat Node:

1. Make sure the Read node you added is currently selected.
2. Select **Transform** > **Reformat**.

   The Reformat node is inserted in the script, and its properties panel opens.



3. From the **output format** dropdown menu, select the format to which you want to output the sequence. If the format does not yet exist, you can select **new** to create a new format from scratch. The default setting, **[root.format],** resizes the image to the format indicated on the **Project Settings** dialog box.
4. You can now use the same Reformat node for any other Read nodes in the script. Simply select the Reformat node and **Edit** > **Copy**. Select another Read node in the script and **Edit** > **Paste**.

Image Caching

To ensure fast playback, Nuke uses several ways of caching data. Some of these include the following:

- The Viewer cache (also referred to as disk cache, which shares the DiskCache node's location) saves the scanlines of the displayed image to the disk cache directory. This location can be set in the **Preferences** panel (see The Cache Directory and Defining the Settings for Caching). When the Viewer displays an image, it reads from this cache of pre-rendered scanlines. This way, Nuke can quickly display image data when returning to a previously cached set of scanlines. You can specify the total amount of space for disk cache in **Preferences** > **Performance** > **Caching** > **comp disk cache size** . If the disk cache needs more space, the cache that was least recently used is disposed of to free more space. You can clear all of the disk cache by selecting **Cache** > **Clear Disk Cache** from the top menu bar.

Note: Clearing the Viewer cache (disk cache) also clears the DiskCache node data.

- The Buffer cache stores scanline data as calculated at selected intermediate stages in the Node Graph. This speeds up the end result of the Node Graph by caching parts of the Node Graph that haven't changed. Nuke automatically determines these intermediate cache points based on several things, including the Node Graph structure, which properties panels are open, and so on.

Tip: You can also force this behavior manually by selecting the **cached** checkbox in the node properties panel.

The Buffer cache uses RAM and you can specify the total amount of space in **Preferences** > **Performance** > **Caching** > **comp cache size (%)**. When the buffer cache needs more space, it disposes of the first cached data that was stored (first in, first out). To clear all of the buffer cache you can either select **Cache** > **Clear Buffers** from the top menu bar, or press **F12** on the keyboard.

- The Playback cache also helps improve smooth playback. Playback cache uses RAM to temporarily store the frames in the Viewer. When you first play back a clip in the Viewer, an orange bar appears automatically displaying the progress of the playback cache. After the playback cache is complete (and the orange line is complete) the frames play back smoothly. You can temporarily disable playback caching by either selecting the pause button above the Viewer or by pressing **P**. To clear playback cache select **Cache** > **Clear Playback Cache**.
- You can use the DiskCache node to cache parts of the Node Graph. This caches scanlines to disk from its input as they are requested by its output. DiskCache data is stored in the same place as the Viewer cache data and can share the data provided the Viewer is operating in full float (32-bit). See Using the DiskCache Node for more information.



Note: You can clear all caches by selecting **Cache** > **Clear All** from the menu bar.

- If you find that loading files from a remote server slows down your processing, you can localize the files that you're working with for speedier processing. See Localizing Files for Better Performance.

The Cache Directory

Both the automatic caching and the DiskCache node use the same cache directory defined using the **Preferences** dialog. In the Preferences, you can also set the maximum size you allow the disk cache to consume. For more information on the caching preferences, see Defining the Settings for Caching below.

Note that the cached images have unique names reflecting their point of output location in the script. This means that you can cache images from multiple nodes in the script without overwriting previously cached images.

Defining the Settings for Caching

You can define the settings for caching in the **Preferences** > **Performance** > **Caching**.

1. Select **Edit** > **Preferences** (or press **Shift**+**S**).
   The **Preferences** dialog displays.

2. Under **Caching** > **Disk Caching**, specify where you want Nuke to cache out data to disk. Pick a local disk (for example, **c:/temp**), preferably with the fastest access time available.

   The environment variable NUKE_DISK_CACHE can be used to override this setting. For more information, see Environment Variables.
3. Using the **comp disk cache size** control, pick the maximum size the disk cache can reach. Ensure there is enough space on the disk for this to be reached. The default value is **10 GB**, but a larger value, such as 50 GB, can often be used.

   The environment variable NUKE_DISK_CACHE_GB can be used to override this setting. For more information, see Environment Variables.
4. Click **OK** to update preferences and then restart Nuke.

Once these settings are defined, the automatic caching of images is enabled. The Viewer caches each frame it displays in the directory specified. If you add a DiskCache node into your script, it also uses the same directory.

The disk cache is preserved when you quit Nuke so it can be used again later. However, when the cache becomes full, old items are automatically deleted to make room for newer items.

Clearing the Disk Cache

When the disk cache becomes full, old items are automatically deleted. If necessary, you can also empty the disk cache manually. You may want to do this if, for some reason, wrong images are displayed in the Viewer.

To Empty the Disk Cache

From the menu bar, select **Cache** > **Clear Disk Cache**.

Using the DiskCache Node

The DiskCache node caches to disk scanlines from its input as they are requested by its output. This can be useful, for example, if:

- you are working on a large, complex node tree. Using the DiskCache node, you can break the node tree into smaller sections and cache any branches that you are no longer working on.
- you are reading in images from a network. If you insert a DiskCache node after a Read node, the image is cached locally and displayed faster.

• you are painting or rotoscoping. If you insert a DiskCache node before a RotoPaint node, flipping frames becomes faster.

The cached images are saved in the same directory as the images the Nuke Viewer caches automatically. You can set the location and size of this directory in the Preferences. For more information, see Defining the Settings for Caching.



Note: Even though the DiskCache node and the automatic Viewer caching use the same cache directory, they do not share the same cache files. Therefore, using a DiskCache node does not create cache files for the Viewer and does not necessarily speed up playback. Instead, if placed at strategic, expensive parts of a node tree, it can speed up calculations, as Nuke can reference the cached data rather than recalculate it.
Unlike the images in the Viewer cache, the images created by the DiskCache node affect your rendered output and are always saved as full floating point images.
If you make a change in the nodes upstream, the affected cached images are discarded and automatically recalculated.



Note: When executing a script on the command line, the DiskCache nodes are NOT automatically executed.

To Cache Images Upstream

1. Set the zoom level in the Viewer. By default, only the lines displayed in the Viewer are cached.
2. Select **Other** > **DiskCache** to insert a DiskCache node after the last node in the section of the node tree that you want to cache.
3. From the **channels** dropdown menu, select which channels to cache.

   Nuke caches the selected channels of the current frame at the current zoom level. From this point on, Nuke references the cached data instead of constantly recalculating the output of the preceding nodes.

   As you pan and zoom around, new parts of the image are cached.
4. If you want to cache more than the current frame and zoom level, click the **Precache** button in the DiskCache properties and enter a frame range in the dialog that opens.

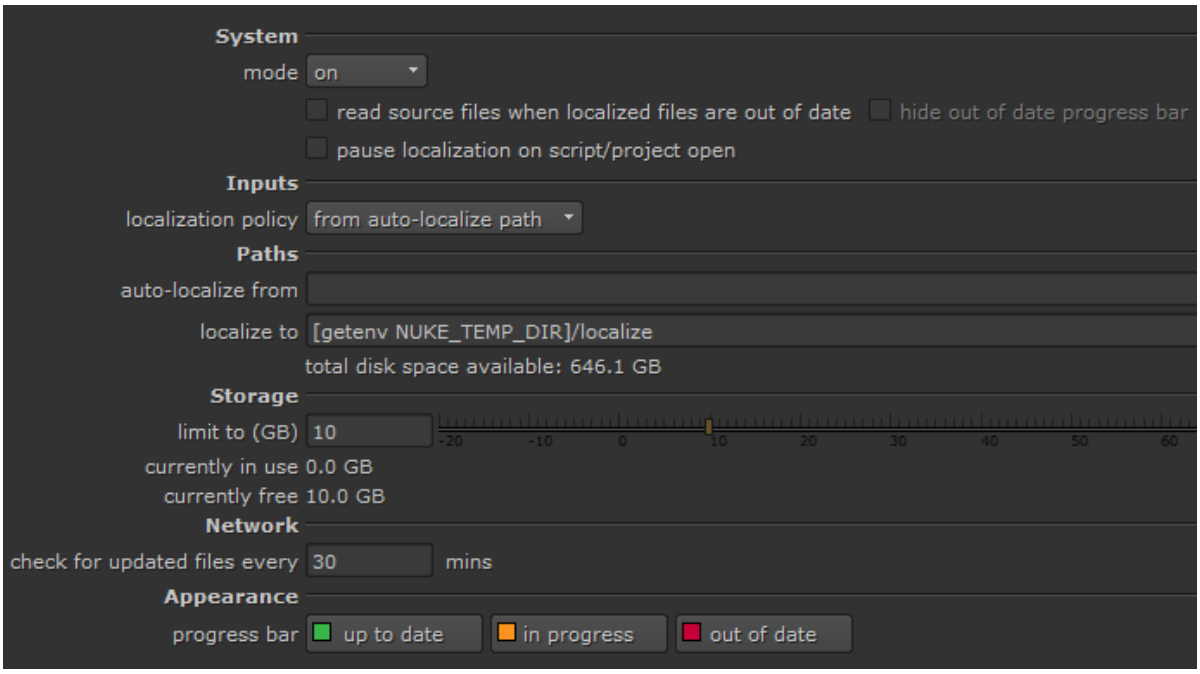
   This forces Nuke to cache all frames specified. All lines are cached regardless of what is shown in the Viewer. Where the required images are partly cached already, Nuke only calculates what is missing.

Localizing Files for Better Performance

You may find that sometimes loading files from a remote server slows down processing. Nuke has the facility to cache files locally, either individually or by setting an automatically localized folder (NUKE_TEMP_DIR/localize, by default), to help guarantee playback stability. Local caching is controlled initially in the **Preferences** dialog, then on a file-by-file basis.

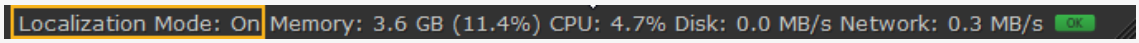
Setting Localization Preferences

1. Press **Shift**+**S** to open the **Preferences** dialog and navigate to **Performance** > **Localization**.



2. Set the required localization **mode:**
   - **on** - checks for updates to source clips for all localization policies, and localizes those files set to **On** or **From auto-localize path** automatically. **On demand** source clips must always be localized manually.
   - **manual** - checks for updates to localized source clips and prompts you to update them manually. Only those with the policy set to **off** are not updated.
   - **off** - no source clips are localized, regardless of the their localization policy.

Note: The current localization **mode** is displayed in the status bar at the bottom-right of the interface.



3. Set the default **localization policy** for new Read nodes using the dropdown:

Note: The **localization policy** for existing Read nodes in the script must be set individually. See Managing Localization for more information.

- **on** - always localize Read nodes with this policy.
- **from auto-localize path** - localize these Read nodes automatically if they reside in the **auto-localize from** directory.
- **on demand** - only localize these Read nodes when you manually update them. See Updating On Demand Clips for more information.
- **off** - never localize these Read nodes.

4. Enter a file path for **auto-localize from**, if required.

   Any files that reside in this directory are automatically cached when brought into Nuke, providing that the Read node's **localization policy** is set to **from auto-localize path**.

Note: Localization in both the timeline and Node Graph is paused during compositing Viewer playback so that performance is not affected.

5. Enter a file path for **localize to**. Leaving this field as the default creates a sub-directory in the **Temp Directory** as the local cache.

Note: On Windows, files saved to the **localize to** directory replace **** (double back slashes) and **:** (colon drive signifiers) with underscores so that the file path works as expected between operating systems. For example:

\\windowspath\to\my\network\file.dpx is saved as **__windowspath\to\my\network\file.dpx**
t:\my\network\path\file.dpx is saved as **t_\my\network\path\file.dpx**

6. Enter a value for **limit to (GB)** to control how much disk space is available in the cache directory.

Note: Negative values in this field reserve the specified amount of space at all times. For example, -2 stops 2 GB of memory being used for caching.

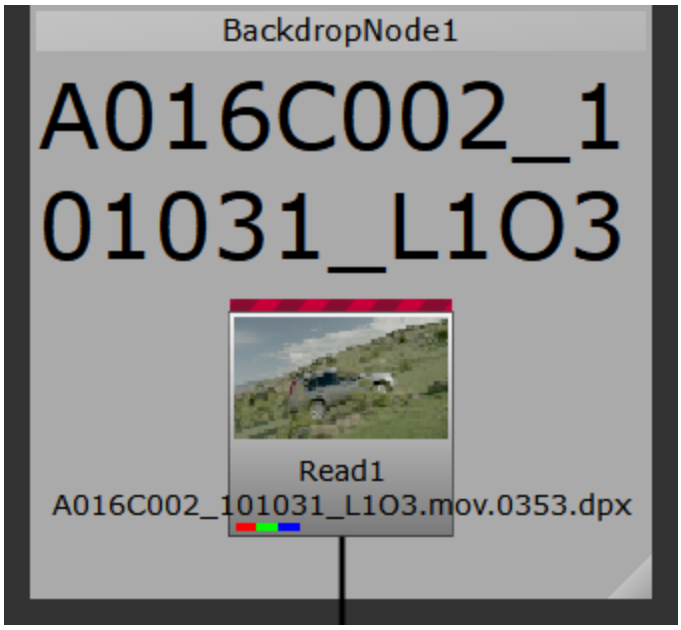
7. You can specify the time interval (in minutes) before localized files are checked for updates using the **check for updated files every ## mins** control.

   The default setting checks for new versions of files every 30 minutes, but you can set this control to any value. Any Read nodes that have changed since the last update check are flagged red in the Node Graph.

Tip: You can change the default localization indicators using the **Appearance** controls.



If **read source files when localized files are out of date** is enabled, Read nodes referencing cached files that have changed since they were localized revert to reading the source files. Read nodes that are reading from the source files are marked with a striped red bar on the Read node.



Enabling **hide out of date progress bar** hides the localization state of out of date files so that the Read node appears the same as regular Read nodes.

Note: The out of date localized files are not discarded, disabling **read source files when localized files are out of date** picks up the out of date files instead of reading the source files.

Managing Localization

As well as the overall **Preferences** for when new Read nodes should be localized, you can set localization on a file-by-file basis. The **localization policy** for any existing Read nodes in your project must be managed individually.

Tip: If you find that localization is slow to copy files, you can increase the number of threads that Nuke uses to process jobs. Set the NUKE_LOCALIZATION_NUMWATCHERS environment variable to the number of threads you want to use. See Environment Variables for more information.

1. Open the Read or ReadGeo node's **Properties** panel.
2. Set the **localization policy** dropdown to one of the following options:
   - **On** - the files are localized, regardless of location, as long as the **limit to (GB)** limit is not breached.
   - **From auto-localize path** - the files are localized if they reside in the **auto-localize from** directory, as long as the **limit to (GB)** limit is not breached.

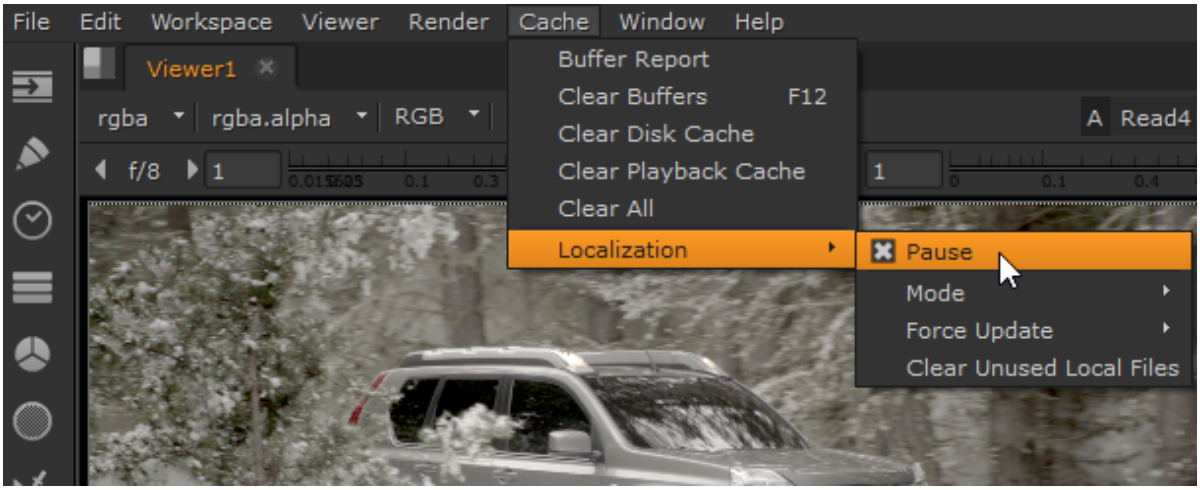
- **On Demand** - the files are only localized when you update them manually. See <XREF> for more information.
- **Off** - the files are never localized, regardless of location.

As Read nodes are localized, an amber progress bar displays in the thumbnail. Fully cached clips are marked with a green bar at the top of the thumbnail and out-of-date clips are marked in red.

Note: Container formats, such as **.mov** and **.r3d**, do not display progress bars during localization. The Read node only shows the green indicator once the entire file is localized successfully.



3. If you need to pause localization temporarily, navigate to **Cache** > **Localization** and select **Pause**.



4. If you find that your cache is filling up regularly, you can:
   - increase the amount of available space for localization by raising the **limit to (GB)** preference,
   - navigate to **Cache** > **Localization** > **Delete Unused Local Files** (see Clearing Localized Files), or
   - manually clear files from the cache directory in NUKE_TEMP_DIR/localize, by default.
5. You can force Nuke to check for updated source files, rather than waiting for the **check for updated files every** interval to expire, by selecting **Force Update** for **All** files, just **Selected** Read nodes, or **On Demand only**.

Note: Each file has its own update time, which is reset whenever the source files are checked.

The following table is a quick reference guide to when and how Read nodes are localized, where:

- **green** - Read nodes are localized automatically.
- **amber** - Read nodes are localized when updated manually.
- **red** - Read nodes are not localized.

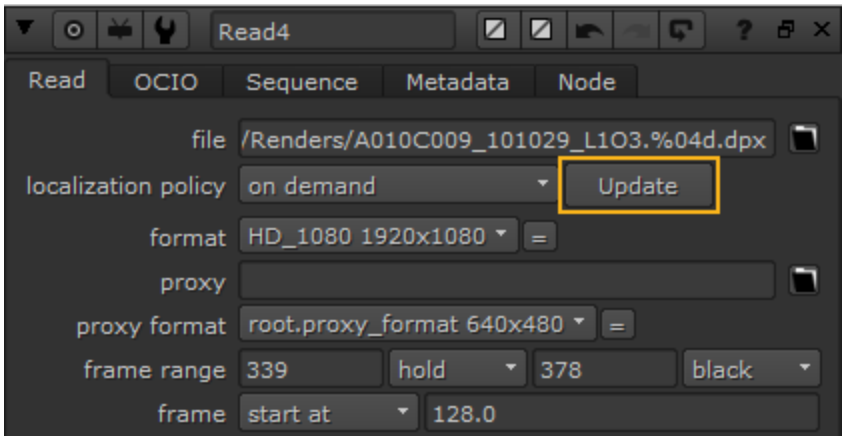
System Preference	Read Node Preference			
	on	auto-path	on demand	off
on				
manual				
off				

Updating On Demand Clips

Read nodes with their **localization policy** set to **on demand** are polled to check for updates at the interval set in the **Preferences**. If the file has changed, the Read node is marked with a red bar to show it is out of date.

To update a single **on demand** Read node:

1. Double-click the Read node in the Node Graph to display its **Properties**.
2. Click **Update**.

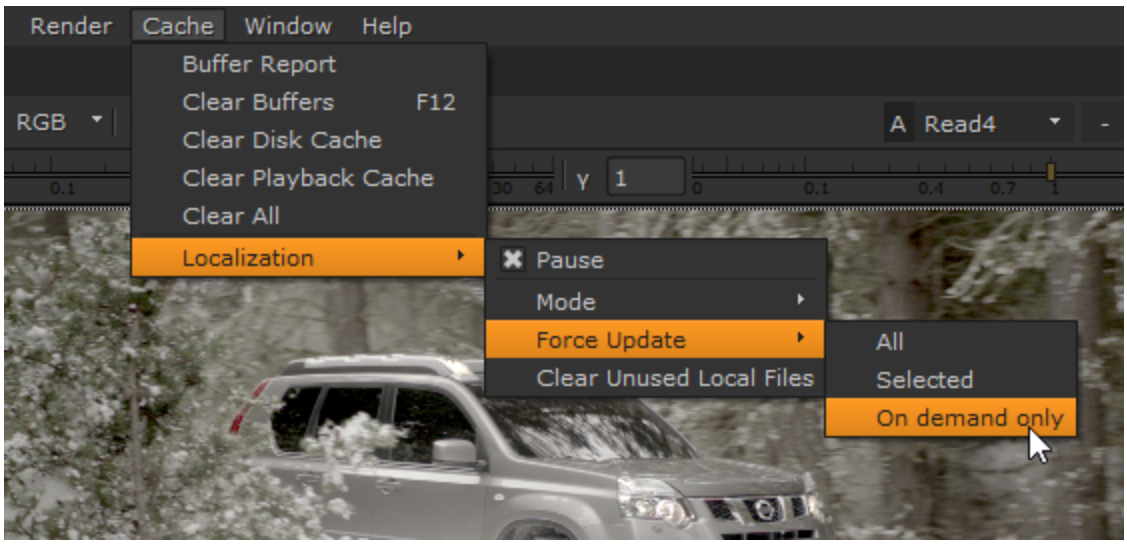


Note: If a Read node's localization policy is set to **on demand** and it hasn't been localized previously, clicking the **Update** button localizes the file.

The local copy of the files is updated from the remote source files.

To update all **on demand** Read nodes:

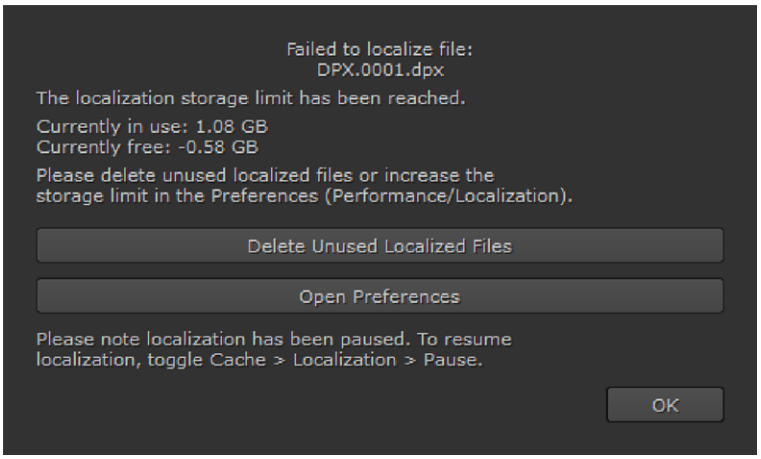
1. Navigate to **Localization** > **Force Update**.
2. Click **On demand only**.



The local copy of all **on demand** Read nodes is updated from the remote source files.

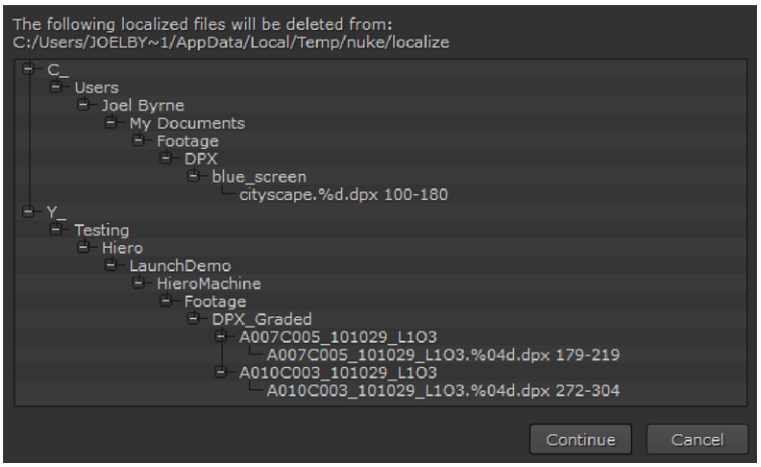
Clearing Localized Files

Localizing a large amount of files can fill up the localization cache quite quickly if you leave the **limit to (GB)** preference at the default 10 GB. When the cache runs out of space, a **Failed to Localize File** dialog displays and localization pauses.



You can delete localized files by clicking **Delete Unused Local Files** (or by navigating to **Cache** > **Localization** > **Clear Unused Local Files**). Nuke displays a dialog containing all the files that are marked for delete.

Tip: You can also open the **Preferences** from this dialog to adjust the localization behavior, such as increasing the **limit to (GB)** preference.



Click **Continue** to delete the localized files or **Cancel** to keep the cached files.

Saving Scripts and Recovering Back-Ups

You know the mantra: save and save often.

Nuke provides three ways to save your scripts, or comps, making it easy to version them. There's also an automatic timed backup, which you can turn off if you're feeling brave - but we sure don't recommend it.

Saving Scripts

There are three ways of saving scripts:

- To save a new script, select **File** > **Save Comp as** (or press **Shift**+**Ctrl**/**Cmd**+**S**).
- To update changes to a script already saved, **File** > **Save Comp** (or press **Ctrl**/**Cmd**+**S**).
- To save and upgrade to the next version, **File** > **Save New Comp Version** (or press **Alt**+**Shift**+**S**).

To Save a Script

1. Select **File** > **Save Comp as**.

   The **Save script as** dialog opens.
2. Browse to the directory where you want to store the script. For instructions on using the file browser, see Using the File Browser.
3. In the field in the bottom of the dialog, enter a name for the script after the folder path, for example **firstscript_v01.nk**.
4. Click **Save**.

Tip: The **_v01** string in the end of a script name allows you to use the **Save New Comp Version** feature. Selecting **File** > **Save New Comp Version** saves the current version of your script and increments its name (that is, saves the different versions under different names using _v01, _v02, _v03, and so on, in the end of file names). This only works when the file name includes a number that can be incremented.

Automatic Back-Up of Scripts

You can define where and how often Nuke makes automatic back-ups your files, or turn off the autosave function.



Article: See Knowledge Base article Q100158 for more information on incremental autosaves.

To Define Autosave Options for a Script

1. Select **Edit > Preferences**.

   The **Preferences** > **General** panel opens.
2. Edit the following settings:
   - **idle comp autosave after** - to define how long (in seconds) Nuke waits before performing an automatic back-up after you have left the system idle.
   - **force comp autosave after** - to define how long (in seconds) Nuke waits before performing an automatic back-up regardless of whether the system is idle.
   - **autosave comp filename** - to define where and under what name Nuke saves your automatic back-up files. By default, the files are saved in the same folder as your project files with the extension **.autosave**. To change the autosave location, enter **/path/to/directory/[file tail [value root.name]].autosave** to use a custom file path.
3. Click **Save**.



Note: For the automatic back-up to work, you must save your script first so that the autosave can reference the file. We'd hate for you to lose your work, so please do this early on in the process!

To Turn off Automatic Back-Up

1. Select **Edit > Preferences**.

   The **Preferences** > **General** panel opens.
2. Set the **idle comp autosave after** and force comp autosave after fields to **0**.

   From now on, Nuke does not perform any automatic back-ups, and you are more likely to lose your work in the case of a system or power failure.

Recovering Back-Ups

After experiencing a system or power failure, you are likely to want to recover the back-up files created by Nuke's autosave function.

1. Relaunch Nuke.

   A dialog opens that asks you if you want to recover the autosave file.
2. Click **OK**.

   Nuke opens the back-up file for your use.

There may be times when you don't want to load the autosave file and rather need to load the last saved version. For example, consider a situation where you modified a script, but decided not to commit the changes and so exited Nuke without saving. In all likelihood Nuke autosaved some or all of your changes, in which case if you open the autosaved file you are not working on the original script, as intended. If you accidentally open an autosaved script, then simply close it and reload the last saved version.



Note: Breakpad crash reporting allows you to submit crash dumps to Foundry in the unlikely event of a crash.

By default, crash reporting is enabled in GUI mode and disabled in terminal mode. You can toggle reporting on and off using the **--crashhandling 1** or **0** option on the command line or by setting the environment variable NUKE_CRASH_HANDLING to 1 or 0.

When crash handling is enabled in GUI mode, you can control whether reports are automatically submitted or not using the **--nocrashprompt** command line option or by setting the environment variable NO_CRASH_PROMPT to 0.

Crashes in terminal mode are automatically submitted when crash handling is enabled.

Closing Scripts

To close a script:

1. Select **File** > **Close Comp** (or press **Ctrl**/**Cmd**+**W**).
2. If you have made any unsaved changes to the script, Nuke prompts you to select whether to save them. Click **Yes** to save your changes or **No** to ignore them.

   Nuke quits and relaunches, as though you ran it again. It does everything it does at start-up, apart from displaying the splash screen. Therefore, you can use **Ctrl**/**Cmd**+**W** as a quick way to clear memory, reread plug-in paths, and reload the **init.py** and **menu.py** files. (The **init.py** and **menu.py** files are files that Nuke runs at start-up and can be used for configuring Nuke. You may want to reload these files if you have made any changes to them.)

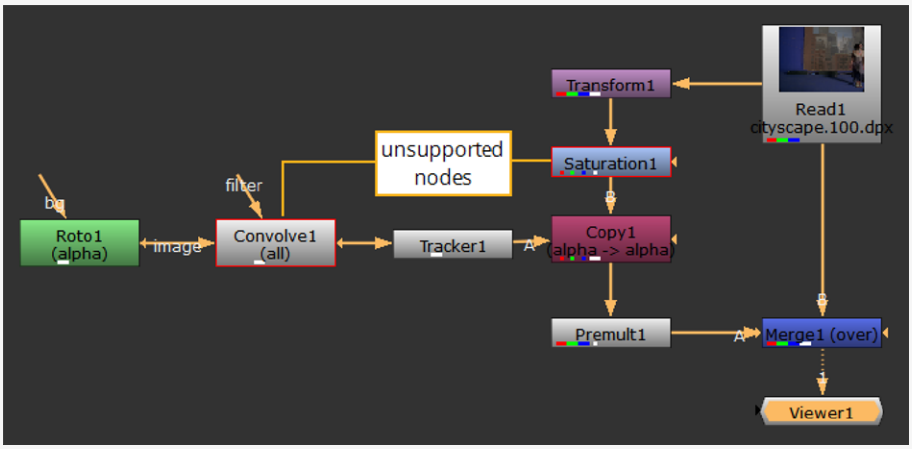
Loading Scripts

When you have built a script, or comp, and saved it and want to come back to it later, you need to load in an entire script file. You recognize Nuke's script files from the extension **.nk** (for example **firstscript.nk**).

1. Select **File** > **Open Comp** (or press **Ctrl/Cmd**+**O**).

   The **Script to open** dialog appears.
2. Browse to the script you want to open. For instructions on using the file browser, see Using the File Browser.
3. Click **Open**.

Note: Some NukeX plug-ins are not supported by Nuke, and likewise, some Nuke nodes are not supported in Nuke Assist. Unsupported plug-ins are displayed in the Node Graph with errors, and unsupported nodes are outlined in red.



The Viewer renders the output of the node tree whether the components are supported or not, but you cannot modify the output of unsupported plug-ins and nodes.

Defining Frame Ranges

Several dialogs in Nuke, such as the **Frames to render** and **Frames to flipbook** dialogs, prompt you for a frame range. To define one, you need to enter a starting frame and an ending frame, separated by a dash.

For example, to restrict an action to frames 1, 2, 3, 4, and 5, you would use **1-5** as the frame range.

The following table gives you more examples of frame ranges you can define.

Frame Range	Expands To
3	3
-3	-3
1 3 4 8	1, 3, 4, 8
1-10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
-3-4	-3, -2, -1, 0, 1, 2, 3, 4
-8--5	-8, -7, -6, -5
1-10×2 (frame range from 1 to 10 in steps of 2)	1, 3, 5, 7, 9
1-10×3 (frame range from 1 to 10 in steps of 3)	1, 4, 7, 10
1-4×1 8-10×1 12-14×1 (multiple ranges separated by spaces)	1, 2, 3, 4, 8, 9, 10, 12, 13, 14

You can use the above ways of defining a frame range everywhere in Nuke. In addition to dialogs, they can be used on the command line (where any frame ranges should be preceded by the -F switch) and in Python statements. For more information, see Command Line Operations and the Nuke Python documentation (**Help** > **Documentation**).

Reformatting Elements

These pages teach you how to reformat images through scaling, cropping, and pixel aspect adjustments. You will also learn to adjust bounding boxes to minimize processing and rendering times.

Using the Reformat Node

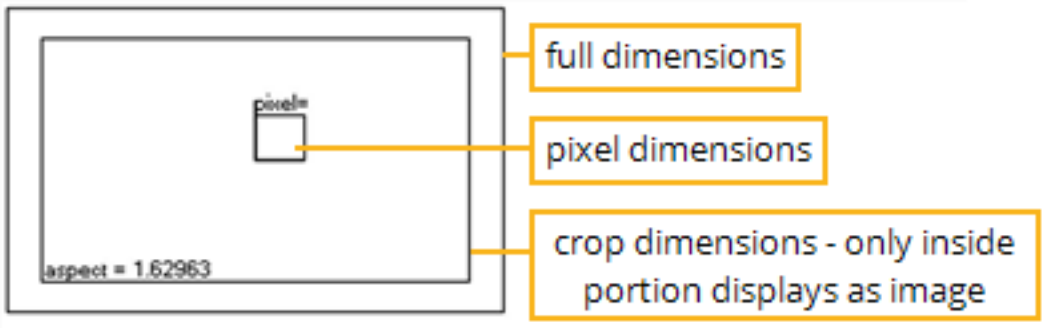
You can use the Reformat node for three different purposes:

1. To generate image sequences that match a desired image format in terms of both resolution and pixel aspect ratio (the width to height ratio of the format's individual pixels).
2. To create thumbnails (low resolution frames which you might post to the web in order to storyboard a sequence). The node scales the frame until it fits inside a rectangle whose dimensions you specify. It also sets pixel aspect ratio to one (square).
3. To scale images. The scale factor is rounded slightly so that the output image has an integer number of pixels in the direction you select in the Reformat node's controls.

Converting Images to a Desired Image Format

When you read in elements, Nuke stores their format settings and makes them available to the Reformat node. You can then apply one of the existing formats to your images, or create, edit, and delete formats yourself.

When creating a new format from scratch, you define the overall resolution, the cropped resolution (optional) and the pixel aspect ratio. As you define these parameters, the Reformat operator graphically displays them for you in the manner shown below.



To Create a New Output Format:

1. Click **Transform** > **Reformat** to insert a Reformat node at an appropriate place in your script (generally before a Write node).
2. Connect a Viewer to the output of the Reformat node so you can see the effect of your changes.
3. Select **new** from the **output format** dropdown menu. The *New format* dialog appears.
4. Type a name for the new format in the **name** field.
5. In the **file size** fields, type the full output resolution (in pixels).
6. If you want to crop the full output resolution (for example, to create a letter box):
   - Check **image area**.
   - Increment the **x** field to define the left boundary of the crop. (The display updates to show you the left boundary of the crop relative to the full-size input.)
   - Increment the **y** field to define the bottom boundary of the crop.
   - Increment the **r** field to define the right boundary of the crop.
   - Increment the **t** field to define the top boundary of the crop.
7. If the destination display device for the image sequence uses non-square pixels, type the appropriate pixel aspect ratio in the **pixel aspect** field (for example, if your destination is a digital video display, type **1.1**).

Note: You can also add formats to Nuke via entries to the **menu.py** file:
1. Open the **menu.py** file (located in same directory as your Nuke executable).
2. Add an entry similar to the following example:

```
nuke.addFormat ("720 486 0 0 720 486 0.9 NTSC_video")
```

where the numbers specify, respectively, the format's full horizontal resolution, full vertical resolution, left crop position, bottom crop position, right crop position, top crop position, and pixel aspect ratio; and where the final text string designates the format's name.
3. Save and close the **menu.py** file. The next time you launch Nuke the format is available for selection from the **Project Settings** dialog, Reformat node properties panel, and elsewhere.

To Edit a Format:

1. Select the format you wish to edit from the **output format** dropdown menu.
2. From the same dropdown menu, select **edit**. The **Edit format** dialog appears.
3. Edit the **name**, **file size**, **image area**, and **pixel aspect** fields as necessary.
4. Click **OK** to save the changes to the format.

To Delete a Format:

1. Select the format you wish to delete from the **output format** dropdown menu.
2. From the same dropdown menu, select **delete**. The format is removed from the menu.

To Apply a Format:

1. If necessary, click **Transform** > **Reformat** to insert a Reformat node at an appropriate place in your script (generally before a Write node).
2. Connect a Viewer to the output of the Reformat node so you can see the effect of your changes.
3. From the **type** dropdown menu, select **to format**.
4. Select the format you wish to apply from the **output format** dropdown menu.
5. From the **resize type** field, select the method by which you want to preserve or override the original aspect ratio. Select:
   - **width** to scale the original until its width matches the format's width. Height is then scaled in such a manner as to preserve the original aspect ratio.
   - **height** to scale the original until its height matches the format's height. Width is then scaled in such a manner as to preserve the original aspect ratio.
   - **fit** to scale the original until its smallest side matches the format's smallest side. The original's longer side is then scaled in such a manner as to preserve original aspect ratio.
   - **fill** to scale the original until its longest side matches the format's longest side. The input's shorter side is then scaled in such a manner as to preserve original aspect ratio.
   - **distort** to scale the original until all its sides match the lengths specified by the format. This option does not preserve the original aspect ratio, so distortions may occur.
6. When cropping the output, check **center** to position the crop area at the center of the frame.
7. Select the appropriate filtering algorithm from the **filter** dropdown menu (see Choosing a Filtering Algorithm).
8. When scaling an image with Key, Simon, and Rifmen filters, you may see a haloing effect which is caused by pixel sharpening these filters employ. If necessary, check **clamp** to correct this problem.

Creating Thumbnails

1. Click **Transform** > **Reformat** to insert a Reformat node at an appropriate place in your script (generally before a Write node).
2. Connect a Viewer to the output of the Reformat node so you can see the effect of your changes.
3. From the type dropdown menu, select **to box**.
4. In the **width** and **height** fields, type the output dimensions. The units are pixels.
5. Use the **resize type** dropdown menu to select the method by which you preserve or override the original pixel aspect ratio. Select:
   - **width** to scale the original until its width matches the value in the **width** field. Height is then scaled in such a manner as to preserve the original aspect ratio (this means that the output you specified in **height** may not match the result).
   - **height** to scale the original until its height matches the value in the **height** field. Width is then scaled in such a manner as to preserve the original aspect ratio (this means that the output you specified in **width** may not match the result).
   - **fit** to scale the original until its smallest side matches the corresponding value in **width/height**. The longer side is then scaled in such a manner as to preserve the original aspect ratio.
   - **fill** to scale the original until its longest side matches the corresponding value in **width/height**. The smallest side is then scaled in such a manner as to preserve the original aspect ratio.
   - **distort** to scale the original until its sides match the values in the **width/height** fields. This option does not preserve the original aspect ratio, so distortions may occur.
6. Select the appropriate filtering algorithm from the **filter** dropdown menu (see Choosing a Filtering Algorithm).
7. When scaling an image with Key, Simon, and Rifmen filters, you may see a haloing effect which is caused by pixel sharpening these filters employ. If necessary, check **clamp** to correct this problem.

Scaling Image Sequences

1. Click **Transform** > **Reformat** to insert a Reformat node at an appropriate place in your script (generally before a Write node).
2. Connect a Viewer to the output of the Reformat node so you can see the effect of your changes.
3. From the type dropdown menu, select **scale**.
4. In the **scale** fields, enter scale factors for the width and the height. To scale each direction separately using different scale factors, click the **2** button.
5. Use the **resize type** dropdown menu to select the method by which you preserve or override the original pixel aspect ratio. Select:

- **width** to scale the original so that it fills the output width. Height is then scaled in such a manner as to preserve the original aspect ratio.
- **height** to scale the original so that it fills the output height. Width is then scaled in such a manner as to preserve the original aspect ratio.
- **fit** to scale the original so that its smallest side fills the output width or height. The longest side is then scaled in such a manner as to preserve the original aspect ratio.
- **fill** to scale the original so that its longest side fills the output width or height. The smallest side is then scaled in such a manner as to preserve the original aspect ratio.
- **distort** to scale the original so that both sides fill the output dimensions. This option does not preserve the original aspect ratio, so distortions may occur.

6. Select the appropriate filtering algorithm from the **filter** dropdown menu (see Choosing a Filtering Algorithm).
7. When scaling an image with Key, Simon, and Rifmen filters, you may see a haloing effect which is caused by pixel sharpening these filters employ. If necessary, check **clamp** to correct this problem.

Cropping Elements

To **crop** a frame is to cut out the unwanted portions of the image area.



The original image.



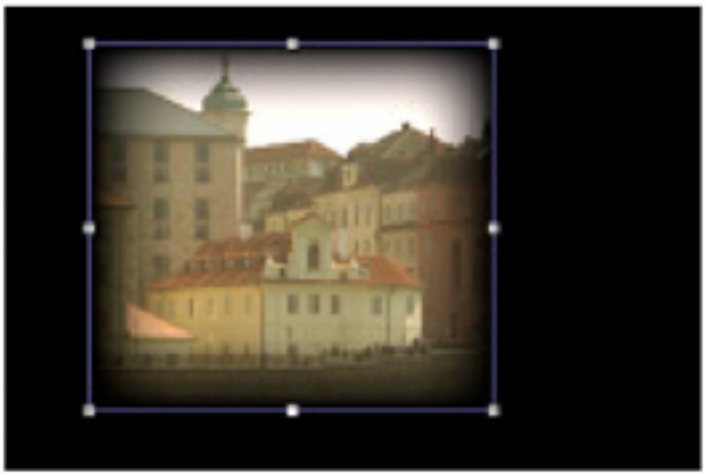
Cropping the image.

1. Click **Transform** > **Crop** to insert a Crop node at an appropriate place in your script.
2. Connect a Viewer to the output of the Crop node so you can see the effect of your changes.
3. Define the crop boundaries:
   - In the Viewer, drag on any side of the frame to reposition it.



- Or, in the Crop properties panel, increment or decrement the **box** field (**x** stands for left side, **y** for bottom side, **r** for right side, and **t** for top side).

4. To fill the cropped portion with black, check **black outside**. To fill the cropped portion by expanding the edges of the image, uncheck **black outside**. To adjust the image output format to match the cropped image, check **reformat**.
5. If you wish to vignette the edges of the cropped portion, increment the **softness** field.



Adjusting the Bounding Box

The bounding box defines the area of the frame that Nuke sees as having valid image data. The larger the bounding box is, the longer it takes Nuke to process and render the image. To minimize processing and rendering times, you can crop the bounding box. Occasionally, the bounding box may also be too small, in which case you need to expand it.

Note: Other Nuke functions, such as Transforms and Merges, can also affect the size of the bounding box, see Transforming in 2D and Merging Images for more information.

Resizing the Bounding Box

To adjust the bounding box, you can use the AdjBBox and CopyBBox nodes. The AdjBBox node crops and expands the bounding box edges, whereas with the CopyBBox node you can copy a bounding box from one input to another. If needed, you can also add a black outside edge to the bounding box using the BlackOutside node.

The AdjBBox node expands or crops the edges of the bounding box by a specified number of pixels.



An expanded bounding box.



A cropped bounding box.

Tip: You can enable a warning to indicate when the bounding box is greater that the format in Nuke's **Preferences**. See Bounding Box Warnings for more information.

For example, if you have an image with lots of black (0,0,0,0), you can adjust the bounding box to contain just the useful area so that Nuke won't waste time computing results where there is no change.

1. Select **Transform** > **AdjustBBox** to insert an AdjBBox node after the image whose bounding box you want to resize.
2. Connect a Viewer to the AdjBBox node, so you can see the effect of your changes.
3. In the AdjBBox controls, adjust the **Add Pixels** slider to increase or decrease the size of the bounding box. By default, 25 pixels are added to the edges of the bounding box.

   Nuke expands or crops the edges of the bounding box. If the bounding box is cropped, whatever is outside the bounding box area is replicated towards the edges of the image.



Copying a Bounding Box from One Input to Another

Some Nuke operations, such as a merge, can cause an expansion of the bounding box area because Nuke does not know that the extra area is going to be black or another constant color. Often, you can fix this by copying the bounding box from one of the inputs to the resulting image, thus cutting off this extra area. For this, you can use the CopyBBox node.

1. Select **Merge** > **CopyBBox** to insert a CopyBBox node after the node whose bounding box you want to use.
2. Connect the image whose bounding box you want to copy to the CopyBBox node's input A, and the image onto which you want to copy the bounding box to input B.

   Nuke copies the bounding box from input A to input B. Whatever is outside the copied bounding box area in image B gets replicated towards the edges of the image.



Adding a Black Outside Edge to the Bounding Box

If you adjust a bounding box with the AdjBBox or CopyBBox node, you may notice that whatever is outside the bounding box area gets replicated towards the edges of the image. If necessary, you can remove these replicated edge pixels and fill everything outside the bounding box area with black. To do this, use the BlackOutside node.



A cropped bbox with replicated edges.



The effect of the BlackOutside node.

1. Select the image whose edges outside the bounding box you want to fill with black.
2. Select **Transform** > **BlackOutside** to add a BlackOutside node in an appropriate place in your script. Nuke fills everything outside the bounding box area with black.

Bounding Box Warnings

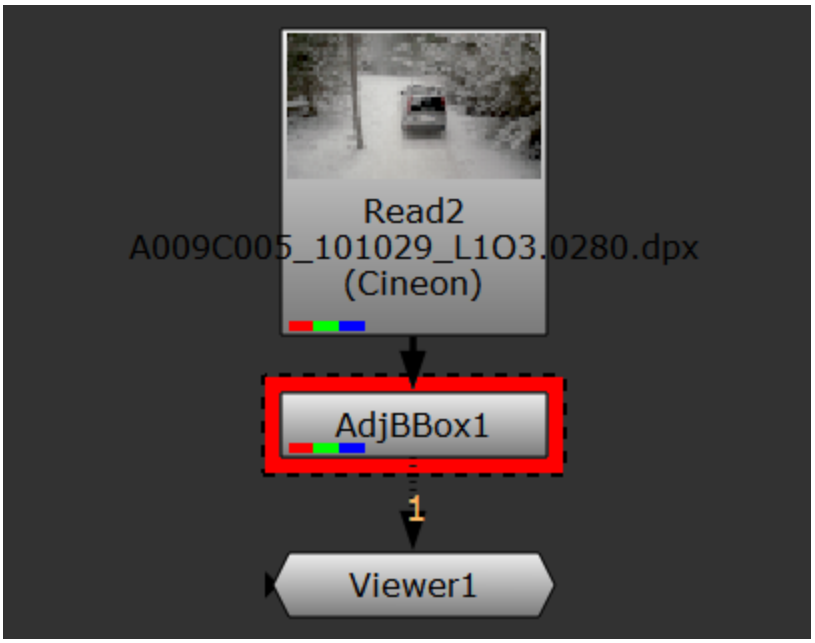
Zooming into the Viewer to work on a shot means that you can't always see the extent of the bounding box in relation to the format, which can result in unnecessary processing.

To make it easier to see the state of your bounding box, Nuke can display visual warnings on the nodes that affect the bounding box.

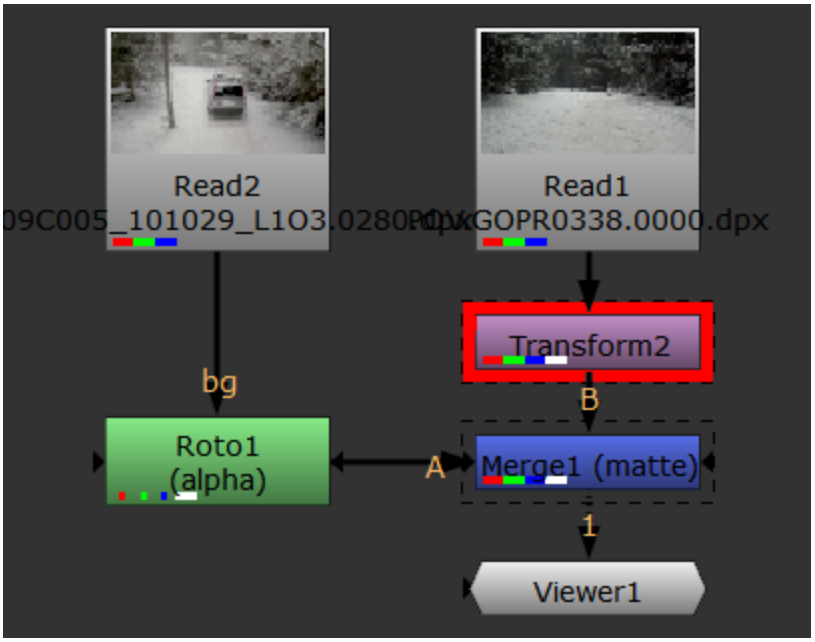
> **Warning:** If you enable the **Bounding Box Warning**, Nuke performs extra processing steps to identify problematic nodes, which may result in performance degradation.

To enable the warnings, in Nuke's **Preferences** under **Panels** > **Node Graph**, enable **Bounding Box Warning**:

• **red rectangle with dotted stroke** - the indicated node creates a bounding box greater than the format.



- **dotted stroke without the red rectangle** - the bounding box size is greater than the format at the indicated node, but the bounding box size has been set by an upstream node.



The bbox warning **threshold** controls how far past the edge of the format the bounding box can expand before the warning is displayed in the Node Graph. For example, if you're working with UHD_4K footage and the default 10% threshold, you can expand the bounding box horizontally by 384 pixels before the warning is displayed.

Tip: You can set the color of the warning rectangle in the **Preferences** under **Panels** > **Node Graph** > **Bounding Box Warning**.

Channels

Digital images generally consist of the four standard channels: red, green, blue, and alpha. Nuke allows you to create or import additional channels as masks, lighting passes, and other types of image data.

Introduction

A Nuke script can include up to 1023 uniquely named channels per compositing script. For example, you can combine multiple render passes from a 3D scene - an image from the red, green, and blue channels, a depth mask (z-depth channel), a shadow pass, a specular pass, lighting passes, and multiple mattes all stored within one image sequence in your composite.

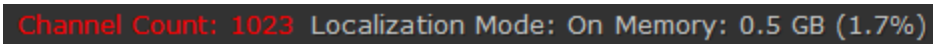
Note: When a script is saved, any channels that are not referenced in the script are discarded automatically.

When creating channels and layers, bear in mind these good practice guidelines:

- Ensure that all layers use the same channel names in the same order. This avoids complications with multilayer **.exr** files imported into Nuke.
- Always use proper names for channels, never just a single letter.
- Always create a custom layer for custom channels, don't add to the existing default layers.
- Never use more than four channels per layer. Nuke only has a four channel interface.

A current **Channel Count** is displayed in the bottom-right of the interface, which changes color as the number of channels increases. The default threshold is 1023, but you can set the limit in the Preferences under **Project Defaults** > **Channel Management** > **Channel Warning Threshold**.

The **Channel Count** turns yellow if you exceed the **Channel Warning Threshold** and red if the **Channel Count** is equal to or greater than the maximum channel value 1023.



Note: Nuke does not remove unused channels until you close and reopen a script, so the **Channel Count** does not decrease when you remove Read nodes from the Node Graph.

Reserved Channels

Nuke reserves certain channel names for internal use to ensure the smooth running of scripts under the hood. If you create a custom channel with a reserved channel name, the custom name is renamed automatically to preserve this functionality. For example, if you create a custom channel called **mask.a**, the channel is renamed **mask_extra**.

The reserved channels are:

rgba.red, **rgba.green**, **rgba.blue**, **rgba.alpha**

depth.Z

forward.u, **forward.v**

backward.u, **backward.v**

deep.front, **deep.back**

disparityL.x, **disparityL.y**, **disparityR.x**, **disparityR.y**

mask.a

rotopaint_mask.a

mask_planartrack.a

mask_splinewarp.a

Quick Start

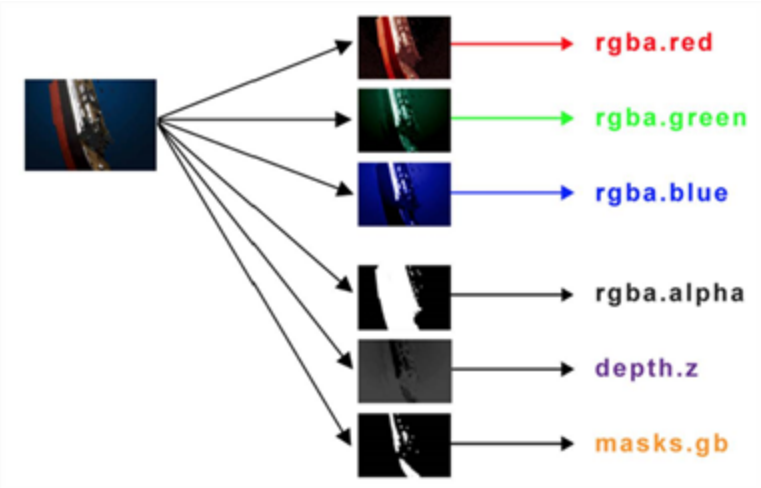
Here's a quick overview of the workflow:

1. Channels in Nuke are always a part of a layer. You can create new channels and layers using the **new** option in the channel selection dropdown menus (such as **output** and **mask**) in a node's properties panel. For more information, see Object Material Properties.
2. Using the channel selection controls you can select which channels the node is processing and outputting, or using as a mask when color correcting for instance. For more information, see Calling Channels and .
3. The channels can also be linked to other channel controls through the **Link menu**. For more information, see Linking Channels Using the Link Menu.

4. Using the Shuffle node, you can rearrange your input channels and apply the result in the output. For more information, see Shuffling Channels.

Understanding Channels and Layers

At a very basic level, **channels** in Nuke carry image data and **layers** are containers for these channels, up to the maximum of 1023 channels per script. When elements are rendered out in the OpenEXR format, for example, you can operate on multiple channels from a single image.



The layer name, in this case **rgba**, **depth**, or **masks**, followed by the channel name.

Channels

Think of a channel as a container that contains image data. Once created or read into your composite, the image data stored in a channel is available downstream in the network until the value is replaced with something else or the channel is removed. The channel may even be "empty" - depending on where you reference it in the compositing network.

Layers

All channels in a script must exist as part of a layer (also called a *channel set*). You're probably familiar with the default layer - *rgba* - which includes the channels with pixel values of red, green, and blue, and also the alpha channel for transparency.

All channels in a composite must belong to at least one layer. Some channels, like alpha, may be available in other layers, too. Channel names always include the layer name as a prefix, like this: `layer_name.channel_name`.

By default, every script has a layer called **rgba**. When you first import an image element, Nuke automatically assigns its channels to the rgba layer - that is, the image channels are named **rgba.red**, **rgba.blue**, **rgba.green**, and **rgba.alpha**.

The **rgba** layer allows for the standard four-channel workflow of most node-based compositing systems. However, you're not limited to these four channels. You can create new channels and assign them to new layers up to the limit of 1023 channels per script.

Creating Channels and Layers

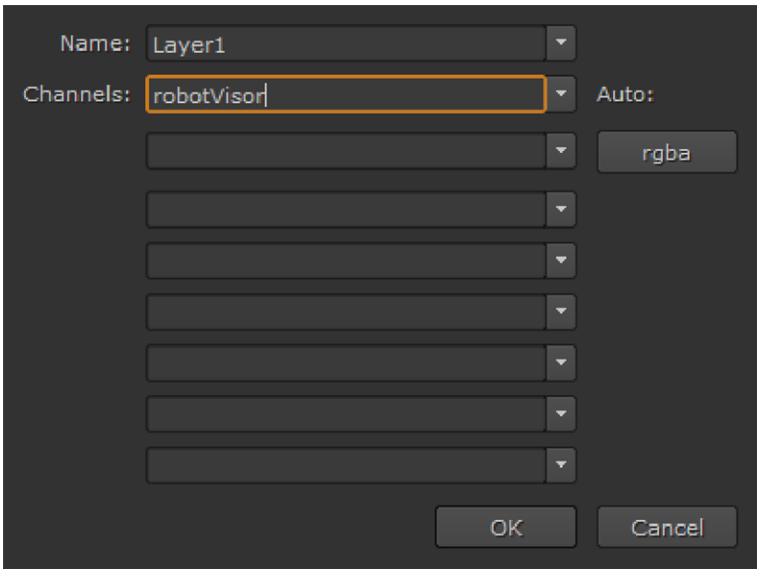
It's important to understand that many types of nodes allow you to direct their output to a specific channel and parent layer. You have the option of processing these channels in each subsequent node, or leaving them unchanged.

Many nodes feature an **output** or **channels** setting, which lets you direct the output of the current node to a specific layer and channel. You can also use the **output** or **channels** dropdown menu to create new layers and channels.

Some nodes do not include an **output** or **channels** setting in their parameters. For these, you can connect other nodes, such as Channel Copy or Shuffle, to create and manage channel output in the node tree.

To Create a New Layer and/or Channel

1. Open the properties panel for the node whose output creates the new channel.
2. From the **output** or **channels** dropdown menu, select **new**.
3. Under **Name**, enter the name of the layer, and under **Channels** the new channel name.



Note: You can either use a new layer name to create a new layer, or enter a layer you've created previously. You can't create new channels into layers that are built into Nuke (such as **mask**).

4. Click **OK**.

Note: You can also create new channels with the Shuffle node. See Shuffling Channelsfor more information.

Reserved Channels

Nuke reserves certain channel names for internal use to ensure the smooth running of scripts under the hood. If you create a custom channel with a reserved channel name, the custom name is renamed automatically to preserve this functionality. For example, if you create a custom channel called **mask.a**, the channel is renamed **mask_extra**.

The reserved channels are:

rgba.red, **rgba.green**, **rgba.blue**, **rgba.alpha**

depth.Z

forward.u, **forward.v**

backward.u, **backward.v**

deep.front, **deep.back**

disparityL.x, **disparityL.y**, **disparityR.x**, **disparityR.y**

mask.a

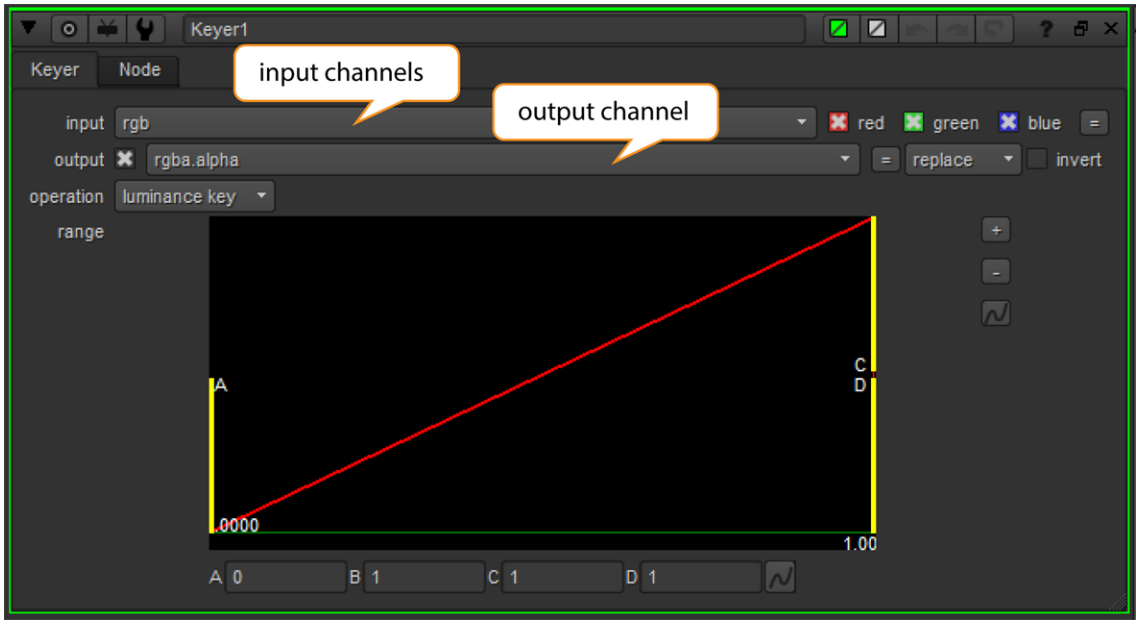
rotopaint_mask.a

mask_planartrack.a

mask_splinewarp.a

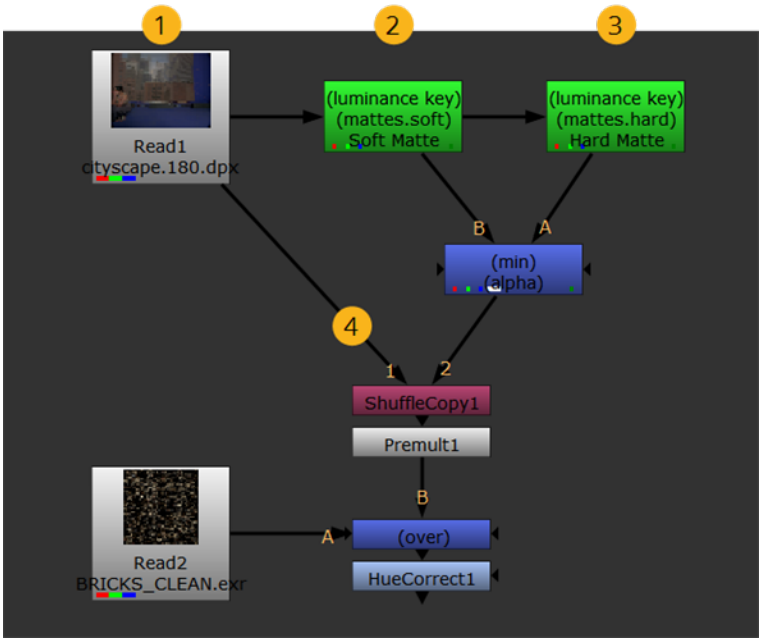
Calling Channels

By default, most nodes in Nuke attempt to process the current channels in the rgba set and place the output in those same channels. However, many nodes also contain an input dropdown menu which lets you select the channels you want to process, and an **output** dropdown menu to select the channel(s) where the results should be stored.



Some nodes also contain **mask** controls and a **mask** input connector, which let you select a channel for use as a matte to limit operations such as color corrections. Using these mechanisms, you can point the output of almost any node in the script to any available channel.

The script below attempts to clarify these concepts. Note the script generates six channels (though it could just as well generate 1023). The steps below describe how each channel was created.



A six-channel script.

1. The script reads in the foreground, creating three channels (**red**, **green**, and **blue**), which are by default assigned to the **rgba** set. Channel count: 3
2. A low contrast key (**soft**) is pulled and assigned to a new layer called **mattes**. Channel count: 4
3. A high contrast key (**hard**) is pulled and also assigned to the mattes layer. Channel count: 5
4. The **mattes.hard** and **mattes.soft** channels are mixed to form the final matte (**alpha**), which is assigned to the **rgba** layer. Channel count: 6

Suppose now that you wanted to perform a color correction using the output of the Soft Matte node as a mask for the correction. There's no need to pipe the output from that Soft Matte node - it already exists in the data stream along with the other five channels that were created.

You simply attach a color correction node, such as HueCorrect, then select the appropriate channel from the **mask** controls, for example **mattes.soft**.

Note: The **mattes** portion of the name indicates the parent layer.

Viewing Channels in the Viewer

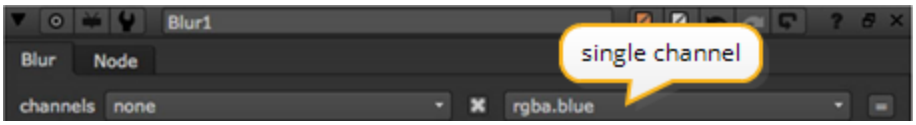
You can view the individual red, green, blue, and alpha channels in the Viewer using the **R,G,B**, and **A** keys on the keyboard. For more information, see Compositing Viewers and Using the Timeline Viewer.

Selecting Input Channels

A node's **channels** field lets you select one or several channels for processing.

To Select a Single Input Channel

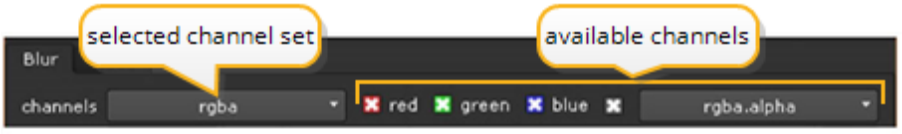
1. Open the properties panel of the node into which you wish to feed a channel.
2. From the **channels** field, select **none**.
3. From the right most channel field - the one which typically calls the alpha channel - select the single channel you wish to process.



To Select Multiple Input Channels

1. Open the properties panel of the node into which you wish to feed channels.
2. From the **channels** field, select the layer containing the channels you wish to process.

   The layer's channels appear with check boxes.



3. Deselect those channels which you don't wish to process. The node processes all those you leave selected.

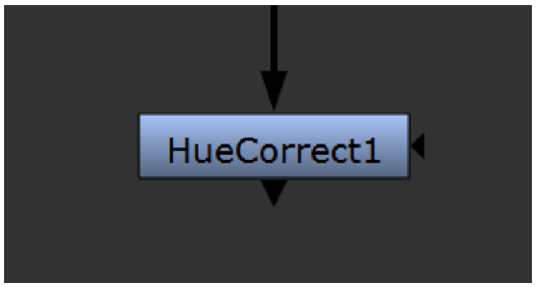
Selecting Masks

The **mask** controls in a node's properties panel let you select a single channel for use as a matte in a given process (typically, a color correction). The given process thereafter is limited to the non-black areas of the

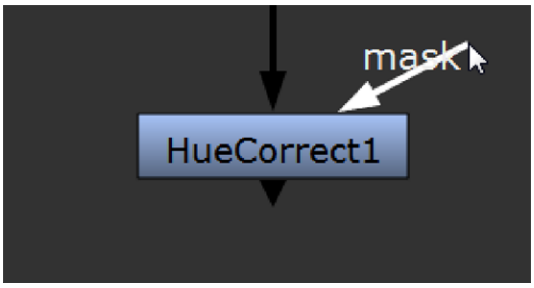
selected channel.

You can use one of the script's existing channels as the matte, or attach a mask to the node with a mask input connector.

You can find mask input connectors on color correction and filter nodes, such as HueCorrect and Blur. At first, they appear as triangles on the right side of the nodes, but when you drag them, they turn into arrows labeled **mask**. You connect them the same way as any other connectors. If you cannot see a mask input connector, open the node's properties panel and make sure **mask** is set to **none**.



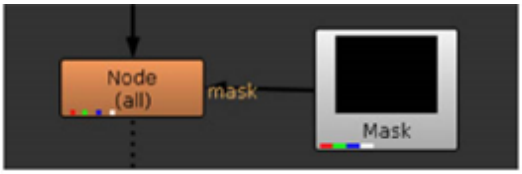
Before dragging the mask input connector.



When dragging the mask input connector.

To Select a Channel for Use as a Matte from the Mask Input

1. Connect a mask to the node with its mask input connector.



If you cannot see the mask input connector, open the node's controls and make sure **mask** is set to **none**.



By default, when a mask input is connected, the node uses the alpha channel from it as a matte.



2. If you don't want to use the alpha channel as the matte, select the channel you want to use from the **mask** dropdown menu.

3. If you want the mask from the **mask** input copied into the predefined **mask.a** channel, check **inject**. This way, you can use the last mask input again downstream. You can also set a stream of nodes to use **mask.a** as the mask, and then change the masking of all of them by simply connecting a new mask into the mask input connector of the first node.
4. If necessary, check the **invert** box to reverse the mask.



Tip: The **invert** control also affects any injected masks from upstream nodes.

5. If the overall effect of the node is too harsh, you can blend back in some of the input image by adjusting the **mix** slider.

To Select a Channel for Use as a Matte from the Main Input

1. Make sure nothing is connected to the node's mask input connector. If you disconnect a mask input, the mask input connector disappears, as it is no longer being used.
2. Select the channel you want to use from the **mask** dropdown menu.
3. If necessary, check the **invert** box to reverse the mask.
4. If the overall effect of the node is too harsh, you can blend back in some of the input image by adjusting the **mix** slider.

Linking Channels Using the Link Menu

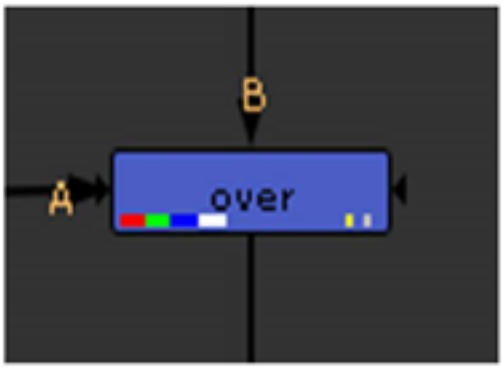
You can create expression links to connect channel and layer controls with other controls in various nodes. Since these controls aren't meant to be animated, you can't use the full range of Nuke expressions, nor can you use Python or Tcl languages. You can link controls using the **Link menu**  next to the control on the properties panel:

1. Click the **Link menu** and select **Set link**. An Expression dialog opens.
2. Enter your expression in the Expression field and click **OK**. For more information on expressions, see Expressions.
3. You can edit an existing link by clicking the **Link menu** and selecting **Edit link**.
4. You can also **Ctrl/Cmd**+drag the **Link menu** to another control to create a link between the two.
5. To remove a link, click the **Link menu** and select **Remove link**.

Tracing Channels

You may have noticed that nodes visually indicate the channels which they are processing (that is, treating in some way) and passing (that is, conveying without any treatment). This is done via a system of colored rectangles, which allows you to trace the flow of channels throughout a script.

Look closer at this example node. The wide rectangles indicate channels which Nuke processes (in this case, the red, green, blue, and alpha channels). The narrow rectangles indicate channels that Nuke passes onto the next node without processing (in this case, the **mattes.soft** and **mattes.hard**).



Visual confirmation of extra channels.

Renaming Channels

In the course of building your script, you may find it necessary to replace certain channels in a layer.

1. Open the properties panel for a node which has the channel selected on the **channels**, **input**, or **output** dropdown menu.
2. Click on the dropdown menu where the channel is displayed and select **rename**.

   The **Rename Channel** dialog appears.
3. Enter a new name for the channel and click **OK**.

Reserved Channels

Nuke reserves certain channel names for internal use to ensure the smooth running of scripts under the hood. If you create a custom channel with a reserved channel name, the custom name is renamed automatically to preserve this functionality. For example, if you create a custom channel called **mask.a**, the channel is renamed **mask_extra**.

The reserved channels are:

rgba.red, **rgba.green**, **rgba.blue**, **rgba.alpha**

depth.Z

forward.u, **forward.v**

backward.u, **backward.v**

deep.front, **deep.back**

disparityL.x, **disparityL.y**, **disparityR.x**, **disparityR.y**

mask.a

rotopaint_mask.a

mask_planartrack.a

mask_splinewarp.a

Removing Channels and Layers

When you are done using a layer or a channel within a layer, you may wish, for the sake of clarity, to remove it so that it is no longer passed to downstream nodes. Note that leaving channels in the stream does not itself cause them to be computed, only channels required are computed.



Note: When a script is saved, any channels that are not referenced in the script are discarded automatically.

To Remove a Layer or a Channel Within a Layer

1. Click **Channel** > **Remove** to insert a Remove node at the appropriate point in your script.
2. In the Remove properties panel, select the layer you wish to remove from the channels fields.
3. If you don't wish to remove the entire layer, uncheck the boxes corresponding to the channels which you still wish to be able to call downstream.
4. Click **OK** to close the properties panel.

   The layer and/or the channels you removed are no longer displayed in node parameters downstream from the Remove node.



Note: Removing layers and or channels does not free up space for the creation of new channels and layers. Once you create a channel, it permanently consumes one of the script's 1023 available channel name slots. You are free, however, to rename channels and/or assign them new outputs.

Shuffling Channels

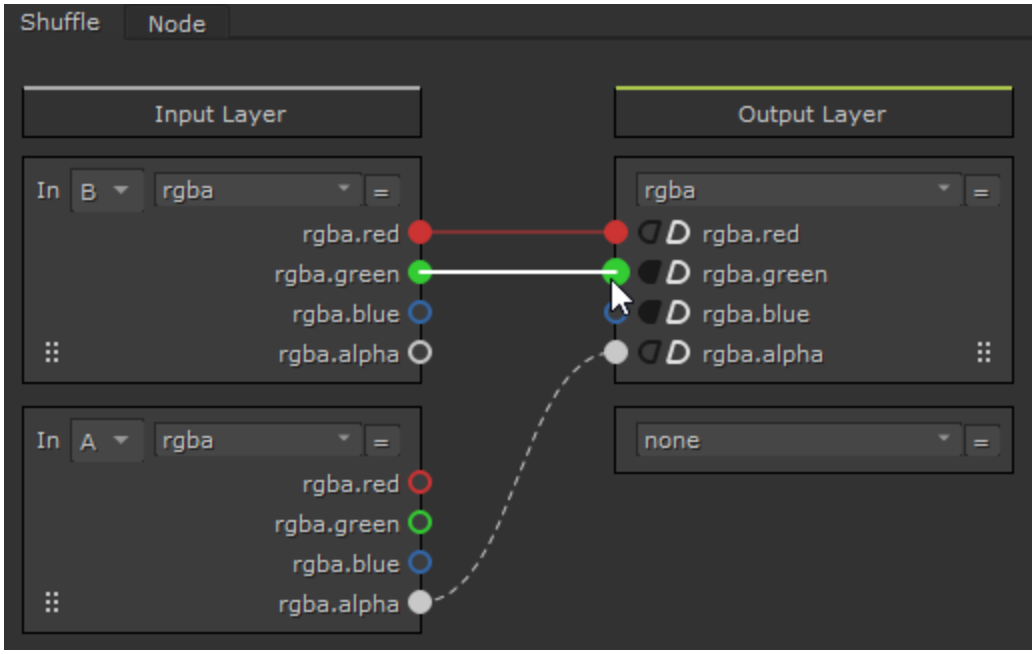
You can use Nuke's Shuffle node to rearrange the channels from a single image (B input) or two images (B and A inputs) and then output the result to the next node in your compositing tree. Select **Channels** > **Shuffle** or press **Tab** in the Node Graph and type **Shuffle** to create a Shuffle node.

The Shuffle node can:

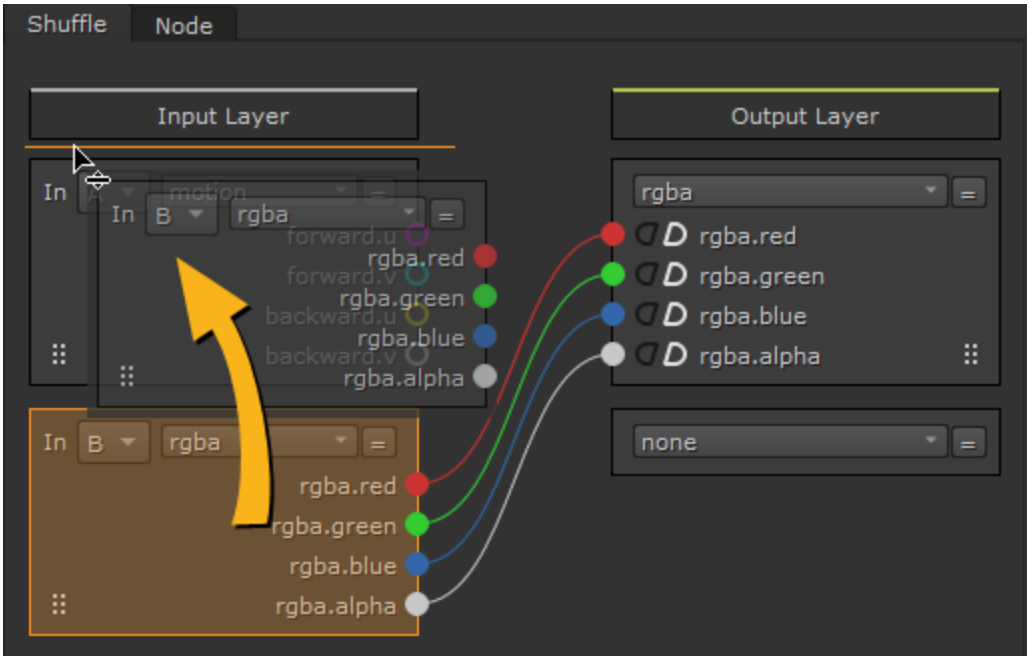
- rearrange up to eight channels from a single image (B input). For example, you can use it to swap **rgba.red** for **rgba.green**,
- rearrange channels between two separate nodes (A and B input), like a foreground and background branch,
- replace a channel with black (removing the alpha channel, for example) or with white (making the alpha solid, for example),
- create new channels.

The **Input Layer** represents the channel sets connected to the **B** and **A** inputs and the **Output Layer** represents what is passed down the node tree. You can drag noodles from sockets in either direction to connect the input and output layers. Nuke displays a preview of the resulting noodle connection in white when you drag sockets.

Note: Channels that are valid, but are not connected to the current node tree are displayed as dotted lines.

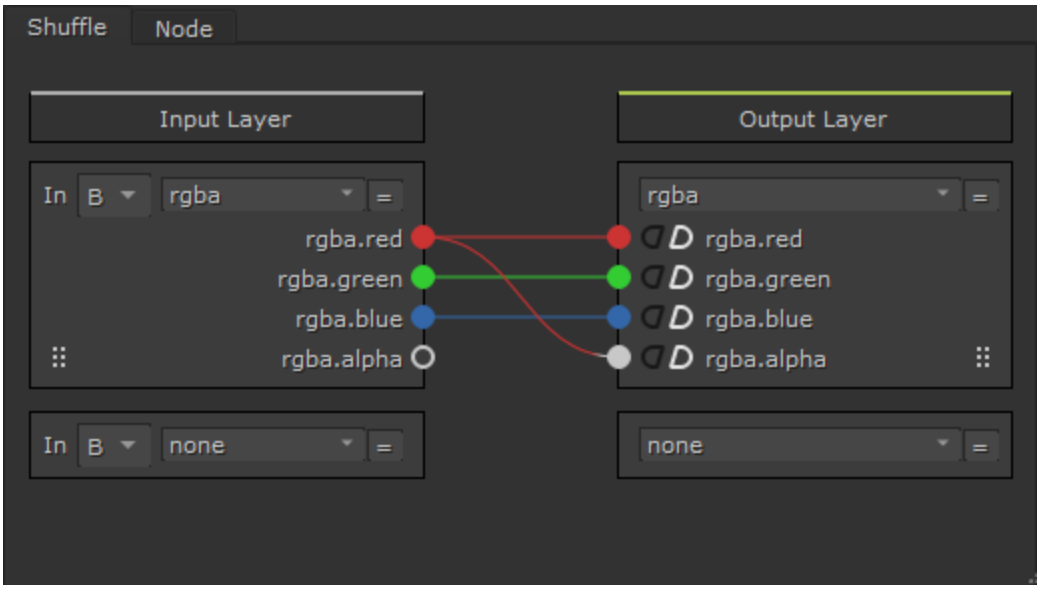


You can change the layer order by dragging the ⠿ icon and dropping the layer in its new position. Changing the order of layers can help with script organization so that you can keep the first output layer for swapping channels in the current data stream and the second output layer for creating new channels.



Shuffle Examples

A simple Shuffle in a single image might be copying the **red** channel of an **rgba** layer into its **alpha** channel. Using Shuffle, drag the **red** input socket to the **alpha** output socket to copy the channel.



Tip: If you just need to copy a channel from one data stream into another, you can also use **Channel** > **Copy**, instead of Shuffle. Then, specify the channel to copy and the destination channel for output.

A slightly more complex Shuffle from two images might be replacing the B input's **alpha** channel with the A input's **alpha** channel.



The Shuffle node's **Properties** panel.

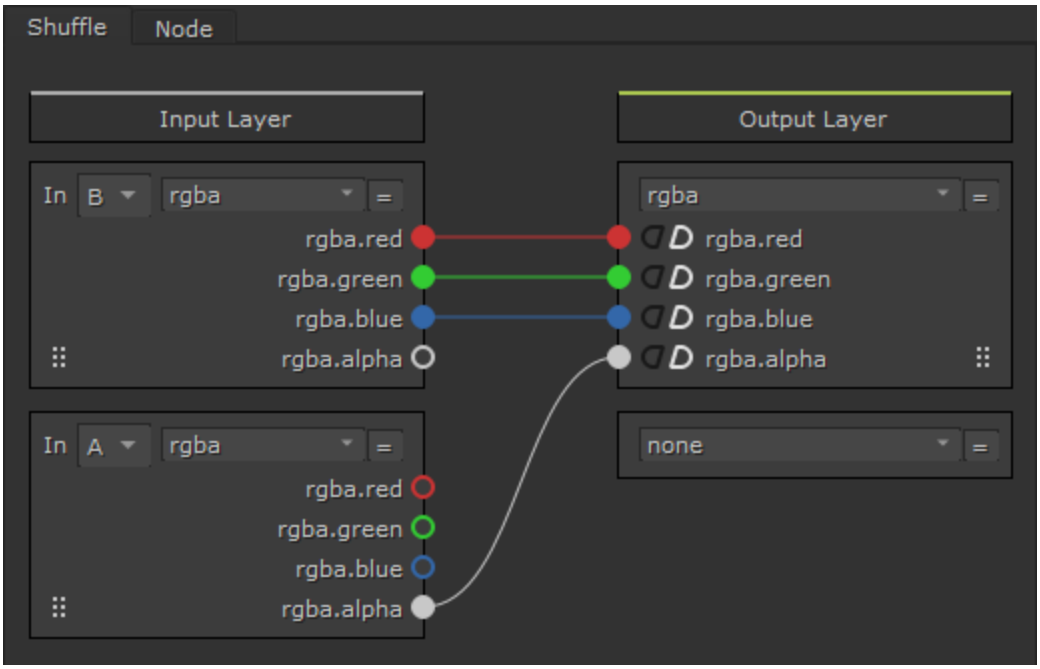
- **Channels from B Input** - the first group of channels are supplied by the B input on the node.
- **Channels from A Input** - the second group of channels are supplied by the A input on the node.

To shuffle the **alpha** channel from A into the output layer and ignore the alpha from B, do the following:

1. Click **Channel** > **Shuffle** to insert a Shuffle node.
2. Select the incoming channels from the **Input Layer**. You can select up to eight channels in this manner.
3. Select the layer to which you wish to direct the incoming channels from the dropdown menus on the right. You can select up to eight channels here as well.

Note: If the outgoing layer to which you wish to direct channels does not yet exist, create it using the **new** option on the dropdown menus on the right.

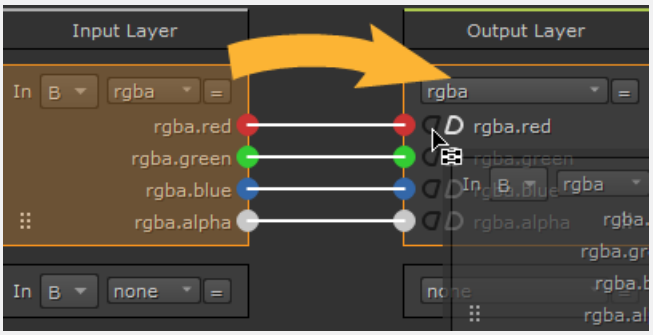
4. Connect the noodles as required to shuffle the channels. In this example, we disconnect the B input's **alpha** channel and connect the A **input**'s alpha channel. The resulting output layer consists of the **rgb** channels from B and the **alpha** channel from A.

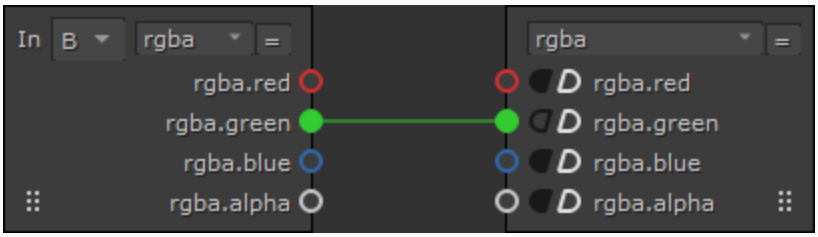
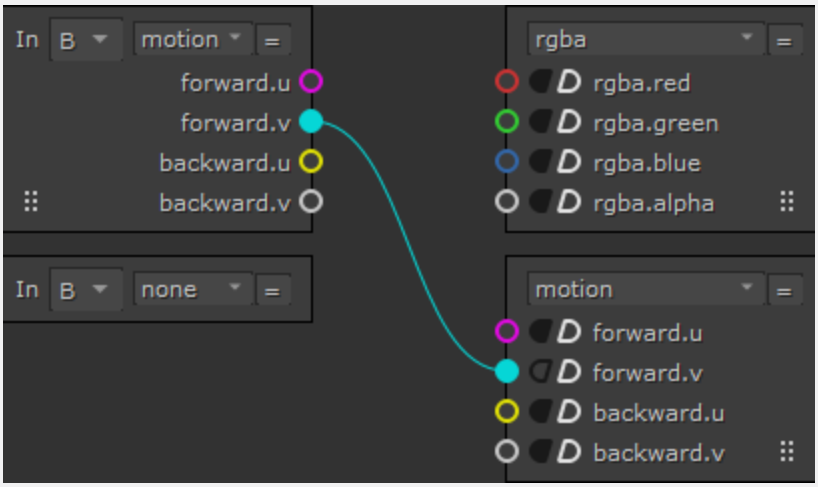
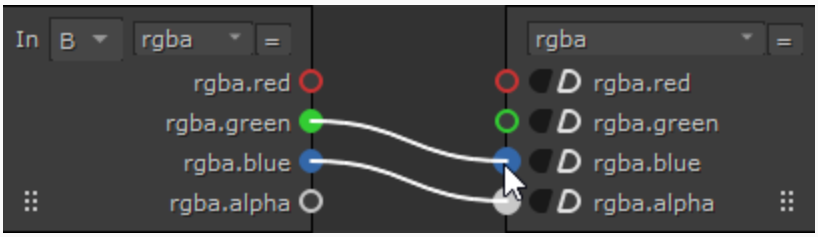
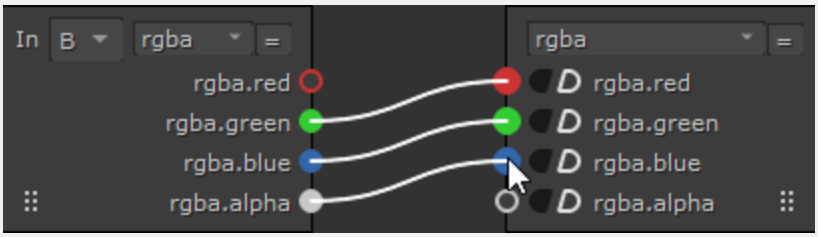
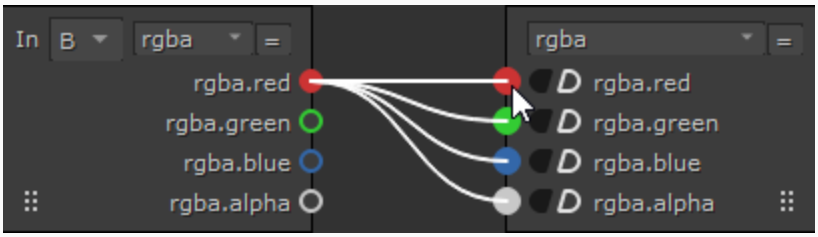


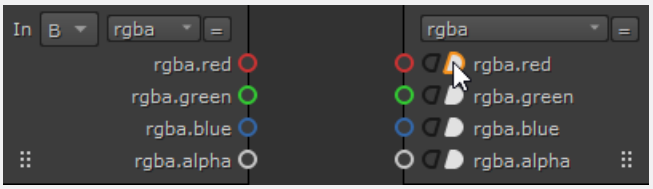
Tip: While not required, it's good practice to use the first output layer for swapping channels in the current data stream and the second output layer for creating new channels. This protects the default **rgba** channel set from unintentional overwriting, and makes it easier for other artists to understand the workings of your script.

Shuffle Keyboard Shortcuts

Dragging noodles between sockets can be time consuming in large scripts, so Shuffle includes a number of time-saving shortcuts.

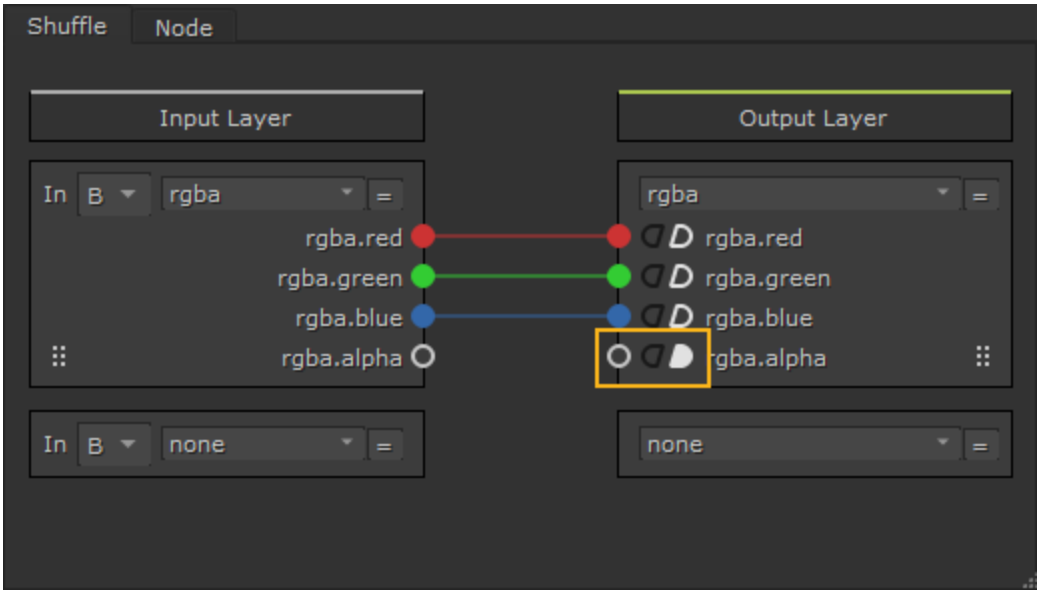
Description	Action	Result
Auto-connect all channels	Drag and drop	

Description	Action	Result
Auto-connect by order	double-click	
Auto-connect by channel name	**Ctrl/Cmd** + double-click	
Connect a channel and all lower channels in the layer	**Ctrl/Cmd** + drag	
Connect a channel and all higher channels in the layer	**Ctrl/Cmd** + **Shift** + drag	
Broadcast an input channel to all output channels	**Alt** + drag	

Description	Action	Result
Set all output channels to full black (0) or full white (1)	**Ctrl/Cmd** + click	

Assigning Constants

The Shuffle node can also assign black (**0**) or white (**1**) constants to any incoming channel. So, for example, to reset the **alpha** channel to a full-frame image, enable the full white button:



Assigning constants to channels.

Tip: You can also use **Ctrl/Cmd** + click to set all output channels to full black (0) or full white (1).

Creating Swap Layers

Finally, if the layer to which you wish to output channels does not yet exist, you can create it using the **new** option on the dropdown menus on the right. Once you select the **new** option, you follow the same process for creating layers as is described under Creating Channels and Layers.

Merging Images

With Nuke, you can merge images in a wide variety of ways. In these pages, we teach you how to use the Merge, ContactSheet, and CopyRectangle nodes.

Layering Images Together with the Merge Node

The Merge node with its compositing algorithms allows you to control just how your images are combined.



Note: When using most of the available merge algorithms, Nuke expects premultiplied input images. However, with the **matte** operation you should use unpremultiplied images.

To Layer Images with the Merge Node

1. Select **Merge** > **Merge** (or press **M** on the Node Graph) to insert a Merge node after the images you want to layer together.
2. Connect your images to the Merge node's **A** and **B** inputs.
3. If necessary, you can connect multiple A images to the Merge node. Once you have got the A and B inputs connected as instructed in step 2, drag more connectors from the left side of the Merge node to the images you want to use as additional A inputs.

   Each input is merged in the order connected, for example **A3**, **A2**, **A1**, **B**.
4. Connect a Viewer to the output of the Merge node so you can see the effect of your merge operation.
5. In the Merge node's controls, select how you want to layer the images together from the **operation** dropdown menu. The default and the most common operation is **over**, which layers input A over input B according to the alpha of input A. For descriptions of all the available operations, see Merge Operations.
6. Set which input's bounding box you want to use for the Merge output:
   - **union** - resize the output bbox to fit both input bboxes completely.
   - **intersection** - use only those parts of the image where the input bboxes overlap.
   - **A** or **B** - use the selected input's bbox for the output.

   See Adjusting the Bounding Box for more information on bounding boxes.

Tip: You can enable a warning to indicate when the bounding box is greater that the format in Nuke's **Preferences**. See Bounding Box Warnings for more information.

7. Select which input's metadata and frame range is passed down the tree using the **metadata from** and **range from** dropdowns.
8. Using the **A channels** and **B channels** dropdown menus, select which channels to use from the A and B inputs and which channels to use as the A and B alpha. If you want to merge more channels than these and output them into the same channels, select them from the **also merge** dropdown menus and checkboxes.
9. From the **output** dropdown menu, select the channels you want to write the merge of the A and B channels to. Channels named in the **also merge** dropdown menu are written to the same output channels.
10. If necessary, you can also adjust the following controls:
    - To select which input's metadata to pass down the tree, use the **metadata from** dropdown menu.



Note: When **metadata from** is set to **All** and there are keys with the same name in both inputs, keys in **B** override keys in **A**.

For more information on file metadata, see Working with File Metadata.

- To dissolve between the original input B image (at 0) and the full Merge effect (at 1), adjust the **mix** slider. A small light gray square appears on the node in the node graph to indicate that the full effect is not used.
- If you want to mask the effect of the Merge operation, select the mask channel from the **mask** dropdown menus. To invert the mask, check **invert**. To only apply the effect at the edges of the mask, check **fringe**.

  Note that you should not use the alpha of the inputs for the mask. It produces erroneous results (though the error is often hard to see); you can achieve better results by turning on alpha masking.
- From the **Set BBox to** dropdown menu, select how you want to output the bounding box. The default is **union**, which combines the two bounding boxes. You can also select **intersection** to set the bounding box to the area where the two bounding boxes overlap, **A** to use the bounding box from input A, or **B** to use the bounding box from input B.
- By default, Nuke assumes that images are in linear color space. However, if you want to convert colors to the default 8-bit color space defined in the **LUT** tab of your project settings (usually, sRGB), check **Video colorspace**. The conversion is done before the images are composited together, and the results are converted back to linear afterwards. Any other channels than the red, green, and blue are merged without conversion.

Checking this option can be useful if you want to duplicate the results you obtained from an application that uses the standard compositing math but applies it to non-linear images (for example, Adobe® Photoshop®). In this case, you typically also need to make sure **premultiplied** is not checked in your Read node controls.

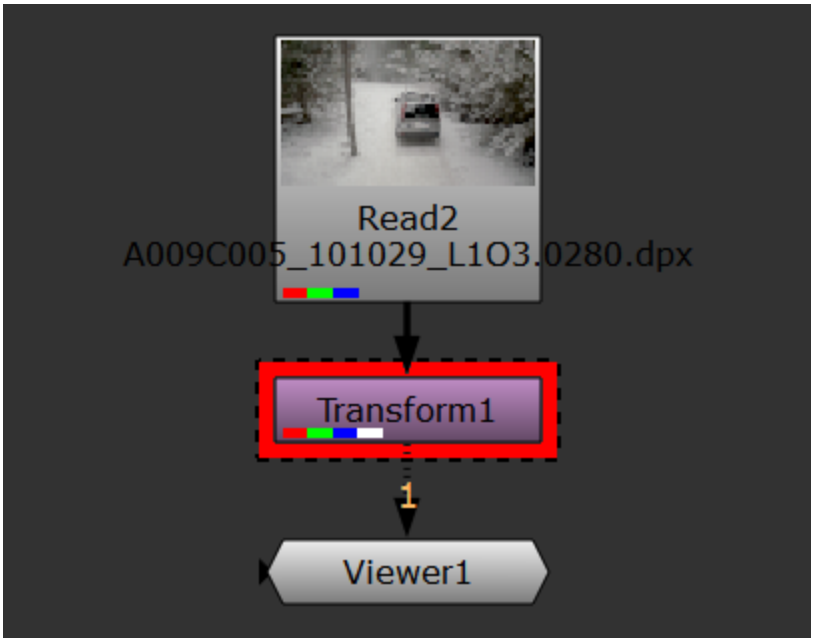
- By default, the same math is applied to the alpha channel as the other channels. However, according to the PDF/SVG specification, many of the merge operations (for example, overlay and hard-light) should set the alpha to (a+b - ab). This way, the input images remain unchanged in the areas where the other image has zero alpha. If you want to enable this, check **alpha masking**.

Bounding Box Warnings

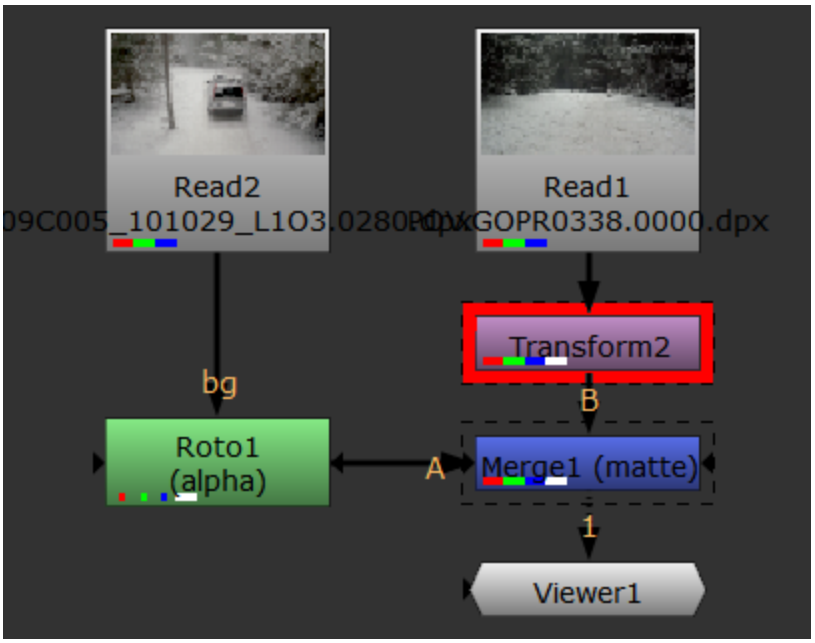
Zooming into the Viewer to work on a shot means that you can't always see the extent of the bounding box in relation to the format, which can result in unnecessary processing.

To make it easier to see the state of your bounding box, Nuke can display visual warnings on the nodes that affect the bounding box. To enable the warnings, in Nuke's **Preferences** under **Panels** > **Node Graph**, enable **Bounding Box Warning**:

- **red rectangle with dotted stroke** - the indicated node creates a bounding box greater than the format.



- **dotted stroke without the red rectangle** - the bounding box size is greater than the format at the indicated node, but the bounding box size has been set by an upstream node.



The bbox warning **threshold** controls how far past the edge of the format the bounding box can expand before the warning is displayed in the Node Graph. For example, if you're working with UHD_4K footage and the default 10% threshold, you can expand the bounding box horizontally by 384 pixels before the warning is displayed.

Tip: You can set the color of the warning rectangle in the **Preferences** under **Panels** > **Node Graph** > **Bounding Box Warning**.

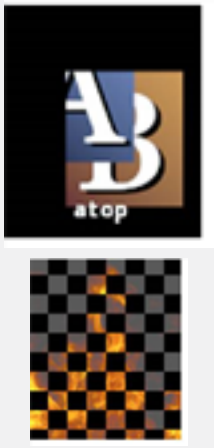
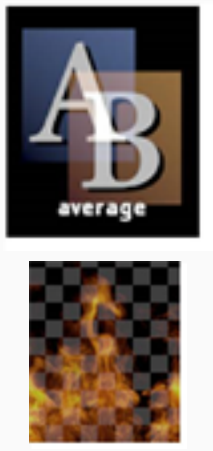
Merge Operations

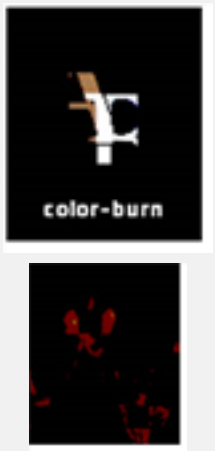
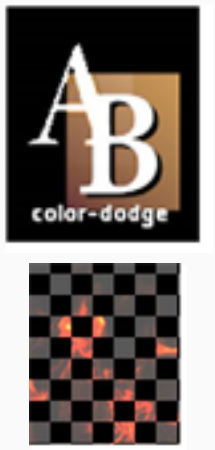
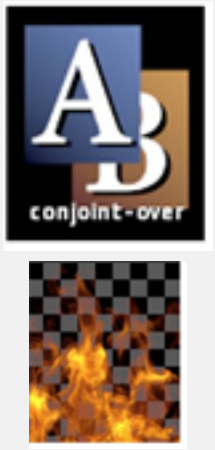
When layering images with the Merge node, you need to select a compositing algorithm that determines how the pixel values from one input are calculated with the pixel values from the other to create the new pixel values that are output as the merged image.

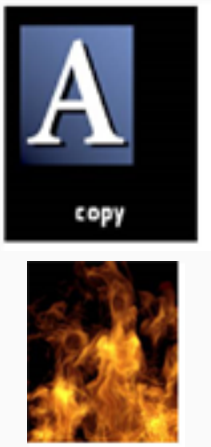
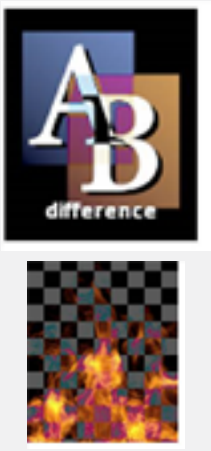
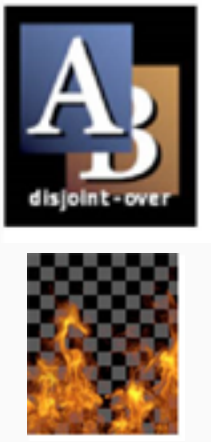
The **operation** dropdown menu in the Merge node's properties panel houses a large number of different compositing algorithms, giving you great flexibility when building your composite. The available algorithms are listed in alphabetical order.

Tip: With many compositing algorithms available, it may sometimes be difficult to find what you're looking for in the **operation** dropdown menu. Luckily, there's a quick way of finding a particular operation. With the menu open, you can type a letter to jump to the first operator that starts with that letter. To move to the second operation that starts with the same letter, press the letter again. For example, to select the **soft-light** operation, open the menu and press **S** twice.

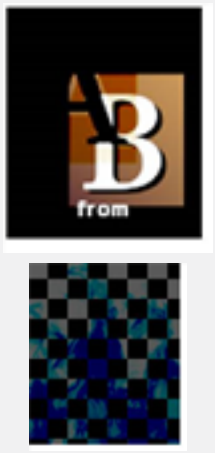
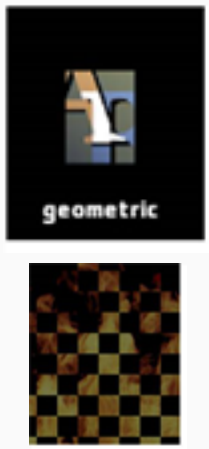
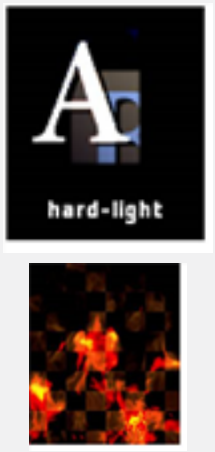
The following table describes each operation and its associated compositing algorithm. There are example images to illustrate the effects, one that combines the letters A and B to a merged image and another that has an image of fire merged with the familiar checkerboard. You may want to spend some time familiarizing yourself with each algorithm in order to be able to determine which operation to use in each situation.

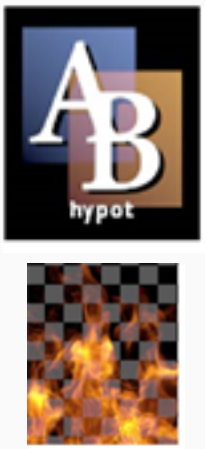
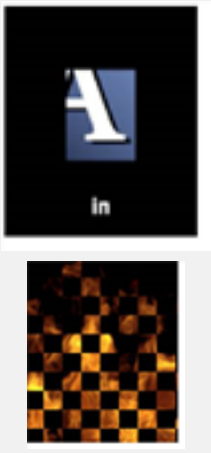
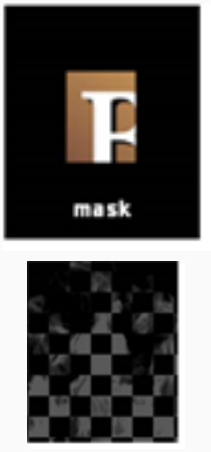
Operation	Algorithm	Description	Illustration	Example Uses
atop	Ab+B(1-a)	Shows the shape of image B, with A covering B where the images overlap.		
average	(A+B)/2	The average of the two images. The result is darker than the original images.		

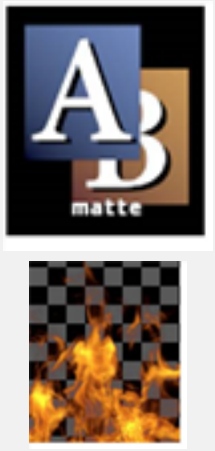
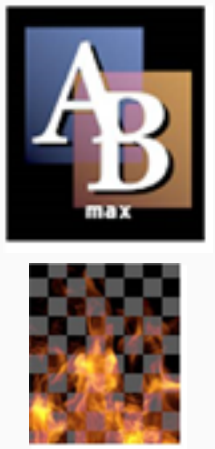
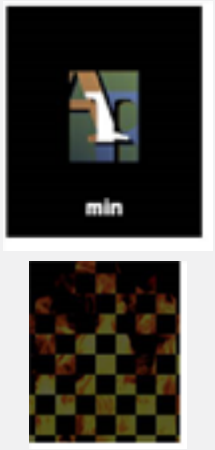
Operation	Algorithm	Description	Illustration	Example Uses
color-burn	darken B towards A	Image B gets darker based on the luminance of A.		
color-dodge	brighten B towards A	Image B gets brighter based on the luminance of A.		
conjoint-over	A+B(1-a/b), A if a>b	Similar to the over operation, except that if a pixel is partially covered by both A and B, conjoint-over assumes A completely hides B. For instance, two polygons where A and B share some edges but A completely overlaps B. Normal over produces a slightly transparent seam here.		

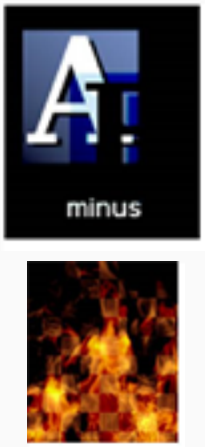
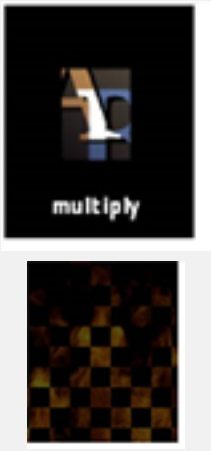
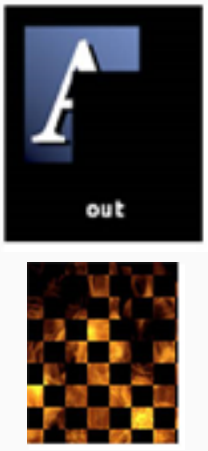
Operation	Algorithm	Description	Illustration	Example Uses
copy	A	Only shows image A.		This is useful if you also set the **mix** or **mask** controls so that some of B can still be seen.
difference	abs(A-B)	How much the pixels differ. Also available from **Merge** > **Merges** > **Absminus**.		Useful for comparing two very similar images. This mode can also be used as a difference keyer.
disjoint-over	A+B(1-a)/b, A+B if a+b<1	Similar to the over operation, except that if a pixel is partially covered by both a and b, disjoint-over assumes the two objects do not overlap. For instance, two polygons that touch and share an edge. Normal over produces a slightly transparent seam here.		This can be useful if you want to merge element a over element b, and element a has element b already held out. For example, you may have a CG character whose hair, skin, and clothing are rendered separately so that each object has the other

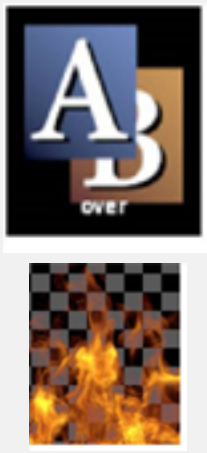
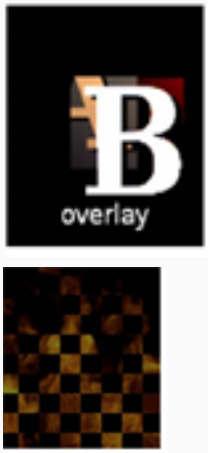
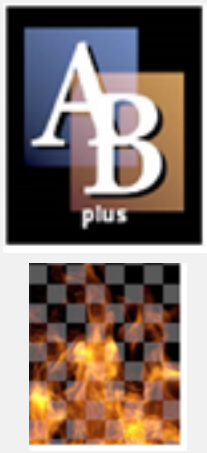
Operation	Algorithm	Description	Illustration	Example Uses
				objects held out of the render. In this case, using the over operation would produce dark lines around the comped objects. This is because over does a hold-out of the background image, meaning the background is held out twice.
divide	A/B, 0 if A<0 and B<0	Divides the values but stops two negative values from becoming a positive number.	 	This does not match any photographic operation, but can be used to undo a multiply.
exclusion	A+B-2AB	A more photographic form of difference.	 	

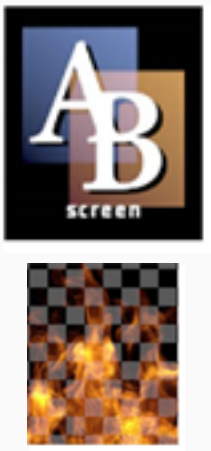
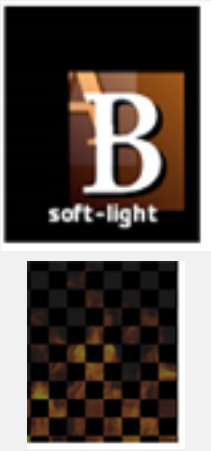
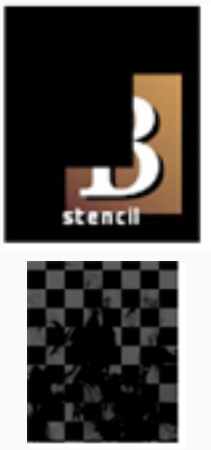
Operation	Algorithm	Description	Illustration	Example Uses
from	B-A	Image A is subtracted from B.		
geometric	2AB/(A+B)	Another way of averaging two images.		
hard-light	multiply if A<0.5, screen if A>0.5	Image B is lit up by a very bright and sharp light in the shape of image A.		

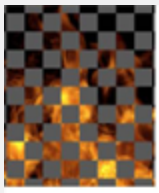
Operation	Algorithm	Description	Illustration	Example Uses
hypot	sqrt (A*A+B*B)	Resembles the plus and screen operations. The result is not as bright as plus, but brighter than screen. Hypot works with values above 1.		This is useful for adding reflections, as an alternative to screen.
in	Ab	Only shows the areas of image A that overlap with the alpha of B. Also available from **Merge** > **Merges** > **In**.		Useful for combining mattes.
mask	Ba	This is the reverse of the in operation. Only shows the areas of image B that overlap with the alpha of A.		

Operation	Algorithm	Description	Illustration	Example Uses
matte	Aa+B(1-a)	Premultiplied over. Use unpremultiplied images with this operation. Also available from **Merge** > **Merges** > **Matte**.		
max	max (A,B)	Takes the maximum values of both images. Also available from **Merge** > **Merges** > **Max**.		This is a good way to combine mattes and useful for bringing aspects like bright hair detail through.
min	min (A,B)	Takes the minimum values of both images. Also available from **Merge** > **Merges** > **Min**.		

Operation	Algorithm	Description	Illustration	Example Uses
minus	A-B	Image B is subtracted from A.		
multiply	AB, A if A<0 and B<0	Multiplies the values but stops two negative values from becoming a positive number. Also available from **Merge** > **Merges** > **Multiply**.		Used to composite darker values from A with the image of B - dark gray smoke shot against a white background, for example. This is also useful for adding a grain plate to an image regrained with F_Regrain.
out	A(1-b)	Only shows the areas of image A that do not overlap with the alpha of B. Also available from **Merge** > **Merges** > **Out**.		Useful for combining mattes.

Operation	Algorithm	Description	Illustration	Example Uses
over	A+B(1-a)	This is the default operation. Layers image A over B according to the alpha of image A.		This is the most commonly used operation. Used when layering a foreground element over a background plate.
overlay	multiply if B<0.5, screen if B>0.5	Image A brightens image B.		
plus	A+B	The sum of image A and B. Also available from **Merge** > **Merges** > **Plus**. Note that the plus algorithm may result in pixel values higher than 1.0.		Useful for compositing laser beams, but you're better off not using this one for combining mattes.

Operation	Algorithm	Description	Illustration	Example Uses
screen	A or B ≤1? A+B-AB: max (A,B)	If A or B is less than or equal to 1 the screen else use the maximum example, resembles Plus. Also available from **Merge** > **Merges** > **Screen**.		This is useful for combining mattes and also for adding laser beams.
soft-light	B(2A+(B(1-AB))) if AB<1, 2AB otherwise	Image B gets lit up. Not as extreme as the hard-light operation.		
stencil	B(1-a)	This is the reverse of the out operation. Only shows the areas of image B that do not overlap with the alpha of A.		

Operation	Algorithm	Description	Illustration	Example Uses
under	A(1-b)+B	This is the reverse of the over operation. Layers image B over A according to the matte of image B.	 	
xor	A(1-b)+B(1-a)	Shows both image A and B where the images do not overlap.	 	

Tip: If you have used older versions of Nuke, you may have seen Merge operations called diagonal, nl_over, and nl_under. Diagonal has been renamed and is now called hypot. To get the results of nl_over and nl_under, you can check **Video colorspace** and use over and under.

Generating Contact Sheets

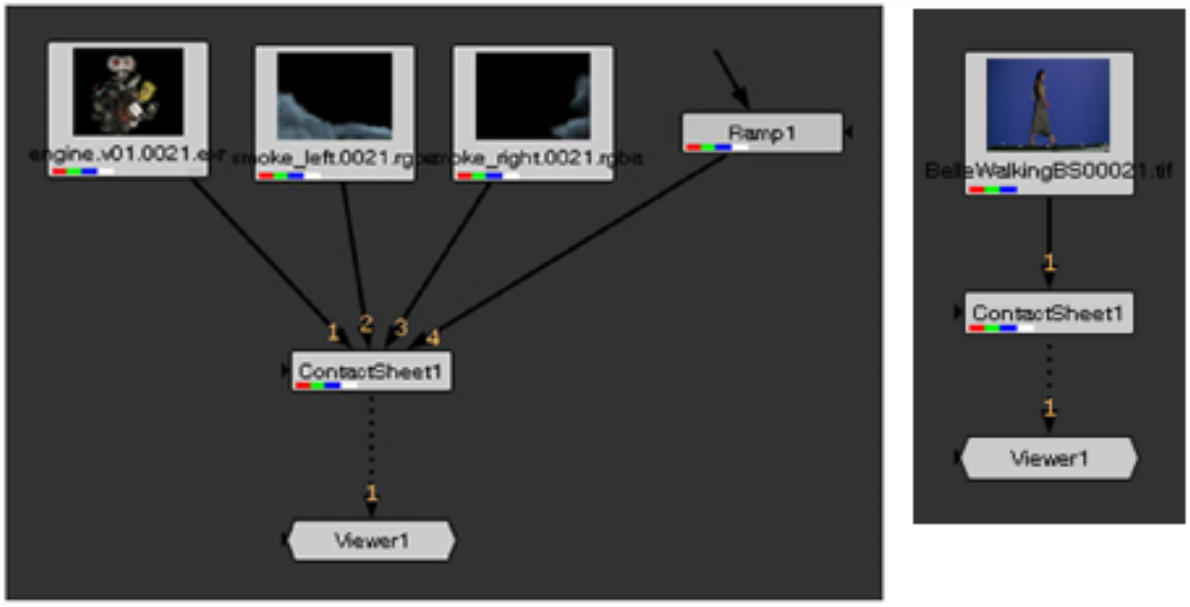
In order to demonstrate, document or manage what you are doing for a project, it can be useful to generate a contact sheet that shows your frame sequence(s) lined up next to each other in a matrix. For this, you can use the ContactSheet node. It generates a contact sheet from all its inputs or from the frames of one input.



A contact sheet generated from
the frames of one image sequence.

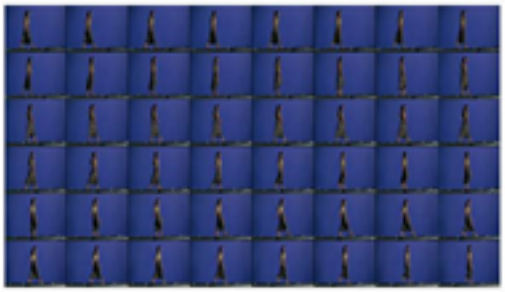
To Generate a Contact Sheet

1. Select **Merge** > **ContactSheet** to insert a ContactSheet node in your script.
2. Connect the image(s) you want to include in your contact sheet to the numbered input(s) of the ContactSheet node. If you want to include several different image sequences in the contact sheet, use multiple inputs. If you want the contact sheet to include the frames of just one image sequence, use only one input.



3. Connect a Viewer to the ContactSheet node so you can see the effect of your changes.
4. In the ContactSheet properties, define the **Resolution** (width and height) of the entire contact sheet in pixels.

5. If you want to create a contact sheet from the frames of one input, check **Use frames instead of inputs**. In the **Frame Range** field, define the frame range you want to include in the contact sheet.
6. In the **rows/columns** field, specify into how many rows and columns you want to arrange the input images or frames.
7. To adjust the size of the gaps between the images in the contact sheet, increment or decrement the **gap** value.

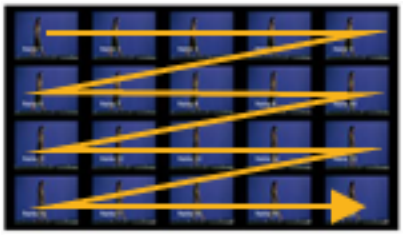


The **gap** value set to 0.

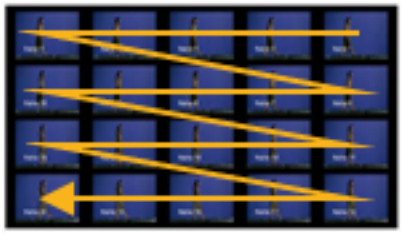


The **gap** value set to 50.

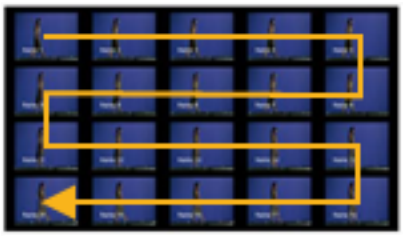
8. From the **Row Order** and **Column Order** dropdown menus, select how you want to order the images or frames in the contact sheet:



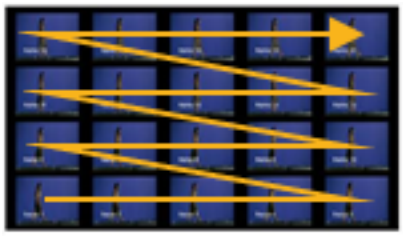
Row order: TopBottom
Column order: LeftRight



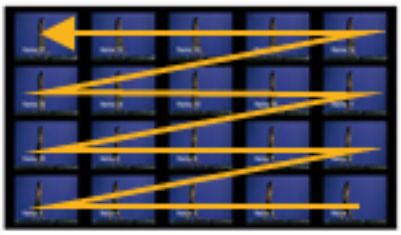
Row order: TopBottom
Column Order: RightLeft



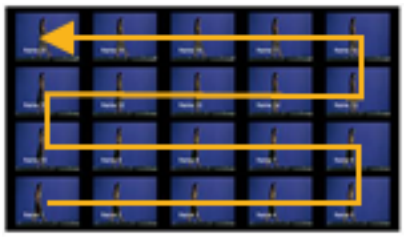
Row order: TopBottom
Column Order: Snake



Row order: BottomTop
Column Order: LeftRight



Row order: BottomTop
Column Order: RightLeft



Row order: BottomTop
Column Order: Snake

Tip: If you want to add any text, such as the frame number, on top of the images in the contact sheet, insert a Text node between the input image(s) and the ContactSheet node.

Copying a Rectangle from one Image to Another

With the CopyRectangle node, you can copy a rectangle from one input on top of another.



Input A.

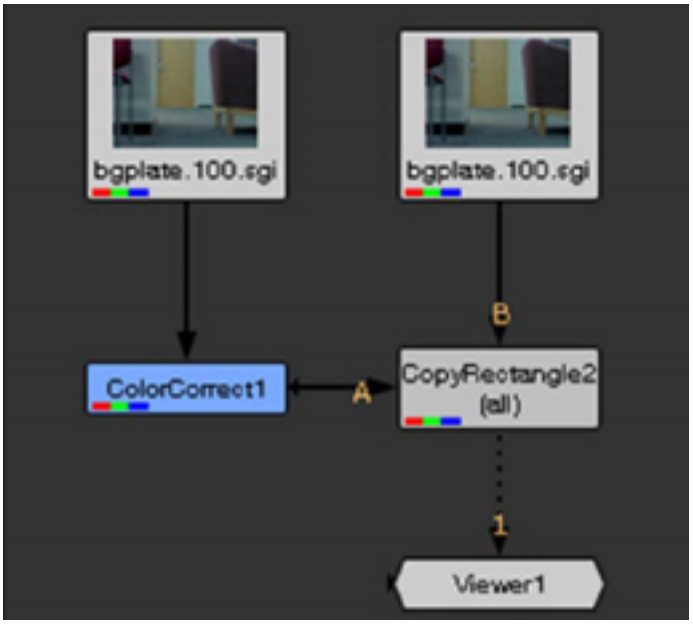


Input B.



The output of CopyRectangle.

The CopyRectangle node can also be used to limit effects, such as color corrections, to a small region of an image. To do so, you need to use the same image in both input A and B and only perform the color correction on one input.



A rectangle from input A color corrected and copied on top of input B.



The original image before CopyRectangle.



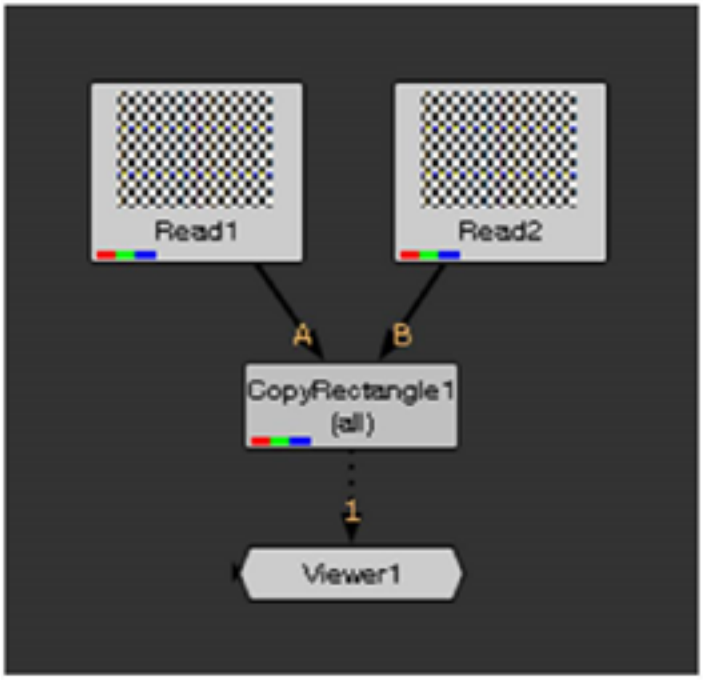
Defining a rectangle with Copy Rectangle.



The color corrected rectangle
on top of the original image.

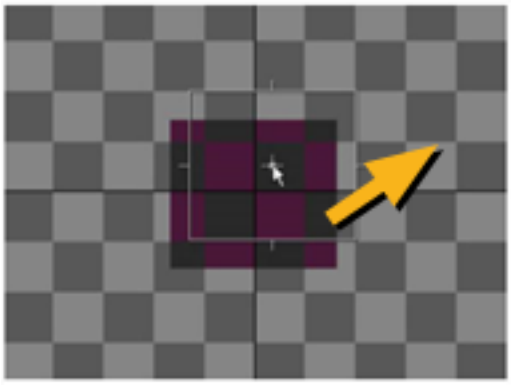
To Copy a Rectangle from One Image to Another

1. Select **Merge** > **CopyRectangle** to insert a CopyRectangle node after the image that has a region you want to copy (input A) and the image you want to copy the region to (input B). Create the following setup:

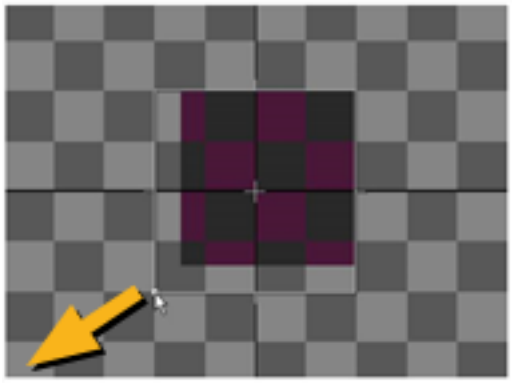


2. In the CopyRectangle controls, use the **channels** dropdown menu to select the channels you want to copy from input A.
3. To define the rectangle you want to copy, resize and reposition the CopyRectangle overlay in the Viewer. Drag the center of the overlay to reposition, and the edges to resize. If you cannot see the overlay in the

Viewer, open the CopyRectangle properties panel and double-click on the Viewer node in the Node Graph.

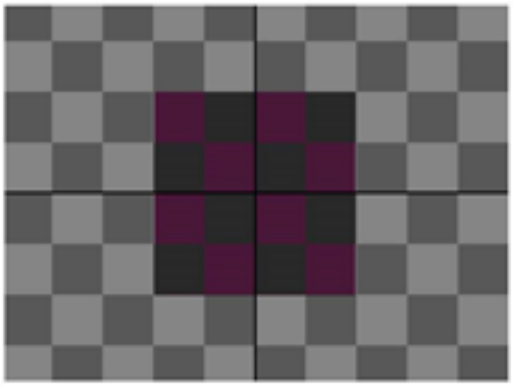


Repositioning the rectangle.

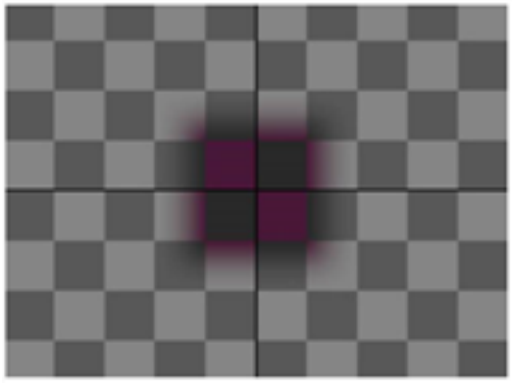


Resizing the rectangle.

4. To control how soft the edges of the rectangle seem, adjust the **softness** slider. The higher the value, the softer the edges.



A low **softness** value.



A high **softness** value.

5. To dissolve between the full CopyRectangle effect and input B, adjust the **mix** slider.

Removing Noise with Denoise

The Denoise node is an efficient tool for removing noise or grain from your footage. It uses spatial or temporal filtering to remove noise without losing image quality.

Quick Start

Here's a quick overview of the workflow:

1. Connect Denoise to the footage from which you want to remove noise.

   See Connecting Denoise.
2. Position the analysis box over a suitable analysis region. Denoise automatically analyzes the noise structure inside this region and removes noise from the footage.

   See Analyzing and Removing Noise.
3. Review the results.

   See Reviewing the Results.
4. If you're not happy with the results, you can fine tune them by adjusting the noise profile, frequencies, and channels.

   See Fine Tuning.



Before Denoise.



After Denoise.

Tip: You can check **Use GPU if available** to have the node run on the graphics processing unit (GPU) rather than the central processing unit (CPU). For more information on GPU support, see the Release Notes for your version of Nuke.

You can select the GPU to use in the Preferences. Press **Shift**+**S** to open the **Preferences** dialog, make sure you're viewing the **Preferences** > **Performance** > **Hardware** tab, and set **default blink device** to the device you want to use. You must restart Nuke for the change to take effect.

If you are using a render license of Nuke, you need to add **--gpu** on the command line.

Connecting Denoise

1. Create a Denoise node by clicking **Filter** > **Denoise**.
2. Connect the Denoise node's **Source** input to the image you want to denoise.
3. If you're working with complex footage that doesn't have a convenient area for analyzing noise (a flat area free from image detail, edges, and luminance variations), you can attach an optional noise clip to the **Noise** input. When a **Noise** clip is supplied, the noise is analyzed in this clip, rather than the **Source** clip. The **Noise** clip should have similar noise characteristics to the **Source** clip.
4. You can attach pre-calculated motion vectors to the **Motion** input. You can generate motion vectors using VectorGenerator, NukeX's SmartVector, or an external vector generator. See VectorGenerator or SmartVector for more information.

   Denoise can generate motion vectors internally, but connecting this input may produce significantly faster results.
5. Attach a Viewer to either the **Source** or **Noise** clip, depending on which you want to use for the analysis.
6. Proceed to Analyzing and Removing Noise.

Analyzing and Removing Noise

1. In the Denoise controls, set **Source** to **Film** or **Digital** depending on the type of footage you're using. **Digital** is the default setting and it works fine in most cases even if the footage is in a film format.
2. In general, you can also leave **Noise Model** set to **Modulated**. However, you may want to try **Constant** if you're working on either:
   - film footage with lots of detail but not too much noise in dark regions, or
   - digital footage with lots of detail but not too much noise in light regions.

3. In the Viewer, scrub to a frame with a suitable analysis region. This should be a flat area free from image detail, so no textures, edges, or shadows. If this is not the case, you may get poor results, as the algorithm thinks the image detail is noise and removes it.
4. You can enable **Temporal Processing** to use more than one frame to denoise the analysis frame. Denoise blends frames either side of the analysis frame to improve the result. The number of frames used either side of the current frame is determined by the **Frames To Blend** control.

Note: **Temporal Processing** requires the **motion**, **forward**, and **backward** channels from the **Motion** input. You can't enable **Temporal Processing** unless the **Motion** input is connected.

You can increase the **Frame Blending** value to force Denoise to blend more regions of the frame, but higher values can lose image detail.

5. Position and resize the analysis box to cover the analysis region. Note that the minimum size for the analysis region is 80x80 pixels. If the analysis region is too small, Denoise cannot analyze the footage or remove any noise.



A good analysis region.



A poor analysis region.

The analysis area selection automatically updates not only the **Analysis Region** parameter in the **Noise Analysis** group, but also the frame from which the sample is taken (**Analysis Frame**). By default, whenever the analysis box is altered, the internal analysis of the noise in that region reoccurs.

6. Connect a Viewer to the Denoise node.

   The output should now show the denoised frame.
7. Proceed to Reviewing the Results.

Tip: By default, Denoise starts analyzing the noise in your input footage when you move the analysis box. If you scrub to a new frame, don't move the analysis box, and want to reanalyze the noise from the new frame, you can force analysis by clicking the **Analyze Noise** button in the **Noise Analysis** parameter group.

If you'd like to disable analysis, you can check **Lock Noise Analysis**.

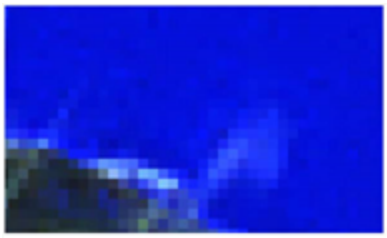
Tip: You can export the analysis profile in an external file. Note that if you've set **Profile** to **Constant**, only the controls that affect the analysis are saved in this file. By contrast, if you've set **Profile** to **Automatic**, both the analysis profile and the automatically calculated noise profile are exported.

To export the profile, use the **Analysis file** control under **Noise Analysis** to set the name and location of the file. The file extension doesn't really matter here - for example **.txt** is fine. Once you have named the profile, click the **Export** button to export it.

To import the same profile later, use **Analysis File** again to navigate to it and click **Import**. This disables any controls that are read from the profile. To re-enable them, you can uncheck **Lock Noise Analysis**.

Reviewing the Results

1. Zoom in to review the results.
2. To compare the denoised image with the original, press **D** on the Denoise node repeatedly to disable and re-enable it.



The original image.



The denoised image.

3. It can also be useful to look at the noise that was removed from the original image. To do so, set **Output** to **Noise**.

   Only noise should be visible in this image. If you can see a lot of picture detail, it means the current settings are making Denoise work too hard and remove too much of the image, which leads to a soft result.



If you can see picture detail in the **Noise** output, too much of the image is being removed.

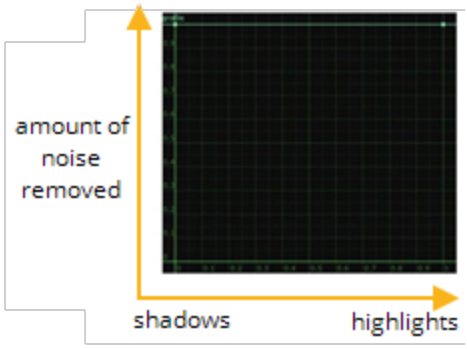
4. If your footage contains a lot of areas of sub-black, areas of black with a value less than 0, enable **Lift Blacks** to lift the blacks towards white. You can also enable **Preserve Edges** to attempt to sharpen the image at edges preventing over-smoothing. **Preserve Edges** can emphasize noise in some cases.
5. To blend the denoised luminance with the image's original luminance, you can temporarily increase **Luminance Blend**. This brings back some of the image detail in the result. You might want to have this set to 1, for example, when you're working on denoising the footage, but for the final result, you'll want to decrease it. The default value is 0.7.
6. If you're not happy with the results, you can try:
   - moving the analysis box to a different, flat area of the image,
   - analyzing on a different frame (by moving to a different frame and clicking **Analyze Noise**),
   - enabling **Temporal Processing** to blend more than one frame to calculate the denoise effect, or
   - tweaking the Denoise controls (**Denoise Amount**, **Roll Off**, and **Smoothness** in particular).

   Proceed to Fine Tuning.

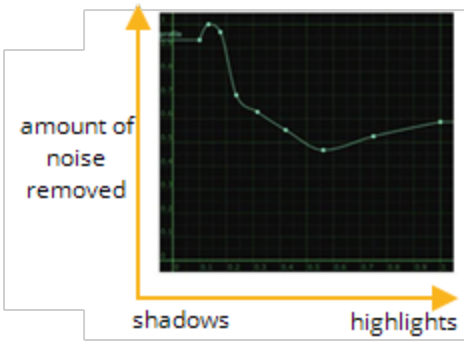
Fine Tuning

1. Set **Denoise Amount** to adjust the overall amount of noise to remove. Increase this to remove more noise, or decrease it if you want to keep more detail. A value of 0 removes no noise.
2. If the denoised image looks too sharp, use **Roll Off** to adjust the smoothness of the denoise thresholding. A value of 1 equals hard thresholding. Any other value produces soft thresholding between:
   - the **Denoise Amount** value and
   - the **Roll Off** value multiplied by **Denoise Amount**.

3. If you're not getting the correct smoothness level by adjusting the **Denoise Amount**, try setting **Smoothness** to a new value. This controls the smoothness of the denoised image, affecting the underlying image rather than the noise detected. In most cases, the default value of 1 works fine though.
4. If the results are too smooth and adjusting the above controls didn't help, try setting **Profile** to **Automatic**. Unlike the **Constant** profile, which looks at the analysis region and removes the same amount of noise across all intensities, automatic profiling looks at the entire **Profile Frame** to estimate a noise profile and removes different amounts of noise from the shadow, midtone, and highlight areas of the **Source** image.



Constant profile.

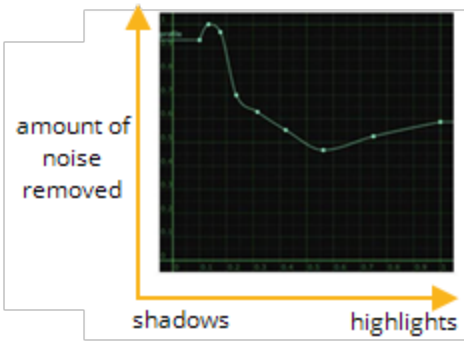


Automatic profile.

When you first switch to automatic profiling, Denoise uses the current frame to calculate the profile. If you'd like to use a different frame, you need to scrub to that frame and click the **Recalculate Profile** button on the **Profile** tab. This updates the **Profile Frame** control, which is read only.

Note: Denoise always bases the noise profile on your **Source** footage even if you've attached another clip to the **Noise** input.

5. You can also tweak the noise profile yourself using the controls on the **Profile** tab. This works in both the **Constant** and **Automatic** profiling mode.
   - Denoise displays the noise profile curve in the curve editor. The x axis represents image intensity, from low frequencies on the left to higher frequencies on the right. The y axis represents the relative amount of noise removed.

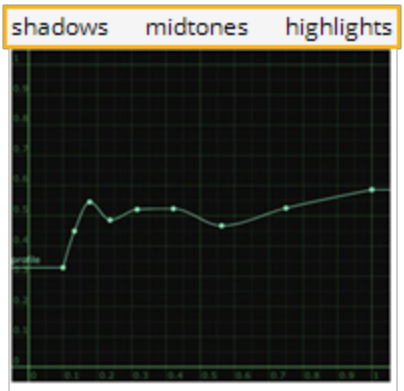


A noise profile curve.

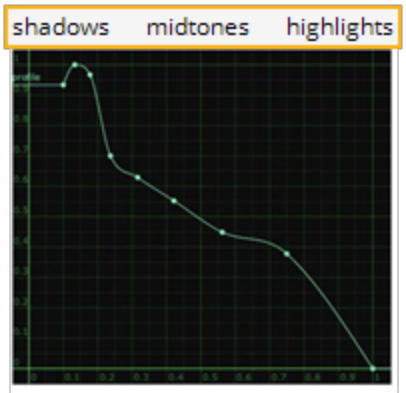
You can adjust the curve manually by dragging the points on the curve to a new location. To add more points to the curve, **Ctrl**/**Cmd**+**Alt**+click on the curve.

If you are not happy with your changes, click **Reset Profile** to reset the curve to its original shape (clicking **reset** works as well).

- Make sure you check **Tune Profile** to enable your changes. Then, adjust **Low Gain**, **Mid Gain**, and **High Gain** to scale the denoising threshold in the shadow, midtone, and highlight areas of the image. For example, a value of **2** multiplies the threshold by 2. Everything below the threshold is considered noise and removed, while everything above the threshold is kept.



The noise profile after decreasing **Low Gain**.



The noise profile after decreasing **High Gain**.

6. Back on the **Denoise** tab, the **Tune Frequencies** section lets you enable, disable, and adjust the denoising in different noise frequencies. This allows you to select the frequencies that contain noise, process them by as much as you think is best, and leave details in the other frequencies untouched.

   Normally, most of the noise occurs in the high and medium frequencies, so often you can disable the low and very low frequencies altogether. Use the **Gain** sliders to remove noise (by increasing the value) or to add more detail and noise (by decreasing the value).

7. Under **Tune Channels**, you can adjust the denoising threshold for the luma and chroma channels. Increase the **Luminance Gain** and **Chrominance Gain** values to remove more noise, or decrease them to remove less.

Keying with ChromaKeyer

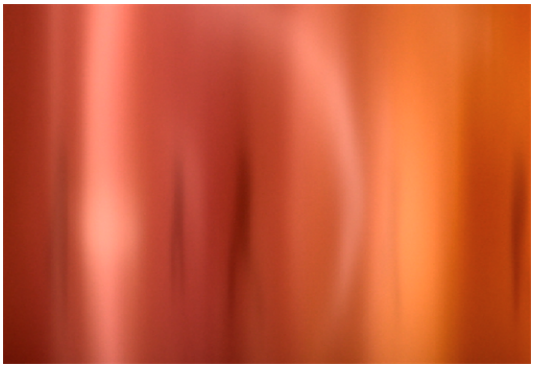
This section explains how to use the blue/green screen keyer, ChromaKeyer, in Nuke. ChromaKeyer can take advantage of modern GPUs and multi-core CPUs to accelerate the keying process when used for compositing in Nuke's Node Graph. ChromaKeyer is also available as a soft effect in Nuke Studio's timeline environment.

Quick Key

The images below show a green screen foreground and the background to be composited over.



Foreground



Background

1. Start Nuke and read in the foreground and background images. From the **Keyer** menu, select **ChromaKeyer** and attach a Viewer.
2. In the ChromaKeyer **Properties** panel, click the color swatch next to **screen color** to activate the eye dropper.
3. **Ctrl**/**Cmd**+**Shift**+click and drag a rectangular area over the green pixels in the Viewer. This averages the pixels in the area selected to produce a better key.





In some cases, this is all you need to do to perform a key, since selecting the screen color creates a screen matte and despills the foreground.

4. Merge the foreground over the background to produce your final comp.



Picking the screen color may be enough for a lot of keys, but there are many more tools within Nuke that can be used to tackle more complicated shots. See Improving Mattes, Despilling and Color Replacement, and Multi-Pass Keying for more information.

Picking the Screen Color

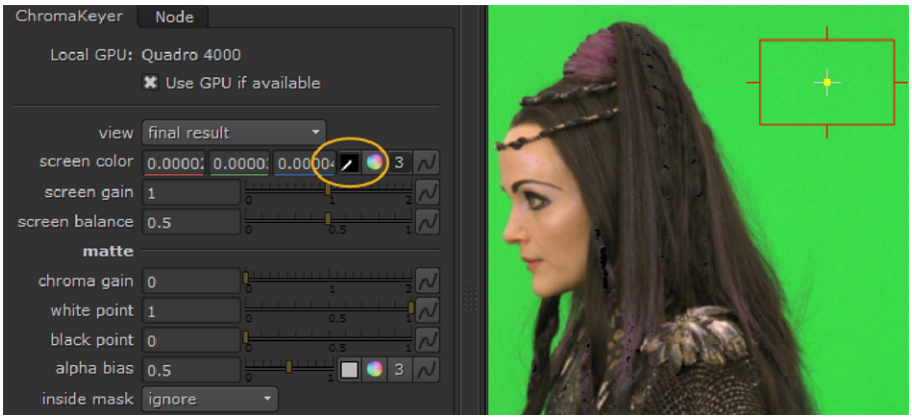
The **screen color** is probably the most important control in keying, and you should always pick the screen color before doing anything else. It should be set to the color of the green or blue behind the foreground object.

1. Start Nuke and read in the foreground and background images. From the **Keyer** menu, select **ChromaKeyer** and attach a Viewer.

2. In the ChromaKeyer **Properties** panel, click the color swatch next to **screen color** to activate the eye dropper.
3. Pick the screen color directly from the image by **Ctrl**/**Cmd**+**Shift** and dragging a rectangle over the green pixels. The average value of the pixels selected is used.



Tip: You can discard sampled pixels by **Ctrl**/**Cmd**+right-clicking in the Viewer.



It's worth picking different shades of green or blue from different parts of the screen to get the best result.



Note: Picking colors repeatedly does not add to previous selections, keying more of the image with each click. To key more of the image, pick different shades of green or blue in additional ChromaKeyer nodes downstream, and set the **inside mask** control to **source alpha**. See Multi-Pass Keying for more information.

Picking the **screen color** does two things:

- It creates the initial **matte** used to composite the foreground downstream. The matte can be improved using the **matte** controls in the **Properties** panel. See Improving Mattes for more information.
- It despills the foreground, but you can use the **despill** controls for greater accuracy. For more information on **despill**, see Despilling and Color Replacement for more information.

Adjusting Screen Gain and Balance

The **screen gain** controls how much of the screen color is removed to make the matte. Increasing this value keys more of the foreground. In the example image, a lower **screen gain** value adds too much of the background back into the output whereas a high **screen gain** value erodes the foreground too much and tends to tint the edges the opposite of the screen color (for green screens, edges become magenta).



A **screen gain** value lower than 1 adds background color back into the image.



A **screen gain** value higher than 1 removes background color from the image.

The **screen balance** controls the bias toward the two non-primary colors after the **screen color** has been chosen. A **screen balance** closer to 0 balances the output toward the stronger of the two colors and a value closer to 1 toward the weaker color. In the example image, the background color is green with RGB values of 0.04, 0.7, and 0.07, so values closer to 0 affect the blue component.



A **screen balance** closer to 0 affecting the blue component.



A **screen balance** closer to 1 affecting the red component.

Improving Mattes

The **matte** controls can improve the basic matte created after picking the **screen color**. The best way to view the matte in ChromaKeyer is to switch the Viewer to display the **alpha** channel in the **R** or **Luminance** display styles.



The matte in the **alpha** channel **R** display.



The matte in the **alpha** channel **Luminance** display.

Adjusting the Gain, White Point, and Black Point

Adjusting the **chroma gain**, **white point**, and **black point** either dilates or erodes the matte to include more or less of the foreground image based on the control used. The example image, concentrating on the area near the character's chin, shows a good example of where a matte can be improved.



Adjusting the **chroma gain** controls how much of the chroma difference between the source image and the screen color is used to correct the matte. Increasing the gain generates a matte that has less transparent areas in the foreground, but can produce harder edges.



The matte with a low **chroma gain** value.

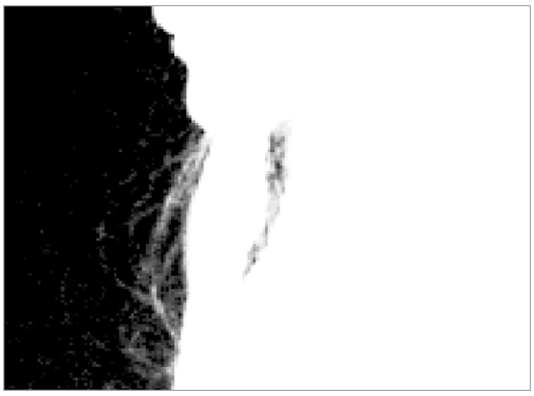


The matte with a high **chroma gain** value.

Adjusting the **white** and **black point** controls the pixels that appear as foreground and background. Any pixels with values higher than the **white point** value are treated as foreground (alpha = 1) and any pixels with values lower than the **black point** value are treated as background (alpha = 0).



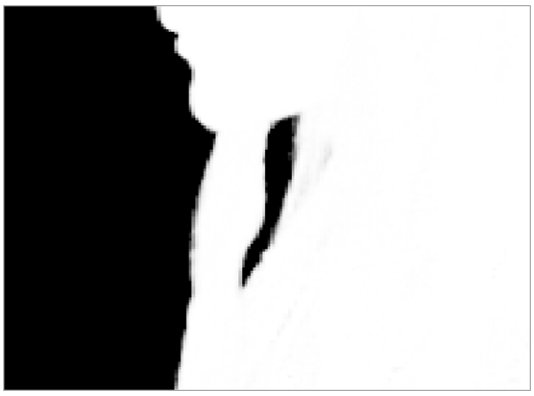
The matte with the default **white point** value.



The matte with a **white point** value of 0.5.



The matte with the default **black point** value.



The matte with a **black point** value of 0.5.

The **chroma gain**, **white point**, and **black point** controls represent a balancing act between removing too much foreground and retaining too much background. Once you're happy with the matte, proceed to Despilling and Color Replacement.

For particularly tricky keying problems, you can improve the matte using multi-pass keying. This method uses multiple ChromaKeyer nodes with all but the first one in the node tree set to **inside mask** > **source alpha**, so that the mattes in the alpha channels are passed down the node tree. See Multi-Pass Keying for more information.

Despilling and Color Replacement

Although ChromaKeyer automatically despills the matte when you select a screen color, you may find that you can improve the matte by manually despilling the key using a **despill bias** or by replacing color at the edges of the matte.

Custom Despilling

The **despill bias** allows you to specify a color from the image, separate from the **alpha bias**, to improve the overall despill for the matte. Typically, you should pick skin tones or hair colors for the **despill bias**.

1. Enable **custom despill bias** to allow you to use the **despill bias** controls.
2. Click the color swatch next to **despill bias** to activate the eye dropper.
3. **Ctrl**/**Cmd**+**Shift**+click and drag a rectangular area over the color you want to replace the highlighted pixels in the Viewer. This averages the pixels in the area selected to produce a better result.



Blue screen pixels present in a default matte.



Selecting the hair color despills the blue pixels, producing a better matte.

Color Replacement

Improving a matte by adjusting the alpha channel can remove the wrong amount of screen color from the pixels where transparency has changed. The **replace** controls instruct ChromaKeyer how to deal with these

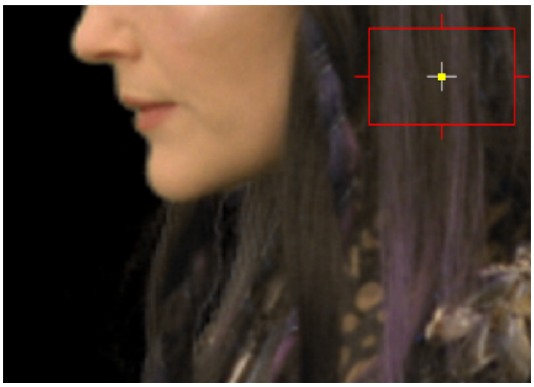
pixels. The **replace mode** controls which pixels inherit the **replace color**.

1. In the ChromaKeyer **Properties** panel, select the required replace mode using the dropdown:
   - **ignore** - the despilled image is left untouched if the alpha is modified. This is the default operation.
   - **edge hard color** - the despilled image has a corresponding amount of the **replace color** added for any increase in alpha.
   - **edge linear color** - the despilled image has a graded amount of the **replace color**, controled by a linear curve, added for any increase in alpha. Pixels closer to the background have more background bias and pixels closer to the foreground have more foreground bias.
   - **edge soft color** - the despilled image has a corresponding amount of the **replace color** added for any increase in alpha, however, it attempts to modulate the luminance of the resulting pixel so that it matches the original pixel. This gives a more subtle result than the **edge hard color** option.
2. Click the color swatch next to **replace color** to activate the eye dropper.
3. **Ctrl**/**Cmd**+**Shift**+click and drag a rectangular area over the color you want to replace the highlighted pixels in the Viewer. This averages the pixels in the selected area to produce a better result.

Note: You can enhance the despill effect by enabling **add-in matte fix** to also apply the replace color to areas eroded or dilated by the **white point** and **black point** controls.



Image with **edge soft color** highlighted.



Highlighted pixels replaced by the color picked from the Viewer.

4. Use the **replace amount** slider to control how much of the **replace color** is applied.
5. Disable the **premultiply** control if you don't want the output premultiplied by the alpha channel.

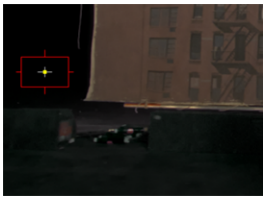
Multi-Pass Keying

In production situations, you may not always get a uniform color across your blue or green screen background, which can make pulling a key difficult. ChromaKeyer allows you to key different regions of the screen additively by picking shades of blue or green in multiple ChromaKeyer nodes. The example image shows a poor background containing different shades of blue as a screen color.



To perform a multi-pass key, do the following:

1. Pick an initial screen color as described in Picking the Screen Color.

   You can see from the image that ChromaKeyer has done a decent job on the area chosen, but the lower-half of the image still contains some noise.
2. Add a second ChromaKeyer node and in the **Properties** panel, set the **inside mask** control to **source alpha**.

   This takes into account the screen matte from the initial ChromaKeyer node when you pull another key.
3. Pick another **screen color** in the problem area using the second ChromaKeyer node.

   The key is improved by adding the second screen matte to the results of the first.
4. You can add as many ChromaKeyer nodes as you like, as long as you remember to set the **inside mask** control to **source alpha** in all but the first ChromaKeyer node.

Keying with Cryptomatte

Cryptomatte creates ID mattes automatically, using organizational information already available at render time. Using the Cryptomatte plug-in in Nuke allows you to select mattes and write out your selection to a channel. This makes it quick and easy to create masks from your mattes and isolate objects or sections of an object to make adjustments.



To use Cryptomatte, your input image or sequence must have a Cryptomatte layer. To add a Cryptomatte layer, you must use Cryptomatte in the application you render from, such as Maya, to create ID mattes and add them as a render layer before rendering. You can then read the image or sequence into Nuke and use the Cryptomatte plug-in.

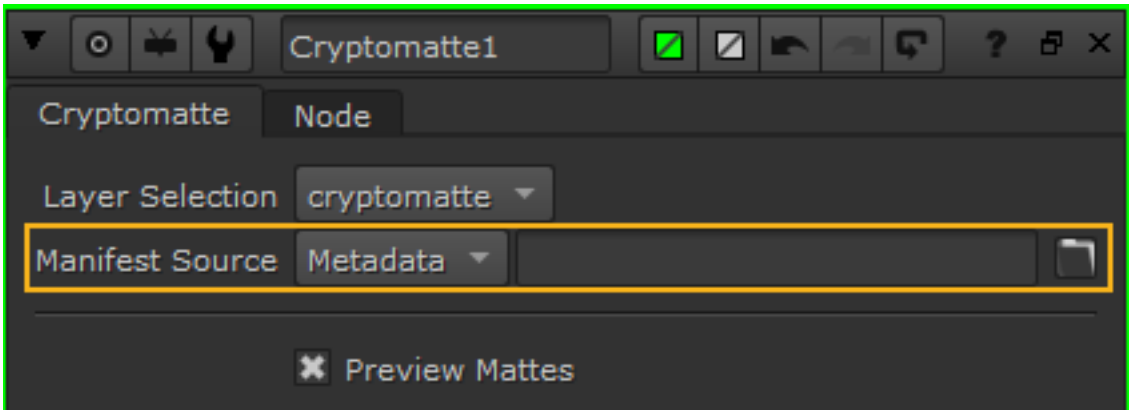
Note: Matte ID layers that have not been generated using Cryptomatte cannot be accessed using the Cryptomatte plug-in in Nuke.

Cryptomatte Metadata and Manifests

Cryptomatte uses metadata contained in multi-channel file types, such as **.exr**, to store information about the name of the Crypomatte, the type of hash used, and the conversion method used to convert the hash value into pixel color data.

Note: Only one hash type, MurmurHash3_32, and one conversion method, unit32_to_float32, are currently supported.

There are two possible sources of the manifest, it can either be embedded into the metadata of an **.exr** file, or stored on disk in a sidecar file. The type of manifest source is normally chosen at the rendering stage. The renderer is configured to render out an **.exr** with the manifest either embedded inside, or generated as a separate sidecar file. The Cryptomatte plug-in allows artists to select the source of the manifest using the **Manifest Source** control in the node **Properties**. This means the source can be overridden if needed which gives artists more flexibility.



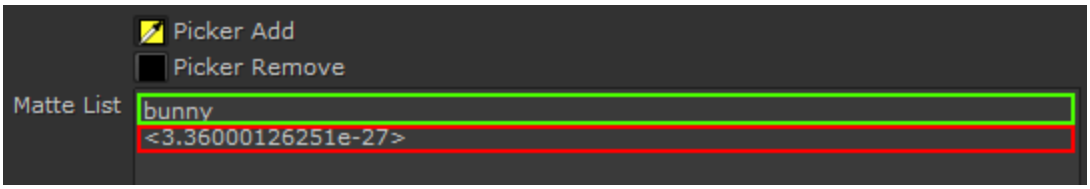
The manifest contains the names and the IDs of all Cryptomatte objects in the form of a JSON dictionary where the names are the keys and the IDs are the values, for example:

"bunny": "13851a76"

Where **bunny** is the name of one Cryptomatte object, and **13851a76** is its ID.

A manifest is not required, and if the Cryptomatte manifest is missing or its content is corrupted, you are still able to select or deselect mattes, however it's not possible to recover the matte name. In this case,

Cryptomatte adds a floating-point value of the selected ID, wrapped by angle brackets, to the **Matte List**. For more information about the **Matte List**, see the Using the Matte List section in this topic.



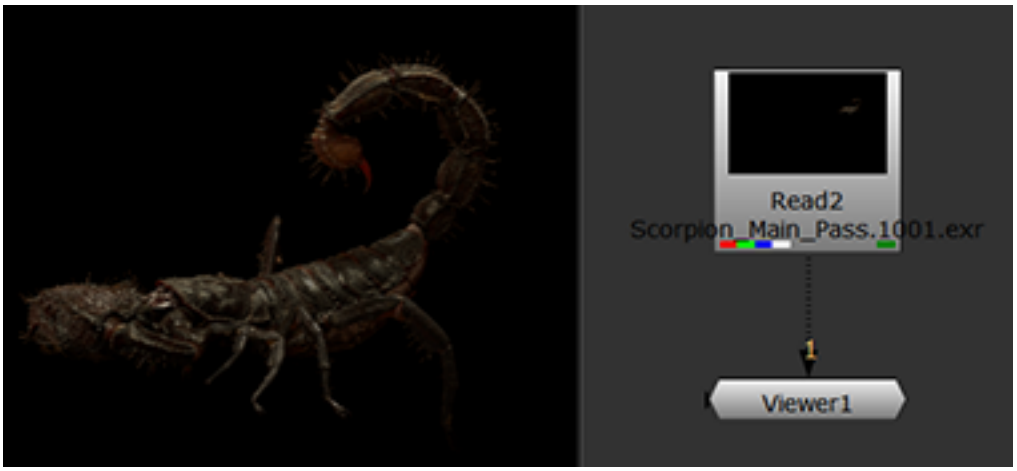
In this example, the green entry, **bunny**, is the original matte name, added while the manifest was present.

The red entry, **<3.36000126251e-27>**, is the floating-point hash value of the matte ID, wrapped by angle brackets, added after the manifest was removed.

Using the Cryptomatte Plug-in

To use the Cryptomatte plug-in:

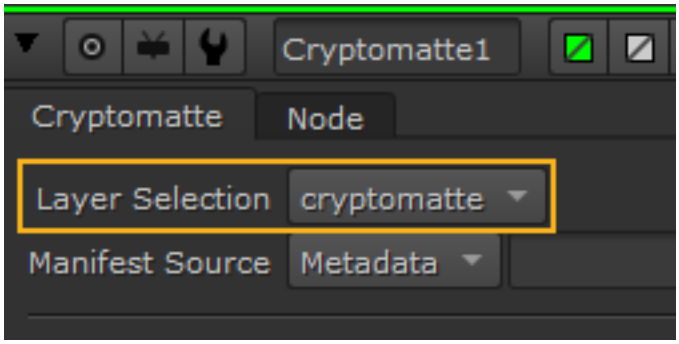
1. Read in an image or sequence with a Cryptomatte layer.



2. Place a Cryptomatte node downstream of the Read node.
   The Cryptomatte layer is visible in the Viewer.

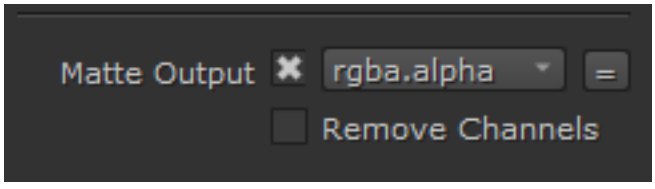


3. In the Cryptomatte **Properties**, choose a Cryptomatte layer from the **Layer Selection** menu.



If you have multiple Cryptomatte layers available, they are listed here. If you have just one Cryptomatte layer, it is selected by default.

4. Choose a manifest source option from the **Manifest Source** dropdown. If your manifest is embedded in the input image, choose **Metadata**, otherwise choose **Sidecar** and navigate to the manifest JSON file in the **sidecar** field.
5. Use the **Preview** checkbox to toggle the Cryptomatte matte boundaries visibility in the **Viewer**. This is useful for quickly switching back and forth between the selected mattes and the original input image to ensure you are selecting the correct parts of an object.
6. Click the **Matte Output** dropdown to choose the channel to which the extracted matte is written out.



If you don't want the matte to be written to a channel, you can select **none** from the **Matte Output** dropdown, or un-tick the **Matte Output** checkbox.

Note: For more information on channels in Nuke, see Channels.

7. Enable the **Remove Channels** option to remove all channels except rgba and the chosen matte output channel.
   This can be useful when using the extracted matte as you may not need all other channels.

Adding and Removing Mattes

You can add and remove mattes from the matte selection in the Cryptomatte **Properties** using the **Picker Add** and **Picker Remove** controls, or through the **Matte List**. You can also remove all mattes from the matte selection using the **Clear** button.

To add mattes using the **Picker Add** option:

1. Click the color swatch next to **Picker Add** in the Cryptomatte **Properties** to enable the eyedropper color picker.
2. **Ctrl/Cmd** + **Click** a matte to add it to the selection.
   Selected mattes are displayed in yellow in the **Viewer** and the matte names are added to the **Matte List**.



3. **Ctrl/Cmd** + **Click** to add more mattes to the selection.



Tip: You can marquee select multiple mattes by holding **Ctrl/Cmd** + **Shift** and dragging in the **Viewer**.

The alpha output is produced as a result of your selections.



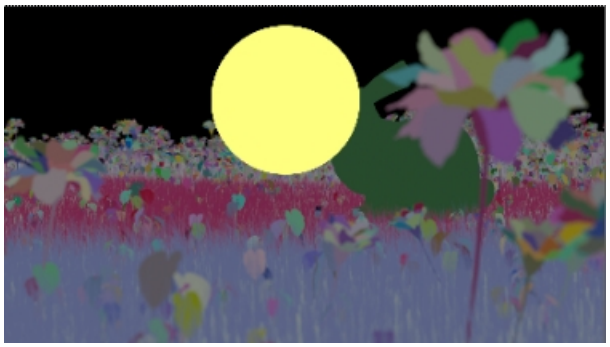
4. When you are finished selecting mattes, click the color swatch  next to **Picker Add** in the Cryptomatte **Properties** to disable the eyedropper color picker. You can also **Ctrl/Cmd** + right click in the **Viewer** to disable the eyedropper color picker.

Adding Additional Mattes Using Encryptomatte

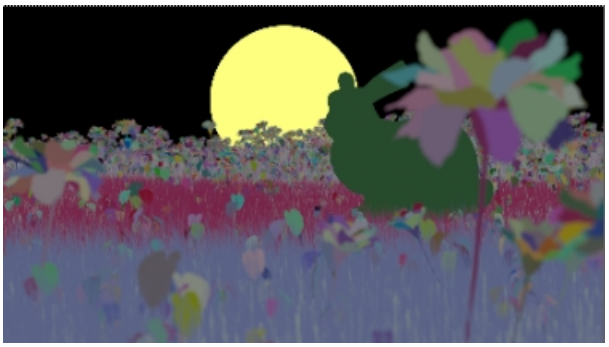
The Encryptomatte node takes any alpha input and converts it into a Cryptomatte-selectable ID, which can be selected downstream using the Cryptomatte node. The new matte can be merged over or under existing Cryptomattes and can be added to an existing Cryptomatte layer or a new layer created in the Encryptomatte node.

1. Add an Encryptomatte node upstream of your Cryptomatte node.
2. Connect a matte to the Encryptomatte node's **Matte** input. This could be a Roto or RotoPaint node or a keyer.

   The **Matte Name** is automatically filled in from the node name connected.
3. You can select **manual** to enter a Matte Name of your choice. Some nodes, such as NoOp don't auto-complete the **Matte Name**.
4. Select the **Merge Operation**, **over** or **under**, to determine how the new matte interacts with those already present in a Cryptomatte layer.



A new matte merged **over** existing mattes in the same layer.



A new matte merged **under** existing mattes in the same layer.

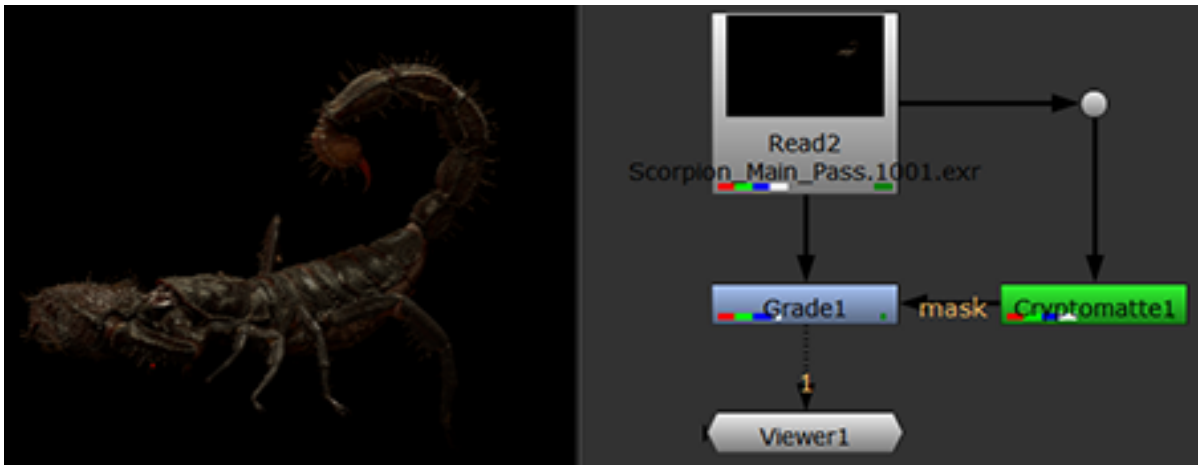
5. The new matte can be added and removed as normal using the **Picker Add** and **Picker Remove** tools.

Using the Matte Output

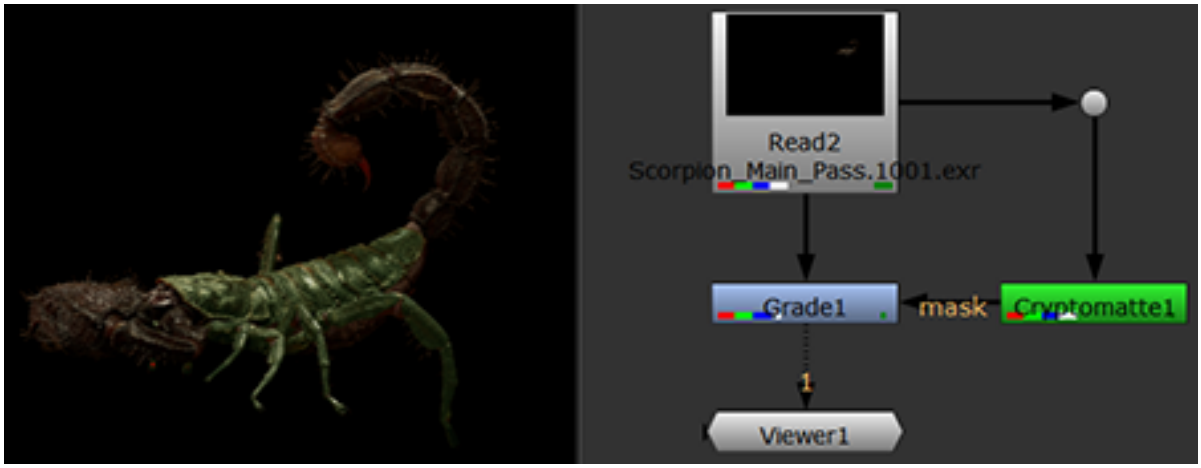
Once you have selected your mattes, you can use the chosen matte output channel as a mask, for example, for grading or to remove objects from view.

To grade your image using the selected mattes as a mask:

1. Add a **Grade** node and connect it to your image or sequence. Connect the **Mask** input to the Cryptomatte node.



2. Grade the image using the **Grade** node controls.

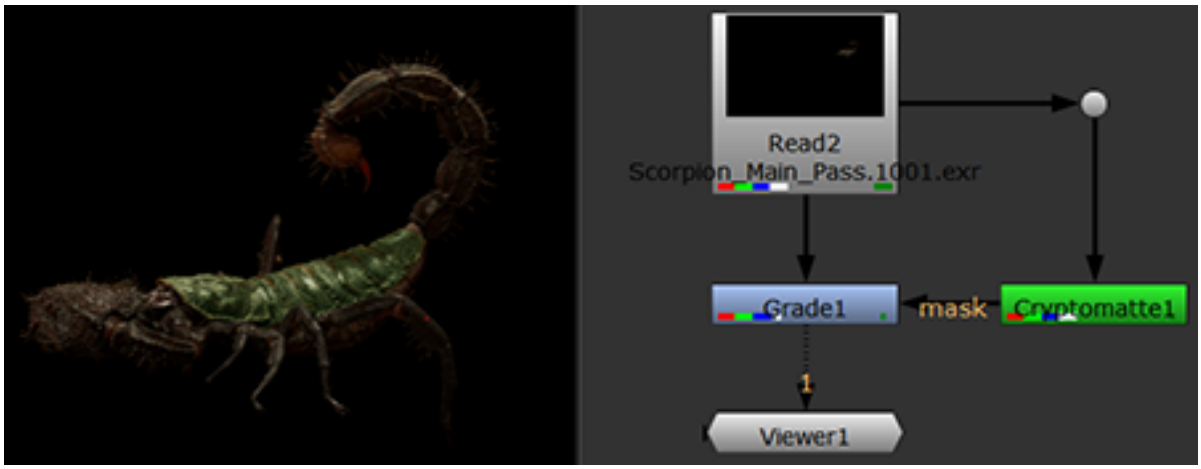


Only the masked areas are graded.

You can now select and deselect mattes to refine which parts of the image are affected by the mask.

3. Click the color swatch ■ next to **Picker Remove** in the Cryptomatte **Properties** to enable the eyedropper color picker.

4. **Ctrl/Cmd** + **Click** a matte to remove it from the selection.

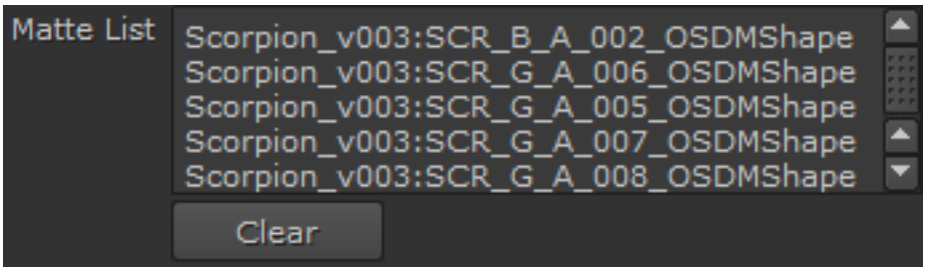


Deselected mattes are removed from the **Matte List**.

Note: You don't need to be viewing the Cryptomatte node to select and deselect mattes.

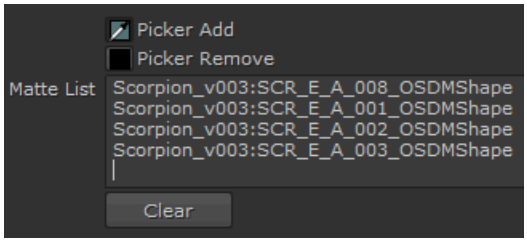
Using the Matte List

The **Matte List** displays a list of currently selected matte names. This list can be modified in text form or using the **Picker Add** and **Picker Remove** controls.

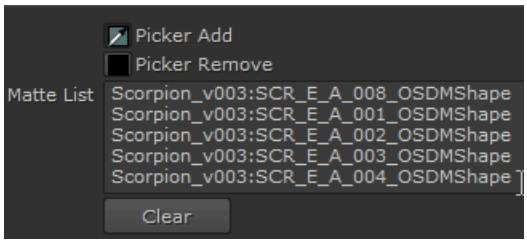


It may sometimes be useful to add or remove matte names from the **Matte List** in text form, as parts of an object may not always be visible in the **Viewer** which means selecting them using the color picker can be difficult.

For example, selecting all the scorpion's legs in this sequence would be difficult as half of them are hidden behind the body. To avoid scrubbing through the timeline to try and located the legs, you can instead add all the legs easily by typing their matte names into the **Matte List**.



In the same way, mattes that are not currently visible can be removed from the matte selection by deleting the **Matte List** entry.



The **Matte List** also supports basic wildcard functionality so asterisks (*****) can be used to select multiple mattes on one line. Asterisks (*****) can be appended to a word to represent any number or character, for example:

scorpion* selects scorpion1, scorpion2, ... scorpionN

When using **Picker Remove**, only lines in the **Matte List** referencing the exact matte ID selected with the color picker are removed from the **Matte List**, for example:

selecting scorpion1 removes scorpion1 from the **Matte List**.

Expressions referencing a set of matte IDs, such as scorpion*, cannot be removed or changed even if the set contains only the selected matte ID, for example:

if scorpion* references only scorpion1, scorpion* is not removed from the **Matte List**. You can also use the asterisk (*****) to reference all matte IDs.

Additionally, you can remove mattes using the **-** (minus) remove expression, which allows you to make more complex selections easily. For example, rather than adding multiple elements of the same object to the **Matte List** individually, you can add all mattes and selectively remove just those you don't want.

scorpion* selects scorpion1, scorpion2, ... scorpionN

-scorpion1 removes just one matte, but leaves the rest selected in the **Matte List**.

Rendering Cryptomatte Files

Nuke renders Cryptomatte layers using the Write node in the same way as other channel data, but there are a few controls in the **Properties** that must be set correctly for the layers to render as expected:

Tip: Click the **Export Write** button in the Encryptomatte node's **Properties** to automatically choose these settings.

- Set the channels dropdown to **all**. Cryptomatte data is not stored in the **rgba** layer, but in dedicated Cryptomatte layers.
- Set the **file type** to **exr** to render Cryptomatte layers. You can't render Cryptomatte layers in any other format.
- Set the **datatype** to **32 bit float**. **16 bit half** is not supported by Cryptomatte.
- Set the **metadata** control to **all metadata**. The metadata is required to determine the layers contained in the file.

All other Write node settings can be used as required. See Write for more details on the Write node's **Properties**.

Keying with Keylight

This section explains how to use the blue/green screen keyer, Keylight, in Nuke.

Quick Key

Consider this shot from The Saint, pictures courtesy of CFC and Paramount British Pictures Ltd.



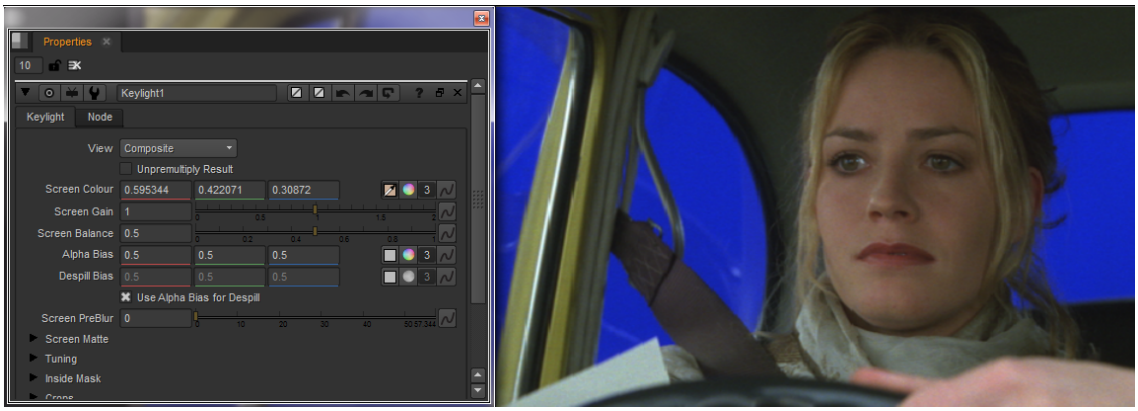
Blue screen.

The figure above is the blue screen foreground that should be composited over the background shown below.



Background.

1. Start Nuke and read in both images. From the **Keyer** menu, apply **Keylight** and attach a Viewer.
2. Click the color swatch next to **ScreenColor** to activate the eye dropper. In the Viewer, **Ctrl/Cmd+Shift+Alt**+click and drag a rectangular area over the blue pixels as shown below.



Picking the screen color also sets the **ScreenBalance**.

That's it. In many cases, this is all you need to do to perform a key, since selecting the screen color creates a screen matte and despills the foreground.

3. Switch output from **FinalResult** to **Composite** to see the foreground keyed over the background. The final composite is shown below.



Picking the screen color may be enough for a lot of keys, but there are many more tools within Nuke that can be used to tackle more complicated shots. These are described later in this chapter.

Basic Keying

The following section describes the parameters you need to do basic keying. This gives you enough to tackle most simple keys. A discussion of advanced parameters to fine tune keys and tackle complex keys can be found under Advanced Keying.

Picking the Screen Color

The screen color is probably the most important parameter and you should always pick the screen color before doing anything else. It should be set to the color of the green or blue curtain behind the foreground object. Pick the screen color directly from the image by **Ctrl/Cmd+Shift+Alt** dragging a rectangle over the blue pixels. The average value of the pixels selected is used.

Tip: If you press **Alt** when sampling a color, Nuke always samples the source image regardless of what you're looking at. This means that you can pick the blue screen color even if you are viewing the matte, status, or composite.

Tip: You can discard sampled pixels by **Ctrl**/**Cmd**+right-clicking in the Viewer.

Tip: Picking different shades of blue from the screen color can give quite different results. It's worth picking from different parts of the screen to get the best result.

Picking the **ScreenColor** creates the screen matte used to composite the foreground over the background. It also sets the **ScreenBalance** (if this has not already been set manually) and despills the foreground.

Screen Matte

Setting the screen color pulls a key, or in other words, creates a matte - the **ScreenMatte**. Setting the screen color also despills the foreground, although you can also use the **DespillBias** to remove more spill. In some cases this is enough to get a decent key. For more information on **ScreenColor**, see Screen Color.

The image below shows a well-lit blue screen behind an actor.



You should note that repeatedly picking colors does not add to previous selections and key more of the image with each click. To key more of the image, try picking different shades of blue then use the screen strength parameter. See Keying More.

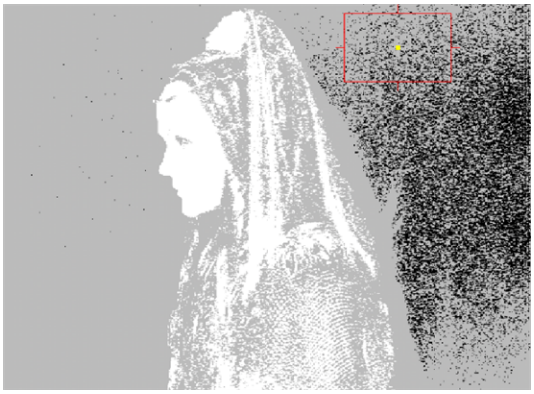
Viewing the Key

After picking the screen color, you have created a matte (the screen matte) and despilled the foreground. The result can be displayed in a number of different ways using the **View** control. You can output the final composite of the foreground over the background as an rgba, or you can output the premultiplied or unpremultiplied foreground for compositing elsewhere in the tree. The screen matte and the status view are the other two options which are useful in fine-tuning the key rather than as an output image in their own right.

The **Status** is one of the options in the **View** dropdown menu and shows an exaggerated view of the key so that you can make a more informed decision when tuning the key. The image on the right shows the **Status** display after the screen color has been picked from the image on the left.



Green screen.



Status.

Three colors are displayed:

- Black pixels show areas that are pure background in the final composite.
- White pixels show areas that are pure foreground.
- Gray pixels are a blend of foreground and background pixels in the final composite. You need gray pixels around the edge of the foreground to get a good key at the foreground edge. Pixels that are a blend between the foreground and background are shown in just one shade of gray. This is done to highlight potential problems with the key. These gray pixels may represent a foreground/background blend of 50/50 or 99/1. No distinction is made as to this ratio.

You may occasionally see other colors in the **Status** view and these are covered under View.

Keying More

To improve the key by firming up the foreground so the background doesn't show through, you should adjust the **ClipWhite** parameter. To key more of the foreground so that the background is clearer, you should use the **ClipBlack** parameter. Look at the **ScreenMatte** and the **Composite** while you're doing this. Don't overdo either of these or the edges between foreground and background become hard.

Advanced Keying

The following section describes how Keylight works under the hood as well as the parameters you need to fine tune keys and get the most out of Keylight. Basic parameters covered previously may also be covered here in more detail.

Under the Hood

Keylight is a 'color difference keyer', which means that for it to figure out a key, it compares every pixel in the image against a single color, known here as the **ScreenColor**.

View

The **View** parameter allows Keylight to render the final composite of the foreground over the background, or the foreground RGBA for compositing further down the tree. Two options, **ScreenMatte** and **Status**, are for viewing the key rather than an output. The options are:

- **Source** - shows the blue/green screen foreground.
- **SourceAlpha** - shows the alpha channel on the foreground input.
- **ScreenMatte** - this is the matte created from picking the **ScreenColor**. It does not include any inside or outside masks.
- **InsideMask** - shows the inside input. This is used to firm up the foreground matte to stop print through.
- **OutsideMask** - shows the outside input. The outside mask is used as a garbage mask to reveal the background.
- **CombinedMatte** - the screen matte, inside mask, and outside masks added together.
- **Status** - this renders an exaggerated view of the key so that minor problems are shown clearly.

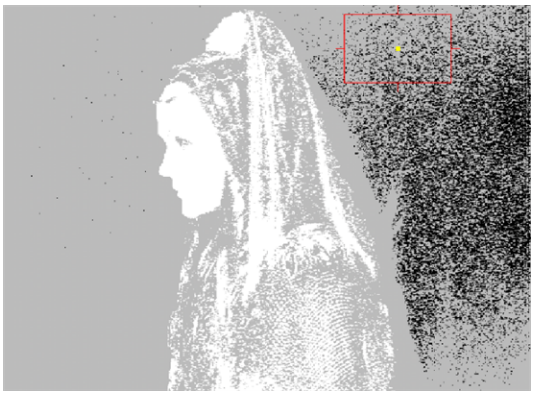
- **IntermediateResult** - use this option on shots that can only be keyed using several different keys on different parts of the image (multi-pass keying). This renders the original source image with the **ScreenMatte** generated in this Keylight node. In Keylight nodes down the tree, you should set the **SourceAlpha** in the **InsideMask** folder to **AddToInsideMask**.
- **FinalResult** - this creates a premultiplied RGBA foreground that can be composited later. There's an **UnpremultiplyResult** toggle you can use if you wish.
- **Composite** - this renders the foreground composited over the background using all mattes, spill and color corrections.

Status

Status is one of the options in the View dropdown menu and shows an exaggerated view of the key so that you can make a more informed decision when fine tuning the composite. The figure on the right shows the Status after the screen color has been picked from the image shown in the figure on the left.



Green screen.



Status.

Three colors are displayed:

- Black pixels represent pure background in the final composite.
- White pixels are pure foreground.
- Gray pixels are a blend of the foreground and background pixels. The gray is just one color to highlight any areas that are not pure foreground or background. Gray pixels do not mean the key is poor - the final composite may be fine.

You may occasionally see other colors in the **Status** view. The figure on the left shows black, white, gray, and green pixels.



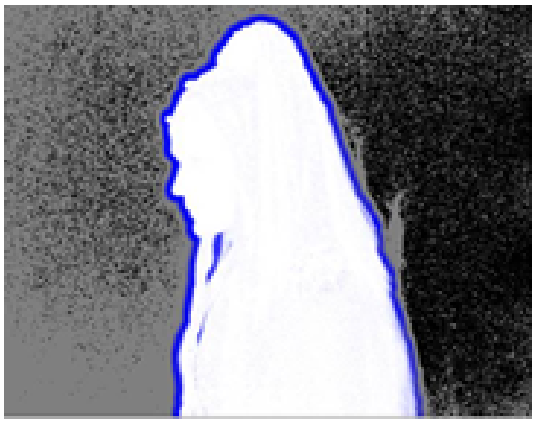
Status showing processing of the alpha channel.



Composite showing **ScreenReplaceColor**.

- Green pixels are a warning. They show you the parts of the alpha that have changed through processing the alpha channel (clipped, softened, or eroded). These areas have had the correct amount of spill removed, but the alpha has subsequently changed and the composite may no longer look right. This can be corrected using the **ScreenReplaceColor** to put back color in these areas. Above, the figure on the right is an extreme example to illustrate the point. The **ScreenReplaceColor** has been set to pure red and you can see that this mirrors the green pixels in the **Status** view.

Similarly, you may see blue pixels in the **Status**.



Status showing how the inside matte affects the foreground.



Composite showing the inside replace color.

- Blue pixels represent processed pixels in the **InsideMask** that affect the despill of the foreground. The **InsideReplaceColor** is used to modify these pixels. Another extreme example is shown above in the figure on the right. The **InsideReplaceColor** is set to pure yellow and the **InsideReplace** is **HardColor**.
- You may also see dark red pixels in the **Status**. Red pixels indicate areas where an outside mask has been used to reduce the transparency of the image.

View

The **View** parameter allows Keylight to render the final composite of the foreground over the background, or the foreground RGBA for compositing further down the tree. Two options, **Screen Matte** and **Status**, are for viewing the key rather than an output. The options are:

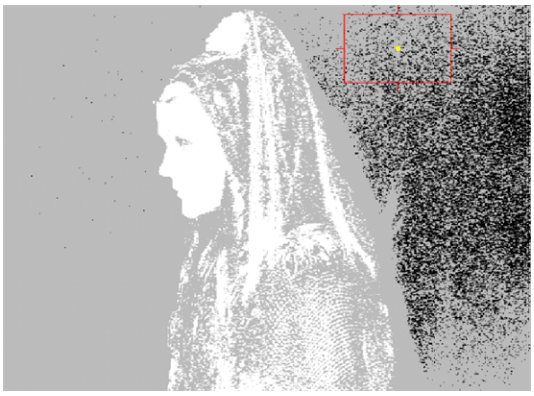
- **Source** - shows the blue/green screen foreground.
- **Source Alpha** - shows the alpha channel on the foreground input.
- **Screen Matte** - this is the matte created from picking the **Screen Color**. It does not include any inside or outside masks.
- **Inside Mask** - shows the inside input. This is used to firm up the foreground matte to stop print through.
- **Outside Mask** - shows the outside input. The outside mask is used as a garbage mask to reveal the background.
- **Combined Matte** - the screen matte, inside mask, and outside masks added together.
- **Status** - this renders an exaggerated view of the key so that minor problems are shown clearly.
- **Intermediate Result** - use this option on shots that can only be keyed using several different keys on different parts of the image (multi-pass keying). This renders the original source image with the **Screen Matte** generated in this Keylight node. In Keylight nodes down the tree, you should set the **Source Alpha** in the **Inside Mask** folder to **Add To Inside Mask**.
- **Final Result** - this creates a premultiplied RGBA foreground that can be composited later. There's an **Unpremultiply Result** toggle you can use if you wish.
- **Composite** - this renders the foreground composited over the background using all mattes, spill and color corrections.

Status

Status is one of the options in the View dropdown menu and shows an exaggerated view of the key so that you can make a more informed decision when fine tuning the composite. The figure on the right shows the Status after the screen color has been picked from the image shown in the figure on the left.



Green screen.



Status.

Three colors are displayed:

- Black pixels represent pure background in the final composite.
- White pixels are pure foreground.
- Gray pixels are a blend of the foreground and background pixels. The gray is just one color to highlight any areas that are not pure foreground or background. Gray pixels do not mean the key is poor - the final composite may be fine.

You may occasionally see other colors in the **Status** view. The figure on the left shows black, white, gray, and green pixels.



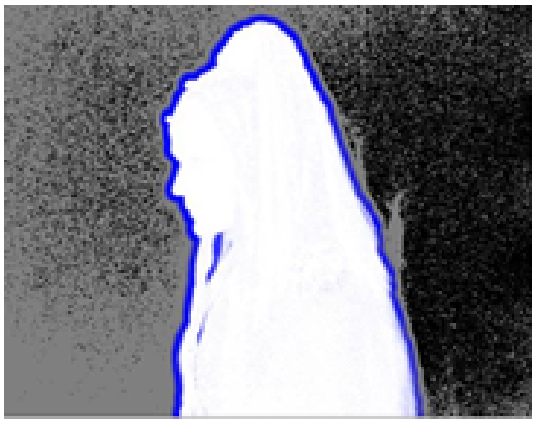
Status showing processing of the alpha channel.



Composite showing **Screen Replace Color**.

- Green pixels are a warning. They show you the parts of the alpha that have changed through processing the alpha channel (clipped, softened, or eroded). These areas have had the correct amount of spill removed, but the alpha has subsequently changed and the composite may no longer look right. This can be corrected using the **Screen Replace Color** to put back color in these areas. Above, the figure on the right is an extreme example to illustrate the point. The **Screen Replace Color** has been set to pure red and you can see that this mirrors the green pixels in the **Status** view.

Similarly, you may see blue pixels in the **Status**.



Status showing how the inside matte affects the foreground.



Composite showing the inside replace color.

- Blue pixels represent processed pixels in the **Inside Mask** that affect the despill of the foreground. The **Inside Replace Color** is used to modify these pixels. Another extreme example is shown above in the figure on the right. The **Inside Replace Color** is set to pure yellow and the **Inside Replace** is **Hard Color**.
- You may also see dark red pixels in the **Status**. Red pixels indicate areas where an outside mask has been used to reduce the transparency of the image.

Screen Color

The screen color represents the color of the pure blue (or green) screen. The first thing you should do when pulling a key is pick the **Screen Color**.

Note: If you press **Alt** when sampling a color, Nuke always samples the source image regardless of what you're looking at. This means that you can pick the blue screen color even if you are viewing the matte, status or composite.

Tip: You can discard sampled pixels by **Ctrl**/**Cmd**+right-clicking in the Viewer.

Picking the **Screen Color** creates the screen matte used to composite the foreground over the background. It also sets the **Screen Balance** and despills the foreground.

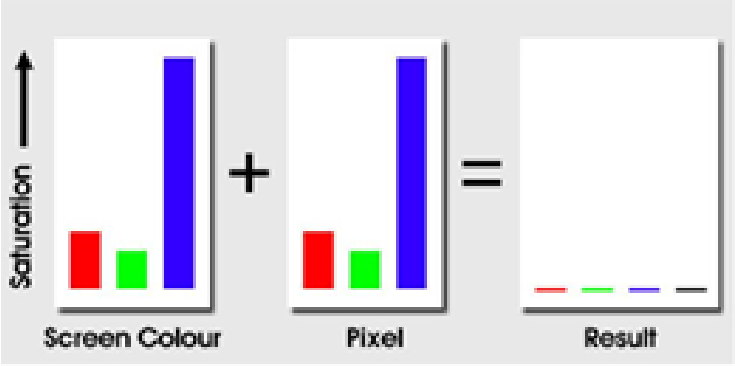
The **Screen Color** is a single color. It has a primary component, blue or green, and that has a saturation. Once the screen color has been picked, Keylight analyzes all the pixels in the image and compares the saturation of the primary component in each of these pixels with the corresponding saturation of the screen color. Keylight uses this comparison to do two things.

1. It calculates the transparency of that pixel and puts it in the alpha channel.
2. It removes the screen color from the pixel, a process known as despilling.

Tip: It's worth sampling a selection of screen (blue or green) colors and viewing the result. Picking different colors gives different results.

Background Pixel

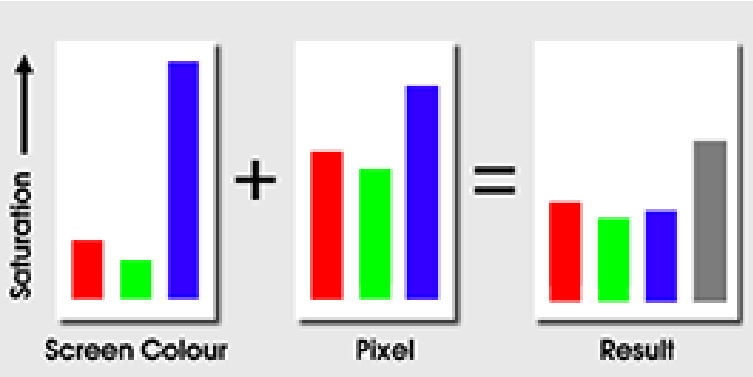
If the saturation of the pixel in the image is as strong or greater than the screen color, then it'll be a pixel from the blue screen background, and that pixel is set to completely transparent and black. See the figure below.



Blue screen pixel set alpha to zero.

Edge Pixel

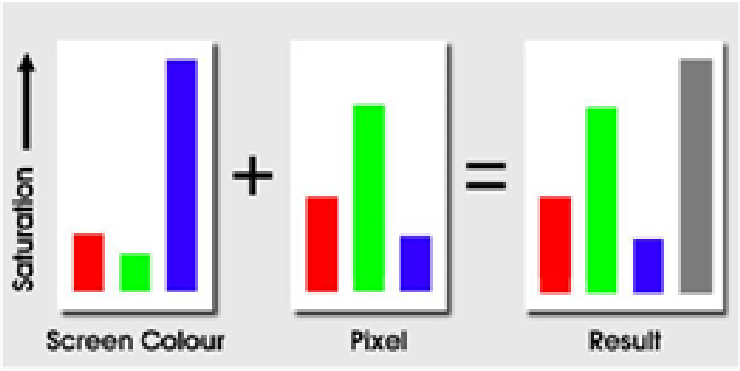
If the saturation of the pixel is less than the screen color, then it'll be the edge of the foreground object, and we subtract some of the screen color from the pixel (despilling) and set the image to semi-opaque. See the figure below.



Edge pixel gives partial alpha.

Foreground pixel

If the primary component in the pixel is not the same as the primary component of the screen color, we have a foreground pixel, and the alpha is set to completely opaque. The pixel color is not modified. See the figure below.



Foreground pixel gives full alpha.

Note: You should note that the **ScreenColor** is a single color. You are not picking lots of colors that are keyed out.

Biasing

What's biasing all about? Biasing in Keylight was originally developed for a shot in the motion picture "Executive Decision". The foreground consisted of reddish browns, but a combination of factors led to the 'green screen' being lit so that its primary component was actually slightly red.

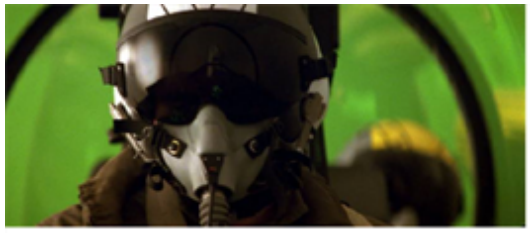


So what happens when we pick the screen color? Well because the screen was 'red', as is the foreground, our pilot ends up being keyed out as shown below.



Not a great result, I'm sure you'll agree, and much pressure was applied to the lowly programmers to get around the problem.

A work around to this is to manually color correct the image so that the background is properly green, pull the key from this corrected image, then 'un-correct' the result of that so that the foreground colors match the original. A corrected image would look something like the one shown below. The green screen is now strongly green and distinct from the foreground colors. Notice also the red cast on the pilots mask has been removed and turned into a neutral gray.



This is effectively how the Keylight developers got around the problem. They introduced the concept of a 'bias' color, which is a color cast that is removed from the source image and screen color, then a key is pulled from this modified image, then the color cast is put back. In essence, this automates the work around described above, however, it is done in a way that does not slow Keylight down at all.

For our Executive Decision shot, an appropriate color is the red cast on the pilot's mask in the source footage. Setting our bias to this now gives us the far better result as shown below.



The Bias Colors in Everyday Use

It also turns out that the bias color is actually useful for situations without strong casts, typically where there is some color spill around the edge of keys. By setting the biases to the main color that occurs near the edge of the foreground (typically flesh tones or hair tones), you allow Keylight to better discriminate between foreground and background.

Picking a bias color

To pick a bias color, click the color swatch next to **Alpha Bias** to activate an eye dropper and **Ctrl/Cmd+Shift+Alt**+drag a box over the image foreground. The average color under the box is used for the bias you have selected.

Note: If you press **Alt** when sampling a color, Nuke always samples the source image regardless of what you're looking at. For instance, you may be looking at the blue screen keyed over the background but you are picking colors from the Source image.

Tip: You can discard sampled pixels by **Ctrl**/**Cmd**+right-clicking in the Viewer.

Why are there two bias colors?

Remember that Keylight does two things, calculates a transparency and removes the screen color from the foreground. By default, one bias color, the Alpha Bias, is used for both operations. This works fine in most situations, for example, the Executive Decision shot above.

However, sometimes you can pick a bias that gives a great alpha, but performs a poor despill, and another bias that gives a great despill, but a poor alpha. Consider the blue screen from the TV series Merlin, courtesy of CFC Framestore shown below in the figure on the left.

We pick the strong blue of the background without selecting an alpha bias, and end up with the lovely alpha shown on the right, but the despill resulting from this key is poor as shown below.



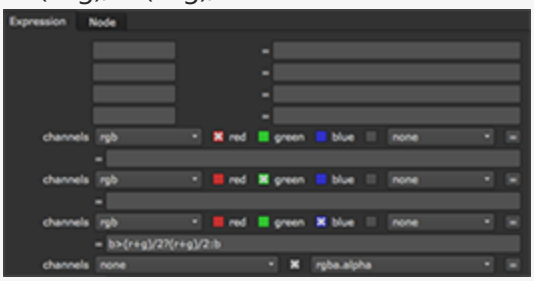
Merlin blue screen.



Nice Alpha.

Tip: There are several nodes in Nuke you can use for spill removal. For example, if you are using a greenscreen image, you can add an Expression node after your foreground image and set the expression field for the green channel to:
g>(r+b)/2?(r+b)/2:g

Similarly, you can despill a bluescreen image by setting the expression field for the blue channel to:
b>(r+g)/2?(r+g)/2:b



You can also use the HueCorrect node for despill. For more information, see Correcting Hue Only.

We can pick an alpha bias to get a better despill, but this destroys our nice alpha. The way around this is to turn off the **Use Alpha Bias for Despill**, which gives you a separate bias factor to use solely for despill calculations. If you then pick the **Despill Bias** to be something from Miranda Richardson's hair or skin tone, you keep the nice alpha, and get a good despill as well (see the figure on the right).



Poor despill.



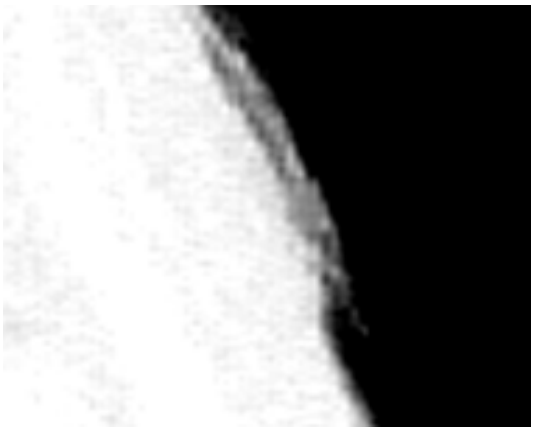
Final key, using separate **Despill** and **Alpha Biases**.

Clip Black and White

The clip levels are adjusted using two parameters - **Clip Black** and **Clip White**. Any alpha value at or below **Clip Black** is set to zero and any alpha value at or above **Clip White** is set to 1. The figure on the left shows the original alpha of an image, and the figure on the right shows the result of clipping it.



Clip Black = 0.



Clip Black = 0.5.

Notice how the gray areas in the black background have been reduced and that the gray edges have hardened up considerably. When compositing, the **Clip Black** control can be used to improve the background image if parts of the foreground are showing through. The **Clip White** control, on the other hand, can be used to firm up the center of the matte, making it less transparent to the background.



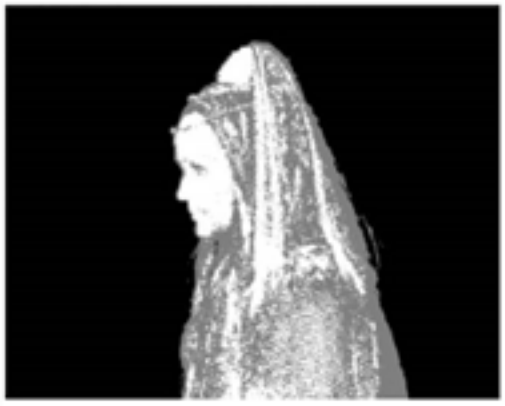
Note: If you choose to use **ClipBlack** and **ClipWhite**, you need to be really careful that you don't destroy the edges on your foreground. It is possible to use **ClipRollback** to compensate for this.

Screen Gain

The screen gain controls how much of the screen color is removed to make the screen matte. Increasing this value keys more. The figure on the left shows the **Status** after picking the **Screen Color**.



Status after picking the **Screen Color**.



Status showing the increase in **Screen Gain**.

You can clearly see that parts of the background are gray where they should be black. When composited, you may see faint pixels from the foreground where you should be seeing pure background. Increasing the screen gain fixes this, as shown in the figure on the right (above), but increasing it too much destroys your good work. Like many keying parameters it's a balance - not too much, not too little. Increasing the screen gain too much leads to the background showing through the foreground and edge detail can be destroyed. Below, the figure on the right shows this quite well.



Screen Gain = 1.05 giving a good Screen Matte.



Screen Gain = 1.50 giving background show-through

and over eroded edges.

Note the steering wheel is black when it should be white. If you look at the composite, you can see the background showing through here. Also, some of the fine hair detail on the actor, visible in the figure on the left, has been eroded in the figure on the right.

Screen Balance

The **ScreenBalance** is set automatically after picking the **ScreenColor**.

Saturation is measured by comparing the intensity of the primary component against a weighted average of the two other components. This is where the Screen Balance control comes in. A balance of 1 means that the saturation is measured against the smallest of the other two components in the screen color.

A balance of 0 means that the saturation is measured against the larger of the other two components. A balance of 0.5 measures the saturation from the average of the other two components.

The appropriate balance point for each image sequence you key is different depending on the colors in that image. Generally speaking, blue screens tend to work best with a balance of around 0.95 and green screens with a balance of around 0.5. These values are selected automatically the first time you pick the screen color. If the key is not working too well with these settings, try setting the balance to about 0.05, 0.5 and 0.95 and see what works best.

PreBlur and Tuning

Some shots can be improved by softening the foreground image that is used to generate the key. The original image is then used in the composite and color corrections. The **Screen PreBlur** parameter is used to do this. DV footage or grainy shots may benefit from subtle use of this control.

Tuning

Keylight creates the screen matte after the screen color has been picked. You can make fine adjustments to this matte using the **Gain** controls. Increasing the gain controls makes the screen matte more transparent by increasing the amount of screen color showing through the matte. This tends to tint the edges the opposite of the screen color (for blue screens, edges become yellow). Decreasing the gain makes the main matte more opaque by reducing the amount of screen color showing through the matte.

The matte can be adjusted independently in the shadows, midtones, and highlights, giving more control than the clipping levels.

The level of the midtones can be adjusted too. For example, if you are working on a dark shot you may want to set the midtone level to a dark gray to make the gain controls differentiate between tones that would otherwise all be considered shadows.

Screen Processing

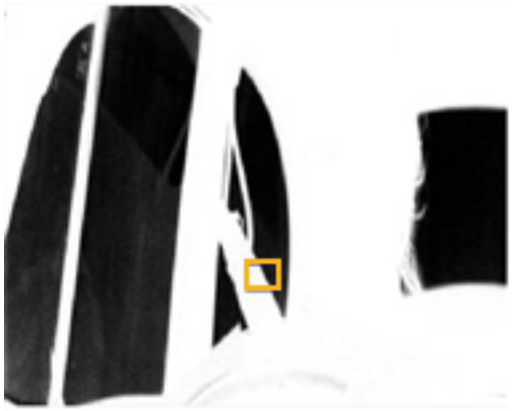
Once you have picked the screen color and got the screen matte, you may wish to process this matte using the parameters in the **Screen Matte** group. The matte can be adjusted using clipping levels; it can be eroded or grown, despotted, and softened.

Two-stage keying

Consider this example. Having applied Keylight and picked the screen color, you have good edges to your matte but the background is showing through the foreground. You could fix this by tweaking the **Clip White**, but in doing so it ruins your edges. One way round this is a two stage key (another way is using **Clip Rollback**). In the first key, you process the screen matte using the clipping levels to give a harsh black and white matte, then soften and erode it. Switch **View** to **Intermediate Result** to output the original green screen with the eroded matte as an RGBA. Then, use this as the input to another Keylight node. In this second node, pick the screen color to give good edges but with transparent foreground. Don't process this matte, instead use the input alpha channel to fix the transparent foreground. Just set **Source Alpha** in the **Inside Mask** folder to **Add To Inside Mask**.

Clip Rollback

Pulling a screen matte (the figure on the left) typically produces lots of transparency (gray) in the matte at the edges. This is good since this is what you need to key hair well. You may also get transparency in the foreground as shown in the figure on the right. This is bad as your subject appear slightly see-through, and this should be corrected.



Screen matte highlighting the close up view as shown in the figure on the right.



Close up screen matte showing unwanted (gray) transparency in the (white) foreground.

You can do this by connecting a matte into the third (**InM**) input, or you can use the **Clip White** parameter to turn these gray pixels white. This cleans up the foreground (the figure on the left) but it also destroys the edge detail you want to keep. This is where **Clip Rollback** comes in. This is used to put back the edges to restore the detail that was lost. A rather exaggerated clip rollback is shown in the figure on the right to illustrate the point.



Clip White has been used to remove the unwanted gray pixels in the white matte.



Clip Rollback has been used to restore the unwanted erosion of the edge.

Dilate

This control should not normally be used as eroding the edges can produce a very poor key. However, the **Screen Dilate** parameter allows you to grow (if greater than zero) or shrink (if less than zero) the alpha in the **Screen Matte**. These controls are sub-pixel accurate.



Screen Matte.



Eroded Matte.

Softness

Occasionally, it is useful to be able to blur the matte. Use **Screen Softness** for this. The most common example would be to pull a very harsh matte that you would use as an inside matte further down the tree. For this, you'd soften and erode the screen matte.

Despot

This controls how much to simplify the matte. It coagulates similar regions so that, for example, black specks in the white matte can be absorbed by the surrounding white areas. Increasing the **Screen Despot Black** removes isolated spots of black in the white matte. Increasing **Screen Despot White** removes isolated spots of white in the background up to that size.



Eroded matte.



Despot.

Mattes

There are 4 mattes in Keylight.

1. Screen Matte
2. Inside Mask
3. Outside Mask
4. Alpha (Composite Alpha)

The **Screen Matte** is generated by the Keylight algorithm after the screen color has been picked. It can be processed (clipped, eroded, etc) by the screen matte processing tools.

The **Inside Mask** is the hold out matte. It is used to confirm areas that are definitely foreground. If your subject has blue eyes and is being shot in front of a blue screen, this mask can be used to put back the eyes. This mask is taken from the **InM** input to Keylight. The embedded alpha channel of the foreground input can be added to this mask using the **Source Alpha** parameter in the **Inside Mask** folder.

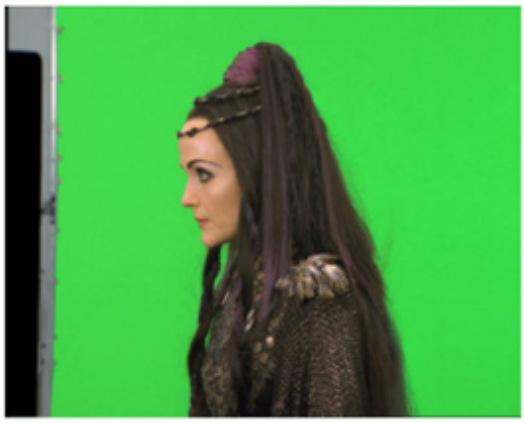
The **Outside Mask** is the garbage mask and is used to remove unwanted objects (lighting rigs, etc) from the foreground. The mask is taken from the **OutM** input to Keylight. The luminance or the alpha of this input is set using the **OutM Component** parameter.

The matte used to blend the foreground and background in the final composite is the alpha displayed in the alpha channel of the composite. This matte is the combination of the screen matte, inside matte, and outside matte.

Inside and Outside Masks

If you can't adequately improve the screen matte using the clip levels, you can create a matte in Nuke around the pixels you definitely want to be foreground or background and use this as a mask input. The inside mask makes the foreground less transparent and the outside mask is used to clean up the background that might have bits of the foreground showing through. It is sometimes referred to as the hold out mask.

The outside mask (garbage mask) is often used to clean up screens that are not a constant color or have lighting rigs in shot by forcing the alpha transparent.



The inside mask can be used to keep elements in the foreground that you don't want to lose (an actor's blue eyes in front of a blue screen). These masks should normally be softened externally to blend into the screen matte.

The below image shows the Bezier spline drawn around the lighting rig on the left side of the screen.



Connect the mask to the **OutM** input of Keylight and switch the parameter **OutM Component** to **Alpha**. The outside mask forces that part of the image to be in the background, thus keying out the rig.

Source Alpha

This parameter determines what to do with any embedded alpha in the original source image. You need this if you are doing multiple keys on different parts of the image with the View output set to **Intermediate Result**.

- **Ignore** - this does not add any embedded alpha to the screen matte.
- **Add To Inside Mask** - the embedded alpha is added to the inside mask. You should select this when multipass keying with **Output View** set to **Intermediate Result**.

• **Normal** - the embedded alpha is used to composite the image.

Color Replacement

Remember that Keylight does two things - it removes the screen color to despill the image and generates an alpha (**Screen Matte**) to composite the foreground over the background layer.

If you then process the **Screen Matte**, for example, by eroding the alpha or changing the clip levels, Keylight would be removing the wrong amount of screen color from the pixels whose transparency has now changed. The **Screen Replace** instructs Keylight how to deal with such pixels. The **Status** displays which pixels use a replace method. Those pixels that use a replace method because the alpha processing tools modified the transparency are green, whilst those pixels whose transparency was modified by the inside matte are blue. See the Status View under View.

There are four options to the replace method. These are:

1. **None** - the despilled image is left untouched if the alpha is modified.
2. **Source** - the image has a corresponding amount of the original pixel (screen color and all) reintroduced/removed if the alpha is changed.
3. **Hard Color** - the despilled image has a corresponding amount of the **Screen Replace Color** added for any increase in alpha.
4. **Soft Color** - the despilled image has a corresponding amount of the **Screen Replace Color** added for any increase in alpha, however, it attempts to modulate the luminance of the resulting pixel so that it matches the original pixel. This gives a more subtle result than the **Hard Color** option.

Inside mask

If the changes to the screen matte are due to an inside mask, the **Inside Replace** and **Inside Replace Color** parameters can be used to modify the color in these areas just like the **Screen Replace** parameters described above.

Edges

Built-in crop tools are included to quickly remove parts of the foreground at the edges of the image. It can also be useful in tidying up a matte at the edges where luminance changes in the blue screen are proving difficult to key out.

With **X Method** and **Y Method** set to **Color** and **Edge Color,** set to pure blue (for a blue screen), set the **Left** to crop out the left-hand side of the image revealing the background. The figures below show the changes to the **Combined Matte** with cropping.



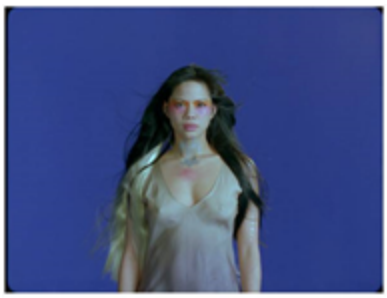
Left = 0.



Left = 0.35.

InM component

The component (luminance or alpha channel) of the inside mask input that is used in the calculations. Typically, this is a garbage matte that covers the area you know to be 100% foreground.



A bluescreen image.



An inside matte.

Tip: To avoid having to roto the inside matte throughout the clip, you can connect another Keylight node to the **InM** input and use it to create a hard, dilated key (set **Screen Dilate** to a low value).

OutM component

The component (luminance or alpha channel) of the outside mask. Typically, this is a garbage matte that covers the area you know to be 100% background.



A bluescreen image.



An outside matte.

Keying with Primatte

This section explains how to use the blue/green screen keyer, Primatte, in Nuke.

Quick Start

1. Connect the Primatte node to your background and foreground images. See Connecting the Primatte Node.
2. Click on the **Auto-Compute** button. Primatte attempts to automatically sample the backing screen color and perform the cleanup phases of keying with Primatte.

   If you get good results, jump ahead to spill removal. See Spill Removal - Method #1.

   If you don't get the results you wanted from **Auto-Compute**, continue to step 3.
3. Click the **operation** dropdown, select **Smart Select BG Color**, and manually sample the targeted background color by holding **Ctrl/Cmd** and clicking in the Viewer.



Tip: You can discard sampled pixels by **Ctrl**/**Cmd**+right-clicking in the Viewer.

   See Smart Select BG Color for more information.
4. Click the **operation** dropdown, select **Clean BG Noise**, and clean up any remaining white regions in the dark, bluescreen area by sampling in the Viewer.

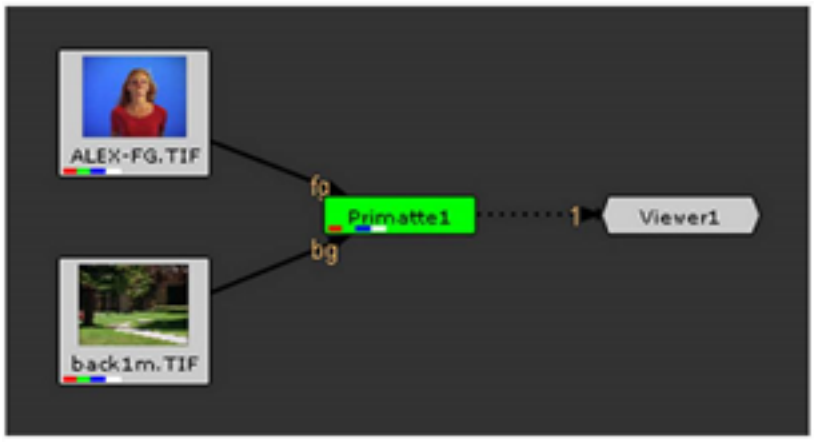
   See Clean BG Noise for more information.
5. Click the **operation** dropdown, select **Clean FG Noise**, and clean up any dark regions in the foreground by sampling in the Viewer.

   See Clean FG Noise for more information.
6. Remove any spill using the spill removal tools. See Spill Removal - Method #1.

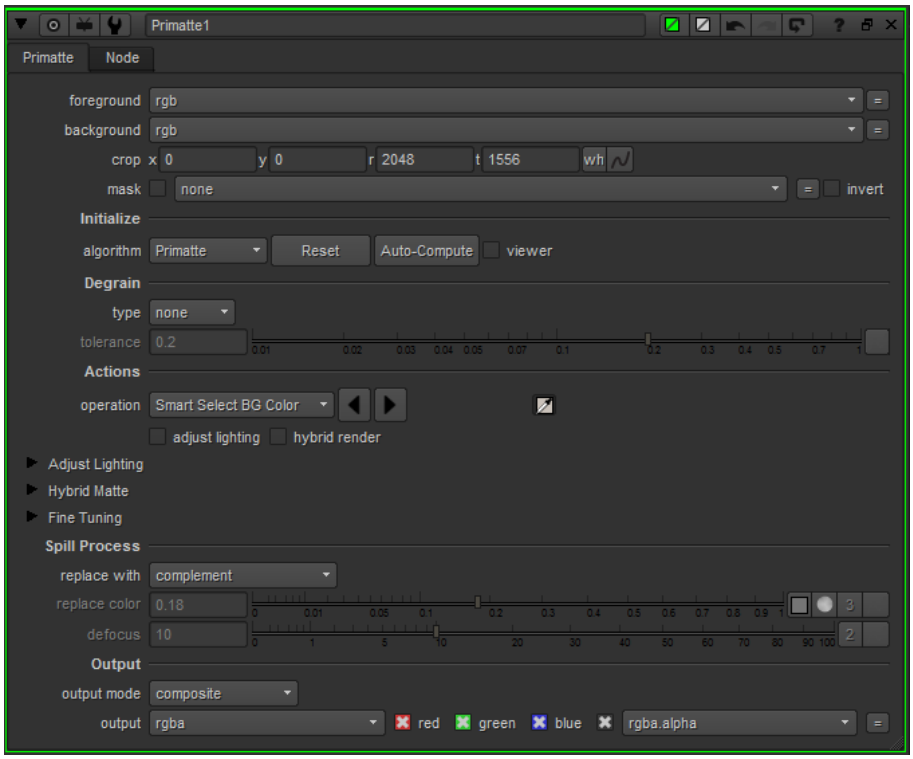
Connecting the Primatte Node

1. Start up Nuke and create a Primatte node (**Keyer** > **Primatte**).
2. Connect a foreground image to the Primatte node's **fg** input and a background image to the **bg** input.

3. Add a Viewer node so you can see the result.



4. When you select the Primatte node, the Primatte properties panel displays.



Primatte Basic Operation Tutorial

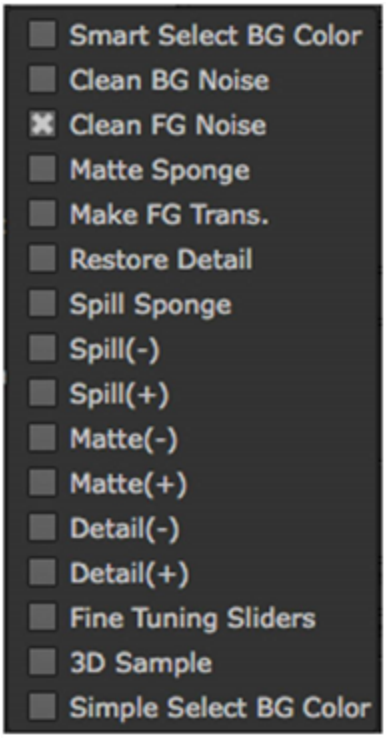
This describes the operation of the Primatte node in Nuke. A more detailed explanation of how the Primatte algorithm actually works can be found under The Primatte Algorithm .

Auto-Compute

Primatte has a feature that attempts to eliminate the first three steps of the more standard keying procedure. The **Auto-Compute** button is a good starting point and it may make your keying operation much easier.

1. Click on the **Auto-Compute** button. Primatte attempts to automatically sense the backing screen color, eliminate it, and even get rid of some of the foreground and background noise that would normally be cleaned up in the **Clean BG Noise** and **Clean FG Noise** phases of keying with Primatte.
2. If you get good results then jump ahead to the spill removal tools. See Spill Removal - Method #1.
3. If you don't get the results you wanted from **Auto-Compute**, please continue from this point on to get the basic Primatte operation procedures. See Smart Select BG Color

The basic functionality for the Primatte interface is centered around the **operation** dropdown menu and the Viewer window.



There are four main steps to using the Primatte and **Select BG Color** is the first step.

Smart Select BG Color

Ensure that the **Smart Select BG Color** action is selected (it should be at this time as it is the default **operation** mode).

Position the cursor in the bluescreen area (or whatever background color you are using), usually somewhere near the foreground object. Hold the **Ctrl/Cmd** key down and sample the targeted background color. Release the mouse button and Primatte starts the compositing process. If the foreground shot was done under ideal shooting conditions, Primatte has done 90-95% of the keying in this one step and your image might look like this.



However, if you have a very unevenly lit backing screen, you may not be getting the results you're after. If this is the case, enable **adjust lighting** on the Primatte properties. For more information, see Actions Section.

Note: Primatte works equally well with any color backing screen. It does not have to be a specific shade of green or blue.

Tip: If you dragged the cursor in the blue area, Primatte averages the multi-pixel sample to get a single color to adjust to. Sometimes Primatte works best when only a single pixel is sampled instead of a range of pixels. The color selected at this point in the Primatte operation is critical to the operation of the node from this point forward. Should you have difficulties further along in the tutorial after selecting a range of blue shades, try the **Smart Select BG Color** operation again with a single dark blue pixel or single light blue pixel. You can also switch to the alpha channel view and click around in the bluescreen area and see the different results you get when the initial sample is made in different areas.

Tip: If you would rather make a rectangular selection and not use the default 'snail trail' sampling method, you can do a **Ctrl+Shift+**drag sample.

Tip: You can discard sampled pixels by **Ctrl/Cmd**+right-clicking in the Viewer.

Tip: If the foreground image has a shadow in it that you want to keep it in the composite, do not select any of the dark blue pixels in the shadow and the shadow comes along with the rest of the foreground image.

The second and third steps in using Primatte require viewing the matte or alpha view in the Viewer window. Press the **A** key on the keyboard to change to the alpha view. The image displayed changes to a black and white matte view of the image that looks like this.



Clean BG Noise

Set **operation** to **Clean BG Noise**. If there are any white regions in the dark, bluescreen area, it is noise (or shades of blue that did not get picked up on the first sample) and should be removed. Sample through these white noise regions and release the pen or mouse button to process the data and eliminate the noise. Repeat this procedure as often as necessary to clear all the noise from the background areas. Sometimes increasing the brightness of your monitor or the screen gamma allows you to see noise that would otherwise be invisible.

Note: You do not need to remove every single white pixel to get good results. Most pixels displayed as a dark color close to black in a key image become transparent and virtually allow the background to be the final output in that area. Consequently, there is no need to eliminate all noise in the bluescreen portions of the image. In particular, if an attempt is made to meticulously remove noise around the foreground object, a smooth composite image is often difficult to generate.

Tip: When clearing noise from around loose, flying hair or any background/foreground transitional area, be careful not to select any of areas near the edge of the hair. Leave a little noise around the hair as this can be cleaned up later using the **FineTuning** sliders.



Before background noise removal.



After background noise removal.

Clean FG Noise

If there are dark regions in the middle of the mostly white foreground object, that is, if the key is not 100% in some portion of the targeted foreground, select **Clean FG Noise** from the **operation** dropdown menu. Use the same techniques as for **Clean BG Noise**, but this time sample the dark pixels in the foreground area until that area is as white as possible.



Before foreground noise removal.



After foreground noise removal.

Tip: After sampling the backing screen color and finding that the edges of the foreground object look very good, you sometimes find that an area of the foreground object is transparent. This is due to the foreground containing a color that is close to the backing screen color. When this transparency is removed using the **Clean FG Noise** operation, the edge of the foreground object picks up a fringe that is close to the backing screen color and it is very hard to remove without sacrificing quality somewhere else on the image. To tackle this problem, you can enable **hybrid render**. For more information, see Actions Section.

These were the steps necessary to create a clean matte or key view of the image. With this key, the foreground can be composited onto any background image. However, if there is spill on the foreground object from light that was reflected off the background, a final operation is necessary to remove that background spill get a more natural looking composite.

For the fourth step in the Primatte operation, return the **RGB** view to the Viewer window by clicking again on the **A** keyboard key. This turns off the alpha channel viewing mode; the Viewer window again displays the **RGB** view with the background image (if you connected one to the Primatte node).

The sample image below has gone through the first three steps and has examples of spill. Notice the blue fringe to her hair and a blue tint on her right cheek, arm and chest.



Blue spill visible.

Spill Removal - Method #1

There are three ways in Primatte to remove the spill color. The quickest method is to select **Spill Sponge** from the **operation** dropdown menu and then sample the spill areas away. By just positioning the cursor over a bluish pixel and sampling it, the blue disappears from the selected color region and is replaced by a more natural color. Additional spill removal should be done using the **Fine Tuning** tools or by using the **Spill(-)** operation. Both are explained further on in this chapter.

Note: All spill removal/replacement operations in Primatte can be modified using the **Spill Process replace with** tools. Spill can be replaced with either the **complement** of the background color, a **solid** color you've selected, or by colors brought from a **defocused background**. Depending on the spill conditions, one of these options should provide the results you are looking for. See the information in Replacing Spill for more details.

Note: Primatte's spill removal tools work on color regions. In the image in Clean FG Noise, samples should be made on the light flesh tones, the dark flesh tones, the light blonde hair, the dark blonde hair, and the red blouse color regions. One sample in each color region removes spill from all similar colors in the foreground image.

If the spilled color has not been totally removed using the **Spill Sponge** or the result of the **Spill Sponge** resulted in artifacts or false coloring, the **Spill(-)** tool should be used instead for a more subtle and sophisticated removal of the spilled background color. This is discussed in Actions Section.

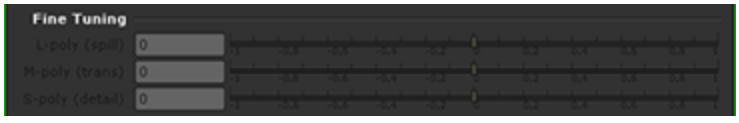
Tip: There are several nodes in Nuke you can use for spill removal. For example, if you are using a greenscreen image, you can add an Expression node after your foreground image and set the expression field for the green channel to:
g>(r+b)/2?(r+b)/2:g

Similarly, you can despill a bluescreen image by setting the expression field for the blue channel to:
b>(r+g)/2?(r+g)/2:b

You can also use the HueCorrect node for despill. For more information, see Correcting Hue Only.

Spill Removal - Method #2

1. Select the **Fine Tuning Sliders** in the **operation** dropdown menu. This activates the **Fine Tuning** sliders.



2. In the Viewer, zoom into an area that has some blue edges or spill.
3. Using the cursor, sample a color region that has some spill in it. When you let up on the pen or mouse button, Primatte registers the color selected (or an average of multiple pixels) in the current color swatch.

4. For most images, the **L-poly (spill)** slider is all that is required to remove any remaining blue spill. Move the slider to the right to remove spill color from the sampled pixels. Move it to the left to move the selected pixels toward the color in the original foreground image.

   When using the **L-poly (spill)** slider, spill color replacement is replaced based on the setting of the **Spill Process replacement with** control. For more information on these tools, see the section of this chapter on Replacing Spill .



Tip: It is better to make several small adjustments to the blue spill areas than a single major one.

5. You can use the other two sliders in the same way for different key adjustments. The **S-poly (detail)** slider controls the matte softness for the color which is closest to the background color. For example, you can recover lost smoke in the foreground by selecting the **Fine Tuning Sliders** action, sampling the area of the image where the smoke just starts to disappear and moving the **S-poly (detail)** slider to the left. The **M-poly (trans)** slider controls the matte softness for the color which is closest to the foreground color. For example, if you have thick and opaque smoke in the foreground, you can make it semi-transparent by moving the **Transparency** slider to the right after selecting the pixels in the **Fine Tuning Sliders** mode.

Tip: If the foreground image changed color dramatically during the fine tuning process, you can recover the original color by selecting an area of the off-color foreground image and moving the **L-poly (spill)** slider slightly to the left. This may introduce spill back into that color region. Again, use the **Fine Tuning Sliders** option to suppress the spill, but make smaller adjustments this time.

Spill Removal - Method #3

If these final spill suppression operations have changed the final compositing results, you may have to return to earlier operations to clean up the matte. If the composite view looks good, it is a good idea to go back and take a final look at the alpha channel view. Sometimes in the Primatte operation, a 100% foreground area (all white) becomes slightly transparent (gray). You can clean those transparent areas up by using the **Matte Sponge** tool.

1. Select the **Matte Sponge** tool in the **operation** dropdown menu.
2. Click on the transparent pixels and they become 100% foreground. All of the spill-suppression information remains intact.

Note: The **Matte(+)** operation also works to solve this problem. For more information, see Sampling Tools.

Sampling Tools

Primatte's spill, matte, and detail sampling tools allow you to make fine adjustments to balance between these aspects of your composite. You can find these tools in the **operation** dropdown menu in the Primatte properties panel.

The Spill Sampling Tools

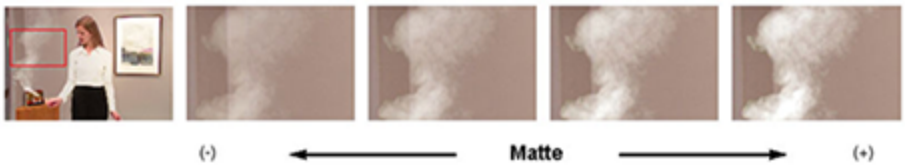
Using the **Spill(+)** and **Spill(-)** modes, you can gradually remove or recover the spill intensity on the foreground object by sampling the referenced color region repeatedly. The difference to the **Spill Sponge** tool is that **Spill Sponge** removes the spill component in a single action at one level and does not allow sampling the same pixel a second time. Even though just a small amount of spill needs to be removed, the **Spill Sponge** removes a preset amount without allowing any finer adjustment. To use the **Spill(+)** and **Spill(-)** tools:



1. Select the **Spill(-)** sampling tool from the **operation** dropdown menu.
2. In the Viewer, zoom into an area that has some blue edges.
3. Click on a pixel with some spill on it.
4. Repeated clicking incrementally removes the spill. Continue this operation until you reach a desired result.
5. To add spill, select the **Spill(+)** tool and repeat steps from 2 to 4.

The Matte Sampling Tools

The **Matte(+)** and **Matte(-)** modes are used to thicken or attenuate the matte information. If you want a thinner shadow on a foreground object, you can use the **Matte(-)** mode as many times as you like to make it more transparent. On the other hand, you can use the **Matte(+)** mode to make the matte thicker in that color region.



The Detail Sampling Tools

The **Detail(+)** and **Detail(-)** modes are a refined version of **Clean BG Noise** and **Restore Detail**. For example, when you see some dilute noise in the backing area but don't want to remove it completely because it affects some fine detail in a different area, try using **Detail(-)**. It attenuates the noise gradually as multiple samples are made on the pixel. You should stop the sampling when important fine details start to disappear.



Replacing Spill

The proper processing of spill on foreground objects is one of the many useful features of Primatte. You can move between four modes to see how they affect the image clip you are working with. Under **Spill Process**, in the **replace with** dropdown menu, you can select the following options:

- **no suppression** - In this mode, no suppression is applied.
- **complement** - This is the default spill replacement mode. This mode maintains fine foreground detail and delivers the best quality results. If foreground spill is not a major problem, this mode is the one that should

be used. The **complement** mode is sensitive to foreground spill. If the spill intensity on the foreground image is rather significant, this mode may often introduce serious noise in the resulting composite.



The **complement** mode maintains fine detail.



Serious noise in the composite.

- **solid color** - In the **solid color** mode, the spill component is replaced by a palette color that you can pick. While the **complement** mode uses only the backing color complement to remove small amounts of spill in the original foreground, the **solid color** mode tries to assuage the noise using the user-defined palette color. Changing the palette color for the solid replacement, you can apply good spill replacement that matches the composite background. Its strength is that it works fine with even serious blue spill conditions.

  On the negative side, when using the **solid color** mode, fine detail on the foreground edge tends to be lost. The single palette color sometimes cannot make a good color tone if the background image has some high contrast color areas.

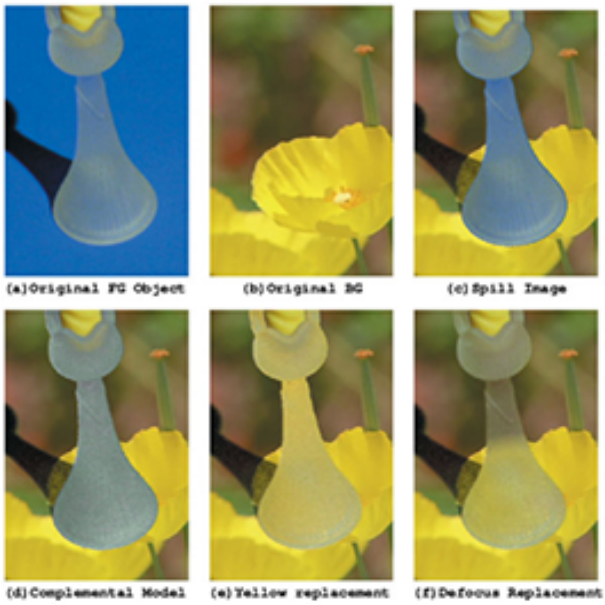


Smooth spill processing with solid color replacement.

- **defocused background** - The **defocused background** mode uses a defocused copy of the background image to determine the spill replacement colors instead of a solid palette color or just the complement color. This mode can result in good color tone on the foreground object even with a high contrast background. As in the example below, spill can even be removed from frosted glass using this feature and still retain the translucency.

  On the negative side, the **defocused background** mode sometimes results in the fine edge detail of the foreground objects getting lost. Another problem could occur if you wanted to later change the size of the

foreground image against the background. Since the background/foreground alignment would change, the applied color tone from the defocused image might not match the new alignment.



Blue suppression of a frosted glass object.

Primatte Controls

On the Primatte properties panel, you can further adjust several controls.

Initialize Section

In the **Initialize** section, you can select which **algorithm** Primatte uses to calculate your keying result:

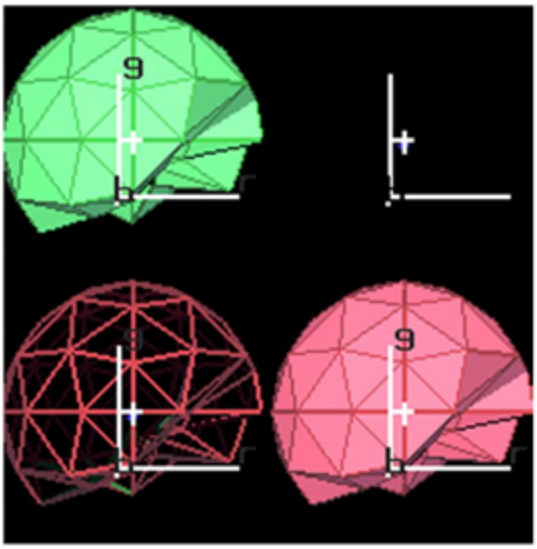


Primatte algorithm dropdown menu.

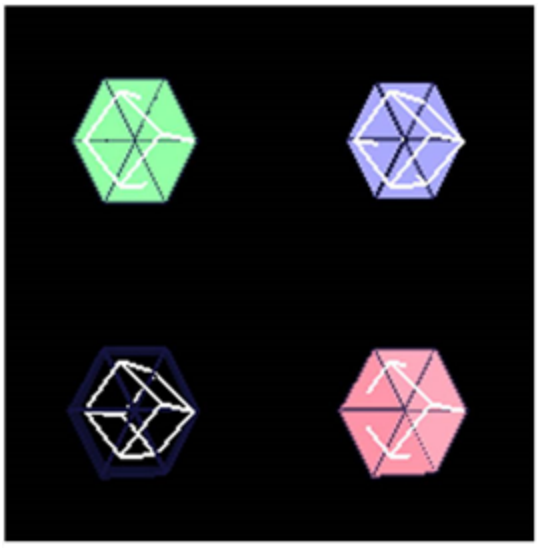
- **Primatte** - The **Primatte** algorithm delivers the best results and supports both the **solid color** and the **complement** color spill suppression methods. It is the algorithm that uses three multi-faceted polyhedrons (as described further down in the this chapter) to separate the 3D RGB colorspace. It is also the default algorithm mode and, because it is computationally intensive, it may take longer to render.

- **Primatte RT** - is the simplest algorithm and therefore, the fastest. It uses only a single planar surface to separate the 3D RGB colorspace (as described further down in this chapter) and, as a result, does not have the ability to separate out the foreground from the backing screen as carefully as the above Primatte algorithm. Other disadvantages of the **Primatte RT** algorithm is that it does not work well with less saturated backing screen colors and it does not support the **complement** color spill suppression method.
- **Primatte RT+** - this is in between the above two options. It uses a six planar surface color separation algorithm (as described further down in this document) and delivers results in between the other two in both quality and performance. Other disadvantages of the **Primatte RT+** algorithm is that it does not work well with less saturated backing screen colors and it does not support the **complement** color spill suppression method.
- **Reset** - Clicking this resets all of the Primatte properties to their initial values.
- **Auto-Compute** - This can be used as the first step in the Primatte operation. Its purpose is to try and do the first three steps of the Primatte operation for you. It tries to automatically detect the backing screen color, remove it, and do some clean-up on the foreground and background noise. If the clip was shot with an evenly lit, well-saturated backing screen, the **Auto-Compute** button leaves you with an image that may only need some spill removal to complete your keying operation. See Auto-Compute.
- **viewer** - This opens a Primatte Viewer that displays a graphical representation of the Primatte algorithms and allows you to see what is happening as the various Primatte tools are used. It is a passive feature that has no adjustment capabilities, but it may prove useful in evaluating an image as you operate on it. See Primatte Viewer Tools.

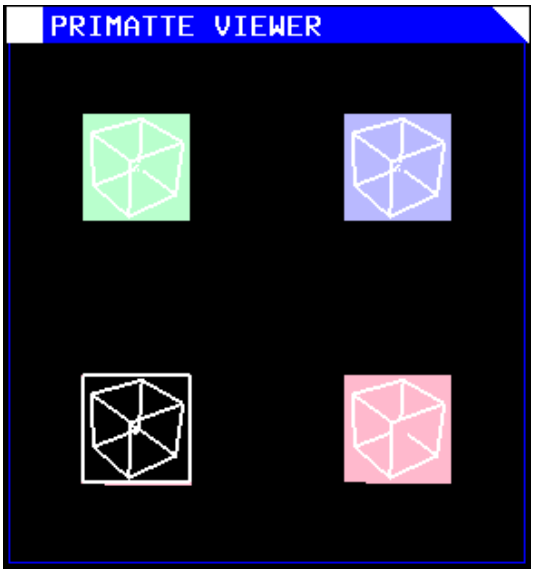
When you select **viewer**, you are presented with a window that may look similar to one of these images (depending on which Primatte **algorithm** you have selected).



Primatte algorithm.



Primatte RT+ algorithm.



Primatte RT algorithm.

The different algorithms are described in more detail in a later section of this chapter, see The Primatte Algorithm . For a description of the Primatte viewer tools, see Primatte Viewer Tools.

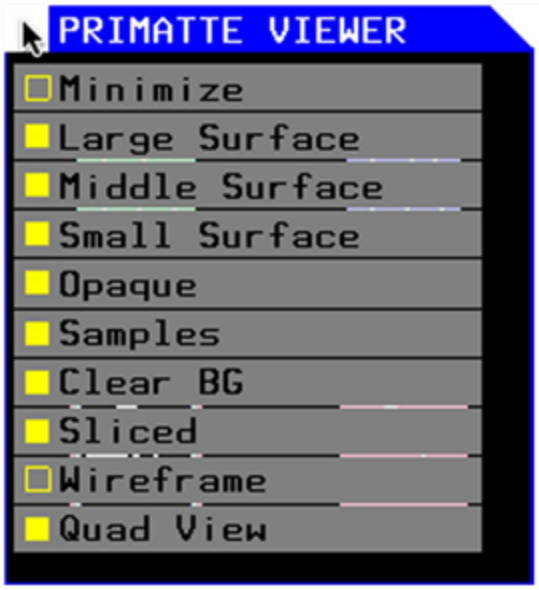
Primatte Viewer Tools

To navigate within the Primatte Viewer, you can:

- drag to rotate the polyhedrons,
- **Shift**+drag to zoom in and out of the polyhedrons, and
- **Ctrl/Cmd**+drag to pan the polyhedrons.

At the top of the Primatte Viewer window, there are also three areas that you can click on:

- Clicking and dragging on the blue center area allows you to move the window around on the screen.
- Clicking and dragging on the triangular white region in the upper right corner allows you to scale the Primatte Viewer window.
- Clicking on the square white region in the upper left of the window displays a dropdown menu that looks like this:



Note: A selected feature has a solid yellow square next to it. An unselected feature has a hollow yellow square next to it.

- **Minimize** - This feature, when selected, makes the Primatte Viewer window disappear. Only the blue title bar at the top of the window remains.
- **Large Surface** - This feature, when selected, displays the large Primatte polyhedron in the Viewer window.
- **Middle Surface** - This feature, when selected, displays the medium Primatte polyhedron in the Viewer window.
- **Small Surface** - This feature, when selected, displays the small Primatte polyhedron in the Viewer window.
- **Opaque** - This feature, when selected, makes the selected polyhedrons opaque. De-selecting it makes them semi-transparent.
- **Samples** - This feature, when selected, allows you to sample color regions on the image window using the 3D Sample **operation** mode and see where those regions are in relation to the polyhedron and the algorithm. The colors are displayed as a spray of pixels in the color selected. This button only allows you to see or hide the sampled colors.

Note: The **3DSample** mode must be selected in the **operation** dropdown menu for this feature to work.

- **Clear BG** - This feature changes the background color of the Primatte Viewer window from black (when unselected) to transparent (when selected).
- **Sliced** - This feature, when selected, slices open the large and medium polyhedrons so that the inner polygons can be seen. When unselected, the largest polyhedron selected becomes a completely closed

polyhedron and you might not be able see the inner polyhedrons (unless the **Opaque** feature is deselected).

- **Wireframe** - This feature, when selected, changes the polyhedrons from shaded-surface objects to wireframe objects.
- **Quad view** - This feature, when selected, displays the different polyhedrons in different parts of the Primatte Viewer:
  - The upper left corner of the Viewer shows the medium polyhedron.
  - The upper right corner of the Viewer shows the small polyhedron.
  - The lower left corner of the Viewer shows the large polyhedron as a wireframe object.
  - The lower right corner of the Viewer shows the large polyhedron.

Degrain Section

The **Degrain** tools are used when a foreground image is highly compromised by film grain. As a result of the grain, when backing screen noise is completely removed, the edges of the foreground object often become harsh and jagged leading to a poor key. These tools were created to, hopefully, help when a compositing artist is faced with a grainy image.

Degrain type

The **Degrain type** dropdown menu gives you a range of grain removal from **none** to **large**. If the foreground image has a large amount of film grain induced pixel noise, you may lose a good edge to the foreground object when trying to clean all the grain noise with the **Clean BG Noise operation** mode. These tools allow you to clean up the grain noise without affecting the quality of the key. A short tutorial explaining when and how to use these tools is at the end of this section. Select:

- **none** - When **none** is selected, you get the color of the exact pixel sampled. This is the default mode.
- **small** - When **small** is selected, you get the average color of a small region of the area around the sampled pixel. This should be used when the grain is very dense.
- **medium** - When **medium** is selected, you get the average color of a medium-sized region of the area around the sampled pixel. This should be used when the grain is less dense.
- **large** - When **large** is selected, you get the average color of a larger region of the area around the sampled pixel. This should be used when the grain is very loose.
- **tolerance** slider - Adjusting the **tolerance** slider should increase the effect of the **Clean BG Noise** tool without changing the edge of the foreground object.

Degrain Tools Tutorial

If you have a noisy image as in the example below...



...you will find that the matte is also noisy:



You can use the **Clean BG Noise** operation to remove the noisy pixels, but this can also modify the edge of the foreground object in a negative manner.

Using the **Degrain** tools in the following way may help you clean up the image and still get a good edge on the matte:

1. Use the **Clean BG Noise** operation just a small amount to remove some of the white noise in the alpha channel view but do use it so much that you affect the edge of the foreground object.
2. Then set the **Degrain type** dropdown menu to **small** as a first step to reduce the grain:



With the degrain **tolerance** slider set at **0**, move it around some. This should increase the affect of the **Clean BG Noise** tool without changing the edge of the foreground object.

Sometimes, this may not be enough to totally remove the grain, so by adjusting the degrain **tolerance** slider, you can tell the Primatte algorithm what brightness of pixels you think represents grain. You should try not to use too high of a value; otherwise, it affects the overall matte. For an example of an over-adjusted image, see below.



The Primatte degrain algorithm uses a defocused foreground image to compute the noise.

Note: The **small**, **medium** and **large** settings for the degrain tools all produce defocused foregrounds that have larger or smaller blurs respectively.

Note: It is important to make sure that the **crop** settings are correctly applied; otherwise, when the defocus image is generated, if there is garbage on the edges of the images, then that garbage is blurred into the defocused foreground.

As a review:

1. Select the **Select BG Color operation** mode and click on a backing screen color.
2. Select the **Clean BG Noise operation** mode and use it sparingly so that it has minimum affect to the edge of the foreground object.
3. If there is still grain in the backing screen area, then use the **Degrain type** functionality starting at the **small** setting to reduce the grain.
4. If the grain is still present, then try increasing the **tolerance** slider a little - not too much.
5. If grain is still a problem, then try changing the **type** to **medium** or **large** and also changing the grain tolerance until the desired effect is achieved.

Note: The grain functionality does not always remove grain perfectly, but is sometimes useful to minimize its effects.

Actions Section

In the **operations** dropdown menu, you can select from the following options:

Smart Select BG Color

This operational mode gets the sampled backing screen color, analyzes the original foreground image, and determines the foreground areas using the Primatte foreground detection routine. Then, using the newly determined foreground areas, performs a **Clean FG Noise** operation internally and determines a more desirable shape for the middle and outer polyhedrons, before finally rendering the composite using the generated polyhedrons.

For keying operations, this is the first step and should be followed by the steps described immediately below.

Clean BG Noise

When you select this operational mode, you sample pixels in the Viewer known to be 100% background. White noisy areas in the 100% background region become black. This is usually the second step in using Primatte.

Clean FG Noise

When you select this operational mode, you sample pixels on the image window known to be 100% foreground. The color of the sampled pixels is registered by Primatte to be the same color as in the original foreground image. This makes dark gray areas in the 100% foreground region white. This is usually the third step in using Primatte.

Matte Sponge

When this operational mode is selected, the color sampled in the Viewer becomes 100% foreground. However, if the sampled color is already keyed out and removed, it leaves the current suppressed color. It only affects the key or matte information. This tool is usually used to quickly remove stray transparent pixels that have appeared during the chromakeying procedure. It is a quick and easy way to make final adjustments to a composite.

Make FG Trans

When this mode is selected, the opaque foreground color region sampled in the Viewer becomes slightly translucent. This operation is useful for the subtle tuning of foreground objects which are otherwise 100% covered with smoke or clouds. It can only be used one time on a particular color. For a more flexible way to thin out a color region and be able to make multiple samples, you should use the **Matte (-)** tool. It expands the medium polyhedron slightly.

Restore Detail

With this mode selected, the completely transparent background region sampled in the Viewer becomes translucent. This operation is useful for restoring lost hair details, thin wisps of smoke and the like. It shrinks the small polyhedron slightly.

Spill Sponge

When you select this operational mode, the background color component in the sampled pixels (or spill) within the image window is keyed out and removed for the color region selected. This operation can only be used once on a particular color and the amount of spill suppression applied is not adjustable. It is the fastest way to remove spill from a composite image. For more accurate spill suppression, a **Fine Tuning** or **Spill (+)** operation should follow or be used instead. This can usually be the fourth (and final) step in using Primatte unless additional adjustments are necessary.

Spill (-)

When this operational mode is selected, color spill is removed from the sampled pixel color (and all colors like it) in the amount of one Primatte increment. If spill color remains, another click using this tool removes more of the color spill. Continue using this tool until all color spill has been removed from the sampled color region. This tool expands the Primatte large polyhedron in the color region sampled.

Spill (+)

When this operational mode is selected, color spill is returned to the sampled pixel color (and all colors like it) in the amount of one Primatte increment. This tool can be used to move the sampled color more in the direction of the color in the original foreground image. It can be used to nullify a **Spill (-)** step. This tool dents the Primatte large polyhedron in the color region sampled.

Matte (-)

When this operational mode is selected, the matte is made more translucent for the sampled pixel color (and all colors like it) in the amount of one Primatte increment. If the matte is still too opaque, another click using this operational mode tool makes the sampled color region even more translucent. This can be used to thin out smoke or make a shadow thinner to match shadows in the background imagery. This tool expands the Primatte medium polyhedron in the color region sampled.

Matte (+)

When this operational mode is selected, the matte is made more opaque for the sampled pixel color (and all colors like it) in the amount of one Primatte increment. If the matte is still too translucent or thin, another click using this operational mode tool makes the sampled color region even more opaque. This can be used

to thicken smoke or make a shadow darker to match shadows in the background imagery. It can only make these adjustments to the density of the color region on the original foreground image. It can be used to nullify a **Matte (-)** step. This tool dents the Primatte medium polyhedron in the color region sampled.

Detail (-)

When this operational mode is selected, foreground detail becomes more visible for the sampled pixel color (and all colors like it) in the amount of one Primatte increment. If detail is still missing, another click using this operational mode tool makes detail more visible. This can be used to restore lost smoke or wisps of hair. Sample where the smoke or hair just disappears and it returns to visibility. This is for restoring color regions that were moved into the 100% background region. It may start to bring in background noise if shooting conditions were not ideal on the foreground image. This tool dents the Primatte small polyhedron in the color region sampled.

Detail (+)

When this operational mode is selected, foreground detail becomes less visible for the sampled pixel color (and all colors like it) in the amount of one Primatte increment. If there is still too much detail, another click using this operational mode tool makes more of it disappear. This can be used to remove smoke or wisps of hair from the composite. Sample where detail is visible and it disappears. This is for moving color regions into the 100% background region. It can be used to nullify a **Detail (-)** step. This tool expands the Primatte small polyhedron in the color region sampled.

Fine Tuning Sliders

If the spilled color has not been totally removed at this point, use the **Fine Tuning Sliders** for more subtle removal of spilled background color. For more information, refer to the section on Fine Tuning.

3D Sample

When this operational mode is selected and **viewer** is enabled in the Primatte properties, the sampled colors are displayed as a spray of white pixels in the Primatte Viewer. This allows you to see where the selected backing screen colors reside within the 3D RGB colorspace.

Simple Select BG Color

Although using the **Smart Select BG Color** operation is preferred, the **Simple Select BG Color** operation may still be useful in some cases. This operation uses the older method of taking the sampled backing screen color, projecting a line in the opposite direction on the hue wheel and generating artificial pixels that may represent the FG object. Then, using the artificially generated foreground pixels, it performs a **Clean FG Noise** operation internally and creates the shape of the middle and outer polyhedrons, before finally rendering the composite using the generated polyhedron.

Current Color Chip

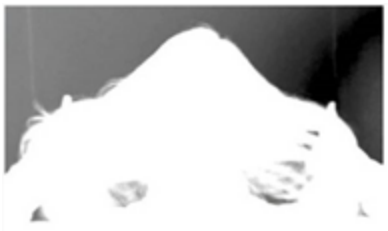
This shows the current color selected (or registered) by the **Fine Tuning** operational mode.

Adjust Lighting

This enables Primatte's light adjusting feature, which, based on the currently selected BG color, generates a clean, evenly lit backing screen to use in the keying operation. This can improve the results if you have a very unevenly lit backing screen. For more information, see Adjust Lighting.



An uneven backing screen.



The resulting matte when **adjust lighting** is disabled.



The resulting key. Areas near the sampled color on the right look good, but on the

left the hair is chunky.



The resulting matte when **adjust lighting** is enabled.



The resulting key is clean on both sides.

Hybrid Render

To tackle this problem, you can enable **hybrid render** to have Primatte create two keys internally and combine them for the best results. For more information, see Hybrid Matte.

Adjust Lighting

When **adjust lighting** is enabled, Primatte first creates a grid on the foreground image and samples the grid to determine which squares contain foreground information and which contain some shade of the background color. Using the remaining shades of blue, it uses extrapolations of the existing backing screen colors, and then uses this technique to fill in the area previously covered by the foreground object. This results in a clean backing screen that has no foreground object (to see this, you can set **output mode** to **adjust lighting BG**). Primatte uses this clean backing screen as the reference data to process the original foreground image to generate light-adjusted foreground that has a much more even shade of blue with the foreground object (this is displayed if you set **output mode** to **adjust lighting FG**). This allows Primatte to perform the keying operation using the light-adjusted foreground, resulting in a clean key around all areas on the foreground object.

The default settings under **Adjust Lighting** should detect all the areas of the grid that contain foreground pixels and deliver a smooth, artificially created, optimized backing screen for the keying. Should it fail to do this, you can adjust the settings of the algorithm by moving the **threshold** slider. This determines if a grid pixel should be treated as a pure background sample, a simulated background sample, or a foreground sample. Increasing the **threshold** value brings more of the foreground into the adjusted lighting.

If necessary, you can also set the **grid size** used in the **adjust lighting** algorithm. Increasing this value increases the grid resolution used in the adjusted lighting calculation.

Hybrid Matte

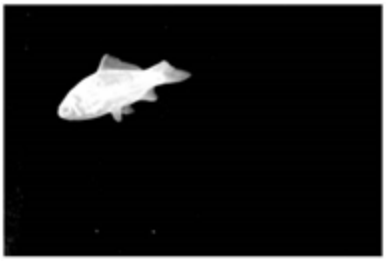
After sampling the backing screen color and finding that the edges of the foreground object look very good, you sometimes find that an area of the foreground object is transparent. This is due to the foreground containing a color that is close to the backing screen color. When this transparency is removed using the **Clean FG Noise** operation, the edge of the foreground object picks up a fringe that is close to the backing screen color. This is very hard to remove without sacrificing quality somewhere else on the image.

When **hybrid render** mode is enabled, Primatte internally creates two keys from the same image:

- **core** - This matte has the transparency removed, but suffers from the bad edges on the foreground object.



- **edge** - This matte has a clean edge on the foreground, but suffers from transparency within the foreground object.



The **core** matte with the bad edges is then blurred and eroded before it is composited over the **edge** matte that has the transparency, resulting in a composite with the best of both options.

The controls under **Hybrid Matte** allow you to adjust the operations performed on the **core** matte:

- **erode** - Adjust the amount of erosion performed on the **core** matte. To view the results, set **output mode** to **hybrid core** and view the alpha channel.
- **blur radius** - Adjust the blur radius used when blurring the **core** matte. To view the results, set **output mode** to **hybrid edge** and view the alpha channel.

Fine Tuning

When the **Fine Tuning Sliders** operational mode is selected, the color of the sampled pixel within the Viewer window is registered as a reference color for fine tuning. It is displayed in the current color chip in the **Actions** section. To perform the tuning operation, sample a color region on the image, select a Fine Tuning slider and move the slider to achieve the desired effect. See the **Fine Tuning Sliders** tool descriptions further on in this section for more details on slider selection.

Spill or L-poly slider (Spill Removal)

When in the **Fine Tuning operation** mode, this **Spill** slider can be used to remove spill from the registered color region. After selecting the **Fine Tuning Actions** mode and registering a color region, this slider can be moved to remove spill from the registered color region. The more to the right the slider moves, the more spill is removed. The more to the left the slider moves, the closer the color component of the selected region is to the color in the original foreground image. If moving the slider all the way to the right does not remove all the spill, re-sample the color region and again move the slider to the right. These slider operations are additive. This result achieved by moving the slider to the right can also be achieved by clicking on the color region using the **Spill(-)** operational mode. This slider bulges the Primatte large polyhedron near the registered color region.

Transparency or M-poly slider (Adjust Transparency)

When in the **Fine Tuning operation** mode, this **Transparency** slider can be used to make the matte more translucent in the registered color region. After selecting the **Fine Tuning operation** mode and selected a color region, moving this slider to the right makes the registered color region more transparent. Moving the slider to the left makes the matte more opaque. If moving the slider all the way to the right does not make the color region translucent enough, re-sample the color region and again move the slider to the right. These slider operations are additive. This result achieved by moving the slider to the right can also be achieved by clicking on the color region using the **Matte(-)** operational mode. This slider bulges the Primatte medium polyhedron near the registered color region.

Detail or S-poly slider (Add/Restore Lost Detail)

When in the **Fine Tuning operation** mode, this **Detail** slider can be used to restore lost detail. After selecting the **Fine Tuning operation** mode and selected a color region, moving this slider to the right

makes the registered color region more visible. Moving the slider to the left makes the color region less visible. If moving the slider all the way to the right does not make the color region visible enough, re-sample the color region and again move the slider to the left. These slider operations are additive. This result achieved by moving the slider to the left can also be achieved by clicking on the color region using the **Detail(-)** operational mode. This shrinks the small polyhedron (which contains all the blue or green background colors) and releases pixels that were close to the background color. The **S-poly** slider in the **Fine Tuning** mode is useful for restoring pixels that were lost because they were so similar to the background color. This slider dents the Primatte small polyhedron near the registered color region.

Spill Process Section

The controls in the **Spill Process** section of the Primatte properties panel let you specify how to replace the spill color.

Replace with

This allows you to select between the three methods of spill replacement:

- **no suppression** - No spill suppression is applied.
- **complement** - Replaces the spill color with the complement of the backing screen color.
- **solid color** - Replaces the spill color with a solid color of your choice.
- **defocused background** - Replaces the spill color with colors from a defocused version of the background image.

Replace color slider

When **solid color** is selected, this sets the color to use to replace the spill.

Defocus slider

When **defocused background** is selected, this area allows you to adjust the amount of defocus applied to the background buffer image.

Output Section

These are the formats for the output of the node (**output mode**):

• **composite** - outputs the composite result of the Primatte node, along with the calculated matte.



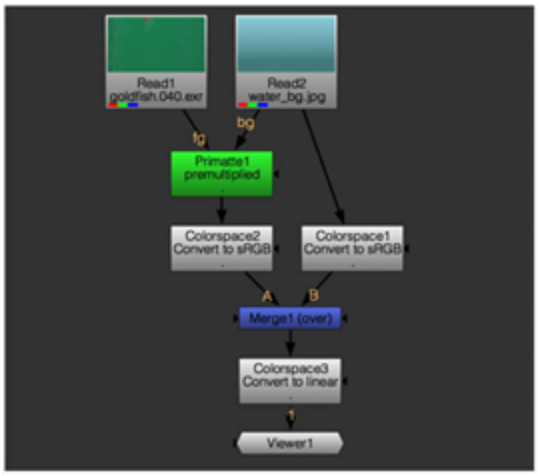


• **premultiplied** - outputs the premultiplied result of the Primatte node, along with the calculated matte.





This can be useful if you want to do your compositing using a Merge node (with **operation** set to **over**) rather than Primatte. This allows you to color correct, transform, and otherwise process your image before compositing it over the background. Note, however, that Primatte works within the sRGB colorspace, whereas Nuke works within a linear colorspace. This means you need to add a Colorspace node after both Primatte and your original background image to convert their colorspaces to sRGB, then do your color corrections or transforms, merge the images together, and, finally, use another Colorspace node to convert the result back to linear.

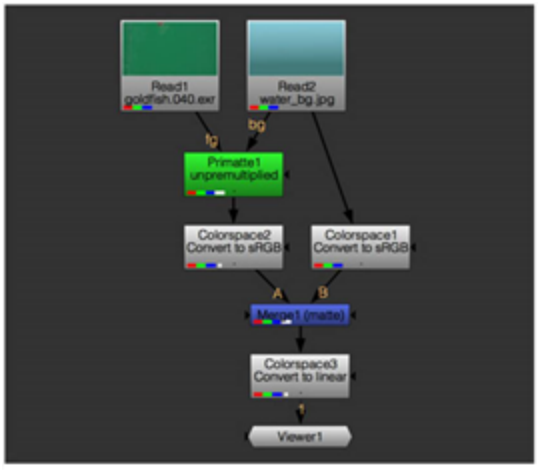


- **unpremultiplied** - outputs the unpremultiplied result of the Primatte node, along with the calculated matte.





This can be useful if you want to do your compositing using a Merge node (with **operation** set to **matte**) rather than Primatte. This allows you to color correct, transform, and otherwise process your image before compositing it over the background. Note, however, that Primatte works within the sRGB colorspace, whereas Nuke works within a linear colorspace. This means you need to add a Colorspace node after both Primatte and your original background image to convert their colorspaces to sRGB, then do your color corrections or transforms, merge the images together, and, finally, use another Colorspace node to convert the result back to linear.



- **not premultiplied** - outputs the unpremultiplied original foreground (rather than the result of the Primatte node), along with the calculated matte.





- **adjust lighting FG** - outputs the light-adjusted foreground that the **adjust lighting** mode creates (this has a more even shade of the backing color with the foreground object). If **adjust lighting** is disabled, this option simply displays the un-optimized, original foreground image. For more information, see Actions Section.



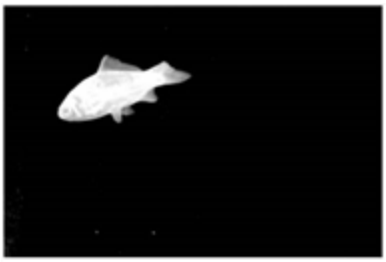
- **adjust lighting BG** - outputs the optimized artificial backing screen that the **adjust lighting** mode creates (a clean backing screen that has no foreground object). For more information, see Actions Section.



- **hybrid core** - outputs the internally generated **core** matte, used when **hybrid render** is enabled. For more information, see Actions Section.



- **hybrid edge** - outputs the internally generated **edge** matte, used when **hybrid render** is enabled. For more information, see Actions Section.



The Primatte Algorithm

There are three Primatte algorithms. Here is a chart that shows the main differences between them.

	Primatte	Primatte RT Plus	Primatte RT
Number of Separating Surfaces	128 (one for each color vector)	6	1
Saturated FG Support	OK	Not Supported	Not Supported
Color Suppression Model	Replacement/ Complement	Replacement	Replacement
Pixel Calculation Cost	Heavy	Light	Very Light

For a description of the Primatte algorithm, see Explanation of How Primatte Works.

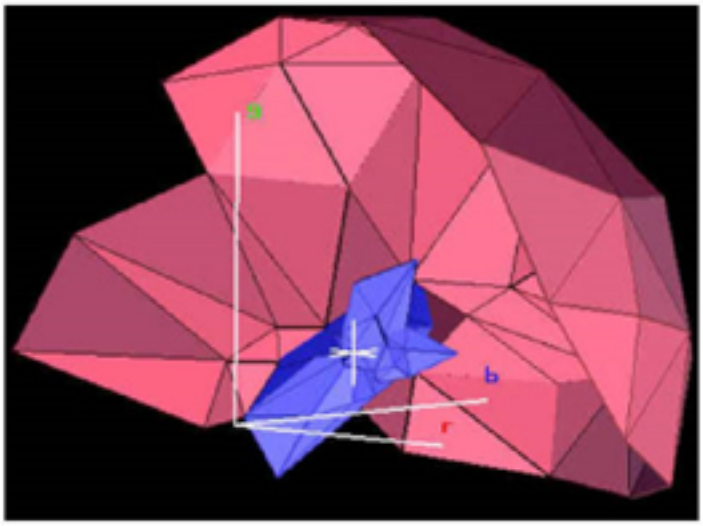
For a description of the Primatte RT+ algorithm, see Explanation of How Primatte RT+ works.

For a description of the Primatte RT algorithm see Explanation of How Primatte RT works.

Explanation of How Primatte Works

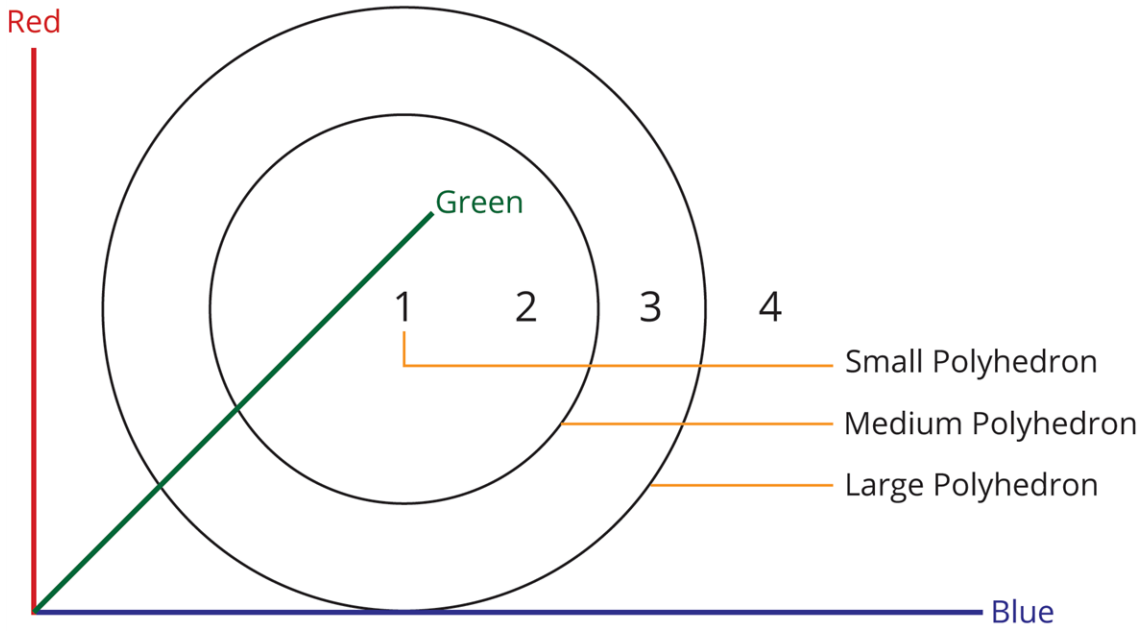
The Primatte chromakey algorithm is a sophisticated method of color space segmentation that can be easily explained to help a user achieve maximum effectiveness with the tool. Basically, Primatte segments all the colors in the foreground image into one of four separate categories. The result is a spill suppressed foreground image and a matte which is used to apply the modified foreground to a suitable background.

Primatte works in 3D RGB color space. Here is a visual representation of the Primatte algorithm after an image has been processed.



By operating the Primatte interface, you essentially create three concentric, multi-faceted polyhedrons. These can be pictured as three globes (or polyhedrons or polys), one within the other, which share a common center point. The creation of these polyhedrons separates all possible foreground colors into one of four regions; inside the small polyhedron **(1)**, between the small and medium polyhedrons **(2)**, between the medium and the large polyhedrons **(3)** and outside the large polyhedron **(4)**.

Primatte Polyhedrons Overview in 3D RGB Colorspace

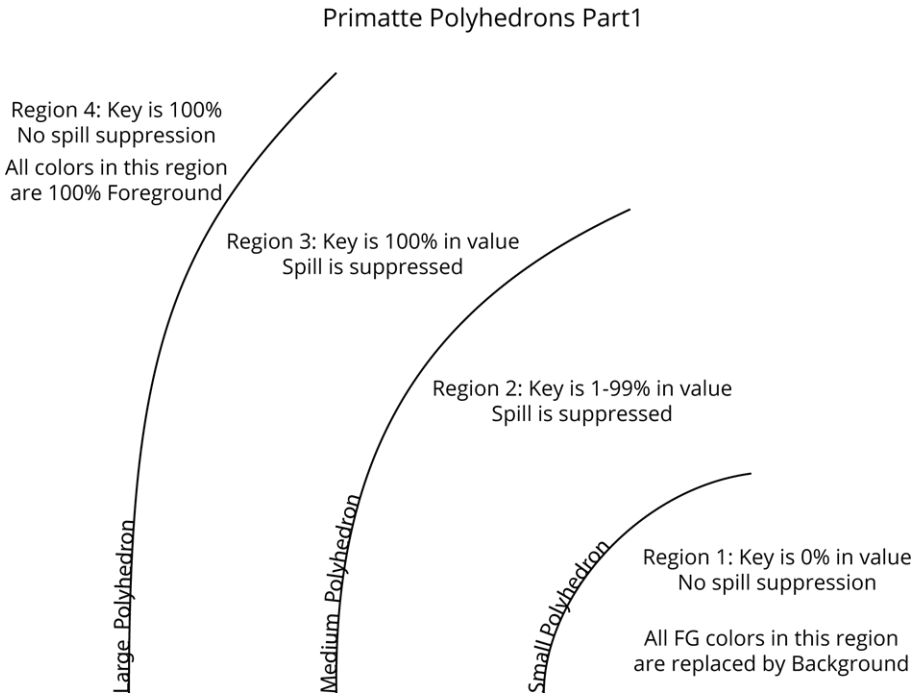


The four regions created are described as follows:

Region 1 (inside the small polyhedron) - This region contains all of the foreground image colors that are considered 100% background. These are the green or blue or whatever colors that were used as the backing color of the foreground image.

Region 2 (between the small and medium polyhedrons) - This region contains all the foreground colors that are at the edges of the foreground object(s), in glass, glass reflections, shadows, sheets of water and other transparent and semi-transparent color regions. These color regions also have spill suppression applied to them to remove color spill from the backing screen.

Region 3 (between the medium and large polyhedrons) - This region contains all the foreground image colors that are 100% foreground but have spill suppression applied to them to remove color spill from the backing screen. Otherwise they are 100% solid foreground colors.



Region 4 (outside the large polyhedron) - This region contains all the 100% foreground image colors that are not modified from the original foreground image. There is no spill suppression applied to these colors.

In the first step in using Primatte (**Smart Select BG Color**), you are asked to indicate the backing color on the original foreground image. The sample should usually be taken from a medium shaded area near the foreground object. By medium shaded area, it is meant that if green is the backing color and the green area of the foreground image has many shades of green ranging from very pale green to almost black, a shade of green in-between these extreme ranges should be chosen. If you're not getting good results using this sample, you should reset Primatte and take another sample using a slightly darker or lighter shade of green. The first sample of Primatte often determines the final result as the center point of all three polyhedrons is created based on this first sample.

A single pixel may be selected or a range of pixels (snail trail or rectangular sample). If a range of pixels is taken, the sample is averaged to get a single color sample. This single pixel or averaged color sample then becomes the center of the small polyhedron. A few other shades around that color are included in the original small polyhedron.

Note: It is recommended that a single pixel be selected as the first sample as you then have some idea where the center point of the polyhedrons is located. If a box sample or a long snail trail sample is made, you can only guess at the average color that ends up being the center point. You can get an idea how this sample affects the algorithm by resetting the Primatte node, viewing the alpha channel, and clicking around on the green or blue screen area while in the **Smart Select BG Color** operation mode. You can immediately see the results of the initial settings of the polyhedrons in this way.

The second step in using Primatte is to clean up the backing color area by adding additional shades of green or blue to the small poly. This second step (**Clean BG Noise**) is usually executed while viewing the black and white alpha channel.



Before background noise removal.



After background noise removal.

While in the **Clean Bg Noise** sampling mode, you sample the white milky regions as shown in the left-hand image above. As you sample these regions, they turn to black as shown in the right-hand image above.

What is happening in the Primatte algorithm is that these new shades of green (the white milky areas) are added to the small poly where all the shades of green or blue are moved. The advantage of this technique is that the polyhedron distorts to enclose only the shades of green that are in the backing screen. Other shades of green around these colors are left undisturbed in the foreground.

Now that you have created a small polyhedron, you need to shape the medium and large polys. A default medium and large poly are both automatically created and are then modified based on the next couple of Primatte operations. The third Primatte step (**Clean FG Noise**) is to sample and eliminate gray areas in the 100% foreground area of the image.



Before foreground noise removal.

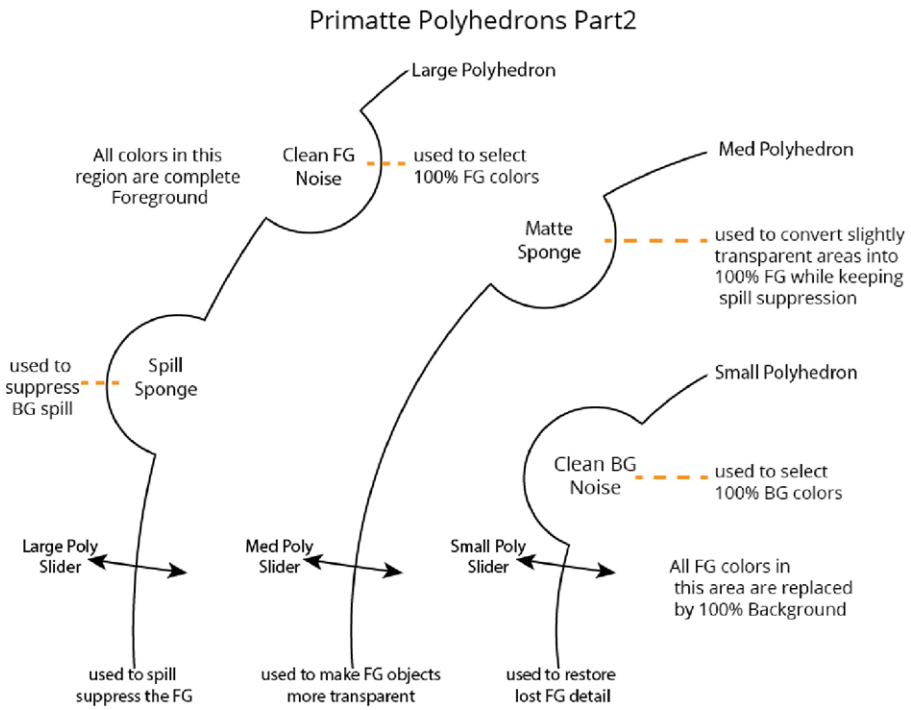


After foreground noise removal.

Again, you make several samples on the dark, grayish areas on the foreground object until it is solid white in color. Primatte is shaping the large polyhedron with each color region that is sampled. Care should be taken in both this and the previous steps to not sample too close to the edges of the foreground object. Getting too close to the foreground object's edges results in hard edges around the foreground object. Primatte uses these samples to modify and shape the medium and large polys to the desired shape. At this point, the matte or key has been created and would allow the foreground objects to be composited into a new background image.

If you now view the RGB channels, there is usually color spill on the edges (and sometimes the center) of the foreground objects. When on the edges of the foreground object, this spill comes from where the edges of the foreground object blended into the backing color. If it is on the center of the foreground object, it usually results from reflected color from the backing screen. The next Primatte step, either **Spill Sponge**, **Fine Tuning** or **Spill(-)**, can now be used to eliminate this spill color.

Let's take a look at what is happening in the Primatte algorithm while this next step is performed. Here is what the various tools in Primatte do to the Polyhedrons when they are used:



As you can see above, the **Spill Sponge** bulges the large polyhedron in the color region specified. A color region is specified by clicking on the image in a particular area with spill present. For example, if you click on some spill on the cheek of a foreground person, Primatte goes to the section of the large polyhedron closest to that particular flesh tone and bulges the polyhedron there. As a result, the flesh tones move from outside the large poly to in-between the medium and large polys. This is **Region 3** and, if you remember, is 100% foreground with spill suppression. As a result of the suppression, the spill is removed from that cheek color and all other shades of that color on the foreground. You would then continue to sample areas of the image where spill exists and each sample would remove spill from another color region.

When all spill has been removed, you should have a final composite. As a last step, you should view the alpha channel and make sure that gray, transparent areas have not appeared in the foreground area. If there are any, you should select the **Matte Sponge** operation mode and sample those gray pixels until they have all turned white again.

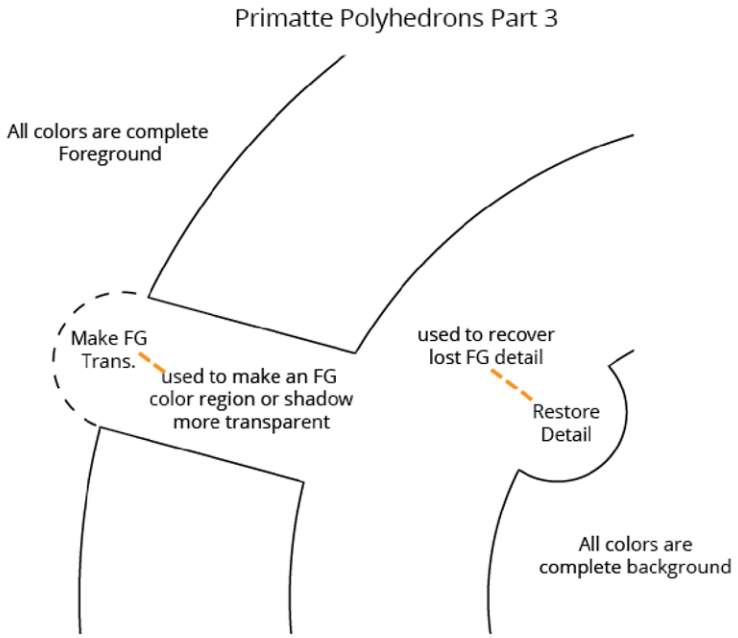
The **Matte Sponge** and **Spill Sponge** tools bulge or dent the polyhedrons a pre-selected amount. If the desired results are not achieved or the results are too extreme for the image, a manual method can be applied. You should select the **Fine Tuning** sliders, select a color region of interest and then move the appropriate slider to get the desired results.

For example, to remove spill, select a region of the composite image with spill on it. Move the spill or large poly slider to the right a little bit, the large poly bulges and the spill should disappear. Move it a little more, if necessary. Moving this slider to the right removes spill (moves the colors from outside the large poly to between the medium and large polyhedrons) and moving it to the left, dents the large poly and moves that color region to outside the large poly.

If you sample a foreground object shadow and then move the **M-poly (trans)** slider to the right, the shadow becomes more transparent. This is useful for matching composited shadows to shadows on the plate photography. It can also be used to make clouds or smoke more transparent.

If some foreground detail disappears during the composite, you can select where the detail should be and then move the **S-poly (detail)** slider to the left. This dents the small poly in that color region and releases the detail pixels from the small poly into the visible region between the small and medium polyhedrons.

The **Spill Sponge** and **Matte Sponge** tools are shortcut tools that automatically move the sliders a pre-selected amount as a timesaving step for you. Other shortcut tools include the **Make FG Trans.** tool and the **Restore Detail** tool.



These shortcut tools are one-step operations where you click on a color region of interest and Primatte performs a pre-calculated operation. Hopefully, most operations using Primatte would only require these tools, but the manual operation of the sliders is always an option.

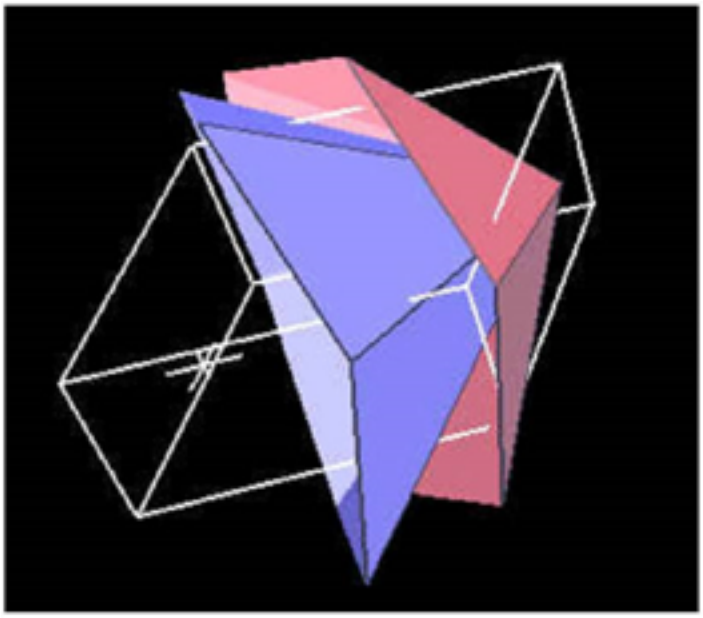
The **Spill(-)** tool bulges the large poly a small amount incrementally in the color region that is clicked on and the **Spill(+)** tool dents it a small amount with each click. The **Matte(-)** and **Matte(+)** tools do the same to the medium poly and the **Detail(-)** and **Detail(+)** do it to the small poly.

Explanation of How Primatte RT+ works

The **Primatte RT+** algorithm differs from the Primatte algorithm in that it has a six surface color separator instead of the 128-faceted polyhedrons. This makes the **Primatte RT+** algorithm much simpler and, therefore, faster to calculate. The results and performance of **Primatte RT+** falls in between the **Primatte**

and **Primatte RT** options. Where the **Primatte RT+** algorithm might not work well is with less saturated backing screen colors and it also does not support the **complement** color spill suppression method (which is the spill suppression method that delivers the best detail). For a well-lit and photographed image or clip, this algorithm produces good results and render quickly.

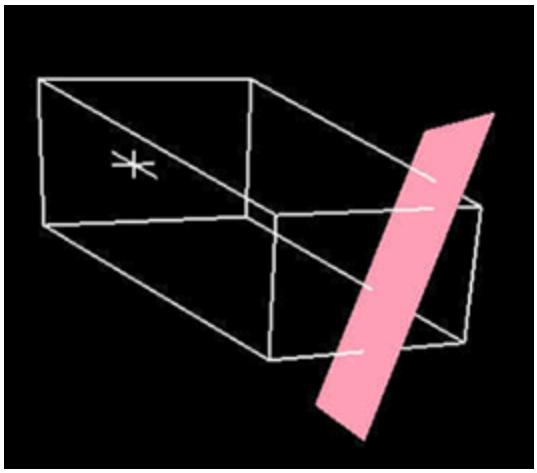
Here is what a visual representation of the Primatte RT algorithm looks like after an image has been processed:



Explanation of How Primatte RT works

Primatte RT is the simplest algorithm and, therefore, the fastest. It uses only a single planar surface to separate the 3D RGB colorspace and, as a result, does not have the ability to separate out the foreground from the backing screen as carefully as the above Primatte algorithm. Like the **Primatte RT+** algorithm, **Primatte RT** might not work well with less saturated backing screen colors and it too does not support the **complement** color spill suppression method (which is the spill suppression method that delivers the best detail). For a well-lit and photographed image or clip, this algorithm produces good results and render very quickly.

Here is what a visual representation of the **Primatte RT** algorithm looks like after an image has been processed:



Contact Details

Here are the contact details for the Photron main office and Primatte office.

Main Office

Photron USA, Inc., 9520 Padgett Street, Suite 110, San Diego, CA 92126, USA

Primatte Office

Photron USA, Inc., 3113 Woodleigh Lane, Cameron Park, CA 95682. Phone: 1-530-677-9980, FAX: 1-530-677-9981, Cell: 1-530-613-3212, E-mail: sgross@photron.com, Website: http://primatte.com

Proprietary Notices

Primatte is distributed and licensed by Photron USA, Inc., San Diego, CA, USA. Primatte was developed by IMAGICA Corp., Tokyo, Japan. Primatte is a trademark of IMAGICA Co IMAGICA Digix Inc., Tokyo, Japan.

Keying with Ultimatte

This section explains how to use the blue/green screen keyer, Ultimatte, in Nuke.

Ultimatte Quick Start

Here's a quick overview of the workflow:

1. Sample the screen (backing) color. For more information, see Sampling the Screen Color.
2. Refine the overlay using the **overlay** tools and, if needed, adjust the controls on the **Ultimatte** tab. For more information, see Using Overlay Tools and Screen Correct.
3. Next refine the matte density using the **matte** tools and, if needed, adjust the controls on the **Density** tab. For more information, see Adjusting the Density of the Matte .
4. If necessary, activate Shadow processing, and use Shadow tool and, if needed, adjust the controls on the **Shadow** tab. For more information, see Retaining Shadows and Removing Noise.
5. If necessary, improve Spill Suppression using **spill** tools and, if needed, adjust the controls on the **Spill** tab. For more information, see Adjusting Spill Controls.
6. If necessary, adjust the **Cleanup** controls, but in general you'll get better results by using **ScreenCorrect** and **MatteDensity** controls. For more information, see Retaining Shadows and Removing Noise.
7. If necessary, you can adjust the controls on the Color tab to match your blacks, whites and gammas between the foreground and the background. For more information, see Adjusting Color Controls.
8. If necessary, activate film processing and adjust its settings on the Film tab. For more information, see Adjusting Film Controls.

Connecting the Ultimatte Node

Start up Nuke, create an Ultimatte node by clicking **Keyer** > **Ultimatte** and connect a foreground and a background image (and any additional inputs you want) to it. Add a Nuke Viewer node so you can see the result.

To use the Ultimatte inputs

In Ultimatte, you have a number of inputs you can use to connect images and mattes that you need to key your footage. Connect the following inputs to your images and/or mattes as necessary:

- **foreground (fg)** - Connect your foreground image to this input.
- **background (bg)** - Connect your background image to this input.
- **clean plate (cp)** - Connect your clean plate to this input. For more information on clean plates, see Using Overlay Tools and Screen Correct.
- **garbage matte (gm)** - Connect your garbage matte to this input. A garbage matte is often used to clean up screens that are not a constant color or have lighting rigs in shot by forcing the alpha transparent.
- **holdout matte (hm)** - Connect your holdout matte to this input. A holdout matte is used when foreground objects are the same color or very close to the backing color. The holdout matte is used to let Ultimatte know that the pixels in the holdout matte region should be considered 100% opaque. Note that, unlike the other inputs, this input is hidden and appears as a small triangle on the left hand side of the node.

Sampling the Screen Color

The first step when keying with Ultimatte is to sample the screen color, or tell Ultimatte what color your blue or green screen is. Do the following:

1. When you've connected the Ultimatte node, the foreground image is displayed in the Viewer.
2. Click **Screen** in the Ultimatte toolbar above the Viewer, and select the screen color by holding down **Alt**+**Ctrl**/**Cmd** and clicking in the Viewer.

   You should select an area on the green screen near important subject detail that is not obscured in any way. The image is rendered and a composite is displayed. If further adjustments are needed, use the controls described below.

Adjusting the Controls on Ultimatte Tab

After you've sampled your screen color, you can start adjusting your result. With the controls on the **Ultimatte** tab you can select different sets of controls you want to enable on other properties panel tabs. There's also an **enable** checkbox on each tab that you can use to activate the controls on that tab and the corresponding tools in the Viewer.

- **Film** - check this to reduce the effects of cyan undercutting on the **Film** tab.
- **screen correct** - check this to use the controls on **Screen Correct** tab.

- **shadow** - check this to use the controls on the **Shadow** tab.
- **spill suppression** - check this to use the controls on the **Spill** tab.
- **cleanup** - check this to use the controls on the **Cleanup** tab.
- **color conformance** - check this to use the controls on the **Color** tab.

Using Overlay Tools and Screen Correct

Screen Correct compensates for anomalies in the backing area such as uneven lighting, smudges, seams, variation in the backing color, green set pieces, and unwanted shadows cast by set pieces, boom arms, etc. This technique assumes that an exact copy of the problematic green screen element with the subject matter omitted, usually called a Clean Plate, is supplied.

Although this technique gives great results by differentiating between foreground elements and backing flaws, you often haven't got the necessary reference materials. In that case, you can create a synthetic Clean Plate with Ultimatte.

To achieve the best results, use a reference Clean Plate as an input to Ultimatte in addition to allowing Ultimatte to generate a synthetic plate. In this way, the reference plate allows for the best shadow separation, while the synthetic plate is used to reduce artifacts such as film grain or video noise (which is rarely accurately reproduced when the clean plate is shot during the time of principal photography). Switch the view to **screen** by clicking the **overlay** dropdown menu and selecting **screen**.

With **screen correct** selected, use the **add overlay** tool to scrub on areas that represent the green screen, including shadows. Usually, it's best to press **Alt**+**Ctrl**/**Cmd** while scrubbing, so that you are picking the color in the input image at that pixel, instead of the color of the processed image.

The overlay is used in the final process to fully eliminate these areas from the final composite. Continue in this manner until the foreground subject and its shadows are surrounded with an overlay. Make sure the overlay does not cover any foreground detail that is to be retained. If the overlay does cover foreground subject matter, then use the **remove overlay** tool to scrub on those areas that should not be covered by the overlay. Repeat these two processes until the overlay covers the green screen areas and no subject matter. View the composite in the Viewer to see the screen-corrected composite with anomalies in the backing area removed. To learn which controls were automatically set by Ultimatte, click the **Screen Correct** tab and note the position of the controls.

Note: When scrubbing on the image using **add overlay**, the RGB value of the points selected are accumulated in a keep list and **removeoverlay** points are accumulated in a drop list. If both lists have equivalent values, a conflict may occur, resulting in no visible change in the overlay. If a conflict occurs, try using Undo (**Ctrl/Cmd**+**X**) which flushes the last set of points added to the appropriate list. If that doesn't help, you can also press the **Reset** button on the **ScreenCorrect** tab to clear the lists and start over.

Under some circumstances, it may be difficult to completely cover the screen area with the overlay. This does not mean that the screen cannot be removed in that area, but that other controls may need to be adjusted to help reduce anomalies in those areas that were not covered by the overlay. It may be advantageous to resample the screen using the **Screen** tool above the Viewer in the area where covering the screen with the overlay is unsuccessful, but be careful about selecting in shadow areas if the final composite requires the retention of shadows.

Additionally, it may be difficult in some circumstances to remove the overlay from very dark areas of foreground detail that may be close to values in dark or shadow areas of the screen. If the overlay does include some dark foreground areas, these areas may be recovered by enabling the **Shadows** controls.

Note: Since Ultimatte produces a synthetic clean plate using an interpolation process, it is important to exclude non-video areas of the image, such as the black blanking lines that may appear in video or the black areas in letterbox images. You can use other nodes, such as Crop, to exclude these areas from the overlay and final composites. Otherwise the synthetic clean plate may contain black (or other colors) which can result in an inaccurate clean plate being generated.

Adjusting overlay controls

You can choose your overlay view, color and output mode in the properties panel, on the **Ultimatte** tab.

- overlay - The overlay control is used to show the calculated overlay on the Viewer and it's helpful for debugging purposes. It helps tune the screen correct controls and also the add/remove overlay tools. In this way you can see immediately if something is wrong and needs to be adjusted. In the overlay dropdown menu, change
  - **off** - to not view the overlay.
  - **screen** - to view the subject in clear, and the preliminary matte area blended with the overlay color.
  - **subject** - show the subject blended with overlay color, and the preliminary matte area in clear.
- **show image as monochrome** - check this to make the input image appear in grayscale so that the overlay areas can be more easily distinguished.

- **overlay color** - use this to change the color of the overlay. You can adjust the alpha channel to modify the opacity of the overlay.

Adjusting the screen correct controls

- **screen tolerance** - This is used to adjust the color range or tolerance to be included or excluded from the screen overlay.
- **shrink** - Use this control to shrink the screen overlay.
- **darks**(**reds smaller**) - Use this control to include or exclude dark areas from the screen overlay in those areas where the blue value (when using green screen) is greater than the red value in the original foreground image.
- **darks**(**reds larger**) - Use this control to include or exclude dark areas from the screen overlay in those areas where the red value is greater than the blue value (when using green screen) in the original foreground image.
- **brights**(**reds smaller**) - Use this control to include or exclude bright areas from the screen overlay in those areas where the blue value (when using green screen) is greater than the red value in the original foreground image.
- **brights**(**reds larger**) - Use this control to include or exclude bright areas from the screen overlay in those areas where the red value is greater than the blue value (when using green screen) in the original foreground image.
- **orphans** - Use this to include or exclude neighboring "rogue" pixels from the screen overlay.

Adjusting the Density of the Matte

The controls on the **Density** tab are used to adjust the density or opacity of the foreground objects. The density of a foreground object is determined by its matte (alpha) value. A completely opaque object's matte is white, a completely transparent object's matte is black, and a partially transparent object's matte is gray. Use the **add matte** dropper to scrub on areas in the matte that appear gray, but should be white (fully opaque) in the matte. These areas are described as "print-through", meaning that the opacity of the subject is too low in this area and the background is visible through the foreground in this area. Be careful not to select those objects that should have a gray matte value such as fine hair, smoke or partially transparent objects, as they become opaque. To learn which controls were automatically set by Ultimatte, click the **Density** tab in the properties panel and note the position of the controls.



Note: If there are sections of the matte which should be opaque but are exhibiting gray values that don't look like typical transparency, then there is a chance that there is a remainder of overlay in this area. If overlay exists on subject matter, return the **Overlay** dropdown menu to **Screen** and use the **removeoverlay** tool to scrub on that area. Check on **show image as monochrome** on the **Ultimatte** tab to aid in determining the extent of the overlay.

Adjusting Density controls

Use these controls to adjust the density of your matte:

- **brights** - Use this control to adjust density in bright foreground objects. Advancing this control too far can cause hard, dark edges around foreground subjects.
- **darks** - Use this control to adjust density in black glossy or dark foreground objects.
- **edge kernel** - Use this control to adjust number of pixels to use as a kernel to reduce dark edges that may exist in transition areas due to an over-dense matte. Advancing this control too far, may cause too much print-through at the edges.
- **warm** - Use this control to adjust density in warm colors (flesh tones). Note that reducing this control too much can cause print-through in reddish foreground objects.
- **cool** - Use this control to adjust density in cool colors. Note that reducing this control too much can cause print-through in bluish foreground objects.

Adjusting Spill Controls

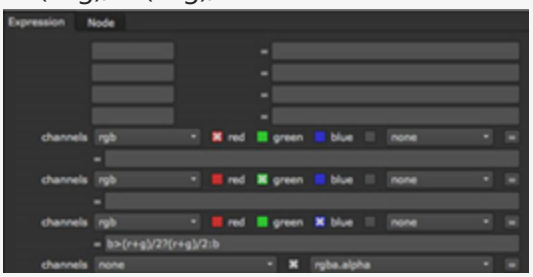
Ultimatte automatically suppresses spill from the backing onto foreground subject matter. The spill controls are used to suppress excessive spill or to retain color similar to spill that has been over-suppressed. To adjust the spill controls, check the **spill suppression** box on the **Ultimatte** tab and click **Spill** tab:

- **cool** - Use this control to adjust the amount of spill in cool colored foreground objects. Used to reproduce blue, green or cyan colors that changed through the spill suppression algorithms.
- **warm** - Use this control to adjust the amount of spill in warm colored foreground objects. Used to reproduce pink, purple and magenta colors for bluescreen, or yellow and orange colors for green screen that changed through the spill suppression algorithms.
- **midtones** - Use this control to adjust the amount of spill in mid-range foreground objects.
- **brights** - Use this control to adjust the amount of spill on bright foreground objects.
- **darks** - Use this control to adjust the amount of spill on dark foreground objects.

- **ambience** - Use this control to select a color to subtly influence the foreground objects in areas that may have contained spill.
- **strength** - Use this control to adjust the intensity of the selected **ambience** color.
- **background veiling** - This control is used to override the automatic suppression of the backing color. Ultimatte uses the selected backing color to suppress the backing to black. An indication that the automatic settings did not suppress enough backing is "veiling" or a colorized haze in some background areas. An indication that the automatic settings suppressed too much backing is darkened or mis-colored foreground edges and transparencies. In most cases this control should be left at the default setting.

Tip: There are several nodes in Nuke you can use for spill removal. For example, if you are using a greenscreen image, you can add an Expression node after your foreground image and set the expression field for the green channel to:
g>(r+b)/2?(r+b)/2:g

Similarly, you can despill a bluescreen image by setting the expression field for the blue channel to:
b>(r+g)/2?(r+g)/2:b



You can also use the HueCorrect node for despill. For more information, see Correcting Hue Only.

Retaining Shadows and Removing Noise

Use the **hold shadow** dropper (only available when **screen correct** and **shadow** are enabled) to scrub on the shadows that you'd like to preserve. These shadows may best be seen in the foreground image.

If unwanted shadows remain, then use the **remove noise** dropper to reduce or eliminate those shadows. If the area that you scrub does not reside under the overlay, then erasing adjusts the appropriate **Cleanup** controls to reduce anomalies in the screen area. This might result in losing some fine detail. Repeat using

these two tools until the shadows you want are retained and any shadows or noise you don't want to keep are removed.

The density, sharpness, and tint of the shadow may be altered by manually adjusting the controls in **Shadow** controls. To learn which controls were automatically set by Ultimatte, click the **Shadow** and/or **Cleanup** tab in the properties panel and note the position of the controls.

Adjusting Shadows controls

Check the **shadow** box on the **Ultimatte** tab and adjust the following controls:

- **high** - Decrease this to reduce or eliminate unwanted shadows (which must be lighter than those shadows that are to be retained). All preserved shadows are lighter.
- **low** - Increase this to restore the density of the darkest part of the shadows that are to be retained.
- **density** - Use this to change the density of the shadows that are retained to better match shadows in the background scene.
- **blur** - Use this to blur the shadows.
- **tint** - Use this to tint the shadows.

Adjusting Cleanup controls

The following controls are used to adjust the black and gray areas of the matte channel. This dramatically affects the nature of foreground objects' edges, the opacity of transparent objects, and the noise in the foreground image. Use these controls sparingly as they might result in the loss of foreground detail. Using **Screen Correction** for dealing with imperfections in the screen is a good alternative for using **Cleanup**.

- **cleanup** - Use this control to reduce imperfections or small amounts of noise in the backing. Note that adjusting this control too far results in a "cut and paste" look with a hard, unnatural edge. Background noise (as well as some foreground detail) is reduced. An alternative method for dealing with green screen imperfections is to use the **Screen correct** controls.
- **shrink** - Use this control to choke or reduce the size of the cleaned-up matte.
- **blur** - Use this control to soften the cleaned-up matte.
- **recover** - Use this control to set a threshold below which the **Cleanup** control does not have influence.

Adjusting Color Controls

Check the **color conformance** box on the **Ultimatte** tab, and set these controls on the **Color** tab. A good keying result requires matched blacks, whites, and gammas between the foreground and background elements. With color conformance you can select blacks, whites, and gammas and can automatically match the foreground to the background (or vice versa).

- **darks** - You can use this control to influence the darkest parts of the image. This is a global control that affects the entire image, but the greatest effect is seen in the darkest areas.
- **midtones** - You can use this control to influence the mid-range parts of the image. This is a global control that affects the entire image, but the greatest effect is seen in the mid-range areas.
- **brights** - You can use this control to influence the brightest parts of the image. This is a global control that affects the entire image, but you can see the greatest effect in the brightest areas.
- **hue** - This control changes the color contents of the image without changing its brightness or color intensity (purity) values. At default setting (0), the image hue is not altered. The range of the control extends from -300 to +300.
- **saturation** - This control changes the color intensity or purity values of the image without altering its color contents or brightness values. At default setting (0), the image saturation is not altered. At minimum setting (-200), the color intensity is reduced to zero and the image is monochrome, or shades of gray.
- **brightness** - This control changes the overall intensity of the image. At the default setting, the image brightness is not altered.

Adjusting Film Controls

Use the controls on the **Film** tab to reduce the effects of cyan undercutting. These controls are only available when **Film** box is checked on **Ultimatte** tab. You can view the results better when you select **subject** in the **overlay** dropdown menu on the **Ultimatte** tab.

Due to the nature of film's emulsion layers, a phenomenon know as cyan undercutting exists that reveals itself as a smeared red edge in areas of sharp transitions, which can best be seen by viewing the individual RGB channels of a film image. Normally, this phenomenon is not a problem until bluescreen compositing techniques are applied. Since the red emulsion layer tracks at a slower rate than the blue and green emulsion layers, the artificial red edge it produces are retained as foreground detail resulting in an unacceptable composite.

- **transparency** - Use this control to adjust the amount of Film Correction in partially transparent foreground objects (such as hair detail).

- **correction** - Use this control to adjust the amount of correction in the individual RGB channels of the foreground image.
- **strength** - Use this control to adjust the overall amount of Film Correction that is applied to the foreground image.
- **shrink** - Use this control to shrink the subject overlay.
- **brights** - Use this control to include or eliminate bright areas from the subject overlay.
- **darks** - Use this control to include or eliminate dark areas from the subject overlay.

Choosing an Output Mode

You can output the merged foreground and background as a final composite, or you can output the premultiplied or unpremultiplied foreground for compositing elsewhere in your node tree. In the **output mode** dropdown menu, select:

- **composite** - to output a merged layer of the background input with the extracted subject.
- **premultiplied** - to output the extracted subject premultiplied by the final matte.
- **unpremultiplied** - to output the extracted subject and final matte unpremultiplied.

Using RotoPaint

Nuke features a vector-based RotoPaint node for help with tasks like rotoscoping, rig removal, garbage matting, and dustbusting. You can draw Bezier and B-Spline shapes with individual and layer group attributes, including per-point and global feather, motion blur, blending modes and individual or hierarchical 2D transformations. This chapter gives full instructions on its usage.

Roto or RotoPaint?

There are two similar nodes in Nuke for rotoscoping, Roto and RotoPaint. The main difference between these two is that you can only create and edit Bezier and B-spline shapes with Roto, while RotoPaint allows you to draw paint strokes too with various brushes. So the Roto node is an optimal choice if you're doing rotoscoping only, whereas RotoPaint gives you a broader scale of tools to use.

All tools and controls in the Roto node work the same way as they do in RotoPaint, so you can learn about using them in the RotoPaint instructions in this chapter. For instance, see:

- Working with the Toolbars for information about the toolbars in Roto.
- Working with the Stroke/Shape List for information about the shape list in the Roto properties panel.
- Drawing Shapes for information about using the Bezier and B-Spline Tools.
- Selecting Existing Strokes/Shapes for Editing for information about selecting shapes.
- Editing Shape-Specific Attributes for information about editing Bezier and B-Spline attributes.
- Editing Existing Stroke/Shape Timing for information about changing the timing of your shape.
- Animating Strokes/Shapes for information about editing your shapes.

RotoPaint Quick Start

Here's a quick overview of the workflow:

1. Connect the RotoPaint node to a Viewer and possible backgrounds. For more information, see Connecting the RotoPaint Node.
2. Select a stroke or a shape tool from the RotoPaint toolbar on the left side of the Viewer. For more information, see Drawing Paint Strokes or Drawing Shapes.
3. Use the RotoPaint tool settings on the top of the Viewer to adjust the stroke/shape you're about to draw. For more information, see Working with the Toolbars.

4. Draw one or more strokes/shapes in the Viewer window. For more information, see for example Using the Brush tool or Using the Bezier and Cusped Bezier Tools.
5. Select an existing stroke/shape using the Select tools or the stroke/shape list. For more information, see Selecting Existing Strokes/Shapes for Editing.
6. Use the control panel controls to adjust your existing stroke/shape. For more information, see Editing Existing Stroke/Shape Attributes.

In addition you can:

- Adjust the splines of your stroke/shape. For more information, see Editing Existing Stroke/Shape Splines.
- Animate your strokes/shapes. For more information, see Animating Strokes/Shapes.
- Use RotoPaint in stereoscopic projects. For more information, see RotoPaint and Stereoscopic Projects.
- Set your favorite RotoPaint tool as your default tool. For more information, see Setting Default RotoPaint Tools and Settings.

Connecting the RotoPaint Node

The RotoPaint node accepts one primary input. Even if you have no inputs, you can still use RotoPaint and in that case, you can use the **format** control to select your output format (by default, the **format** control is hidden, but you can display it by clicking on the black triangle above **color**).

To Connect the RotoPaint Node

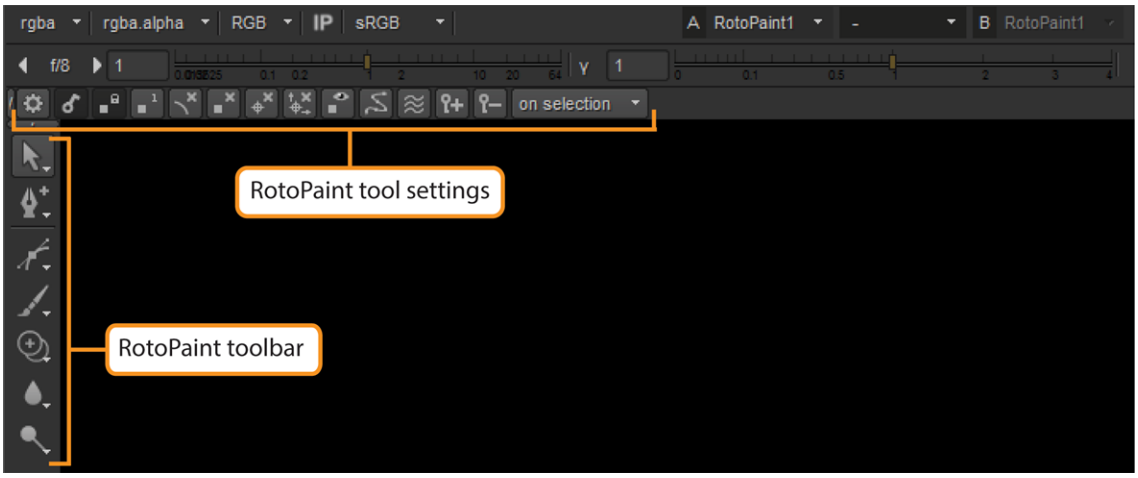
1. Click **Draw** > **RotoPaint** to add a new RotoPaint node. You can also press **P** on the Node Graph. To create a Roto node, you can press **O** on the Node Graph.
2. Drag the **bg** input to the node that you want to apply RotoPaint to.
3. Connect any additional background elements you wish to use. If you plan to reveal pixels from a background element, drag the **bg1** input from the left side of the node to the node whose output you wish to use.

Working with the Toolbars

In the RotoPaint node, you can use two toolbars to define the type of stroke/shape you want to start drawing. These toolbars are placed in the Viewer. The vertical RotoPaint toolbar is for selecting the tool you

want to use and the horizontal one, RotoPaint tool settings, is for adjusting the currently selected tool's settings before drawing new strokes/shapes.

Note: You can't use RotoPaint tool settings to adjust an already existing stroke/shape. For any changes you want to make to a stroke/shape after you've created one, you can use the controls in the RotoPaint properties panel.



RotoPaint toolbar and tool settings.

In the RotoPaint toolbar, you can select your stroke/shape tool. The tools are grouped under the toolbar icons. You can click any tool to make it active and view a tool group by right-clicking (or left-clicking and holding) the icon. The tool that is currently selected is highlighted.

Tip: When the Viewer has mouse-over focus, you can also use the **S** keyboard shortcut to cycle through the modes of the currently selected tool.

In the RotoPaint tool settings on the top of the Viewer, you can define the settings for the tool that you've selected. The controls in this toolbar change depending on which tool you have selected at any given time.

Tip: You can hide the toolbars by clicking the hide button next to them. You can also press **[** (square bracket) to hide the RotoPaint toolbar and **{** (curly bracket) to hide the tool settings (and the entire top toolbar of the Viewer).

Working with the Stroke/Shape List

After you've drawn strokes/shapes, you can edit their order and group them with the Stroke/Shape list in the RotoPaint properties panel. By default, the newest stroke/shape/group appears on top of the list, and your strokes/shapes are named according to their type (for example, "Bezier1" or "Smear2").

Using the list, you can also select strokes/shapes/groups and edit them with the various controls in the properties panel. Some controls can be applied to groups, some can't. Some controls also can only be applied to strokes or shapes (these are grouped under the **Stroke** and **Shape** control tabs respectively). If a control can't be applied to a group, it is grayed out if you have a group selected in the stroke/shape list.

The stroke/shape list provides an overview of existing parameters, showing, for example, whether the item is locked, set invisible, or whether it has motion blur applied to it. Some controls can also be edited directly in the overview by clicking on their icon.

To Use the Stroke/Shape List

You can edit the stroke/shape list in many ways, and use it to adjust strokes/shapes and how they're displayed in the Viewer.

- You can reorder the columns in the stroke/shape list by dragging and dropping them.
- You can create groups for sets of strokes/shapes in the stroke/shape list by clicking the **Add** button  below the list. This creates a subfolder, named "Layer1" by default, and you can drag and drop strokes/shapes to this folder to group them. After strokes/shapes have been grouped, you can edit them as a group and they move together if you change their place in the Viewer. Every group also has its own transformation overlay, which you can use to move the group.
- You can remove a stroke/shape or a group by clicking the **Remove** button  under the stroke/shape list.
- If you want to rename any of the strokes, shapes, or groups, double-click on the name while the item is selected, and give your item a new name. The name has to be unique, so you can't give two items the same name.
- You can also cut, copy, and paste strokes and shapes by right-clicking on them in the control panel and using the **copy** > **curve**, **cut** > **curve**, and **paste** > **spline** options in the menu that appears.

  For information on copying, cutting and pasting shapes and points, see Copying, Pasting, and Cutting Stroke Positions; or for animated shapes and points, see Animating Strokes/Shapes. Alternatively, to copy, cut, or paste shape and point attributes only, see Editing Existing Stroke/Shape Splines.

Note: You cannot use **Ctrl/Cmd+C** or **Ctrl/Cmd+V** in the control panel to copy and paste shapes. You must use the right-click menu.

- You can duplicate strokes/shapes by right-clicking on them and selecting **duplicate**. A new stroke/shape is created with the same spline, animation, and attributes as the one you selected.
- You can hide a stroke, shape, or group by clicking the **Visible** icon  in the stroke/shape list. You can still edit an invisible stroke/shape and view its position in the Viewer.
- You can lock strokes/shapes to prevent them from being edited. To lock an item in the stroke/shape list, click the **Lock** column in the list. A lock icon  appears next to the **Visible** icon.
- You can select the color in which you want the outline of your stroke/shape to appear in the Viewer. Click the **Overlay** column  and select your overlay color. To be able to see the splines for all paint strokes in the Viewer, you need to activate one of the Select tools in the RotoPaint toolbar and click  in the tool settings.
- You can change the color of your stroke/shape in the stroke/shape list by clicking the **Color** column  and using the color picker to select the color.
- You can invert a shape using the **Invert** column . With your shape selected, click in the **Invert** column to toggle between inverted and uninverted modes.
- You can select a blending mode for your stroke/shape using the **Blending** column . With your shape selected, click the Blending column and select the mode you want to apply.
- You can apply shape motion blur using the **Motion blur** column . With your shape selected, click the **Motion blur** column to toggle the motion blur effect.

  See Adding Motion Blur for more information on **shape** and **global** motion blur.
- You can specify the frame range of certain strokes and shapes by navigating to the stroke/shape list of the RotoPaint node, and right-clicking **all** under the **Life** column. Select **frame range** from the pop-up menu, and specify the frame range desired in the **Set frame range** dialog.

  By default, new shapes and strokes are set to all on creation.

Tip: To undo and redo any changes you've made with the RotoPaint node, use the **Undo** and **Redo** buttons on top of the properties panel. Undo uses a cache memory of your changes, so at any time you can undo and redo all the changes you've made since you last opened your project.

Drawing Paint Strokes

Any given RotoPaint node can hold many paint strokes and shapes. You can apply paint strokes using any of the following tools (see also Drawing Shapes).

Icon	Tool	Keyboard Shortcut	Function
	Brush	N (toggles between Brush and Eraser)	Applies colors atop the current plate, or blends colors with the current plate. You can also clone from surrounding frames.
	Eraser	N	Removes pixels from existing strokes and brings background back.
	Clone	C (toggles between clone and Reveal)	Applies pixels from one region of the current plate to another region of the current plate.
	Reveal	C	Applies pixels from a source plate to a destination plate in the corresponding place.
	Blur	X (toggles between Blur, Sharpen and Smear)	Blurs the image in the area of the brush stroke.
	Sharpen	X	Sharpens the image in the area of the brush stroke.
	Smear	X	Smears the area of the smear brush stroke, stretching the selected pixels over their surrounding area.
	Dodge	D (toggles between Dodge and Burn)	Brightens the background color on the area of the brush stroke to reflect the brush stroke. Using this tool on black produces no change. No part of the stroke area is darkened.
	Burn	D	Darkens the background color on the area of the brush stroke to reflect the brush stroke. No part of the stroke area is lightened.

Tip: You can choose to have your RotoPaint always open with a particular tool selected. If you want to open it with the Brush tool selected, for example, do the following:
1. Create a file called **menu.py** in your plug-in path directory if one doesn't already exist. For more information on plug-in path directories, see Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts.
2. To select the Brush tool as your default tool, save the following in your **menu.py**:
nuke.knobDefault('RotoPaint.toolbox','brush')
3. Restart Nuke.

For more information on default tools, see Setting Default RotoPaint Tools and Settings.

Tip: If you are using a tablet, you can tie a new stroke's opacity, size, or hardness to pen pressure by clicking the corresponding **pressure alters** checkbox in the RotoPaint tool settings. You can also later change which attribute the pressure alters for an existing stroke on the **Stroke** tab of the properties panel.

Separate Select tools in the RotoPaint toolbar let you select strokes/shapes once they've been drawn and after that you can make changes to them using the properties panel controls (see Selecting Existing Strokes/Shapes for Editing, and Editing Existing Stroke/Shape Attributes).

These pages discuss the general steps for using each of these tools, and give an overview on editing their attributes and timing.

Using the Brush tool

The **Brush** tool lets you apply colored or blended strokes to the current plate.



Painting with the Brush tool.

To Use the Brush Tool

1. Click the **Brush** tool  in the RotoPaint toolbar.
2. Set color, opacity, brush type, brush size, and brush hardness in the RotoPaint tool settings at the top of the Viewer. (For information on the available options, see Editing Existing Stroke/Shape Attributes.) You can also change the size of the brush using **Shift**+drag directly in the Viewer as shortcut.
3. Optionally, set the lifetime of the stroke in the RotoPaint tool settings. (For information on the available options, see Editing Existing Stroke/Shape Timing.)
4. Apply strokes as necessary.

Tip: If you need to draw a completely straight line with the brush tool, try this: Draw a freehand stroke, activate the **Select All** tool, enable **show paint stroke splines** mode  in the tool settings, and marquee select and delete all but the first and last point on the stroke.

Using the Eraser Tool

The **Eraser** tool lets you remove pixels from existing paint strokes.

To Use the Eraser Tool

1. Set opacity, brush type, brush size, and brush hardness in the RotoPaint tool settings at the top of the Viewer. (For information on the available options, see Selecting Existing Strokes/Shapes for Editing.)
2. Optionally, set the lifetime of the stroke in the RotoPaint tool settings. (For information on the available options, see Selecting Existing Strokes/Shapes for Editing.)
3. Right-click the **Brush** tool in the RotoPaint toolbar and select the **Eraser** tool .
4. Apply strokes as necessary. You can also erase multiple strokes, if you have drawn more than one.



Unwanted paint strokes.



Using the Eraser tool to remove the unwanted paint strokes.

Tip: If you're using a graphics tablet, Nuke automatically switches to Eraser mode when you use the erase end of the pen.

Using the Clone Tool

The **Clone** tool lets you remove unwanted features from the plate or from a different input by painting over them with pixels offset from the pointer or a transformation of the pointer.



Painting with the Clone tool.

To Use the Clone Tool

1. Click the **Clone** tool in the RotoPaint toolbar.
2. To view all the settings for the Clone tool, enable **show clone settings** in the RotoPaint tool settings at the top of the Viewer.

3. In the RotoPaint tool settings, set the paint source dropdown menu to the input you want to clone pixels from. (For information on the available options, see Editing Stroke-Specific Attributes.)

   You can also use the transform controls in the clone settings  to transform the clone source and reset it back to original with the **reset** button.
4. Set opacity, lifetime, brush type, size, and hardness for your stroke in the tool settings. (For more information, see Editing Existing Stroke/Shape Attributes and Editing Existing Stroke/Shape Timing.) If you've got more than one view set up, you can check the **single view** box to only clone in one view, or uncheck it to clone in all of them.
5. To set the clone offset, hold down **Ctrl/Cmd** and left-click at the source location, drag to where you want to paint, and release. Alternatively, you can enter the offset numerically using the **translate** controls in the RotoPaint tool settings. If you'd like the offset amount to be rounded to an integer (whole number of pixels), check **round**. Rounding to a pixel can be useful if you don't want to soften the image by partially blending pixels together.
6. Start painting. The pointer overlay depicts the source of the offset as a crosshair within a circle and the destination as a circle (the diameter of which represents the breadth of the stroke).

   You can use **/** (forward slash) and ***** (asterisk) on the numeric keypad to zoom your clone source in and out, and **0** (zero) and **.** (decimal point) to rotate it right and left. You can also use the number keys on the numeric keypad to nudge the clone source.
7. To reset the clone offset, you can use **Ctrl/Cmd**+drag to adjust the offset you set before, or **Ctrl/Cmd**+**Shift**+drag to start a new offset from the brush pointer's location.

Tip: If you're cloning from the current plate (**foreground**), you're also cloning all the strokes/shapes you've previously drawn. If you want to clone from the original background or a different picture, you need to set the paint source dropdown menu to pull from that input.

Tip: To clone pixels from another frame of the input clip, you can use the **time offset** slider to define which frame you want to clone from. See Editing Clone or Reveal Attributes.

Using the Reveal Tool

The **Reveal** tool lets you pull pixels from background elements onto the current plate. The **Reveal** tool requires at least two inputs (see Connecting the RotoPaint Node); otherwise, your strokes draw in white. You can also view which input you are using as the source for your strokes in the **Source** column in the stroke/shape list.



The current plate.



The source input.



Painting with the Reveal tool.

To Use the Reveal Tool

1. Right-click the **Clone** tool in the RotoPaint toolbar and select the **Reveal** tool .
2. In the RotoPaint tool settings at the top of the Viewer, set the paint **source** dropdown menu to the input you want to pull pixels from. (For information on the available options, see Editing Stroke-Specific Attributes.)
3. Set opacity, brush size, and brush hardness in the RotoPaint tool settings. (For information on the available options, see Editing Existing Stroke/Shape Attributes.)
4. Optionally, set the lifetime of the stroke in the RotoPaint tool settings. (For information on the available options, see Editing Existing Stroke/Shape Timing.)
5. You can also reveal pixels from another frame of the input clip by using the **time offset** slider to define which frame you want to reveal from. See Editing Clone or Reveal Attributes.
6. If you want, you can view your revealing source image in a Viewer overlay. To do this, check the **onion** box in the RotoPaint tool settings and enter an onion skin value to adjust the opacity of the overlay. This can help you better see what you are revealing from the source image. You can also toggle the keyboard shortcut **T** to enable or disable onion skin.
7. Start painting.



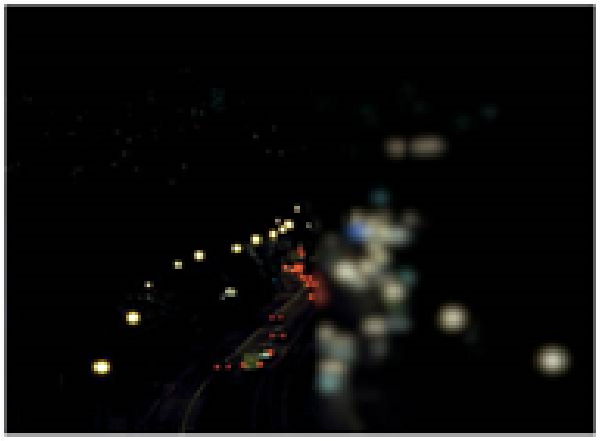
Onion skin disabled.



Onion skin enabled.

Using the Blur Tool

The **Blur** tool lets you blur parts of the plate.



Painting with the Blur tool.

To Use the Blur Tool

1. Click the **Blur** tool  in the RotoPaint toolbar.
2. Set opacity, brush type, brush size, and brush hardness in the RotoPaint tool settings. (For information on the available options, see Editing Existing Stroke/Shape Attributes.)
3. Optionally, set the lifetime of the stroke in the RotoPaint tool settings. (For information on the available options, see Editing Existing Stroke/Shape Timing.)
4. Apply strokes by clicking on the part of image you want to blur.

Tip: You can use the **effect** control in the RotoPaint tool settings to adjust the strength of the tool you're using. With the **Blur** tool, it controls the amount of blur.

Using the Sharpen Tool

With the **Sharpen** tool, you can sharpen the image within the area of the brush stroke.



Painting with the Sharpen tool.

To Use the Sharpen Tool

1. Right-click the **Blur** tool  and select **Sharpen** tool.
2. Set opacity, brush type, brush size, and brush hardness in the RotoPaint tool settings at the top of the Viewer. (For information on the available options, see Editing Existing Stroke/Shape Attributes.)
3. Optionally, set the lifetime of the stroke in the RotoPaint tool settings. (For information on the available options, see Editing Existing Stroke/Shape Timing.)
4. Apply strokes by clicking on the part of image you want to sharpen.

Tip: You can use the **effect** control in the RotoPaint tool settings to adjust the strength of the tool you're using. With the **Sharpen** tool, it controls how much the image is sharpened.

Using the Smear Tool

With the **Smear** tool, you can smear or stretch pixels over the surrounding pixels.



Painting with the Smear tool.

To Use the Smear Tool

1. Right-click the **Blur** tool and select **Smear** tool .
2. Set opacity, brush type, brush size, and brush hardness in the RotoPaint tool settings at the top of the Viewer. (For information on the available options, see Editing Existing Stroke/Shape Attributes.)
3. Optionally, set the lifetime of the stroke in the RotoPaint tool settings. (For information on the available options, see Editing Existing Stroke/Shape Timing.)
4. Apply strokes by clicking and dragging on the part of image you want to smear.

Using the Dodge Tool

With the **Dodge** tool, you can lighten the pixels in the area of the brush stroke. This makes the background color brighter on the area of the brush stroke to reflect the brush stroke.



Painting with the Dodge tool.

To Use the Dodge Tool

1. Click the **Dodge** tool .
2. Set opacity, brush type, brush size, and brush hardness in the RotoPaint tool settings at the top of the Viewer. (For information on the available options, see Editing Existing Stroke/Shape Attributes.)
3. Optionally, set the lifetime of the stroke in the RotoPaint tool settings. (For information on the available options, see Editing Existing Stroke/Shape Timing.)
4. Apply strokes as necessary.

Using the Burn Tool

With the **Burn** tool, you can darken the pixels in the area of the brush stroke. This makes the background color darker on the area of the brush stroke.



Painting with the Burn tool.

To Use the Burn Tool

1. Right-click the **Dodge** tool and select **Burn** tool .
2. Set opacity, brush type, brush size, and brush hardness in the RotoPaint tool settings. (For more information on the available options, see Editing Existing Stroke/Shape Attributes.)
3. Optionally, set the lifetime of the stroke in the RotoPaint tool settings. (For information on the available options, see Editing Existing Stroke/Shape Timing.)
4. Apply strokes as necessary.

Drawing Shapes

Any given RotoPaint node can hold several shapes, and you can draw them using any of the following tools.

Icon	Tool	Keyboard Shortcut	Function
	Bezier	V (toggles between Bezier, B-spline, Ellipse, and Rectangle)	Applies a Bezier shape. Bezier shapes are defined using control points and tangents.
	Cusped Bezier	V	Applies a Bezier shape with sharp corners and no tangents.
	B-Spline	V	Applies a B-spline shape. Unlike Bezier shapes, B-splines are created by only using control points. The position of the points in relation to each other determines what kind of splines the shape consists of.
	Ellipse	V	Applies an ellipse-shaped Bezier shape.
	Rectangle	V	Applies a rectangle-shaped Bezier shape.
	Cusped Rectangle	V	Applies a rectangle-shaped Bezier shape with sharp corners and no tangents. You can see the

Icon	Tool	Keyboard Shortcut	Function
			difference between rectangles and cusped rectangles if you move at least one of the control points.
	Open Spline	V	Lets you draw curves in a similar way to other shapes, except that they don't need form a closed shape.

Separate Select tools in the RotoPaint toolbar let you make changes to a stroke/shape once it's been drawn (see Selecting Existing Strokes/Shapes for Editing).

Tip: You can choose to have your RotoPaint always open with a particular tool selected. If you want to open it with the Bezier tool selected, for example, do the following:
1. Create a file called **menu.py** in your plug-in path directory if one doesn't already exist. For more information on plug-in path directories, see Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts.
2. To select Bezier as your default tool, save the following in your **menu.py**:
nuke.knobDefault('RotoPaint.toolbox','bezier')
3. Restart Nuke.

For more information on default tools, see Setting Default RotoPaint Tools and Settings.

Tip: If you begin drawing shapes and then decide you want to planar track these with the PlanarTracker node, you can do that easily by right-clicking on the shape in the stroke/shape list and selecting **planar-track**. This creates a new planar tracking layer for the shape and attaches the shape to it. For more information about PlanarTracker, see Tracking with PlanarTracker.

Using the Bezier and Cusped Bezier Tools

The Bezier and Cusped Bezier tools let you draw Bezier shapes. Cusped Beziers are Bezier shapes with sharp corners and no tangents.



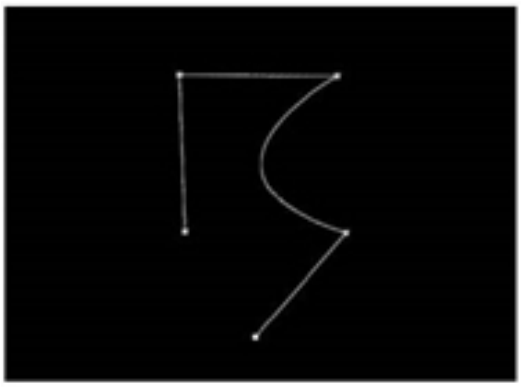
Drawing a Bezier shape.

To Use the Bezier or Cusped Bezier Tools

1. Do one of the following:
   - Click the **Bezier** tool  in the RotoPaint toolbar.
   - Right-click the **Bezier** tool in the RotoPaint toolbar and select the **Cusped Bezier** tool .
2. Select color, blending mode, opacity, and other settings for the shape in the RotoPaint tool settings at the top of the Viewer. (For information on the available options, see Editing Existing Stroke/Shape Attributes.)
3. Optionally, set the lifetime of the shape in the RotoPaint tool settings. (For information on the available options, see Editing Existing Stroke/Shape Timing.)
4. Draw a shape in the Viewer by clicking to create the outlines that make up your shape.
5. If you're using the Bezier tool, you can click+drag while drawing to create a point and adjust its tangent handles. With the tangent handles, you can adjust the spline of your shape.
   - You can move individual handles to adjust their length, keeping the angle consistent.
   - Press **Shift** while moving the tangent handles to move both handles at the same time, keeping the angle consistent.
   - Press **Ctrl**/**Cmd** to temporarily break the angle.
6. You can also use other shortcuts to adjust your shape while drawing:
   - **Shift**+click to create a sharp exit point on the previous point (this has no effect when using the Cusped Bezier tool, as all the points are sharp anyway).



Curved exit point.



Sharp exit point.

- **Ctrl/Cmd**+click to sketch your shape freely.

7. To close your shape, press **Return** or click the first point of your shape. Changing to a different tool also closes a shape. By default, closing a shape activates the Select tool.

   If you're using the Bezier tool and close your shape by clicking its first point, you can also drag the point to create tangent handles for adjusting it.
8. With the Select tool active, you can **Shift**+click on multiple shape points to bring up the transform box, which you can use to further transform your shape or particular points in your shape.
9. You can also cusp and smooth a point in your Bezier shape to create a sharp corner on your shape or smooth it again. To do this, right-click on a point in your Bezier shape and select **cusp/de-smooth** or **smooth**.

Tip: You can also apply animation from a Tracker node to a point in your Bezier shape. From the Tracker node, **Ctrl/Cmd**+drag the transformation information into the Viewer, on the point you want to animate. See also Linking Expressions for creating links between the Tracker node and RotoPaint controls.

Using the B-Spline Tool

The **B-spline** tool lets you draw B-spline shapes.



Drawing a B-spline shape.

To Use the B-Spline Tool

1. Right-click the **Bezier** tool in the RotoPaint toolbar and select **B-Spline** tool .
2. Select color, blending mode, opacity, and other settings for the shape in the RotoPaint tool settings at the top of the Viewer. (For information on the available options, see Editing Existing Stroke/Shape Attributes)
3. Optionally, set the lifetime of the shape in the RotoPaint tool settings. (For information on the available options, see Editing Existing Stroke/Shape Timing.)
4. Click in the Viewer to create the splines that make up your shape.
   - **Ctrl/Cmd**+drag to sketch the spline freely.
   - Click+drag the cursor left or right to adjust the tension (that is, the distance between the spline line and a point) *on the previous point*.
   - To adjust the tension of *any point* in the B-spline shape you're drawing, select a point, press **Ctrl/Cmd+Shift,** and drag left or right: right to increase the tension, left to decrease it.
5. To close your shape, press **Return**. Changing to a different tool also closes a shape. By default, closing a shape activates the Select All tool.
6. With the Select All tool active, you can **Shift**+click on multiple shape points to bring up the transform box, which you can use to further transform your shape or particular points in your shape.
7. You can also cusp and smooth a point in your B-spline shape to create a sharp corner on your shape or smooth it again. To do this, right-click on a point in your B-spline shape and select **cusp/de-smooth** or **smooth**.

Tip: You can convert your B-spline shape into a Bezier shape after creating it. Select the B-spline shape using the Select All tool, right-click on it and select **convert bspline to bezier**. You can now edit your shape in the same way as other Bezier shapes.

Using the Ellipse, Rectangle, and Cusped Rectangle Tools

The **Ellipse**, **Rectangle**, and **Cusped Rectangle** tools let you draw an ellipse- or rectangle-shaped Bezier shape. Rectangles and cusped rectangles look exactly the same initially, but if you move one or more of the control points, you can see that cusped rectangles always have sharp corners and no tangents.

After creation, ellipses, rectangles, and cusped rectangles can be edited like normal Bezier shapes.



Here, an ellipse and a rectangle have been used to create a quick garbage matte.

To Use the Ellipse, Rectangle, and Cusped Rectangle Tools

1. Right-click the **Bezier** tool in the RotoPaint toolbar and select the **Ellipse** tool , **Rectangle** tool , or **Cusped Rectangle** tool .
2. Select color, blending mode, opacity, and other settings for the shape in the RotoPaint tool settings at the top of the Viewer. (For information on the available options, see Editing Existing Stroke/Shape Attributes.)
3. Optionally, set the lifetime of the shape in the RotoPaint tool settings. (For information on the available options, see Editing Existing Stroke/Shape Timing.)
4. Click+drag across the Viewer to draw an ellipse or a rectangle shape.

   If you want to draw a shape from the center out, press **Ctrl**/**Cmd**+**Shift** as you're dragging.

If you want to create a perfect circle with the **Ellipse** tool or a square with the **Rectangle** tool, hold down **Shift** while drawing your shape.

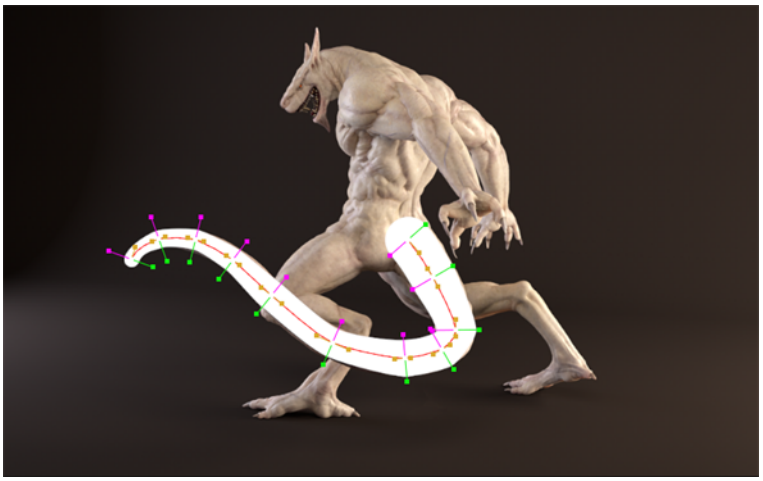


Holding down **Shift** while drawing an ellipse creates a perfect circle.

Using the Open Spline Tool

The **Open Spline** tool lets you draw curves in a similar way to other shapes, except that they don't need form a closed shape.

After creation, you can edit points on open splines using the standard smooth and transform handles, but they also have individual thickness and feather handles. The image shows an open spline and each point's smooth (orange), thickness (green), and feather (magenta) handles.



Here, an open spline is used to roto a creature's tail.

To Use the Open Spline Tool

1. Right-click the **Bezier** tool in the RotoPaint toolbar and select the **Open Spline** tool .



Tip: You can also press **V**, when the Viewer has focus, to cycle through the available tools.

2. Select color, blending mode, opacity, and other settings applicable to all RotoPaint tool at the top of the Viewer. (For information on the available options, see Editing Existing Stroke/Shape Attributes.)

   RotoPaint has a few additional open spline-specific controls:

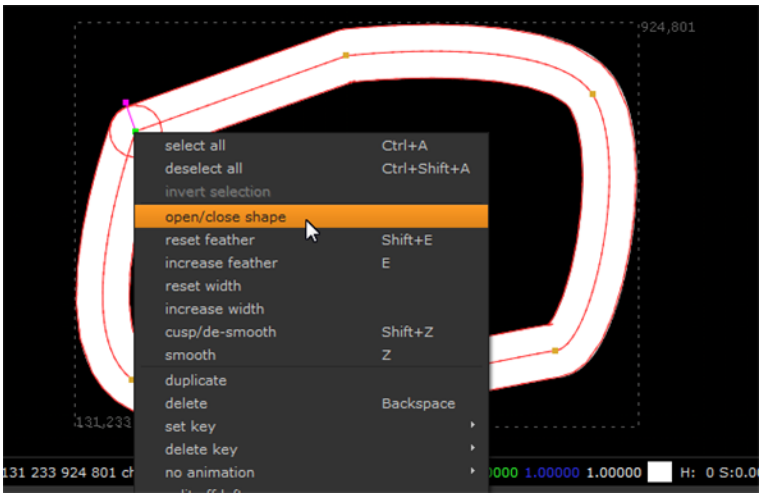
   • **width** - controls the overall thickness of the spline.

   • **start type** - sets the state of the first point in the spline to either **rounded** or **square**.

   • **end type** - sets the state of the last point in the spline to either **rounded** or **square**.
3. Click in the Viewer to place the starting point of the spline and then add the required number of points on the feature you're working on.
4. Press **Return** or select another tool to finish the open spline,

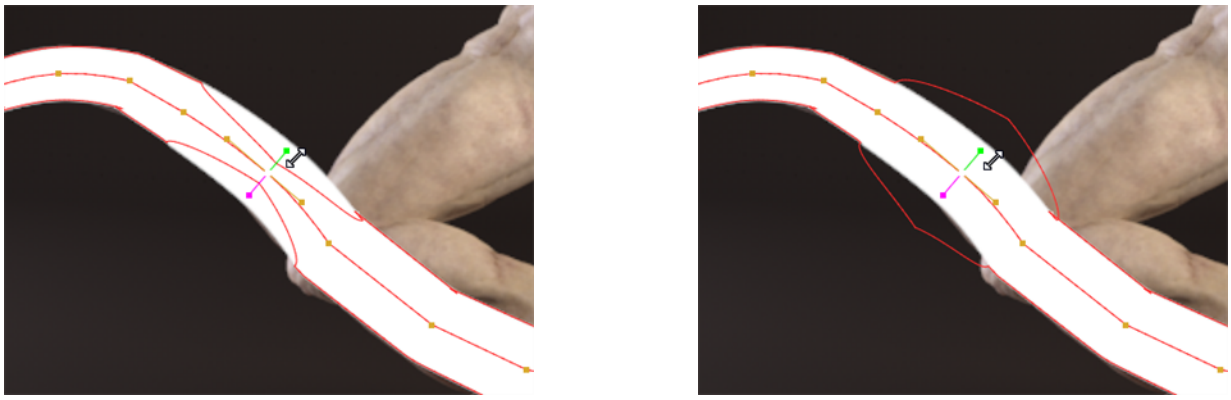
   OR

   Click on the first point in the open spline to close the shape. Closed splines are not filled in the same way as other shape tools, such as **Bezier** and **Ellipse**. You can open a closed spline by right-clicking a point on the spline and selecting **open/close shape**.

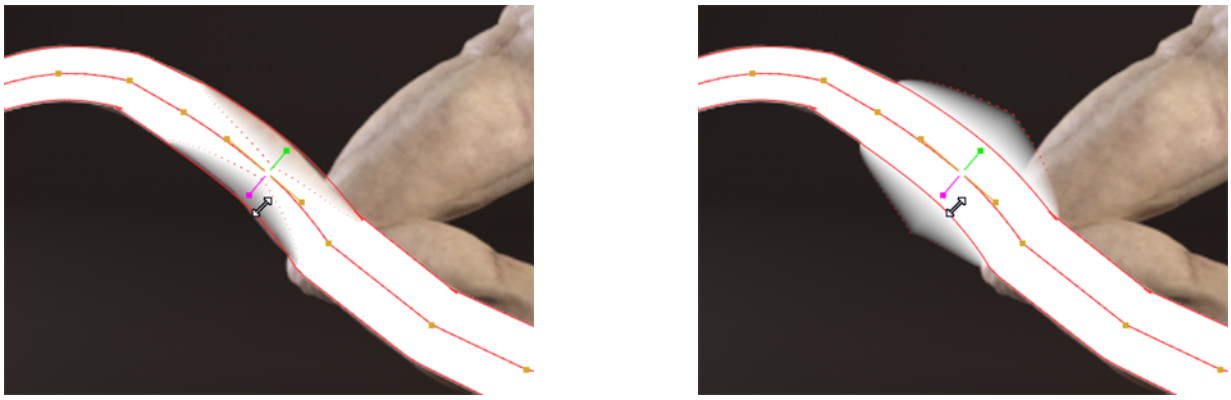


5. You can adjust the thickness and feather of individual points on the spline by selecting the required points and using the handles displayed in the Viewer.

   Drag the handle and release to apply the change to the selected points.



Decreasing and increasing spline thickness (green handle).



Decreasing and increasing feathering (magenta handle).

6. Optionally, set the lifetime of the shape in the RotoPaint tool settings. (For information on the available options, see Editing Existing Stroke/Shape Timing.)

Setting Default RotoPaint Tools and Settings

You can choose to have your RotoPaint always open with a particular tool selected, or open a particular tool with the settings you want.

To Set a Tool as Your Default Tool:

1. Create a file called **menu.py** in your plug-in path directory if one doesn't already exist. For more information on plug-in path directories, see Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts.
2. Select your default tool from the following list and add the Python line in your **menu.py**:

- Select All
  `nuke.knobDefault('RotoPaint.toolbox','selectAll')`
- Select Splines
  `nuke.knobDefault('RotoPaint.toolbox','selectCurves')`
- Select Points
  `nuke.knobDefault('RotoPaint.toolbox','selectPoints')`
- Select Feather Points
  `nuke.knobDefault('RotoPaint.toolbox','selectFeatherPoints')`
- Add Points
  `nuke.knobDefault('RotoPaint.toolbox','addPoints')`
- Remove Points
  `nuke.knobDefault('RotoPaint.toolbox','removePoints')`
- Cusp Points
  `nuke.knobDefault('RotoPaint.toolbox','cuspPoints')`
- Smooth Points
  `nuke.knobDefault('RotoPaint.toolbox','curvePoints')`
- Remove Feather
  `nuke.knobDefault('RotoPaint.toolbox','removeFeather')`
- Open/Close Curve
  `nuke.knobDefault('RotoPaint.toolbox','closeCurve')`
- Bezier
  `nuke.knobDefault('RotoPaint.toolbox','createBezier')`
- Cusped Bezier
  `nuke.knobDefault('RotoPaint.toolbox','createBezierCusped')`
- B-Spline
  `nuke.knobDefault('RotoPaint.toolbox','createBSpline')`
- Ellipse
  `nuke.knobDefault('RotoPaint.toolbox','createEllipse')`
- Rectangle
  `nuke.knobDefault('RotoPaint.toolbox','createRectangle')`
- Cusped Rectangle
  `nuke.knobDefault('RotoPaint.toolbox','createRectangleCusped')`
- Brush
  `nuke.knobDefault('RotoPaint.toolbox','brush')`
- Eraser
  `nuke.knobDefault('RotoPaint.toolbox','eraser')`
- Clone

```
nuke.knobDefault('RotoPaint.toolbox','clone')
```

- Reveal
```
nuke.knobDefault('RotoPaint.toolbox','reveal')
```
- Dodge
```
nuke.knobDefault('RotoPaint.toolbox','dodge')
```
- Burn
```
nuke.knobDefault('RotoPaint.toolbox','burn')
```
- Blur
```
nuke.knobDefault('RotoPaint.toolbox','blur')
```
- Sharpen
```
nuke.knobDefault('RotoPaint.toolbox','sharpen')
```
- Smear
```
nuke.knobDefault('RotoPaint.toolbox','smear')
```

3. Restart Nuke.

To Set Your Default Tool Properties:

1. Create a file called **init.py** in your plug-in path directory if one doesn't already exist. For more information on plug-in path directories, see Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts.
2. Define your RotoPaint tool and the default setting you want to set. To get the specific tool and setting names, you can copy your RotoPaint node in Nuke and paste it into a text editor. The lines with curved brackets after "toolbox" indicate your current tool settings.

   For example:

   - to set the **brush size** for paint to **30**, and for **clone** to **280**, add this Python line in your **init.py**:
   ```
   nuke.knobDefault("RotoPaint.toolbox", '''clone {
   { brush bs 30 }
   { clone bs 280 }
   }''')
   ```
   OR
   - to set the **brush hardness** for paint to **1.0** by default, add this Python line:
   ```
   nuke.knobDefault("RotoPaint.toolbox", '''brush {
   { brush h 1 }
   }''')
   ```
   - to set the **source transform** round on clone to **on** by default, add this Python line:
   ```
   nuke.knobDefault("RotoPaint.toolbox", '''clone {
   { clone str 1 }
   }''')
   ```
3. Restart Nuke.

Note: As you can see from the above examples, the different RotoPaint tools have their own default tool settings. If you attempt to set a default but don't specify for which tool, the default gets ignored as soon as you switch to a different tool. For example, nuke.knobDefault ('RotoPaint.toolbar_brush_hardness','1.0') sets **brush hardness** to **1.0** initially, but as soon as you activate a different tool, you get the default for that tool.

Note: You cannot set up multiple defaults through RotoPaint.toolbox. For example, if you run the following three commands, the default is only set for the last one (sharpen):

```
nuke.knobDefault("RotoPaint.toolbox", '''clone {{ clone ltt 0}}''')
nuke.knobDefault("RotoPaint.toolbox", '''blur {{ blur ltt 0}}''')
nuke.knobDefault("RotoPaint.toolbox", '''sharpen {{ sharpen ltt 0}}''')
```

To set the defaults for clone and blur as well, you should instead use this:

```
nuke.knobDefault("RotoPaint.toolbox", '''clone {
{ brush ltt 0 }
{ clone ltt 0}
}''')
```

Selecting the Output Format and Channels

In the RotoPaint properties panel, you can select one output channel or many to indicate the channels where the results of your changes should be stored.

1. From the **output** field, select the layer containing the channels you want to use. By default, **rgba** is selected, and the red, green, blue, and alpha channels are checked on the right.
2. Uncheck any channels that you don't want to process. The node processes all those you leave checked. For more information on selecting channels, see Calling Channels.
3. If you want to, you can use the **output mask** dropdown menu to select a channel where RotoPaint outputs a mask for what it rendered. By default, the channel is **none**, but if you select a channel in the menu, the **output mask** box is automatically checked.

   The mask can be useful, for example, if you need to apply grain to the areas you've painted, but you don't want to double up the grain in other areas.

   By default, the **output mask** control is hidden, but you can display it by clicking on the black triangle above **color.**

4. If necessary, select your **premultiply** value.

   Premultiply multiplies the selected input channels with a mask representing the paint strokes and shapes. For example, where there are no paint strokes or shapes (the paint matte is black or empty) the input channels are set to black, and where the paint strokes or shapes are opaque (the paint matte is white or full) the input channels keep their full value.

   Note that selecting **rgba** premultiplies the alpha against itself (a*a). If you don't want this to happen, set **premultiply** to **rgb** instead.

Tip: You can use the **Link Menu**  to link the channel controls with other controls. For more information, see Linking Channels Using the Link Menu.

5. From the **clip to** dropdown menu, select how you want to restrict the output image:
   - **no clip** - Do not restrict the output image.
   - **bbox** - Restrict the output image to the incoming bounding box.
   - **format** - Restrict the output image to the incoming format area (the default).
   - **union bbox+format** - Restrict the output image to a combination of the bounding box and the incoming format area.
   - **intersect bbox+format** - Restrict the output image to an intersection of the bounding box and incoming format area.

   You can also check the **Replace** box if you want to clear the channels to black before drawing into them. You might find **Replace** useful, for instance, if you're creating a mask in the alpha channel, but the incoming image already has an alpha channel which you want to throw away.
6. If you haven't connected an input to RotoPaint, select your **format** value. This is the format which the node should output in the absence of any available input format. If an input is connected, this control has no effect.

   By default, the **format** control is hidden, but you can display it by clicking on the black triangle above **color**.

Selecting Existing Strokes/Shapes for Editing

If you've already drawn a stroke/shape but wish to make changes to it, you can select it or certain parts of it with the RotoPaint selection tools. You can also toggle between all of them in the RotoPaint toolbar using the shortcut **Q**.

When you click on a point in a stroke/shape, the point changes color to indicate whether it's in focus (green by default) and whether it has an expression set (red by default). You can change the default colors on the **Viewers** tab in the Preferences.

You can also use the controls in the RotoPaint tool settings to display and hide information such as points, point numbers, splines, and transform handles. See To Adjust Display Properties for Selected Shapes/Strokes below.

To Adjust Display Properties for Selected Shapes/Strokes

Whenever you have a Select tool active, the RotoPaint tool settings allow you to control what information is displayed for the visible shapes/strokes and points in the Viewer:

- To view or hide the numbers for visible shape/stroke points, toggle the **label points** button . Feather points are marked with a bracketed number corresponding their shape point, so for example, if a shape point is marked with the number 2, its feather point is marked with [2].
- To view or hide the splines in visible shapes, toggle the **hide splines** button .
- To view or hide the splines in visible paint strokes, enable **show paint stroke splines**  and toggle the **hide splines** button .
- To view or hide the points (and tangent handles) in visible shapes/strokes, toggle the **hide points** button .
- To view or hide the transform handle jack or transform box for visible shapes/strokes, toggle the **hide transform handles** button  (or press **T** on the Viewer).
- To hide the transform handle jack or transform box when moving a selection, click the hide transform handles on move button . This may make it easier to correctly position your selection.

To Select an Entire Stroke/Shape

1. Click the **Select All** tool , or press the keyboard shortcut **Q**.
2. Select the stroke/shape you wish to edit either by clicking on it in the Viewer or by clicking on its name in the stroke/shape list. To select several strokes/shapes, **Ctrl**/**Cmd**+click or **Shift**+click (to select a range) their names in the stroke/shape list. You can have both splines and points selected simultaneously.

When selecting strokes/shapes in the Viewer, you can invert your selection by right-clicking and selecting **invert selection**. All strokes/shapes you didn't have selected before are now selected.

If you want to move the selected stroke/shape, drag a selection box around the entire shape until a box forms around the shape. Pulling on the cross at the center of the shape allows you to move it around the Viewer. A keyframe is automatically added when you move a stroke/shape, if autokey is enabled. For information on autokey and creating keyframes, see Animating Strokes/Shapes.

Tip: To select points in a paint stroke and view your selection in the Viewer, you have to enable **show paint stroke splines** in the tool settings.

Tip: By default, if you have selected a stroke/shape in the Viewer or the stroke/shape list, clicking on an empty area in the Viewer does not deselect it. If you'd like to change this behavior, you can disable **constant selection** mode in the RotoPaint tool settings.

To Select a Spline

1. Right-click the **Select All** tool and select the **Select Splines** tool.
2. Make sure **hide splines** is disabled in the RotoPaint tool settings.
3. Select the spline you wish to edit either by clicking on it in the Viewer or by clicking on its name in the stroke/shape list. You can also click-and-drag a selection box around the shape with **Select Splines** active. Selecting a spline only selects the spline, not points within it.

Tip: Using the **Select Splines** tool, you can also duplicate the stroke/shape you've selected. Just right-click on one of the points, and select **duplicate**. A new stroke/shape is created with the same spline and attributes as the one you selected.

To Select Only Points on a Stroke/Shape

1. Right-click the **Select All** tool and select the **Select Points** tool.
2. Make sure **hide points** is disabled in the RotoPaint tool settings.

3. Select the stroke/shape you wish to edit by clicking on its name in the stroke/shape list and then select a point in the Viewer. To select several points, **Ctrl**/**Cmd**+click on them in the Viewer or use marquee selection to create a transform box.

   Using the **Select Points** tool restricts selection to one stroke/shape only.

To Select Feather Points

1. Right-click the **Select All** tool and select the **Select Feather Points** tool .
2. Make sure **hide points**  is disabled in the RotoPaint tool settings.
3. Select the stroke/shape you want to edit by clicking on its name in the stroke/shape list and then select a feather point in the Viewer. To select several feather points, **Ctrl**/**Cmd**+click on them in the Viewer or use marquee selection to create a transform box.

   Using the **Select Feather Points** tool restricts selection to one stroke/shape only.

To Select Points Using a Transform Handle

1. Right-click the **Select All** tool and select the **Select Points** tool .
2. Make sure **hide transform handles**  is disabled in the RotoPaint tool settings.
3. Select points in a stroke/shape with **Shift**+click or by clicking and dragging across the points you want to select. A transform box appears.
4. You can also use shortcut **T** to toggle viewing the transform handle.

Editing Existing Stroke/Shape Attributes

After selecting a stroke/shape using the stroke/shape list or one of the Select tools, you can edit and animate its attributes in the properties panel. For more information on selecting strokes/shapes, see Selecting Existing Strokes/Shapes for Editing and Working with the Stroke/Shape List.

If you want to edit the attributes of a stroke/shape prior to drawing one, you should do that in the RotoPaint tool settings in the Viewer. (See Working with the Toolbars.)

Editing Color

When drawing a stroke/shape, you can set the RGBA color values of the stroke/shape using the **color** controls on the **RotoPaint** tab of the RotoPaint properties panel (for more information on the color controls, see the Using the Compositing Environment chapter). You can also adjust color directly in the stroke/shape list using the  control in the **color** column.



Color set to white (the default).



Color set to green.

Editing Opacity

You can set the opacity of the stroke/shape using the **opacity** slider. If you set the opacity of a shape to zero, the outline for it won't be drawn unless the shape is selected. You can also temporarily make the stroke/shape invisible (that is, completely transparent) by toggling the **visible** box  in the stroke/shape list.



A low **opacity** value.



A high **opacity** value.

Tip: When drawing brush strokes, you can tie their transparency to pen pressure. Just check the **opacity** box next to **pressure alters** on the **Stroke** tab of the properties panel.

Selecting a Source for Your Stroke/Shape

You can select a source for your stroke/shape and define whether it's a color, a background, or a foreground image. With your stroke/shape selected, choose a source for your stroke/shape from the **source** dropdown on the **RotoPaint**, **Shape**, or **Stroke** tabs.

Editing Blending Mode

By selecting different blending modes from the **blending mode** dropdown in the properties panel, you can select how the colors in your strokes/shapes blend with the underlying image. You can also apply blending modes to your strokes, shapes, or groups directly in the stroke/shape list using the **blending mode** column .

Each of the blending modes blends the primary color, that is the color of the current stroke/shape/group you're editing with the secondary color, which is the combined color of your background image and any previously rendered strokes/shapes/groups.

The different modes are as follows:

- **Color burn** - Darkens the primary color to reflect the secondary color by increasing the contrast. No part of the image becomes lighter.
- **Color dodge** - Brightens the primary color to reflect the secondary color by decreasing the contrast. No part of the image is darkened.
- **Difference** - Subtracts either the secondary color from the primary color or vice versa, depending on which is brighter. Blending with white inverts the primary color, while blending with black produces no change. Similar colors return black pixels. Difference is a useful mode when working with mattes.
- **Exclusion** - Creates a result similar to the Difference mode but lower in contrast. Like with Difference, blending with white inverts the primary color. Blending with black produces no change.
- **From** - Subtracts the primary color from the secondary color.
- **Hard Light** - Lightens highlights and darkens shadows. If the secondary color is lighter than 50% gray, the result lightens as if it were screened. If the secondary color is darker than 50% gray, the result is darkened as if it were multiplied.
- **Max** - Selects the lighter of the two colors as the resulting color. Only areas darker than the secondary color are replaced, while areas lighter than the secondary color do not change.

- **Min** - Selects the darker of the two colors as the resulting color. Any parts that are lighter than the secondary color are substituted. Any parts of the image that are darker than the secondary color don't change.
- **Minus** - Subtracts the secondary color from the primary color.
- **Multiply** - Multiplies the primary color by the secondary color. The result is always darker. Blending with black gives black, and with white returns the color unchanged.
- **Over** - This mode is the default. The colors of the two images do not interact in any way, and Nuke displays the full value of the colors in the primary image.
- **Overlay** - Depending on the primary color, multiplies or screens the colors. The secondary color brightens the primary color while preserving highlights and shadows.
- **Plus** - The sum of the two colors. Increases brightness to lighten the primary color and reflect the secondary color. Plus is similar to the Screen blending mode, but produces a more extreme result.
- **Screen** - This is a soft Plus making everything brighter but ramping off the whites. Light colors have more of an effect than dark colors. The result is always a lighter color. Blending with black leaves the pixel unchanged, blending with white always returns white. The result is similar to projecting multiple slides on top of each other.
- **Soft Light** - Depending on the primary color, darkens or lightens the colors. Less extreme than the Hard Light mode.

Tip: Note that changing the stack order of your primary and secondary colors might have an impact on your result. For example, if you have two Bezier shapes overlapping each other with a blending mode active, the result depends on which shape is on top of the other. You can change the stack order of strokes/shapes in the stroke/shape list.

Transforming Strokes/Shapes/Groups

To apply spatial transformations to your strokes, shapes, or groups, you can use the controls under the **Transform** tab. Select a stroke/shape/group from the stroke/shape list and adjust:

- **translate** - to move the stroke/shape on x and y axis.
- **rotate** - to spin a stroke/shape around the pivot point. Use **center** (or **Ctrl**/**Cmd**+drag the transform jack) to position the pivot point.
- **scale** - to resize a spline. Use **center** (or **Ctrl**/**Cmd**+drag the transform jack) to position the pivot point.

- **skew X** - to skew the spline of your stroke/shape along the X axis from the pivot point. Use **center** (or **Ctrl/Cmd**+drag the transform jack) to position the pivot point.
- **skew Y** - to skew the spline of your stroke/shape along the Y axis from the pivot point. Use **center** (or **Ctrl/Cmd**+drag the transform jack) to position the pivot point.
- **skew order** - to set the order in which **skew X** and **skew Y** are applied:
  - **XY** - **Skew X** is applied before **skew Y**.
  - **YX** - **Skew Y** is applied before **skew X**.
- **center x, y** - to set the center for rotation and scaling, adjust the values in center x, y.
- **extra matrix** - enter values you want to get concatenated with the transformation controls above. For more information on concatenating, see How Nodes Concatenate.



Note: While you can drag the matrix to another node's matrix to easily use the values elsewhere, you shouldn't try, for example, to drag a 4 by 4 matrix on a 3 by 3 matrix, as doing that might have unexpected results.

Alternatively, you can also use the transform handle (shortcut **T**) in the Viewer to transform elements. To transform an entire stroke/shape, you need to use the transform handle jack, and to transform points in a stroke/shape, you should use the transform box.

The transform handle appears as a transform jack only when the **Transform** tab is active, when any of the other tabs in the RotoPaint properties panel are active, the transform handle appears as a box.

To Transform a Stroke/Shape Using a Transform Handle Jack:

1. Click one of the Select tools  in the RotoPaint toolbar, with the **Transform** tab active.
2. Make sure **hide transform handles**  is disabled in the RotoPaint tool settings.
3. Select a stroke/shape by clicking it in the Viewer or by selecting it in the stroke/shape list. A transform handle jack appears.
4. Use the jack to rotate, scale or skew your stroke/shape.

To Transform Points Using a Transform Box:

1. Click the **Select All** , **Select Points** , or **Select Feather Points**  tool in the RotoPaint toolbar, with the **RotoPaint** tab active.

2. Make sure **hide transform handles**  is disabled in the RotoPaint tool settings.
3. Select points in a stroke/shape with **Shift**+click or by clicking and dragging across the points you want to select. A transform box appears.
4. Use the box to rotate, scale or skew your stroke/shape, or points.
5. To corner pin using the transform box, press **Ctrl**/**Cmd**+**Shift** and drag the transform box points to move them.



Note: Transforming points changes the actual point position, while transforming an entire stroke/shape/group changes transformation applied to the point positions.

Tip: If you'd like to hide the transform box when moving a selection, enable **hide transform handles on move**  in the RotoPaint tool settings. This may make it easier to correctly position your selection.

To Transform Onion Skin Source

When cloning or revealing, you can use the onion skin control on the RotoPaint tool settings to view and transform your source input on top of your foreground. You can also use onion skinning if you're drawing a stroke/shape with a separate input as the source. To adjust onion skin:

1. With your stroke/shape tool selected, check the **onion** box in the RotoPaint tool settings.
2. Adjust the opacity of the onion skin and transform your source by using the onion skin transform overlay in the Viewer.



Onion skin transform handle.

Adjusting Mask Controls

By default, **mask** is set to **none**, but if you want to use a mask, do the following:

1. Check the **mask** checkbox on the **RotoPaint** tab. By default, the **mask** control is hidden, but you can display it by clicking on the black triangle above **color**.
2. Select a channel to use as a mask from the dropdown menu.
3. If you are using the mask input and want the mask copied into the predefined mask.a channel, check **inject**. This way, you can use the mask input again downstream.
4. If necessary, check **invert** to reverse the mask and/or **fringe** to only apply the effect at the edges of the mask.
5. If you find that the overall effect of the RotoPaint node is too harsh, you can also blend some of the original input image back in by adjusting the **mix** slider.

Editing Shape-Specific Attributes

The RotoPaint properties panel includes a set of controls that you mainly need when you're editing the attributes of a shape. You can find these on the **Shape** tab in the properties panel.

Adding and Removing Feather

To soften the edges of a shape, do the following:

1. With your shape selected, check the **on** box next to the **feather** slider on the **Shape** tab to apply feather to your shape. If you don't want any feather on your shape, uncheck the **on** box.
2. Use the **feather** slider to add outward or inward feather around the whole shape. With positive feather values, your feather effect is outward and, respectively, if your feather values are negative, the feather effect is inward.
3. Use the **feather falloff** slider to adjust the falloff profile. Falloff is measured in pixels. You can also change the type of the falloff using the **falloff type** dropdown menu. Select either **linear**, **smooth0**, **smooth1** and **smooth**. Each of these produces a different rate of falloff that may be helpful for example in matching the soft edge to motion blurred image content.
4. To add feather to a single point, right-click on the point in the Viewer and select **increase feather**. The shortcut for this is **E**. If you press it several times, every press adds more feather.
5. Use the feather handle on the point to add feather into the point. By default, the point angle is locked, and moving the point unlocks the angle.

   You can also select several feather points and move them together.
6. To remove feather from a point, right-click on the point and select **reset feather** or use the shortcut **Shift+E.**
7. If you disable **feather link** mode  (enabled by default) in the RotoPaint tool settings, the feather and shape points move independently.
   - **centered -** to center the shutter around the current frame. For example, if you set the shutter value to 1 and your current frame is 30, the shutter stays open from frame 29.5 to 30.5.
   - **start -** to open the shutter at the current frame. For example, if you set the shutter value to 1 and your current frame is 30, the shutter stays open from frame 30 to 31.
   - **end -** to close the shutter at the current frame. For example, if you set the shutter value to 1 and your current frame is 30, the shutter stays open from frame 29 to 30.
   - **custom -** to open the shutter at the time you specify. In the field next to the dropdown menu, enter a value (in frames) you want to add to the current frame. To open the shutter before the current frame, enter a negative value. For example, a value of - 0.5 would open the shutter half a frame before the current frame.

Editing Stroke-Specific Attributes

The RotoPaint properties panel includes a set of controls that you mainly need when you're editing the attributes of a paint stroke. You can find most of these under the **Stroke** tab in the properties panel.

Selecting a Source Image

Both on the **Stroke** and **RotoPaint** tabs, you can set the **source** control to a specific color or the input you want to pull pixels from for Clone and Reveal brushes. Select:

- **color** - to use a specific color in your stroke/shape.
- **foreground** - to pull pixels from the RotoPaint's **bg** input, including any strokes/shapes drawn on it. This input is mainly used with cloning.
- **background** - to pull pixels from the **bg** input, not including any strokes/shapes drawn on it.
- **background 1**, **background 2** or **background 3** - to pull pixels from the **bg1**, **bg2,** or **bg3** input.

Editing Brush Type

On the **Stroke** tab, select the type of brush you want to use for the stroke. From the **brush type** dropdown, select:

- **paint** - to use a normal paint brush.
- **smear** - to use a smear brush on the plate.
- **blur** - to blur your plate with the brush stroke. You can adjust the strength of the blur effect using the **effect** slider.
- **sharpen** - to sharpen your plate with the brush stroke. You can adjust the strength of the sharpening effect using the **effect** slider.

Editing Brush Size

On the **Stroke** tab, you can set the size of the stroke using the **brush size** slider. You can also tie a stroke's size to pen pressure by checking the **size** box next to **pressure alters**.



A low **brush size** value.



A higher **brush size** value.

Editing Brush Spacing

The **brush spacing** slider adjusts the distance between paint brush dabs. A higher setting increases the space between dabs, creating a dotted line effect when painting. A lower setting decreases the distance and creates a solid brush stroke.



A low **brush spacing** value.



A higher **brush spacing** value.

Editing Brush Hardness

On the **Stroke** tab, you can set the hardness of the stroke using the **brush hardness** slider.



A low **brush hardness** value.



A higher **brush hardness** value.

You can also tie a stroke's hardness to pen pressure by checking the **hardness** box next to **pressure alters**.

Adjusting Write On

When you are animating a stroke or a part of it over a range of frames, you can use the **write on** sliders under the **Stroke** tab in the properties panel to adjust the order in which the dabs on the stroke appear over these frames. For more information on animating parameters, see the Using the Compositing Environment chapter.

- **write on start** - slide to choose where along the stroke length the paint begins. 0 is the start of the stroke, 1 is the end.
- **write on end** - slide to choose where along the stroke length the paint ends.

Editing Clone or Reveal Attributes

When you are using the Clone or Reveal tool, you can adjust the controls under the **Clone** tab to transform the input that's being cloned or revealed. Adjust:

- **translate** - to move the source image on x and y axis.
- **rotate** - to spin the source image around a pivot point.
- **scale** - to resize the source image by adding or removing pixels. Use **center** (or **Ctrl**/**Cmd**+drag the transform jack) to position the pivot point.

- **skew X** - to skew the source image along the X axis from the pivot point. Use **center** (or **Ctrl**/**Cmd**+drag the transform jack) to position the pivot point.
- **skew Y** - to skew the source image along the Y axis from the pivot point. Use **center** (or **Ctrl**/**Cmd**+drag the transform jack) to position the pivot point.
- **skew order** - to set the order in which **skew X** and **skew Y** are applied:
  - **XY** - **Skew X** is applied before **skew Y**.
  - **YX** - **Skew Y** is applied before **skew X**.
- **round to pixel** - check this to round the **translate** amount to the nearest whole integer pixel. This can help you avoid softening when using filtering. If **source** is set to **color** on the **Stroke** tab, this control is disabled.
- **filter** - to select the appropriate filtering algorithm. For more information, see Choosing a Filtering Algorithm.
- **black outside** - When rotating or translating the clone source, a part of the image area may get cropped. To fill the cropped portion with black, check **black outside**. To fill the cropped portion by expanding the edges of the image, uncheck **black outside**.
- **time offset** - to clone or reveal pixels from a different frame. Time offset is either relative to the current frame (-1 is the frame previous to the current one) or absolute (1 is the first frame in the clip).
- **view** - to select which view you want to clone from in a stereoscopic project.

Editing Existing Stroke/Shape Timing

When editing an existing stroke/shape, you can edit the range of frames during which a stroke/shape is visible in the properties **Lifetime** tab. The lifetime of a stroke/shape/group is also visible in the **Life** column in the stroke/shape list. By default, a shape is visible on **all frames**, whereas a stroke is only visible on a **single frame**, the frame it was painted on.

Icon	Lifetime Type	Description
	all frames	Click to make a stroke/shape visible for all frames (the default).
	frame to end	Click to make a stroke/shape visible from the current frame to the last frame.

Icon	Lifetime Type	Description
	single frame	Click to make a stroke/shape visible only on the current frame.
	start to frame	Click to make a stroke/shape visible from the first frame to the current frame.
	frame range	Click to make a stroke/shape visible during a specified range of frames using the frames dialog or **from** and **to** controls.

Editing Existing Stroke/Shape Stack Order

When editing strokes/shapes after you've drawn them, you can edit their foreground to background drawing order.

In the stroke/shape list, you can drag and drop strokes/shapes to change their drawing order, and to group them under folders. For more information on using the stroke/shape list, see Working with the Stroke/Shape List.

Editing Existing Stroke/Shape Splines

To edit a stroke/shape's position, you first need to select the stroke/shape in the Viewer or the stroke/shape list. You can then modify the points that make up the stroke/shape's position.

To Add a Point to a Stroke/Shape

1. Select the **Add Points** tool  from the RotoPaint toolbar.

2. Make sure **hide points** is disabled in the tool settings.
3. In the Viewer, click on the selected stroke/shape to add a new point.

To Move a Point

1. With the **Select All** tool or **Select Points** tool active, select the stroke/shape in the Viewer or the stroke/shape list.
2. Make sure **hide points** is disabled in the tool settings.
3. In the Viewer, drag the points you want to move to a new location.
4. You can also nudge a point using the arrows on your numeric keypad. Nudging a point moves it by one pixel to the direction you choose.

Tip: If you find it difficult to select a single point, you might want to increase the **2D/3D handle size** or the **handle pick size** in the Nuke Preferences dialog **(Preferences** > **Panels** > **Viewer Handles)**.

To Move Several Points Together

1. With the **Select All** tool or **Select Points** tool active, select the stroke/shape in the Viewer or the stroke/shape list.
2. Make sure **hide points** is disabled in the tool settings.
3. In the Viewer, drag a marquee around the points (or an entire stroke/shape) that you want to move. If **hide transform handles** is disabled, a transform box appears.
4. Adjust the transform box as necessary.

Note: Moving a shape with **Select All** active moves the selected points or feather points to the new location. Scrubbing the timeline shows the shape move across the Viewer between keyframes. However, moving a shape with **Select Feather Points** active moves only the feather points to the new location, displaying the feather falloff of the selected points.

To Delete a Point

1. Right-click on the **Add Points** tool and select **Remove Points** tool .
2. Select the stroke/shape in the Viewer or the stroke/shape list.
3. Make sure **hide points**  is disabled in the tool settings.
4. In the Viewer, click the point that you want to delete.

   OR

   In the Viewer, right-click on the point that you want to delete and select **delete**.

To Delete an Entire Stroke/Shape

1. In the stroke/shape list, or in the Viewer with the **Select All** tool , click on the stroke/shape you want to delete.
2. Below the stroke/shape list, click on the minus button (-),

   OR

1. Activate the **Select All** tool . In the **Viewer**, click on the stroke/shape.
2. Right-click on the shape and select **delete** or press the **Delete**/**Backspace** key.

Copying, Cutting, and Pasting Stroke Attributes

Copying, cutting, and pasting stroke/shape attributes is achieved in a similar way to affecting position:

To Copy Stroke/Shape Attributes

1. With the **Select All** , **Select Points** , or the **Select Feather Points** tool  active, select the stroke/shape in the Viewer.
2. Make sure **hide points**  is disabled in the tool settings.

3. In the Viewer, drag a marquee around the points (or an entire stroke/shape) that you want to copy. If **hide transform handles**  is disabled, a transform box appears.
4. Right-click inside the transform box and select **copy** > **curve (attribute values).**
   The attribute values of the stroke/shape, including position and shape, are copied to the clipboard. Values only apply to a specific keyframe.

To Cut Stroke/Shape Attributes

1. With the **Select All** , **Select Points** , or the **Select Feather Points**  tool active, select the stroke/shape in the Viewer.
2. Make sure **hide points**  is disabled in the tool settings.
3. In the Viewer, drag a marquee around the points or click on the stroke/shape for the points that you want to cut. If **hide transform handles**  is disabled, a transform box appears.
4. Right-click inside the transform box and select **cut** > **curve (attribute values).**
   The attribute values of the stroke/shape, including position and shape, are cut from the Viewer and placed in the clipboard. Values only apply to a specific keyframe. Note, that the shape of the cut stroke/shape does not disappear from the Viewer.

To Paste Stroke/Shape Attributes

1. With the **Select All** , **Select Points** , or the **Select Feather Points** tool  active, select the stroke/shape in the Viewer.
2. Make sure **hide points**  is disabled in the tool settings.
3. In the Viewer, drag a marquee around the points or click on the stroke/shape for the points that you want to paste. If **hide transform handles**  is disabled, a transform box appears.
4. Right-click inside the transform box and select **paste** > **spline (attribute values)**.
   This pastes the copied stroke/shape attribute values that were on the clipboard. If you have copied another type of curve, for example, **(values)** or **(animations)**, then all options except for that one are disabled.

Point Cusping, Smoothing, Expressions, and Links

Roto and RotoPaint allow you manipulate curves, by cusping or smoothing the points on the curve, add expressions, and link controls to points.

To Cusp or Smooth Points

You can cusp points on a shape to create sharp corners, and smooth points to replace sharp corners with curved lines.

1. Select the shape you want to edit in the Viewer or the stroke/shape list.
2. Select the **Smooth Points** tool  or **Cusp Points** tool  in the RotoPaint toolbar, depending on whether you want to cusp or smooth your points.
3. Make sure **hide points**  is disabled in the tool settings.
4. In the Viewer, click on the point that you want to cusp or smooth.
5. With your point selected, you can also use the shortcut keys **Z** and **Shift+Z** to smooth and cusp it respectively.

   OR

   Right-click on the point you want to smooth or cusp, and select **smooth** or **cusp/de-smooth**.

To Add Expressions to Points

1. With the **Select All** , **Select Points** , **or Select Feather Points**  tool active, make sure **hide points**  is disabled in the tool settings.
2. Select a point in your stroke/shape. Right-click and select **add expression**.
3. Enter your expression values to the fields in the Expression dialog. Alternatively, you can **Ctrl/Cmd**+drag expression values from another node to the point in your stroke/shape.

   You can add expressions to edit your point's location or the shape of the point's feather effect. For more information on Expressions see Expressions.

To Link Other Controls to a Point's Position

You can copy a point link (that is, a point's x and y position) and use it in other nodes.

To copy a point link:

1. With the **Select All** , **Select Points** , **or Select Feather Points**  tool active, make sure **hide points**  is disabled in the tool settings.
2. Right-click on a point and select **copy** > **point link**.
3. Go to the field you want to use the link in, for example another node, right-click and select **Copy** > **Paste**.

   For example, you can copy a point link, click on a Transform node's animation menu  next to **translate**, and select **Copy** > **Paste**. An expression arrow appears between the RotoPaint node and the Transform node, and when you move the point, the Transform node's **translate** values update with your changes.



Note: When copying, cutting, or pasting from the right-click menu, numbers are displayed next to the curve, spline, or point. These numbers represent how many of each you have selected and thus, how many are copied or pasted. For example, **2 point (values)** denotes that 2 points are selected. Pasting two points into a shape with many more points replaces the lowest two points in the shape, or the highest two splines in the stroke/shape list.

Animating Strokes/Shapes

All strokes/shapes that appear on more than one frame can be animated. By default, the **autokey**  option in the RotoPaint tool settings is on, which means your changes to a stroke/shape automatically creates keyframes and animates your stroke/shape. You can also access all the curves and shapes in the Curve Editor and Dope Sheet.

To animate a stroke/shape using autokey

1. Draw a stroke/shape that appears on more than one frame. By default, the **autokey**  option in the RotoPaint tool settings is selected and a keyframe is automatically created in the first frame your

stroke/shape appears.

2. Move to a new frame.
3. With one of the Select tools, select the points or the stroke/shape you want to animate.
4. Adjust the points in your stroke/shape position or change the stroke/shape's attributes as necessary. A new keyframe is automatically set. The frame marker on the timeline turns blue to indicate the selected stroke/shape is animated.
5. Repeat steps 2, 3 and 4 for all the frames you want to set as key frames.

Tip: Note that if you are translating an entire stroke/shape using the **Transform** tab, RotoPaint also draws a track of the stroke/shape's animated position. You can, for example, use the track in another node (such as Tracker or CameraTracker) by **Cmd**/**Ctrl**+dragging the values from the translate animation button to an appropriate field in another node.

To animate strokes/shapes manually

If you choose to switch the **autokey**  function off, you can still create key frames manually. You can set key frames to the entire stroke/shape, or the stroke/shape's spline, transformation or attributes.

1. Move to the frame where you want to create a keyframe and select your stroke/shape.
2. Do one of the following:
   - If you want to create a key that is set to animate the entire stroke/shape, right-click on the stroke/shape and select **set key > all**.
   - If you want to create a key that is set to animate a position, right-click on the stroke/shape and select **set key > spline.**
   - If you want to create a key that is set to animate transformation, right-click on the stroke/shape and select **set key > transform**.
   - If you want to create a key that is set to animate attributes, right-click on the stroke/shape and select **set key > attributes**.

   If you have **autokey**  turned off, you can only adjust a point in a stroke/shape at a keyframe. In other words, in order to make changes to a point, you either have to move to an existing keyframe on the timeline, or you need to create a new keyframe first.

Viewing Spline Keyframes

You can use the **spline key** controls on the **RotoPaint** tab to view whether there are keyframes set for the spline of your stroke/shape. Do the following:

1. Select a stroke/shape on the stroke/shape list.
2. If there are spline keys set on your stroke/shape, the key boxes are highlighted blue and display where you are currently on the timeline with regard to the keyframes that are set already. You can move backwards and forwards between the keyframes using the arrow keys .
3. If you want to add or remove spline keys for the stroke/shape in a selected frame, use the **Add**  and **Delete**  buttons.

To View Keyframes on the Timeline

You can view different types of keyframes you've set, either automatically or manually, on the timeline. If you've set keyframes on the **RotoPaint** tab, these are visible on the timeline when you have the **RotoPaint** tab open in the properties panel. Similarly, if you've created transformation keyframes on the **Transform** tab, you can see those keyframes on the timeline when you have the **Transform** tab open.

- If you want to delete a key that is set to animate a position, right-click on the stroke/shape and select **delete key > spline.**
- If you want to delete a key that is set to animate a transformation, right-click on the stroke/shape and select **delete key > transform.**
- If you want to delete a key that is set to animate attributes, right-click on the stroke/shape and select **delete key > attributes**.

If no other keys are set for the selected stroke/shape, the frame marker on the timeline turns from blue to the default color.

Deleting or Rippling Keyframes

You can delete keyframes individually or by stroke/shape, or use the ripple function to change a keyframe and apply that same relative adjustment to the point across a range of frames.

To Delete a Keyframe

1. Using the **Viewer** timeline, scrub to the frame where you want to delete a keyframe.
2. In the stroke/shape list, select the stroke/shape whose key you want to delete.
3. Do one of the following:
   - If you want to delete a key that is set to animate the entire stroke/shape, right-click on the stroke/shape and select **delete key > all.**
   - If you want to delete a key that is set to animate a position, right-click on the stroke/shape and select **delete key > spline.**
   - If you want to delete a key that is set to animate a transformation, right-click on the stroke/shape and select **delete key > transform.**
   - If you want to delete a key that is set to animate attributes, right-click on the stroke/shape and select **delete key > attributes**.

If no other keys are set for the selected stroke/shape, the frame marker on the timeline turns from blue to the default color.

To Delete All Keyframes for a Stroke/Shape

You can also delete all key frames you've set for a stroke/shape at one go. Do the following:

1. Select the stroke/shape from which you want to delete key frames in the Viewer.
2. Right-click on the stroke/shape and select **no animation** > **all**, **spline**, **transform** or **attribute** depending on whether you want to delete all key frames for that stroke/shape or only key frames animating shape, transform or attributes. All key frames are removed from the stroke/shape you selected.

To Ripple Keyframes

Rippling keyframes allows you to adjust the position of a stroke/shape point on one frame and have that same relative adjustment applied to the point across all frames or a specified range of frames. This way, you can make non-animated changes to a stroke/shape which is being animated over a set of frames.

1. Activate **Select All** tool  in the RotoPaint toolbar and click the **ripple edit**  button in the RotoPaint tool settings. A red border displays around the Viewer to indicate that the ripple mode is active. In the dropdown menu next to **ripple edit** , select:
   - **all -** to ripple all frames in your sequence.
   - **from start** - to ripple frames from the first frame to the current frame.
   - **to end** - to ripple frames from current frame to the last frame.
   - **range** - to ripple a particular range of frames.
2. Select the stroke/shape you want to edit from the stroke/shape list.
3. Make the necessary changes to the stroke/shape in the Viewer.

Copying, Cutting, and Pasting Animations

Copying, cutting, and pasting can save you time, but can be destructive. For example, pasting over shapes destroys all points and keyframes of the target shape in order to replace it.

To Copy Animations

If you want to copy a shape, including all animation attributes, animated points, and associated keyframes, copying animations is the best way to do this. To copy animations for a stroke/shape:

1. With the **Select All** tool  active, select the stroke/shape in the Viewer.
2. Right-click on the stroke/shape and select **copy** > **curve (spline-key animations)**.
   This copies the current selected animation and key frame information to the clipboard.

To copy only curve attribute information for animated shapes:

1. With the **Select All** tool  active, select the stroke/shape in the Viewer.
2. Right-click on the stroke/shape and select **copy** > **curve (attribute animations)**.
   This copies only the selected attributes for the animated shape to the clipboard.

To copy only the point information for animated shapes:

1. With the **Select All** tool  active, select the stroke/shape in the Viewer. Make sure **hide points**  is disabled in the tool settings.
2. Right-click on the point or points and select **copy** > **points (animations)**.
   This copies only the selected animated points and keyframes to the clipboard.

To Cut Animations

If you want to move a stroke/shape and replace another with all the same animated points and keyframes, cutting the shape is a better option than simply copying it. To cut animations for a stroke/shape:

1. With the **Select All** tool  active, select the stroke/shape in the Viewer.
2. Right-click on the stroke/shape and select **cut** > **curve (spline-key animations)**.
   The shape is cut from the Viewer and placed in the clipboard. The transform box still appears on the Viewer and allows you to amend the shape, until you click on another shape or elsewhere in the Viewer.

To cut only animation attributes for a stroke/shape:

1. With the **Select All** tool  active, select the stroke/shape in the Viewer.
2. Right-click on the stroke/shape and select **cut** > **curve (attribute animations)**.
   This cuts the animation attributes for the selected stroke/shape and placed it them in the clipboard.

To cut only the animated points for strokes/shapes:

1. With the **Select All** tool  active, select the animated point(s) of a stroke/shape in the Viewer that you want to cut. Make sure **hide points**  is disabled in the tool settings.
2. Right-click on the point(s) and select **cut** > **point (animation)**.
   This cuts only the animated points and keyframes and places them in the keyboard.

To Paste Animations

If you want to paste copied or cut animated shape information, consider whether the shape you are pasting over has values that need to be saved. Pasting animated points, spline-keys, and attributes over another shape destroys all points and keyframes of the second shape in order to replace it. To paste animations for a stroke/shape:

1. With the **Select All** tool  active, select the stroke/shape in the Viewer that you want to paste the animated shape over. If there are not the same number of points on the shape as on the copied shape, pasting the animated shape may not paste as expected.

2. Right-click on the stroke/shape and select **paste** > **spline (animations)**.
   This pastes the animation and keyframe information for the copied or cut stroke/shape over the previously selected shape. This shape may need to be moved away from the copied shape, in order to see both.

To paste only animation attributes for a stroke/shape:

1. With the **Select All** tool  active, select the stroke/shape in the Viewer that you want to paste the animated attributes over.
2. Right-click on the stroke/shape and select **paste** > **spline (attribute animations)**.
   This pastes the animation attributes for the copied or cut stroke/shape over the previously selected shape's attributes.

To paste only the animated points for strokes/shapes:

1. With the **Select All** tool  active, select the point(s) of a stroke/shape in the Viewer that you want to paste the animated points over. Make sure **hide points**  is disabled in the tool settings.
2. Right-click on the point(s) and select **paste** > **point (animation)**.
   This pastes only the animated points and keyframes from the clipboard.

Adding Motion Blur

After animating your shapes, you can apply motion blur to individual shapes in the shape list or to the all shapes in the current node using the controls in the properties **Motion Blur** tab.

Adding Blur to Shapes

Shape Blur determines the exposure for each moving shape and blends the resulting blurred shapes. This may be more efficient than the **global** motion blur since each shape will only be blended once.



Note: Shape motion blur may result in artifacts when shapes blur over the same region.

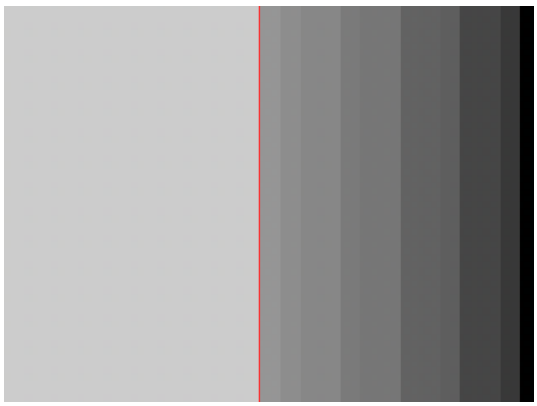
To apply **shape** motion blur:

1. Select the target shapes in the shape list or in the Viewer.
2. In the properties panel, select the **Motion Blur** tab and make sure that **shape** mode is enabled.

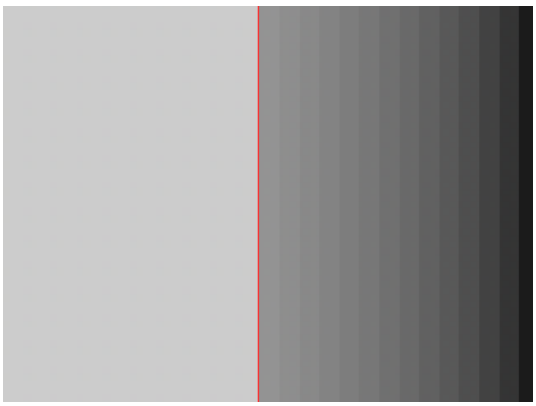
3. Check the **on** box next to the **motionblur** field to apply motion blur to your shape. If you don't want any motion blur on your shape, leave the **on** box unchecked.

> **Tip:** You can also toggle motion blur on and off in the shape list by clicking in the motion blur column for each shape .

4. In the **Shape Blur** > **motionblur** field, enter the sampling rate. This affects the number of times the input is sampled over the shutter time. The higher the rate, the smoother the result. In many cases, a value of 1.0 is enough.

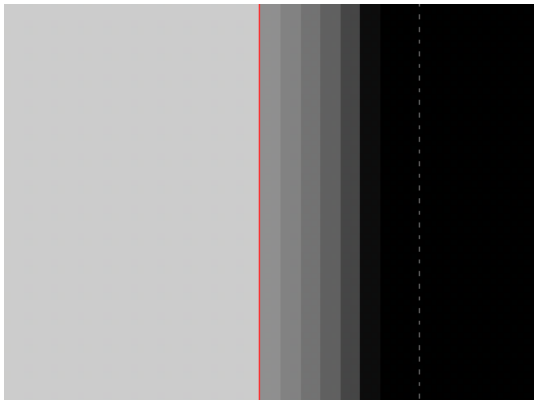


A motionblur value of 0.5 produces less samples.



A higher value, in this case 2.0, produces a smoother blur.

5. In the **shutter** field, enter the number of frames the shutter stays open when motion blurring. For example, a value of 0.5 would correspond to half a frame. Increasing the value produces more blur, and decreasing the value less.



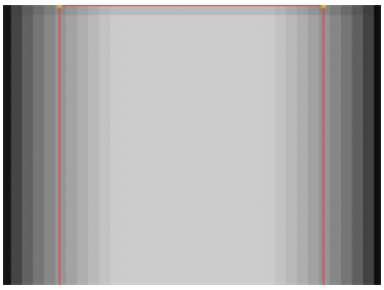
A shutter value of 0.2 produces less blur.



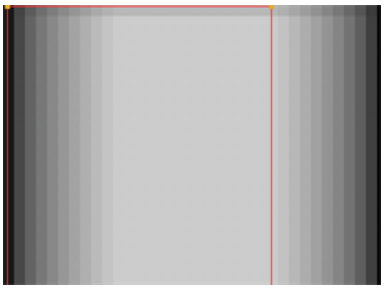
A higher value, in this case 1.0, produces more blur.

6. Adjust the shutter offset using the **shutter offset** dropdown menu. The different options control when the shutter opens and closes in relation to the current frame value. Select:

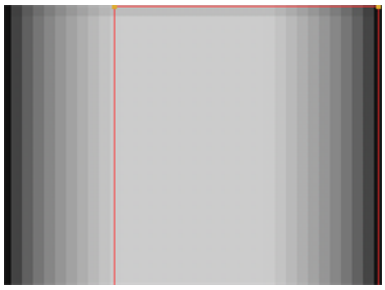
- **centered** - to center the shutter around the current frame. For example, if you set the shutter value to 1 and your current frame is 30, the shutter stays open from frame 29.5 to 30.5.
- **start** - to open the shutter at the current frame. For example, if you set the shutter value to 1 and your current frame is 30, the shutter stays open from frame 30 to 31.
- **end** - to close the shutter at the current frame. For example, if you set the shutter value to 1 and your current frame is 30, the shutter stays open from frame 29 to 30.



Centering the shutter around the current frame.



Opening the shutter at the current frame.



Closing the shutter at the current frame.

- **custom** - to open the shutter at the time you specify. In the field next to the dropdown menu, enter a value (in frames) you want to add to the current frame. To open the shutter before the current frame, enter a negative value. For example, a value of - 0.5 would open the shutter half a frame before the current frame.

Adding Global Motion Blur

Global Blur correctly accounts for interaction between motion blurred shapes. This may be more expensive than the **shape** motion blur since it may blend each shape for every sample, in the same way as TimeBlur. See Applying a TimeBlur Filter for more information.

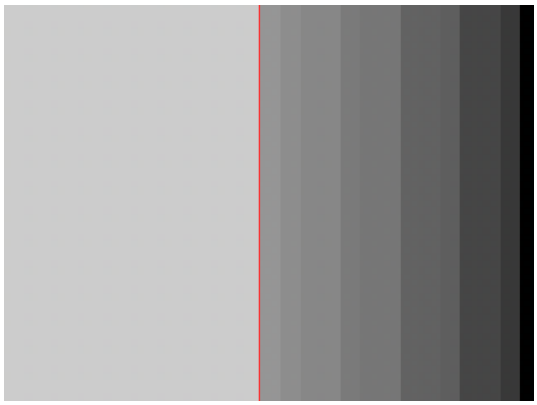
Global motion blur requires that shutter and sampling parameters are the same for all shapes and has been optimized for consecutive shapes with the same properties using the **over** blend mode.



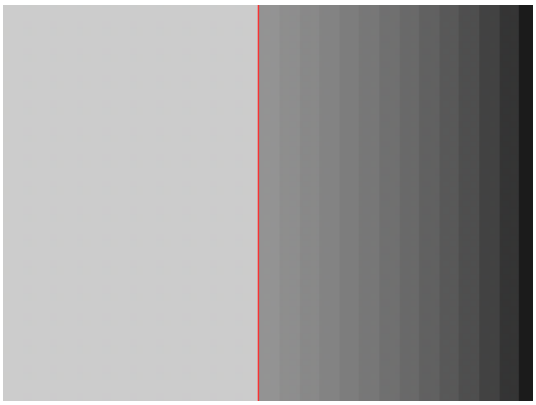
Note: Global motion blur overrides per-shape motion blur, applying your settings to all shapes in the current node's shape list.

To add **global** motion blur:

1. In the properties panel, select the **Motion Blur** tab and make sure that **global** mode is enabled.
2. In the **Global Blur** > **motionblur** field, enter the sampling rate. This affects the number of times the input is sampled over the shutter time. The higher the rate, the smoother the result. In many cases, a value of 1.0 is enough.

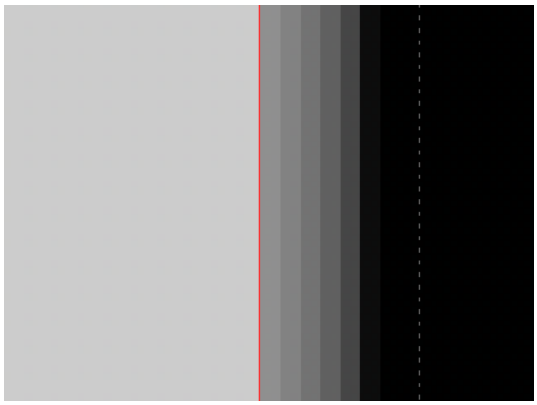


A motionblur value of 0.5 produces less samples.



A higher value, in this case 2.0, produces a smoother blur.

3. In the **shutter** field, enter the number of frames the shutter stays open when motion blurring. For example, a value of 0.5 would correspond to half a frame. Increasing the value produces more blur, and decreasing the value less.

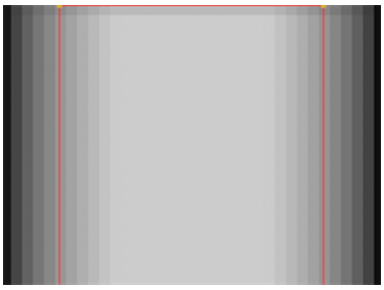


A shutter value of 0.2 produces less blur.

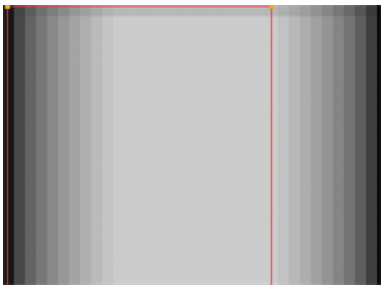


A higher value, in this case 1.0, produces more blur.

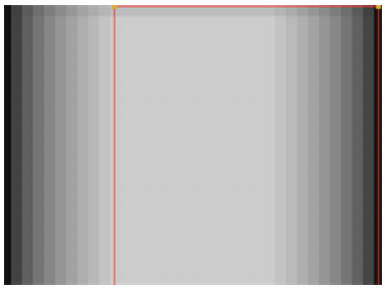
4. Adjust the shutter offset using the **shutter offset** dropdown menu. The different options control when the shutter opens and closes in relation to the current frame value. Select:
   - **centered** - to center the shutter around the current frame. For example, if you set the shutter value to 1 and your current frame is 30, the shutter stays open from frame 29.5 to 30.5.
   - **start** - to open the shutter at the current frame. For example, if you set the shutter value to 1 and your current frame is 30, the shutter stays open from frame 30 to 31.
   - **end** - to close the shutter at the current frame. For example, if you set the shutter value to 1 and your current frame is 30, the shutter stays open from frame 29 to 30.



Centering the shutter around the current frame.



Opening the shutter at the current frame.



Closing the shutter at the current frame.

- **custom** - to open the shutter at the time you specify. In the field next to the dropdown menu, enter a value (in frames) you want to add to the current frame. To open the shutter before the current frame, enter a negative value. For example, a value of - 0.5 would open the shutter half a frame before the current frame.

Viewing Points in the Curve Editor and the Dope Sheet

You can edit your stroke/shape points, their feather points and tangents in the Curve Editor and Dope Sheet. To read more about using the Curve Editor and Dope Sheet, see Animating Parameters.

1. Select your stroke/shape, and right-click on a point.
2. Select **curve editor...** and select:
   - **points** - to add your point to the Curve Editor
   - **points**+**feathers** - to add your point and its feather point to the Curve Editor
   - **points**+**tangents** - to add your point and its tangent to the Curve Editor
   - **all** - to add your point, feather point, and tangent to the Curve Editor.
3. Viewing your points in the Dope Sheet is very similar, just right-click on your point and select one of the same options under **dope sheet...**

Copying, Pasting, and Cutting Stroke Positions

After creating a stroke/shape, you can copy, paste, and cut its position to use the same shape in other strokes/shapes. Copying, cutting, or pasting values applies to only one keyframe.

To Copy Stroke/Shape Positions

You can copy the stroke/shape position values for an entire selected shape. This enables you to use the same values for another shape in the Viewer. Unlike cutting stroke/shape values, described below, copying position values does not delete any keys set. To copy a stroke/shape:

1. With the **Select All** tool  active, select the stroke/shape in the Viewer that you want to copy.
2. Right-click on the stroke/shape and select **copy** > **curve (spline-key values)**.
   This copies the position and point values into the clipboard.

To Cut Stroke/Shape Positions

You can cut the stroke/shape position values for an entire selected shape. This enables you to use the same values to replace another shape in the Viewer. To cut a stroke/shape:

1. With the **Select All** tool  active, select the stroke/shape in the Viewer that you want to copy.
2. Right-click on the stroke/shape and select **cut** > **curve (spline-key values)**.
   This cuts the position and point values and places it into the clipboard. Cutting the position deletes the keyframe, or, if there is no keyframe, deletes the points or spline.

Note: To cut stroke/shape position values, you must be cutting the stroke/shape when it is on a keyframe, unless it has no keyframe. You cannot cut an animated stroke/shape not currently on a keyframe.

To Paste Stroke/Shape Positions

You can paste copied or cut stroke/shape position values from the clipboard onto another stroke/shape. This enables you to create a copy, or replacement, shape in the Viewer. To paste a copied or cut shape:

1. With the **Select All** tool  active, select the stroke/shape in the Viewer that you want to copy.
2. Right-click on a selected stroke/shape and select **paste** > **spline (values)**.

   This pastes the copied or cut position and point values from the clipboard onto another shape. In the case of copied shapes, the new shape may be created over the original. Simply select and drag to move it away from the original shape. Spline key values must have the same number of points between the copied or cut stroke/shape and the stroke/shape you are replacing.

Copying, Pasting, and Cutting Point Positions

You can copy point position values in a stroke/shape you have selected. This enables you to use the same values for another point in a stroke/shape. Unlike cutting point values, described below, copying position values does not delete any keys set.



Note: When copying, cutting, or pasting from the right-click menu, numbers are displayed next to the curve, spline, or point. These numbers represent how many of each you have selected and thus, how many are copied or pasted. For example, **2 point (values)** denotes that 2 points are selected. Pasting two points into a shape with many more points replaces the lowest two points in the shape, or the highest two splines in the stroke/shape list.

To Copy Point Values

1. In the Viewer, scrub to the frame that contains the stroke/shape whose point position you want to copy.
2. Activate the **Select Points** tool .
3. Make sure **hide points**  is disabled in the tool settings.
4. Right-click on the point a stroke/shape and select **copy > point (values).**

Nuke copies the positions of the selected point to the clipboard. Any keys set to animate these positions are not affected.

Pasting Point Positions

You can paste any position you've copied/cut from another point to a selected point in a stroke/shape. If you have **autokey** turned on, this also sets a keyframe at the current frame to animate this point. The attributes of the strokes or any keys set to animate the attributes are not affected.

To Paste Point Values

1. In the **Viewer**, scrub to the frame that contains the stroke/shape to which you want to paste the positions on the clipboard.
2. Activate the **Select Points** tool .
3. Make sure **hide points**  is disabled in the tool settings.
4. Right-click on the point and select **paste** > **point (values).**

   Nuke pastes the positions (but not any attributes) on the clipboard to the selected point and, if you have autokey turned on, sets the current frame as a keyframe.



Note: If you have copied more than one point into the clipboard but have only selected one point to paste into, **paste** > **point (values)** only displays one point and may not look as you expect. To paste both points, ensure that you have two point selected (or however many is needed for your sample).

Cutting Point Positions

You can cut the position values of a point in a stroke/shape. Cut also copies the positions at the current frame to the clipboard. The point attributes or any keys set to animate the attributes are not affected.

To Cut Point Values

1. In the Viewer, scrub to the frame that contains the stroke/shape whose point position you want to cut.

2. Activate the **Select Points** tool .
3. Make sure **hide points**  is disabled in the tool settings.
4. Right-click on the point and select **cut** > **point (values).**

   Nuke deletes any keys set to animate the positions of the selected point, and copies the position to the clipboard. Cutting the position deletes the keyframe, or, if there is no keyframe, deletes the points or spline.



Note: To cut stroke/shape point values, you must be cutting the stroke/shape when it is on a keyframe, unless it has no keyframe. You cannot cut an animated stroke/shape not currently on a keyframe.

RotoPaint and Stereoscopic Projects

If you need to use RotoPaint in a multi-view or stereoscopic project, you may want to draw your strokes/shapes or apply your edits to one view only (for example, the left view but not the right), or create a stroke/shape in one view and have it automatically generated for the other in the correct position.

For more information on multi-view or stereoscopic projects in Nuke, see Stereoscopic Scripts.

Selecting the View to Draw On

When you are using the RotoPaint node to draw a new stroke/shape in a stereoscopic or multi-view project, you can toggle the **singleview** checkbox in the RotoPaint tool settings to draw your stroke/shape on one view only or multiple views. For existing strokes/shapes/groups, you can use the **view** control to select the view the stroke/shape is visible. If you're working on a stereoscopic project, the view you're using for a particular stroke, shape, or group is also visible on the **View** column in the stroke/shape list.

Selecting the View to Clone From

When cloning, you can select the view to use as the clone source. Go to the **Clone** tab and set the **view** dropdown menu to the view you want to clone from. To use the view currently displayed in the Viewer, select **current**.

Reproducing Strokes/Shapes in Other Views

To create a stroke/shape on one view and have it automatically generated for the other:

1. Make sure there is a disparity field upstream from the image sequence you want to paint on. If the image sequence is an .exr file, the disparity field can be included in its channels. Otherwise, you can use a Shuffle node or Ocula's O_DisparityGenerator plug-in to add it in the data stream.
2. In the RotoPaint properties, check all the views in the **view** dropdown menu. Display the view you want to paint on in the Viewer.
3. Draw a stroke/shape in the Viewer.
4. Select the stroke/shape in the stroke/shape list in the RotoPaint node properties.
5. Right-click the stroke/shape and select either:
   - **Correlate points** - to use the disparity at each point of the stroke/shape and translate each point to the corresponding position in the other view.
   - **Correlate average** - to take the disparity at each point, calculate the average disparity of the stroke/shape and then translate it to the corresponding position in the other view.

   The **Correlate** dialog displays.
6. In the **Correlate** dialog, select how to correlate the views in the **correlate from** dropdown. For example, if your stroke/shape was in the correct position in the left view but not the right, you'd set **correlate from** to **left to right**.

   This adds the disparity vectors in the map to the current values, creating the corresponding stroke/shape for the other view.
7. In the Viewer, switch between the two views to compare the original and the correlated strokes/shapes.
8. If you want to adjust the stroke/shape further, you need to adjust both views independently. Adjustments you make to one view are not automatically generated for the other.

Editing Strokes/Shapes in One View Only

Splitting a view off allows you to edit the stroke/shape data in that view only, without affecting any other views that exist in your project settings.

To Edit Stroke/Shape Points

1. Display the view you want to edit in the Viewer.
2. Right-click on a spline or point and select **split off [view]**, for example **[split off left]**.

   Any changes you now make to the stroke/shape points are only applied to the view you chose to split off and are displaying in the Viewer.

   Views that have not been split off can still be edited together. For example, if you have views called **left**, **middle**, and **right**, and choose to split off the left view, any changes to the middle view are also applied to the right view and vice versa.
3. To unsplit a view, right-click on a spline or point again and select **unsplit [view]**.

   The view is unsplit, and all changes you made after splitting it off are lost.

Note: To see splines for paint strokes in the Viewer, you need to activate **show paint stroke splines**  in the RotoPaint tool settings.

Tip: You can also split shapes off by right-clicking on them in the stroke/shape list and selecting **Split off [view]**, for example **Split off left**.

To Edit Stroke/Shape Attributes

1. Display the view you want to edit in the Viewer.
2. In the RotoPaint controls, click the Views menu button  next to the control you want to edit and select **Split off [view name]**. For example, to edit opacity in a view called left, select **Split off left** from the Views dropdown menu next to **opacity**.

Any changes you now make to opacity are only applied to the view you chose to split off and are displaying in the Viewer.

Views that have not been split off can still be edited together. For example, if you have views called **left**, **middle**, and **right**, and choose to split off the left view, any changes to the middle view are also applied to the right view and vice versa.

You can also click on the small arrow on the left side of the control to see the values for each view.

3. To unsplit a view, click the Views menu button  and select **Unsplit [view]**. For example, to unsplit a view called left, you'd select **Unsplit left**.

   The view is unsplit, and all changes you made after splitting it off are lost.

Applying a Stereo Offset

The **stereo offset** control in the RotoPaint properties allows you to move the selected stroke/shape on the x and y axes. This is an extra transform that is applied after all other transforms. Typically, you would position the stroke/shape correctly in the hero view, then split **stereo offset**, and drag the stroke/shape to its correct location in any other views. Note that you can also press **Shift** while dragging to constrain the movement to the x or y axis only.

Stereo offset can be useful, for example, if you have a stroke/shape that is correctly positioned in one view and you want to move it to its correct location in another view, but can't use the **translate** control on the **Transform** tab because that's being driven by Tracker data.

Where Are the Bezier and Paint Nodes?

The pre-6.0 Bezier and Paint nodes have been deprecated in favor of the RotoPaint node. With RotoPaint you have the ability to add more strokes and shapes, group them, and so on. However, Bezier and Paint are still in the application for backwards-compatibility with old scripts. Should you find the need (or just feel nostalgic), you can create the Paint and Bezier nodes in a couple of easy ways:

- Press **X** on the Node Graph, make sure **Tcl** is checked in the dialog that opens, enter **Paint** or **Bezier** in the **Command** field, and click **OK**. Your node appears in the Node Graph.

- You can add a Paint or a Bezier node to your Toolbar menu with a statement in your **menu.py** file like the following:
  - For Bezier:

```
tb = nuke.toolbar("Nodes")
tb.addCommand("Draw/Bezier", "nuke.createNode(\"Bezier\")",
icon="Bezier.png")
```

  - For Paint:

```
tb = nuke.toolbar("Nodes")
tb.addCommand("Draw/Paint", "nuke.createNode(\"Paint\")", icon="Paint.png")
```

Tracking and Stabilizing

Nuke's 2D Tracker allows you to extract animation data from the position, rotation, and size of a pattern. You can then apply the data directly to transform or match-move another element. Or you can invert the data values and apply them to the original element to stabilize the image.

Before Tracking

Before you track, it's important to play back the image several times. This will help you identify the best features for the process, as well as any problems with motion blur, occlusions, or features moving out of frame.

For some images, you may need to apply filters or color-correction to boost the visibility of features before you attempt to track them. Because of the procedural nature of Nuke, you can disable these extra nodes after you get a successful track, or simply reconnect the Tracker node at the appropriate place to apply the transform.

Quick Start

Here's a quick overview of the workflow:

1. Connect a Tracker node to the image you want to track. See Connecting the Tracker Node.
2. Place the required number of track anchors on features in the image. See Adding Track Anchors.
3. Use Automatic Tracking for simple tracks or, for greater accuracy or difficult tracks, use Keyframe Tracking.
4. Apply your tracking data depending on requirements. See Applying Tracking Data.

Connecting the Tracker Node

To connect the Tracker node:

1. Read in and select the clip you want to track.
2. Click **Transform** > **Tracker**.

3. Click **Image** > **Viewer** to insert a Viewer node and connect it to the Tracker node.



Adding Track Anchors

You can add as many track anchors as required, depending on which transformational components you wish to track. For example, when tracking in areas of distortion or noise, it's a good idea to add a lot of tracking anchors and then average the results to get a better overall track.

1. Enable the **fast-add** button and click in the Viewer to add tracking anchors, or click **add track** on the **Tracker** tab in the properties panel to create the required number of anchors.

Note: Holding **Ctrl**/**Cmd**+**Alt** and clicking in the Viewer also enables fast-add mode.

New anchors always appear in the center of the current Viewer. You'll also notice an anchor zoom window in the top left of the Viewer. This allows you to accurately position tracking anchors without zooming the entire Viewer.



2. Temporarily disable tracks using the checkbox on the left of the panel or remove tracks by selecting them in the **Tracks** list and clicking **delete tracks**.

Positioning Track Anchors

Each track anchor consists of a **pattern** and **search** area:

- **Pattern** - the target pixels tracked across multiple frames. This pattern should be as distinct as possible from the surrounding frame and remain visible throughout the majority of the sequence.
- **Search** - the area of the current frame in which the pattern is likely to be found in the next frame. Large search areas can affect performance.



1. Drag the crosshair in the center of the anchor over the pattern to be tracked or manually adjust the **track_x** and **track_y** fields in the **Tracks** list.

   In the example, the corner of a building is directly under the crosshair.

Note: You can enable **Settings** > **snap to markers** to assist you when placing anchors.

Note: Whether you select **blobs** or **corners**, a guide displays in the Viewer as you drag the anchor, highlighting likely patterns.



Release the mouse to snap the anchor to the marker.

2. Adjust the pattern and search areas as required using the following methods:
   - Click and drag the boundaries to adjust on a single axis. For example, dragging a vertical boundary adjusts the area on the x axis.
   - Click and drag the corners of the boundaries to adjust on both axes. Hold **Shift**, click and drag to scale the area.

Note: Hold **Cmd/Ctrl** to adjust boundaries asymmetrically, that is, irrespective of the anchor position.

3. Reset the pattern and search areas by clicking  in the Viewer tools.

Tracking Preferences and Viewer Tools

Before tracking, use the Tracker's properties panel and Viewer tools to control Viewer output and determine tracking behavior.

1. Click the **Settings** tab to display the **General** controls.
2. Select the channels in the image to track using the **track channels** dropdown.
   If this is set to anything other than **all** or **none**, use the checkboxes to select the required channels.
3. Select the **pre-track filter** type to apply before pattern comparison using the dropdown:
   - **none** - no filter is applied.

- **adjust contrast** - the default filter, stretches the image contrast to better suit the tracking algorithm. This is the recommended setting and shouldn't need changing in most circumstances.
- **median** - helps to remove noise from the pattern.

4. Check **adjust for luminance changes** to force Tracker to calculate some extra pre-filtering to compensate for changes in brightness. This option slows the tracking process and can reduce the accuracy of tracks, so only enable this control if there are known changes in brightness.

> **Note:** Enabling **adjust for luminance changes** can occasionally produce better tracks on shots with no differences in luminance, particularly on jittery shots where sub-pixel accuracy is vitally important.

5. **Clamp super-white, sub-zero footage** is enabled by default, restricting the pixel values within the pattern between 0-1.

   If you want to track the sequence using its full dynamic range, disable this control and increase **max_error** to a value large enough to cover the range. For example, a sequence with pixel values of 40 might require a similar **max_error** value.
6. Enable **show error on track paths** to color code tracks showing their pattern matching error values, green for a good match through to red for a poor match.

> **Note:** The Viewer toolbar  button also toggles this control on and off.

7. Enable **hide progress bar** to free up Viewer real-estate while tracking.
8. Enable **snap to markers** to assist you when placing tracking anchors. See Positioning Track Anchors for more information.
9. Set when the zoom window is displayed using the **show zoom window** dropdown:
   - **always** - the window is always displayed.
   - **on track move** - the window is only displayed when a track anchor is moved.
   - **when tracking** - the window is only displayed during tracking.
   - **when tracking or track change** - the window is displayed during tracking and when a tracking anchor is moved.
   - **never** - the window is never displayed.
10. Set the zoom widget's size and magnification using the **zoom size/mag.** dropdowns. You can select **custom** and enter values manually for greater flexibility.
11. A filter is applied to the zoom window on playback by default, but you can enable the filter permanently, or disable it, using the **zoom window filter** dropdown.



Note: The filter applied is the same as that selected on the **Transform** tab, and can produce a more visually stable track. It can make track positioning more difficult, however.

12. Proceed to Automatic vs. Keyframe Tracking to determine which tracking method suits your needs.

Automatic vs. Keyframe Tracking

After placing track anchors and setting your preferences, you're ready to calculate your tracks. Nuke's Tracker provides two calculation methods:

- Automatic Tracking - ideal for simple tracks, there are no extra preparation steps once you've set your tracking anchors and preferences.
- Keyframe Tracking - a more involved method, requiring you to set keyframes on the sequence in order to calculate tracks. Keyframe tracking may be the better option for more complicated patterns and movement.

Automatic Tracking

Calculating tracks automatically uses the tools above the Viewer to control the direction and range of frames to track. Tracking backwards can produce a better track than going forwards if the pattern is clearly visible later in the clip. By default, Auto-Tracking grabs the reference pattern on the first frame of the sequence, from within the pattern anchor, using this pattern throughout the track for comparison between frames.

To help avoid clutter in the Viewer, you can enable or disable the Tracker overlay by right-clicking in the Viewer and selecting **Overlay**, or by pressing **Q** to toggle between the available states:

- overlay off
- overlay on
- overlay on, no animation path

Calculating Auto-Tracks

1. In the Tracker properties panel, select each track you wish to calculate or click **select all** to use all tracks.

2. For each track, select the type of movement the track is intended to output: translation, rotation, or scaling. For example, tracking a feature across the sequence and toward the camera may involve translation and scale.

Note: These controls deal with output from tracking data using the **Transform** controls and are not the same as the **Settings** > **warp type** control, which deals with pattern recognition.

3. Using the tool bar above the Viewer, click either the frame backward (X) or forward (C) buttons  to move to the previous or next frame. Move through a few frames in this manner to ensure that all enabled track anchors are "sticking" to their patterns.

   If a particular track anchor doesn't stick, experiment with a different position.

4. Once all track anchors stick, click the track backward (Z) or track forward (V) buttons  to analyze the whole sequence.
5. To track only a certain frame range, use the range buttons  to enter the required frames.
6. Click stop (Esc) , to cease tracking in either direction.

Tip: When calculating multiple tracks simultaneously, you may find that some tracks stick accurately to the pattern, while others require resetting and reanalysis. When you're happy with a given track, deselect it in the **Tracks** list. This protects it from recalculation, and lets you experiment with better placement for the wayward tracks.

See Troubleshooting Auto-Tracks for help with troublesome tracks.

Troubleshooting Auto-Tracks

No matter how sophisticated tracking becomes, some sequences are inevitably going to cause problems. There are a number of pre-tracking checks you can perform to assist Auto-Tracking:

- Play through the sequence before placing your tracking anchors
- Look for features that are consistent throughout the majority of the sequence
- Avoid occluded features where possible - see Dealing with Occlusions.

You can also adjust the way patterns are tracked and how often they are re-sampled, or grabbed, using the **Settings** tab and **Auto-Tracking** controls:

1. Try adjusting the **max iterations**, **epsilon**, and **max_error** controls to improve the track accuracy:
   - **max iterations** - the maximum number of iterations before the tracking algorithm stops searching for features.

- **epsilon/resolution** - the error level at which Tracker is assumed to have found the feature - no further search for a better match is performed. Higher values may result in a faster but less accurate track.
- **max_error** - the error level at which Tracker stops searching for features.

2. In the Auto-Tracking sub-menu, enable **predict track** to use the animation path to predict where the pattern may appear in the next frame.



Note: If the track fails when prediction is enabled, click the **Clear Forward**  button, or re-tracking follows the same erroneous path.

3. Set the type of movement Tracker should expect in the pattern using the **warp type** dropdown:
   - **Translate**
   - **Translate/Rotate**
   - **Translate/Scale**
   - **Translate/Rotate/Scale**
   - **Affine**

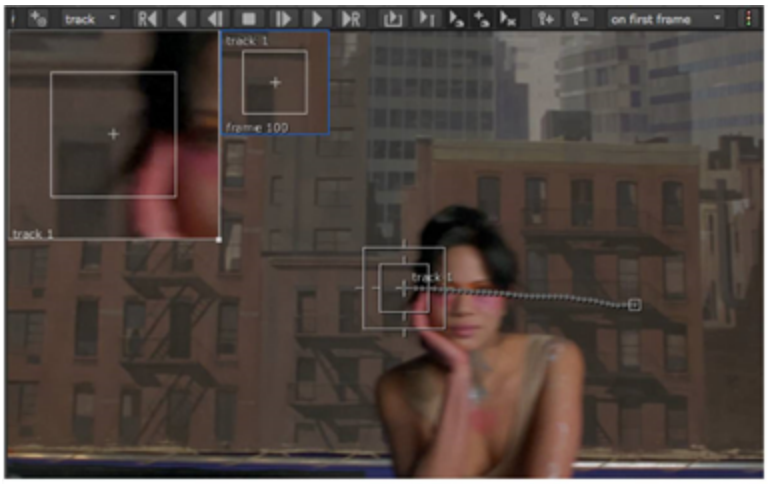
   Translation only is the easiest to calculate, but can lose the pattern if it rotates or scales. Affine can be used to attempt to preserve straight lines and relative distances, compensating for sheering.
4. Try adjusting the pattern grabbing behavior, when or how often a new pattern should be grabbed from the sequence:
   - **on first frame** - the comparison pattern is grabbed on the first frame from with the pattern anchor. You might select this option if the feature translates but doesn't rotate, scale, or sheer.
   - **every frame** - the comparison pattern is grabbed at every frame in the sequence. This option takes longer to process, but can produce a smoother track.
   - **every n frames** - allows you to set the frame interval between pattern grabs using the **every n frames** control.
   - **if error above** - the comparison pattern is grabbed when the error value is greater than that specified by the **when error >** control. Setting this control to a low value grabs the pattern more often.
   - **if error below** - the comparison pattern is grabbed when the error value is less than that specified by the **when error <** control.
   - **custom** - this option enables all the pattern grab behavior controls, allowing you to fine-tune when the comparison pattern in grabbed through out the sequence.
5. Enable **when tracking is stopped** to cause Tracker to re-grab the pattern at the current frame when tracking stops.
6. Enable **when tracker is moved** to cause Tracker to re-grab the pattern at the current frame when a tracking anchor is moved.

Dealing with Occlusions

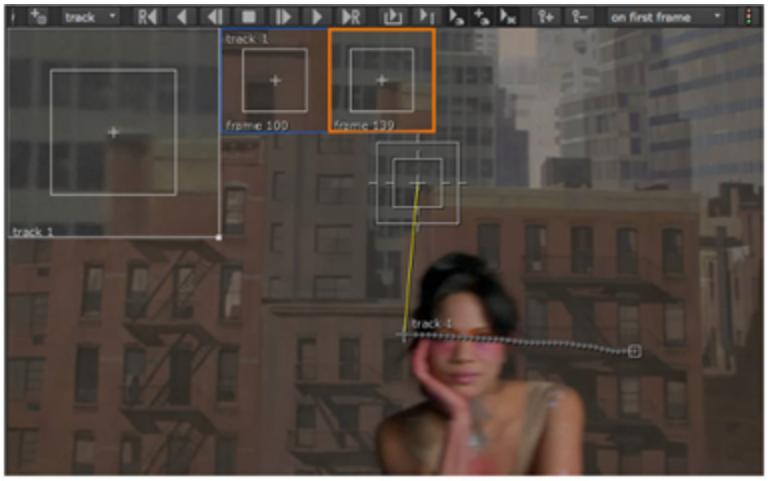
Tracker's offset capability allows you track an obscured feature using the relative position of another feature, providing that the distance between the two points remains constant.

1. Track the pattern normally until the occlusion causes Tracker to fail.

   The zoom window helps to identify the failure point.



2. Play though the sequence to identify a likely offset point - a pattern that remains equidistant from the original pattern grab.
3. Hold down **Ctrl**/**Cmd** and drag the tracking anchor to the offset position.



   The offset amount is recorded in the Tracks list and highlighted in yellow in the Viewer.

4. Continue tracking as normal by clicking the track backward (Z) or forward (V) button. 

   Tracker combines the two tracks into a single continuous track.

5. Use the clear backward and forward buttons  to clear poor keyframes. Click clear all to remove all selected tracks and keyframes, excluding manually placed keyframes.

 **Note:** You can reset tracking anchor pattern and search areas by clicking .

Keyframe Tracking

Calculating tracks using keyframes can be the better option for more complex patterns and movement. It requires a little more work to set up, but can produce more reliable tracks.

Unlike auto-tracks, keyframe tracking compares the current pattern anchor to the patterns of the two nearest keyframes.

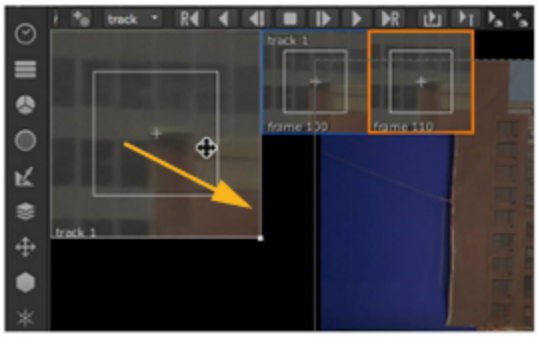
To help avoid clutter in the Viewer, you can enable or disable the Tracker overlay by right-clicking in the Viewer and selecting **Overlay**, or by pressing **Q** to toggle between the available states:

- overlay off
- overlay on
- overlay on, no animation path

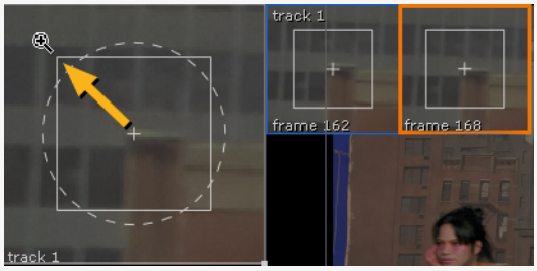
Calculating Keyframe Tracks

To calculate keyframe tracks:

1. In the Tracker properties panel, select each track you wish to calculate in the Tracks list or click **select all**.
2. For each track, select the type of movement the track is intended to output: translation, rotation, or scaling. For example, tracking a feature across the sequence and toward the camera may involve translation and scale.
3. Scrub through the sequence a few frames and adjust the position of the tracking anchor by dragging the anchor to the location of the pattern. You can use the zoom window to fine-tune your positioning. Continue on through the sequence as required.



Tip: You can change the magnification of zoom windows by holding **Shift** and dragging the magnifying glass cursor away from the center of the window.



At each frame, a new keyframe window is added to the right of the zoom window. The keyframe closest to the current playhead frame is highlighted in orange.

It's a good idea to place more keyframes around areas of complexity or greater movement and fewer on straight forward translation. Generally speaking, a greater number of keyframes produces a better track, but at the expense of processing time.

4. When you're satisfied with your keyframes, make sure all your tracks are selected in the **Tracks** list and then click  to track all keyframes.

Tip: When calculating multiple tracks simultaneously, you may find that some tracks stick accurately to the pattern, while others require resetting and reanalysis. When you're happy with a given track, deselect it in the **Tracks** list. This protects it from recalculation, and lets you experiment with better placement for the wayward tracks.

5. You can also force the selected tracks to recalculate between the two nearest keyframes by clicking  in the Viewer toolbar.

   See Troubleshooting Keyframe Tracks for help with troublesome tracks.

Importing Tracking Data

You can import tracking data from third-party software using a plain text file containing three values in each line, frame, x, and y coordinates. Tracker can read files with values separated by spaces, commas, or colons and any blank lines or lines starting with **#**, **;**, and **/** are ignored. For example, importing a file containing:

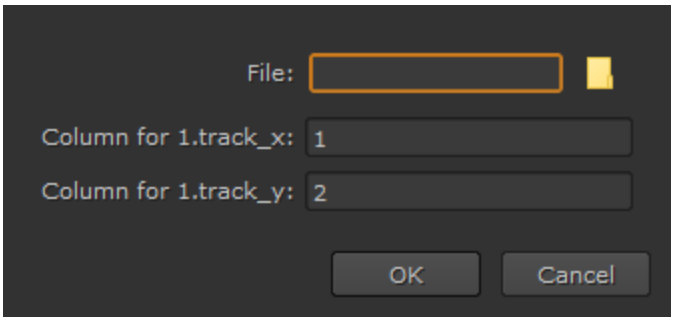
```
# Tracking Data
1,1,1
10,240,240
50,1000,700
90,1800,1200
100,2048,1556
```

Produces five keyframes at frames 1, 10, 50, 90, and 100 with the relevant x,y coordinates.

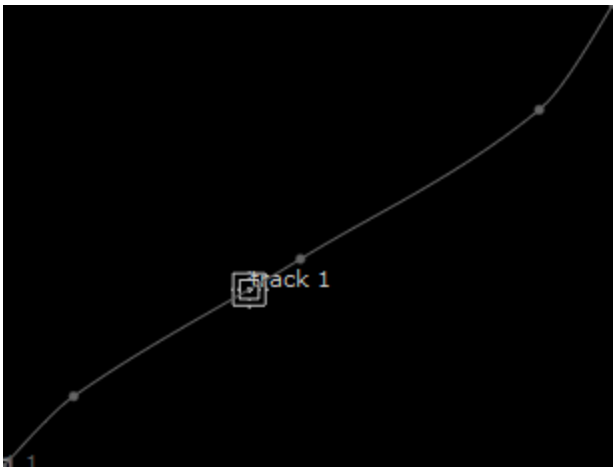
Tip: If you use a particular file format on a regular basis, you might want to create you own importer to parse the **.txt** file. You can use the **<install_dir>/plugins/import_discreet.tcl** file as a guide on how to do this.

To import tracking data:

1. Add a tracking anchor as described in Adding Track Anchors.
   This adds a track to the tracks table in the **Properties** panel.
2. Right-click the new track and select **File** > **Import Time+value Ascii**.
   The **Import discreet** dialog displays.



3. Enter the file path and file name in the **File** field.
4. Set the required column containing the x and y coordinate data.
5. Click **OK** to import the track.
   The points from the file are converted into keyframes and displayed in the Viewer. The track between keyframes is interpolated as normal. The five keyframe example file described earlier produces a track similar to the following image.



Troubleshooting Keyframe Tracks

Again, even with preset keyframes, some sequences are inevitably going to cause problems. There are a number of pre-tracking checks you can perform to assist Auto-Tracking:

- Play through the sequence before placing your tracking anchors,
- Look for features that are consistent throughout the majority of the sequence,
- Avoid occluded features where possible - see Dealing with Occlusions.

Keyframe tracking won't generally stop when a problem is encountered. Tracker attempts to continue using the next keyframe as a reference, which is why placing a lot of keyframes around problem areas is recommended.

Tip: Tracking areas of distortion or noise can produce unreliable results due to the movement of the pixels in the pattern matching box. One way to deal with this is to seed multiple tracks in and around the problem area and then average the resulting tracks together, producing a single more reliable track, by clicking **average tracks** in the properties panel.

1. First, turn on the color-coded error indicator  by clicking the traffic light Viewer tool.
   Each keyframe is colored on a sliding scale from green (good match) to red (poor match).



Bear in mind that a red keyframe doesn't necessarily mean that the tracking result is poor, only that Tracker couldn't reliably match the pattern from one keyframe to the next.

2. Move the tracking anchor to the first of the poor frames, just about the center of the image in the example.
3. Tracker defaults to adding and deleting keyframes automatically when certain conditions are met, but you can toggle these features on and off in the Properties **Tracker** > **Settings** tab **Keyframe Tracking** controls:
   - **re-track when keyframe is moved** - disable this control if you plan to manually position multiple keyframes before re-tracking.
   - **re-track on creation of a new keyframe** - disable this control when placing multiple new keyframes, such as when the track encounters problem areas.
   - **create new key when track is moved** - you could disable this control if you wanted to use the zoom window to examine the sequence more closely without triggering a re-track.
   - **auto-tracks delete keyframes** - when this control is enabled, custom keyframes are deleted during automatic re-tracking.
4. Using the zoom window, drag the anchor to the correct location of the grabbed pattern.

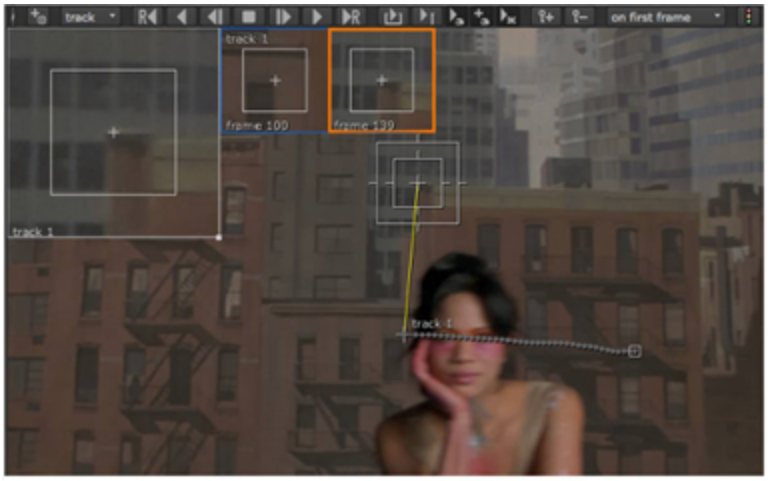
   Tracker attempts to recalculate the track by including your correction.
5. Advance the playhead to the next poor keyframe and repeat until the track is complete.

Dealing with Occlusions

Tracker's offset capability also applies to keyframe tracking, allowing you to track an obscured feature using the relative position of another feature, providing that the distance between the two points remains constant.

1. Place keyframes normally until you reach the occlusion.

2. Play though the sequence to identify a likely offset point - a pattern that remains equidistant from the last keyframe.
3. Hold down **Ctrl/Cmd** and drag the tracking anchor to the offset position.



The offset amount is recorded in the Tracks list and highlighted in yellow in the Viewer.

4. Continue tracking as normal by clicking the track backward (Z) or forward (V) button. 

Tracker combines the two tracks into a single continuous track.

5. Use the clear backward and forward buttons  to clear poor keyframes. Click clear all to remove all selected tracks and keyframes, excluding manually placed keyframes.



Note: You can reset tracking anchor pattern and search areas by clicking .

Applying Tracking Data

You can apply your tracking data to the input image using either the Tracker node's controls, linking expressions, or other Nuke nodes.

Applying Tracking Data Using Tracker's Controls

The simplest way to apply tracking data to the input image or other nodes is to use the controls of the Tracker node itself. Here, we look at using these controls to stabilize, match-move, and remove or apply

jitter.

Stabilizing Elements

The Tracker node's controls let you remove motion, such as unwanted camera shake, from the node's input clip. You can use a single track to stabilize horizontal and vertical motion across the 2D plane, or two or more tracks to eliminate rotation and scale.

1. Create the track(s) you want to use for stabilizing the footage:
   - If you're using a single track, ensure **T** is selected in the Tracks list, so that Tracker only calculates translation.

     On the **Transform** tab, select **transform** > **stabilize 1-pt** to lock the **Filter** to **Keys**. This filter produces the best results when using a single track.
   - If you're using more than one track in the Tracks list, select the transformations that you want Tracker to calculate when stabilizing the image, **T**ranslation, **R**otation, and/or **S**cale.

     On the **Transform** tab, select **transform** > **stabilize**.
2. Set the **reference frame** if you don't want to use the first frame as the transform control frame.
3. Use the **smooth** controls to average frames together to smooth the transforms applied.

   For example, if you're stabilizing using more than one track, you might average frames together for translation and rotation by entering the number of frames in the **t** and **r** fields.
4. Select the required filter. See Choosing a Filtering Algorithm for more information.

   Nuke stabilizes the footage, locking its elements to the same position within the composite.



Note: You can export the transform information to a linked or baked Transform node by selecting the required type from the **Export** dropdown and clicking **create**.

Match-moving Elements

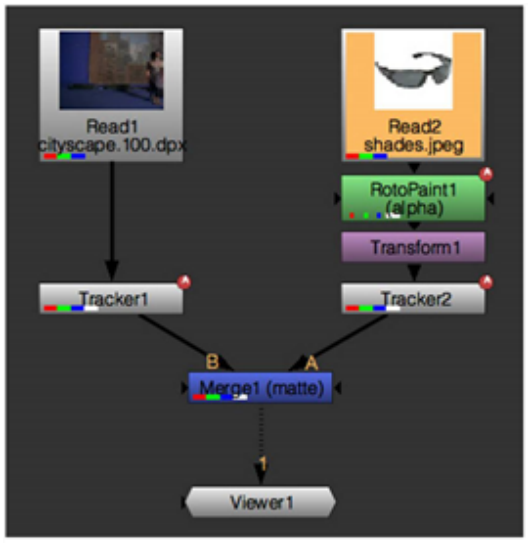
You can use the Tracker node's controls to apply the tracked motion to another image, that is, to match-move an image.

1. Use a Tracker node to track the feature you intend to match.
2. Copy the Tracker node and paste it after the footage you want to match-move.
3. In the second Tracker node's controls, go to the **Transform** tab.
4. From the **transform** dropdown menu, select **match-move**.
5. Set the **reference frame** if you don't want to use the first frame as the transform control frame.
6. Use the **smooth** controls to average frames together to smooth the transforms applied.

For example, if you're using more than one track, you might average frames together for translation and rotation by entering the number of frames in the **t** and **r** fields.

7. Select the required filter. See Choosing a Filtering Algorithm for more information.

   Nuke applies the tracked movement to the footage you want to match-move. A simple script might appear as follows, where Tracker2 is a copy of Tracker1:



A simple match-move script.

Note: You can export the transform information to a linked or baked Transform node by selecting the required type from the **Export** dropdown and clicking **create**.

Removing or Adding Jitter

Tracker can be used to remove high frequency jitter from a sequence, exaggerate existing jitter, or add jitter to another sequence for a consistent look.

To remove jitter:

1. Create the tracks you want to use for jitter removal.
2. In the Tracker node's controls, go to the **Transform** tab.
3. From the **transform** dropdown menu, select **remove jitter**.
4. Set the **reference frame** if you don't want to use the first frame as the transform control frame.
5. Use the **jitter period** to average together frames, adjusting the jitter to achieve the required stability.
6. Use the **smooth** controls to average frames together to smooth the transforms applied.

For example, if you're removing jitter using more than one track, you might average frames together for translation and rotation by entering the number of frames in the **t** and **r** fields.

7. Select the required filter. See Choosing a Filtering Algorithm for more information.
8. Nuke removes jitter from the footage, locking its elements to the same position within the composite.

To exaggerate or add jitter:

1. Create the tracks you want to use for the required jitter operation.
2. In the Tracker node's controls, go to the **Transform** tab.
3. From the **transform** dropdown menu, select **add jitter**.
4. Set the **reference frame** if you don't want to use the first frame as the transform control frame.
5. Use the **jitter period** to average together frames, adjusting the jitter to achieve the required instability.
6. Use the **smooth** controls to average frames together to smooth the transforms applied.

   For example, if you're adding jitter using more than one track, you might average frames together for translation and rotation by entering the number of frames in the **t** and **r** fields.
7. Select the required filter. See Choosing a Filtering Algorithm for more information.

   Nuke exaggerates the tracked jitter in the footage.



Note: If you want to transfer jitter to another sequence, copy the Tracker node and paste it after the footage to which you want to add jitter, then follow steps 3-7 above.

Applying Tracking Data Using Linking Expressions

Nuke's CornerPin2D and Stabilize2D nodes are specifically designed to receive tracking data by linking expressions, but you can apply tracking data in this manner to virtually any Nuke node. For example, you might animate a Bezier or a B-spline shape with tracking data by entering linking expressions into the RotoPaint node's transformation parameters. You can also apply tracking data to individual points.

This section explains the basic procedure for applying tracking data to any node via linking expressions.

Creating Linking Expressions

The Tracker node's Tracker panel displays data related to the position of each track anchor over time (**tracks' x** and **y** fields). This is the data that you most typically apply to other nodes.

To Drag and Drop Tracking Data:

1. Display both the Tracker parameters and the parameters to which you wish to apply the tracking data (the destination control - for example, a RotoPaint node's **translate** control).

2. Select the required track from the **Tracks** list. Only one track can be linked per control.
3. **Ctrl**+drag (**Cmd**+drag on a Mac) from the source control animation button  to the destination control animation button.

When you release, the destination control will turn blue, indicating an expression has been applied. In this case, the drag and drop action has created a linking expression resembling the following example:
`Tracker1.tracks.1.track_x`

Tip: You can also apply tracking (or other transform) data to individual RotoPaint, SplineWarp, or GridWarp points (this is sometimes called per vertex tracking). To do so, **Ctrl**/**Cmd**+drag and drop the track's animation button on a RotoPaint, SplineWarp or GridWarp point in the Viewer.

You can add other components to this linking expression as necessary. For example, you might add a spatial offset to the linking expression by subtracting out the initial frame's tracking values, in which case the final expression would resemble the following:
`Tracker1.tracks.1.track_x-Tracker1.tracks.1.track_x(1)`

See Expressions for more information. Once you enter the linking expression, the destination parameter turns blue.

To Link Animated Parameters with a Tracker Node

You can also link controls with the Tracker node if you use the **Link to** option in the Animation menu. For example, to link the **translate** control of the Roto node with a Tracker node, do the following:

1. Create the Tracker node you want to link to.
2. Go to Tracker's Transform tab and enable **live-link transform**.
   This control enables Tracker to update dynamically as the expression link changes.
3. On the **Transform** tab of the Roto node's properties panel, click on the **translate** animation menu.
4. Select **Link to** > **Tracker linking dialog**.
5. Select the Tracker node you want to use in the **tracker node** dropdown and in the **link to** dropdown, select whether you want to link to the **position** of the track, the **translate** values of the track, or treat the translate value as an offset.
6. Select which tracks you want to use by checking the **track** boxes. The Expression field updates with the appropriate expression syntax. If you select more than one track, the tracks are averaged, for example:
   `(Tracker1.tracks.1.track_x + Tracker1.tracks.2.track_x)/2`
7. Then click **OK**, and your linking is done.
   Your Bezier shape's translate value now changes when the Tracker value is changed.

Transforming Masks with Tracking Data

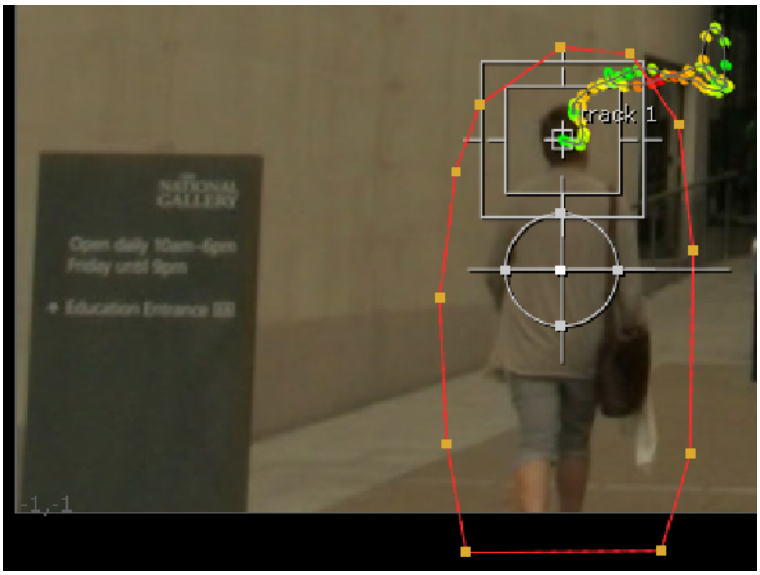
Creating animated masks using Roto and keyframes can be a very time-consuming process, but Nuke's Tracker node can do some of the initial work for you, especially with garbage mattes.

Once you have some solid tracking data, you can drive a roto shape without keyframing individual points:

1. Track a feature in the area you intend mask. In the example, the figure's head serves as the driving point for the matte.



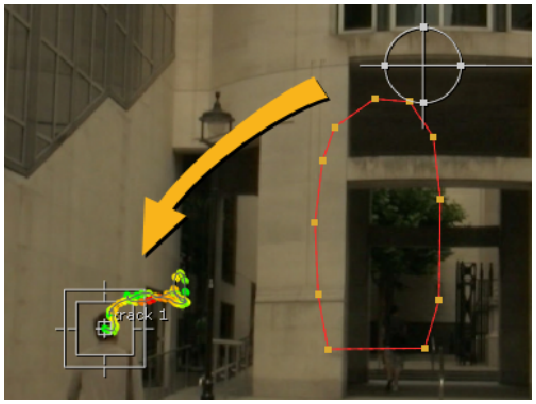
2. Add a Roto node to the script (keyboard shortcut **O**) and draw the shape you intend to drive with the tracking data. In this case, we don't need to be too accurate as we're creating a garbage matte.



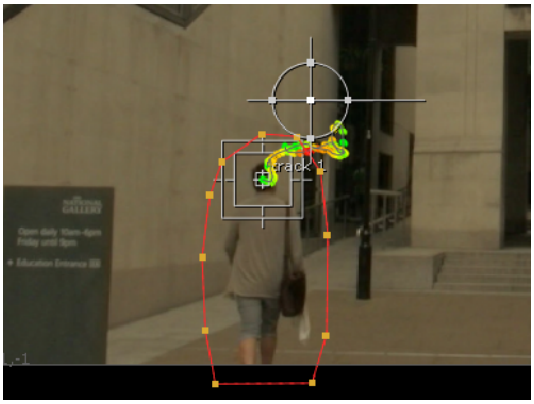
3. In the Roto properties panel, click the **Transform** tab and select the matte in the shapes list.

4. Right-click the **translate** control's animation icon and then select **Link to** > **Tracker 1** > **track 1**.

   The Tracker's **track_z** and **track_y** keyframes are copied into the Roto's **translate** control, applying the same translate *and* offset to the matte shape.
5. To compensate for this, select the **Root** item in the shape list and reposition the Roto shape correctly using the transform handle in the Viewer.



The matte with offset.



The same matte after repositioning.

6. Scrub the playhead to see the matte following the tracked path.

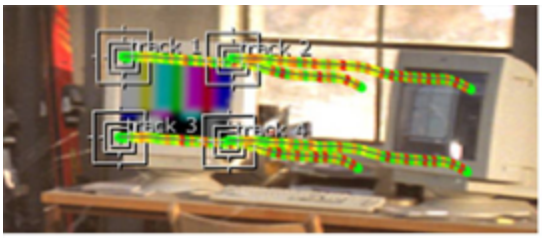
Using the CornerPin2D Node

The CornerPin2D node is designed to map the four corners of an image sequence to positions derived from tracking data. In practice, this node lets you replace any four-cornered feature with another image sequence. For example, suppose you needed to replace the monitor image in the fast-panning shot shown below.



Fast-panning shot requires four-corner tracking.

First, use the Tracker to calculate four separate tracks, one for each corner of the feature.



Generating the four tracks.

Next, attach a CornerPin2D node to the image sequence you want to use as the replacement for the feature, and apply the tracking data. This remaps the image sequence's corners to the correct positions over time. You can create the node manually, or by using Tracker's **Export** dropdown.

The final step is to layer the result over the original element.



The composited image.

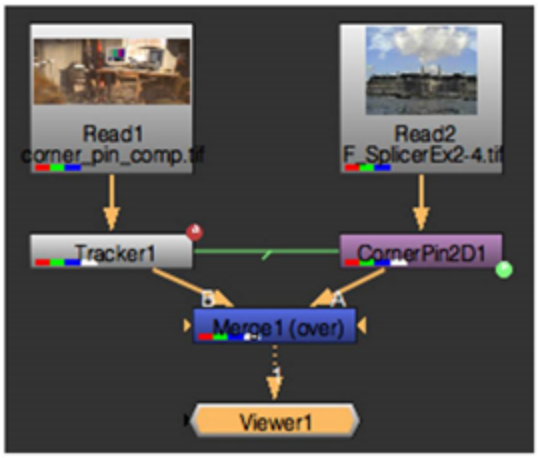
The steps below summarize the use of Tracker's **Export CornerPin2D** workflow.

To Use the CornerPin2D Node

1. Generate four tracks, one per corner, on the feature requiring replacement.
2. Use the current frame or **reference frame** field to specify the frame to use as the starting point. You can also decide whether the CornerPin2D node is expression linked or baked using the **Export** dropdown:
   - CornerPin2D (use current frame)
   - CornerPin2D (use transform ref frame)
   - CornerPin2D (use current frame, baked)
   - CornerPin2D (use transform ref frame, baked)
3. Click **create** to add the CornerPin2D node to the script.
4. Attach the image or sequence to replace the feature tracked to the input of the CornerPin2D node.
5. If necessary, select a different filtering algorithm from the **filter** dropdown menu. See Choosing a Filtering Algorithm for more information.
6. When filtering with **Keys**, **Simon**, or **Rifmen** filters, you may see a haloing effect caused by the pixel sharpening these filters employ. If necessary, check **clamp** to correct this problem.

7. In most cases, you will keep **black outside** checked. This renders black pixels outside the image boundary, making it easier to layer the element over another. (If you uncheck this parameter, the outside area is filled with the outermost pixels of the image sequence.)
8. The final step is to layer the result over the original element.

   A simple script might appear as follows:



Warping with GridWarpTracker

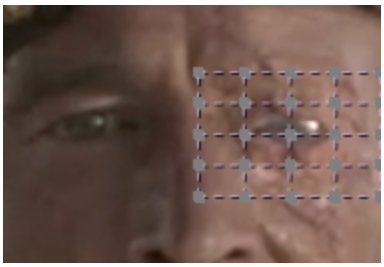
GridWarpTracker allows you to warp and morph using custom grid shapes driven by tracking data, rather than being constrained to rigid transformations using PlanarTracker. If you have a NukeX or Nuke Studio license, you can also use SmartVectors to drive the grids, rather than the usual tracking workflow. The Tracking Using SmartVectors workflow is significantly easier and faster than the vanilla Nuke Tracking Using the Tracker Node workflow.

Note: You can still use GridWarpTracker if you don't have a NukeX or Nuke Studio license, but the tracking process uses the Tracker node rather than SmartVectors.

The **From** and **To** grids allow you to add and copy tracking data between grids so that you can make adjustments without losing your original data and without having to create a backup version of the node. If you have a NukeX or Nuke Studio license, you can add adjustment grids to modify your **From** and **To** grid shapes without altering the original grid data.



Warp source.



Warp grid.



Warp result.

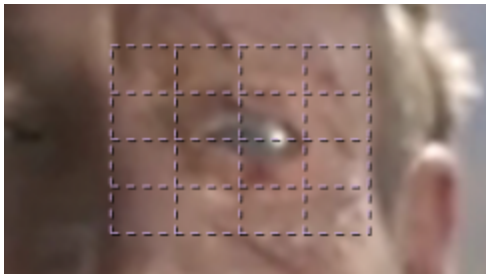
Note: Images illustrating GridWarpTracker courtesy of Little Dragon Studios. All rights reserved in the United States and/or other countries.

Drawing Grids

The **From** and **To** grids control the area and extent of the warp or morph, so drawing accurate grids around the feature you intend to warp can improve the result. You can link the **From** and **To** grid before drawing your grid or copy points between grids using the copy and paste buttons.

The grids covers the entire input image by default, but in most cases you'll want to confine the warp to an area of interest in the image.





1. Enable the Draw Boundary tool and then click and drag in the Viewer to describe the shape of the grid.

Tip: If you want to see your grid in the context of the whole image, set the **Background** control to **Src**, otherwise the rest of the image is masked out.



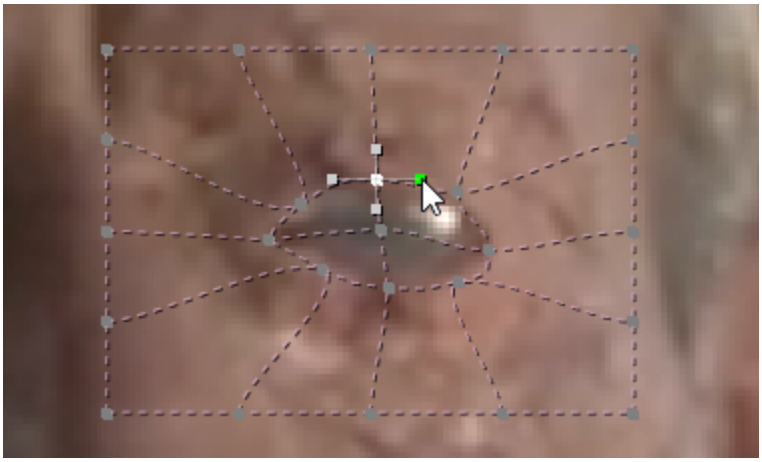
2. You can then use the Insert and Delete tools to add and remove columns and rows until you have the required grid.

Tip: If you prefer, you can use the subdivide buttons to quickly add columns and rows to multiple selections.

3. Link the **From** and **To** grids before adjusting the points by clicking the link icon in the node **Properties** panel. This ensures that both grids have the same starting point after tracking, allowing you to visualize the warp more easily.

   If you want to keep the grids separate for now, you can always copy the keyframes from one to the other using the and paste buttons.

4. Enable the Edit tool to start adjusting the points in the grid to the features in the image. The tangent handles associated with each point can help you to fit the grid to the feature more accurately.



Generally, more accurate grids produce better results, so it's worth taking some time to match the grid to the feature you're warping.

5. You can control how points appear in the Viewer using the label and overlay buttons:
   -  - enable to display a transform handle overlay on all selected points, allowing you to transform them all at once.
   -  - enable to display **x,y** labels on points selected in the grid.
   -  - enable to display the grid a point belongs to, its color, and locked state when you hover over it.
6. Proceed to Tracking Using the Tracker Node if you don't have access to NukeX or Nuke Studio's SmartVector node,

   OR

   Tracking Using SmartVectors if you do have SmartVector data.



Note: SmartVector data can be written out to **.exr** files and used in vanilla Nuke, you don't have to calculate the vectors in context. See Generating Motion Vectors for more information.

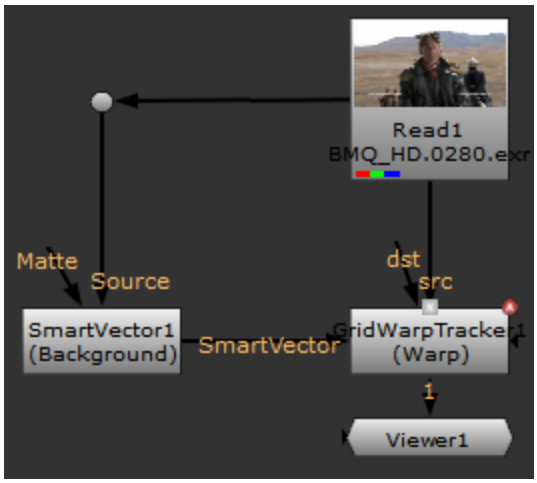
Tracking Using SmartVectors

NukeX and Nuke Studio's SmartVector node produces faster, more accurate tracking data by calculating pixel motion between frames. You can connect a SmartVector node directly to the GridWarpTracker node or write motion vectors to the **.exr** format, which are then used to drive the warp to reduce overheads later on.

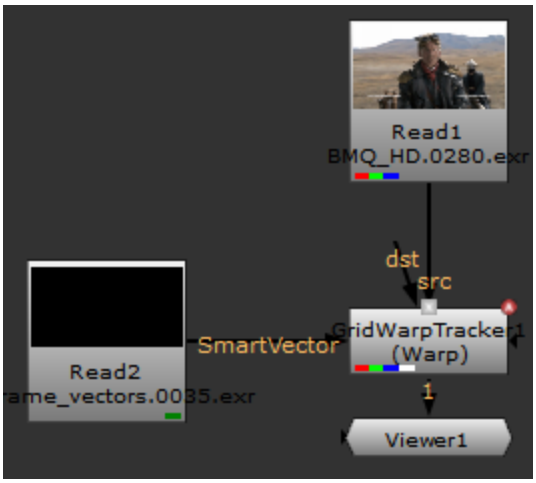
SmartVector data is either calculated in context or baked into the **.exr** format and piped into the GridWarpTracker node to drive the grids.



Note: SmartVector data can be written out to **.exr** files and used in vanilla Nuke, you don't have to calculate the vectors in context. See Generating Motion Vectors for more information.



Calculating vectors in context.



Pre-baked vectors.

Calculating the vectors in context allows you to adjust the SmartVector settings as you go, but using baked vectors is much faster.

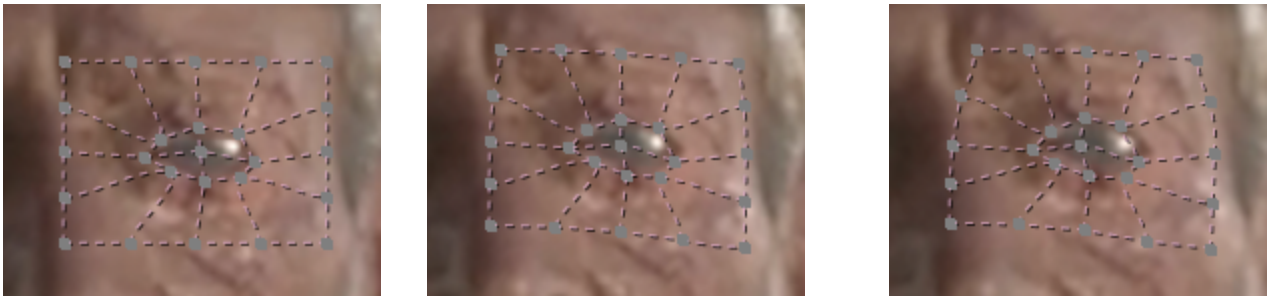
To track the grids using SmartVectors:

1. Click the Track Backward or Track Forward buttons above the Viewer to analyze the whole sequence,

   use the Range buttons and enter the required frames to track only a certain frame range, or

   use the Frame Backward or Frame Forward buttons to track a single frame in either direction.
2. Click Stop, to cease tracking in either direction.
3. Click the Clear Backward or Clear Forward buttons to remove all keyframes in the specified direction from the current frame.
4. Click the Clear All button to remove all keyframes.

   The grids follow the specified feature automatically, using the vector information to move the individual points in both grids.



On simple shots, the grid should be accurate enough to use for warping or morphing. See Warping Shots with GridWarpTracker for more information.

For more complex shots with rapid movement or rotation, tracking might slip causing the grid to deform. Have a look at Correcting Grid Deformation for more information.

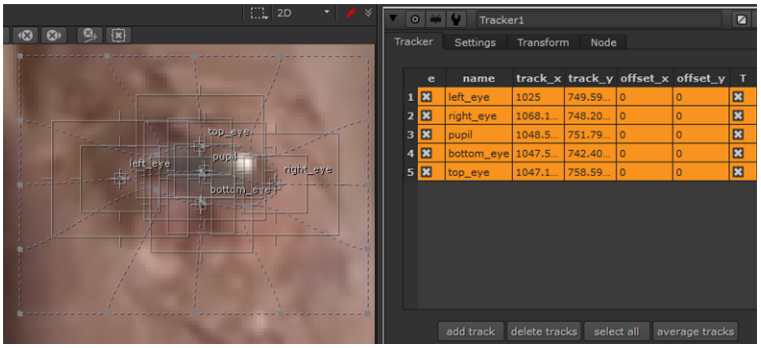
Tracking Using the Tracker Node

If you don't have access to SmartVector data or just prefer to use a familiar workflow, Nuke's Tracker node can drive grid shapes in GridWarpTracker. Tracker set up is slightly more painful, however. For simple shots, you might only need three or four tracks per grid, but creating tracking data for each vertices in the grid produces the best results.

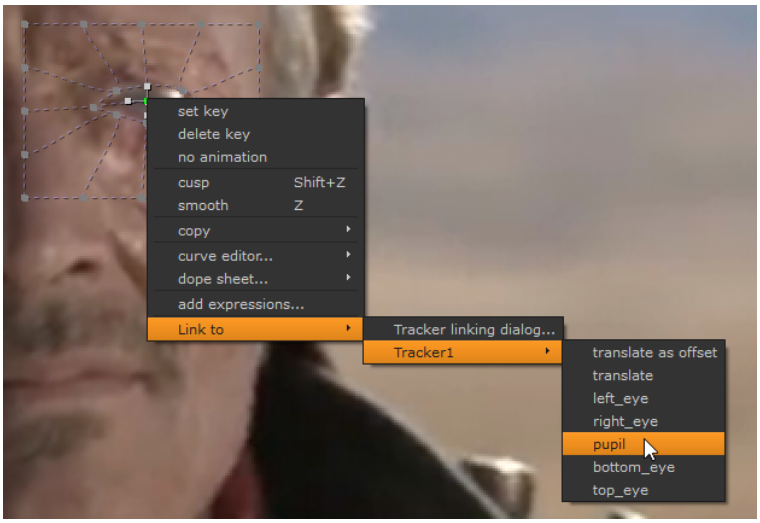
To track the grids using Tracker:

1. Add tracking anchors for the points in the grid you want to track. See Adding Track Anchors for more information on best practice.
2. For each track in the **Properties** panel, select the type of movement the track is intended to output: translation (T), rotation (R), or scaling (S). For example, tracking a feature across the sequence and toward the camera may involve translation and scale.
3. Track the selected features as described in Automatic Tracking or Keyframe Tracking.

   It's a good idea to rename the tracks in the **Properties** panel with names that reflect their position in the image. For example, **left_eye**, **right_eye**, **pupil**, **bottom_eye**, **top_eye**, etc. This can help you assign the correct track to individual grid vertices.

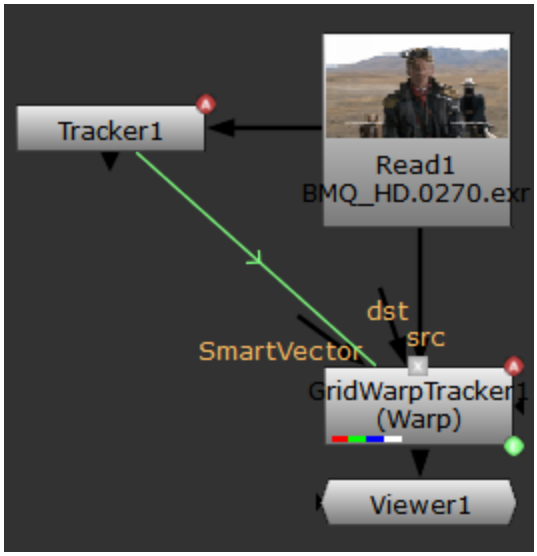


   Tracking can fail on some features, particularly in shots with movement or rotation. Have a look at Troubleshooting Auto-Tracks or Troubleshooting Keyframe Tracks for more information.
4. After tracking your points successfully, right-click a vertex on the grid that corresponds to a tracking anchor and select **Link to** > **Tracker1** and then the name of the track. For example, **pupil** as shown in the example.

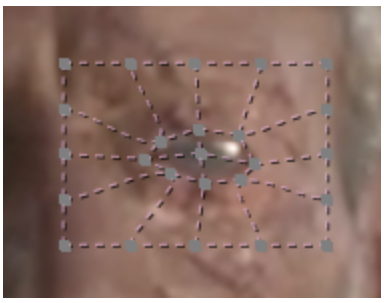
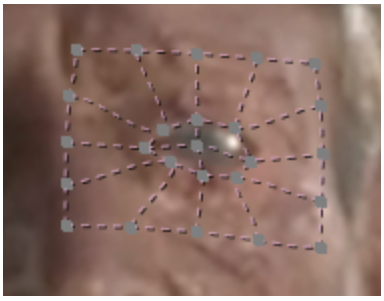
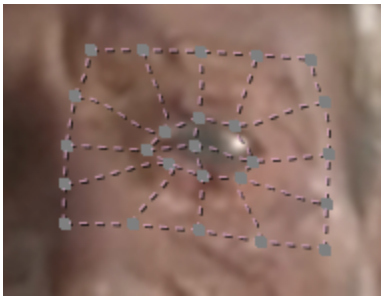


5. Repeat the linking process for all your tracking anchors and grid vertices.

   An expression link, represented by the green arrow in the Node Graph, is created between the tracks and grid vertices so that the Tracker's **translate** keyframes drive the grids automatically.



6. Play through the sequence to check that the grids follow the specified feature automatically, using the tracking information to move the individual points in both grids.

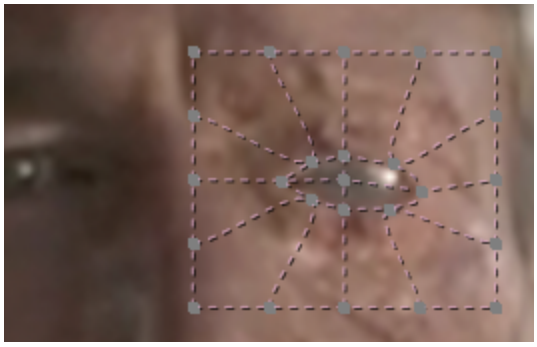
On simple shots, the grid should be accurate enough to use for warping or morphing. See Warping Shots with GridWarpTracker for more information.

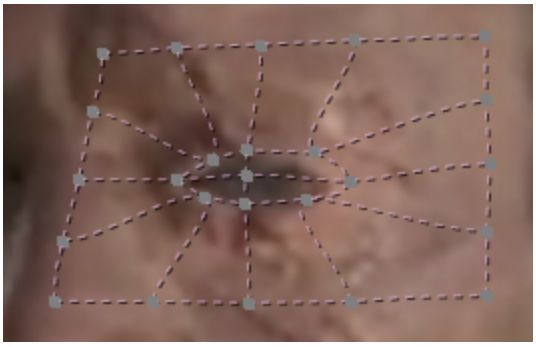
For more complex shots with rapid movement or rotation, tracking might slip causing the grid to deform. Have a look at Correcting Grid Deformation for more information.

Correcting Grid Deformation

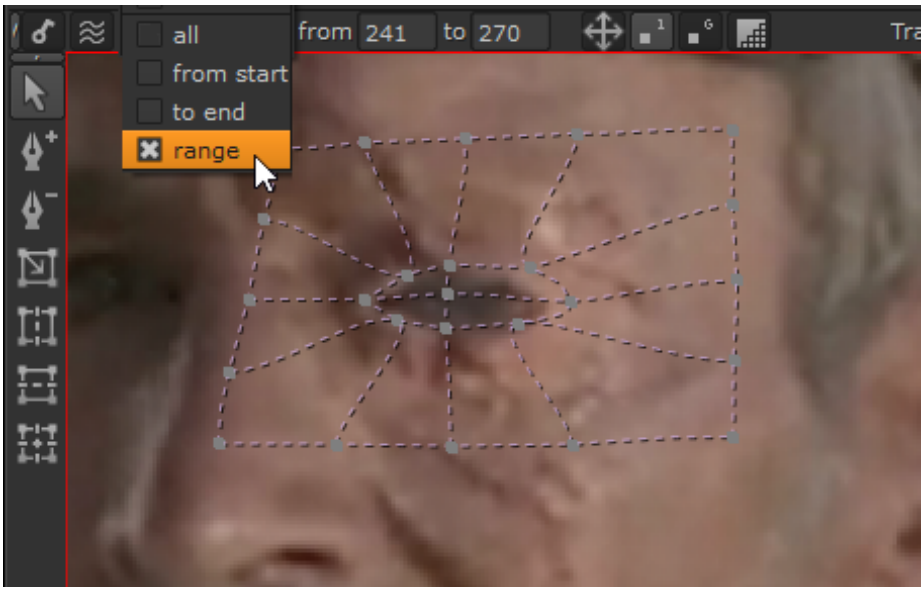
No matter how sophisticated tracking becomes, some sequences are inevitably going to cause problems. If your grids drift or deform, you can adjust the grid points manually and then ripple the relative adjustments to other frames in the sequence.

For example, tracking backwards from the grid's origin in this sequence results in deformation around the actor's eye as his head rotates.





1. Enable ripple above the Viewer and then manually correct the grid points, creating a keyframe at the beginning of the sequence.
2. Set the ripple type to **range** and then enter the frame range between your correction and the point in the sequence where the grid is correct.



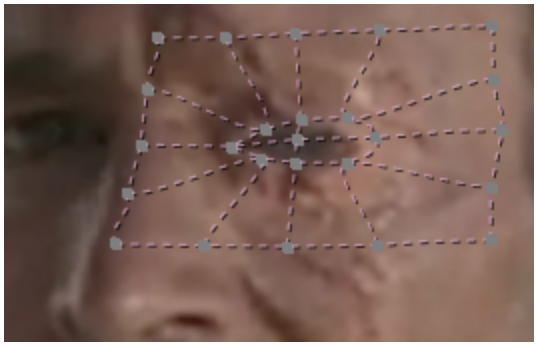
3. Correct the grid using the points in the Viewer.

   Play through the sequence to the grid origin and check that the relative transforms are applied correctly.

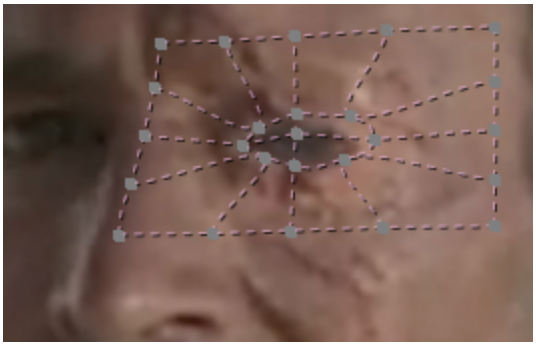
You can also use the **Track Rigidity** control above the Viewer to confine the grids to straighter lines, allowing less deformation. Lower values allow the grid to follow the tracking data more freely, whereas higher values force the grid to retain its original shape.

Note: You must set the **Track Rigidity** value before tracking. Adjusting rigidity after tracking has no effect.

In the example, the grid has deformed as the actor's head rotates. This is not necessarily bad tracking, but for more angular shapes you can increase the **Track Rigidity** to confine the grid to more regular shapes.



Low **Track Rigidity**.



High **Track Rigidity**.

Proceed to Warping Shots with GridWarpTracker when your grids are stable and accurate.

Warping Shots with GridWarpTracker

Images are warped by moving points in the tracked grids to a new location. The **From** grid represents the starting position of pixels in the image and the **To** grid is used to control their destination. Warping an image in GridWarpTracker is the same for both tracking methods, so it doesn't matter if you're using Tracker node data or SmartVector data, the results are similar.

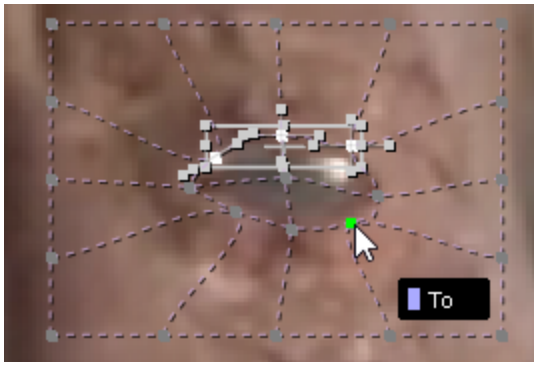
If you have a NukeX or Nuke Studio license, you can use up to six adjustment grids to make incremental changes to grid point positions, giving you greater control and more freedom to roll back any changes you made without affecting the data in the **To** grid. See Warping Using Adjustment Grids for more details.

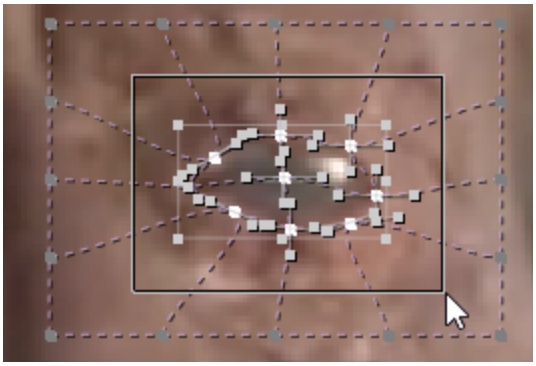
To warp an image in GridWarpTracker:

1. Scrub to the frame within the tracked range where you drew the original grid.
2. Disable the link button in the grids list and select the **To** grid.

> **Note:** Disabling the link allows you to warp the image immediately when you move points. Enabling the link copies the keyframes from one grid to the other, so changing the position of points has no effect as both grids are moving together.

3. Select the points you want to warp by holding **Shift** and clicking the points or by marquee selecting multiple points.



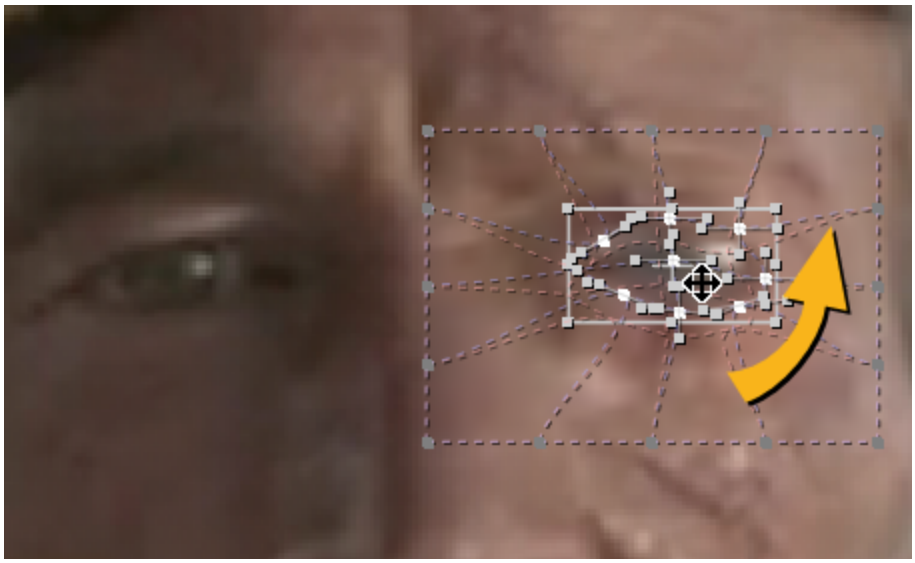


Tip: Enable the transform handle button above the Viewer if the multi-selection handle is not displayed.

4. To apply the warp to other frames, relative to the reference frame, enable the ripple button above the Viewer. Rippling keyframes allows you to adjust the position of a grid vertex on one frame and have that same relative adjustment applied to the point across all frames or a specified range of frames:
   - **off** - ripple edit is disabled.
   - **all** - ripple all frames in the sequence.
   - **from start** - ripple frames from the **first** frame to the **current** frame.
   - **to end** - ripple frames from **current** frame to the **last** frame.
   - **range** - ripple all frames within the **from** and **to** fields.

Note: The Viewer has a red outline when ripple is active to remind you that changes to this frame are applied to other frames in the sequence.

5. Drag the handle to adjust the points and warp the image between the **From** and **To** grids.
   The Viewer displays a low resolution preview of the warp until the scanline render is complete. If you want to turn off the preview, disable the persistent preview button.



Tip: You can hide grids by clicking the eye button in the grids list to reduce clutter in the Viewer.

GridWarpTracker adds keyframes automatically when you move grid points. If you always want to add keyframes manually, disable autokey by clicking the button above the Viewer.

6. Play through the sequence to check that your changes are applied correctly.



Original image.



Warped image.



Rippled keyframes across the sequence.

Warping Using Adjustment Grids

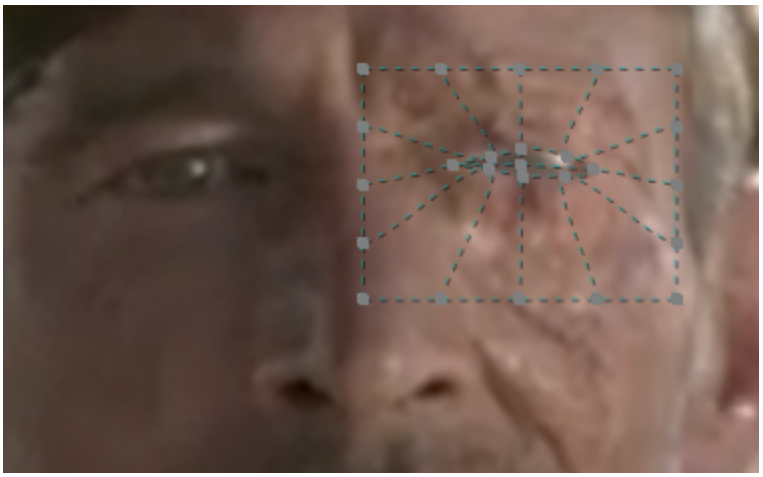
You can add up to six adjustment grids to make incremental changes to grid point positions, giving you greater control and more freedom to roll back any changes you make without affecting the data in the **To** grid.

You can hide and lock adjustment grids in the same way as the **From** and **To** grids, but you can also disable individual adjustment grids to control the warp. For example, you could experiment with creative decisions on the size of an eye with different adjustment grids.

Note: Adjustment grids are only available with NukeX or Nuke Studio licenses.

1. Click the add grid button in the **Properties** panel to add an adjustment grid.
   The **To** grid is copied into **adjustment1** grid.
2. Double-click the adjustment grid and give it descriptive name. For example, **eye_closed**.

In the example, the prosthetic eye is closed by the first adjustment grid.



3. Click the and buttons to disable and hide the **eye_closed** adjustment grid.
4. Add another adjustment and rename it **eye_wide**.

   In the example, the prosthetic eye is opened wide by the second adjustment grid.



5. You can now enable and disable the adjustment grids to close and open the eye in the sequence.









Morphing Shots with GridWarpTracker

Morphing refers to dissolving two images together so that the subject of one image seems to change shape and turn into the subject of the other through a seamless transition. A morph can be easily noticeable or very subtle. An example of a noticeable morph would be a man turning into a woman or one animal turning into another, whereas a transition from an actor to his stunt man would result in a much more subtle morph.

Video: In this video, VFX designer and compositor, Shahin Toosi, talks through the variety of techniques involved in creating a male-to-female morphing effect in NukeX.
https://learn.foundry.com/course/3405/view/nukex-face-morphing

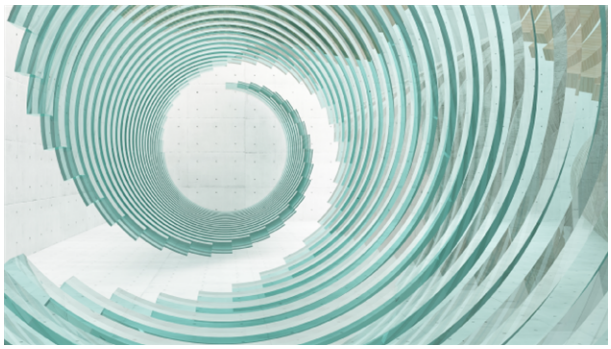
Tip: Click the **Course Files** tab to download the assets in the video.

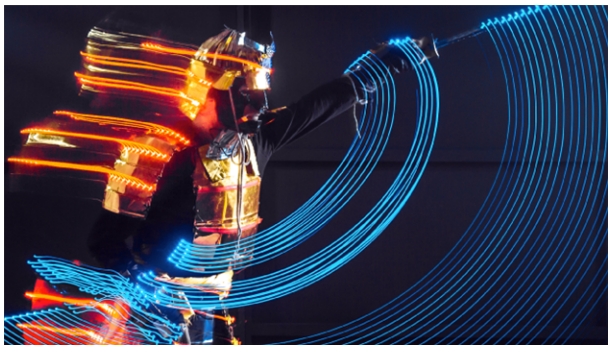
Shahin covers the use of SmartVectors in the GridWarpTracker node as well as techniques for using NukeX's CameraTracker node for stabilization, the 3D Viewer, and a walk-through of the final composite.

- 00:40 - **Optimizing the Canvas**: focusing on the area of interest.
- 01:50 - **Tracking and Stabilizing**: using Camera Tracking to align the plates.
- 06:23 - **Creating SmartVectors**: creating and copying Using the Smart Vector Toolset data.
- 07:32 - **The Morphing Effect**: using Drawing Grids to morph the plates.
- 10:00 - **The Final Composite**: merging onto the background plate.

Transforming Elements

These pages explain how to perform a range of 2D and 2.5D spatial transformations. You learn how to apply geometric transformations (including translations, rotations, scales, and skews) to elements, and how to add motion blur using the nodes in the **Transform** menu.





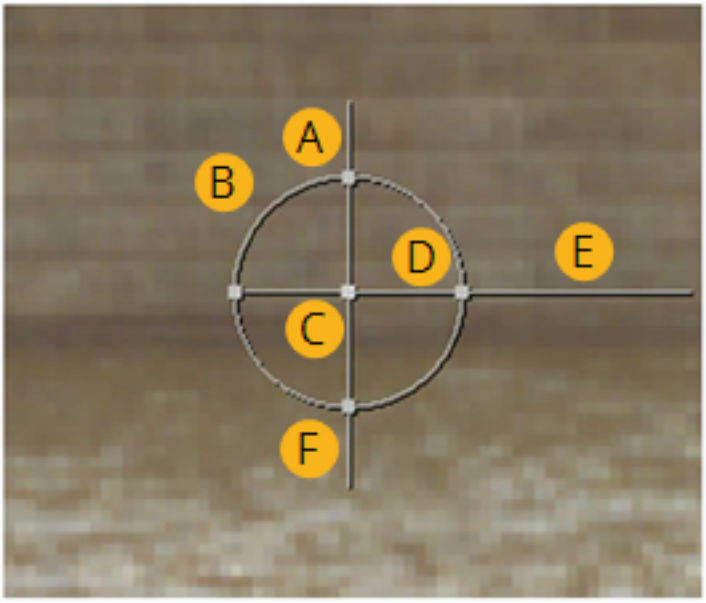
Transforming in 2D

This section describes how to apply 2D transformations including translations, rotations, scales, and skews to elements using a number of Nuke nodes. All transforms, except translations, change the size of the image bounding box.

Tip: You can enable a warning to indicate when the bounding box is greater that the format in Nuke's **Preferences**. See Bounding Box Warnings for more information.

Using the 2D Transformation Overlay

Several of the nodes discussed in this section display a Viewer overlay for executing spatial transformations. This overlay is often a faster alternative to the properties panel. The figure below shows you how to use Nuke 2D transformation overlay.



Transformation overlay.

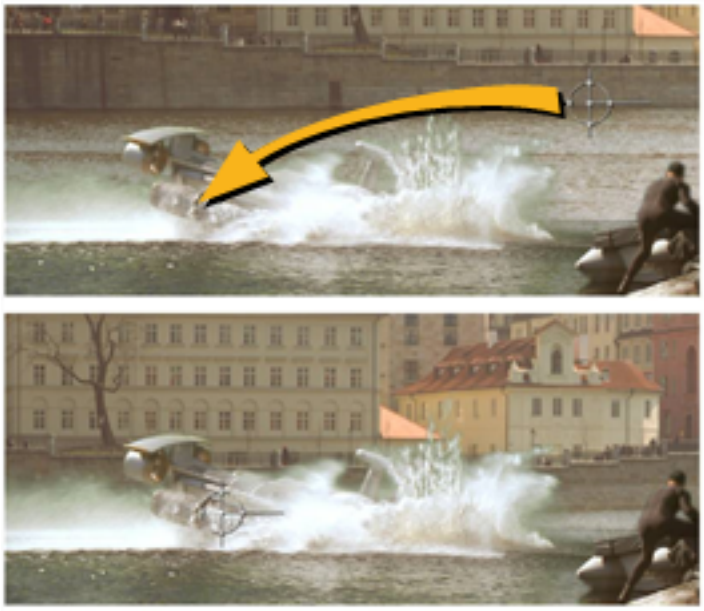
- A) Drag to skew the frame (see Skewing Elements).
- B) Drag to scale the frame uniformly - simultaneously on x and y (see Scaling Elements).
- C) Drag to translate the frame (seeTranslating Elements).

  Shift+drag to constrain the translation to x or y.

  Ctrl/Cmd+drag to reposition the pivot point (the point that acts as the center to transformation operations).
- D) Drag to scale the frame on x.
- E) Drag to rotate the frame around the pivot point (see Rotating Elements). The transform overlay snaps to typical values. To prevent the snapping, press **Shift** while dragging.
- F) Drag to scale the frame on y.

Translating Elements

To *translate* an element is to slide it on x or y.



You can use the Transform, TransformMasked, or Position nodes to translate elements.

Using the Transform Node

The Transform and TransformMasked nodes let you not only translate elements, but also rotate, scale, and skew them from a single properties panel.

TransformMasked is identical to Transform except that it offers controls for assigning a mask to protect certain areas of the frame from translations. For the sake of brevity, this chapter only discusses the use of Transform, but keep in mind you can use TransformMasked any time you need to process a transformation through a mask. The mask controls work in the same fashion as those described in Masking Color Corrections.

To translate an element using the Transform node:

1. Click **Transform** > **Transform** to insert a Transform node at the appropriate place in your script.
2. Connect a Viewer to the output of the Transform node so you can see the effect of your changes.
3. In the Transform properties panel, increment or decrement the **translate x** and **y** fields to slide the element along either axis.

   Or drag on the center of the transformation overlay.

Using the Position Node

The Position node gives you just bare-bones parameters for translating an element.

To translate an element using the Position node:

1. Click **Transform** > **Position** to insert a Position node at the appropriate place in your script.
2. Connect a Viewer to the output of the Position node so you can see the effect of your changes.
3. In the Position properties panel, increment or decrement the **translate x** and **y** fields to slide the element along either axis.

Rotating Elements

To *rotate* an element is to spin it around the pivot point.



To rotate an element using the Transform node

1. Click **Transform** > **Transform** to insert a Transform node at the appropriate place in your script.
2. Connect a Viewer to the output of the Transform node so you can see the effect of your changes.
3. In the Transform properties panel, select the appropriate filtering algorithm from the **filter** dropdown menu (see Choosing a Filtering Algorithm).
4. Position the pivot point as necessary:
   - Increment or decrement the **center x** and **y** fields to move the axis in either direction.
   - Or press **Ctrl** (**Cmd** on a Mac) while dragging on the center of the transformation overlay.
5. Increment or decrement the **rotate** field.

   Or drag on the horizontal bar of the transformation overlay.

Scaling Elements

To *scale* an element is to resize it by adding (upsampling) or removing (downsampling) pixels.



Nuke offers several nodes for scaling elements. Transform is designed primarily for scaling the background plate up or down in a composite. The scaling functions for transform are described below.

Reformat is designed for writing out elements with specific resolutions and pixel aspect ratios. Adding Motion Blur describes the use of this node.

To scale an element using the Transform node

1. Click **Transform** > **Transform** to insert a Transform node at the appropriate place in your script.
2. Connect a Viewer to the output of the Transform node so you can see the effect of your changes.
3. In the Transform properties panel, select the appropriate filtering algorithm from the **filter** dropdown menu (see Choosing a Filtering Algorithm).
4. Position the pivot point as necessary:
   - Increment or decrement the **center x** and **y** fields to move the axis in either direction.
   - Or press **Ctrl** (**Cmd** on a Mac) while dragging on the center of the transformation overlay.
5. To scale the frame uniformly (on both x and y):
   - Increment or decrement the Transform node's **scale** field.
   - Or drag the circle-portion of the transformation overlay.
6. To scale the frame asymmetrically (on x or y):

- Click **scale** parameter's channel chooser 2 to reveal the **x** and **y** fields, then increment or decrement each individually.
- Or drag any of the four points on the circle-portion of the transformation overlay. The top and bottom points scale on y; the left and right points, on x.

Skewing Elements

To *skew* an element is to rotate its pixel columns around the pivot point.



To skew an element using the Transform node

1. Click **Transform** > **Transform** to insert a Transform node at the appropriate place in your script.
2. Connect a Viewer to the output of the Transform node so you can see the effect of your changes.
3. In the Transform properties panel, select the appropriate filtering algorithm from the **filter** dropdown menu (see Choosing a Filtering Algorithm).
4. Position the pivot point as necessary:
   - Increment or decrement the **center x** and **y** fields to move the axis in either direction.
   - Or **Ctrl**+drag (**Cmd**+drag on a Mac) on the center of the transformation overlay.
5. Increment or decrement the **skew** field to rotate the pixel columns around the pivot point.
   Or drag the vertical bar of the transformation overlay.

To Invert a Transform Effect

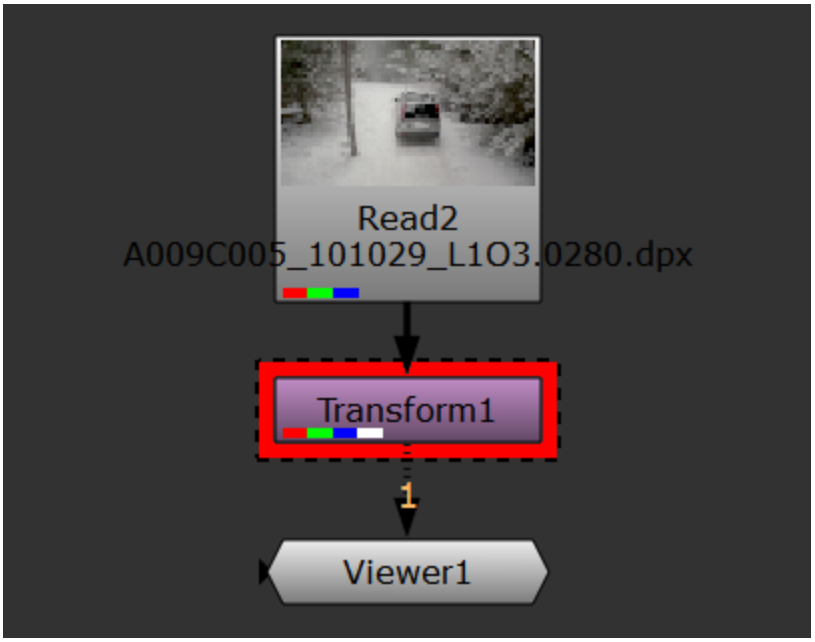
You can invert the effect you've created with the Transform node by checking the **invert** box in the Transform properties panel. This uses the inverse values of your translate, rotate, scale and skew values. When the box is checked, a small transform handle appears next to the standard transform handle in the Viewer.

Bounding Box Warnings

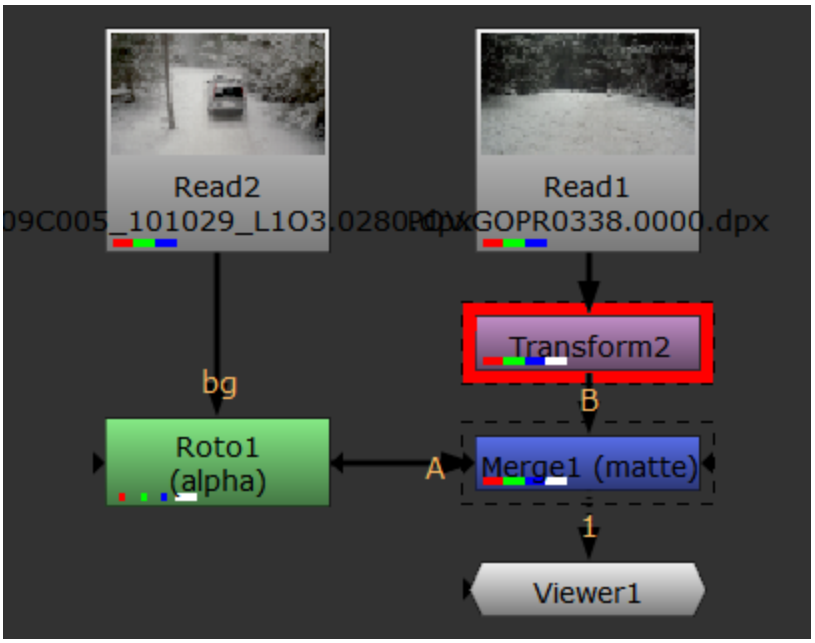
Zooming into the Viewer to work on a shot means that you can't always see the extent of the bounding box in relation to the format, which can result in unnecessary processing.

To make it easier to see the state of your bounding box, Nuke can display visual warnings on the nodes that affect the bounding box. To enable the warnings, in Nuke's **Preferences** under **Panels** > **Node Graph**, enable **Bounding Box Warning**:

- **red rectangle with dotted stroke** - the indicated node creates a bounding box greater than the format.



- **dotted stroke without the red rectangle** - the bounding box size is greater than the format at the indicated node, but the bounding box size has been set by an upstream node.



The bbox warning **threshold** controls how far past the edge of the format the bounding box can expand before the warning is displayed in the Node Graph. For example, if you're working with UHD_4K footage and the default 10% threshold, you can expand the bounding box horizontally by 384 pixels before the warning is displayed.

Tip: You can set the color of the warning rectangle in the **Preferences** under **Panels** > **Node Graph** > **Bounding Box Warning**.

Choosing a Filtering Algorithm

Spatial transformations involve remapping pixels from their original positions to new positions. The question arises as to what values to assign remapped pixels. In the simplest case, they retain their original values, but this can create problems with image quality, particularly in high contrast areas of the frame. For example, the figure below shows a close up of a high-contrast feature that has been rotated clockwise by 45 degrees. The remapped pixels have retained their original values, but the result is a highly aliased, or jagged, edge:



The solution is to apply a more sophisticated *filtering algorithm* to determine the values of remapped pixels - one that takes into account, in some fashion, the values of neighboring pixels.

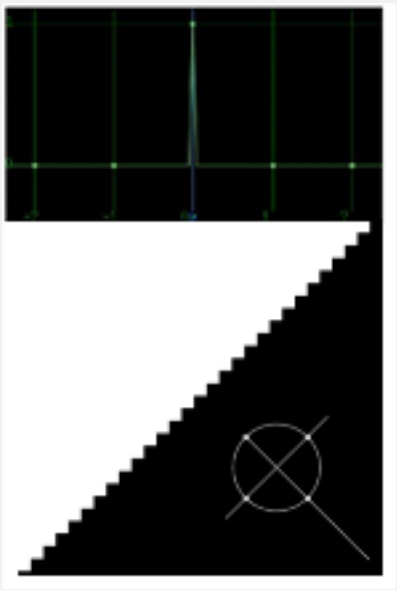
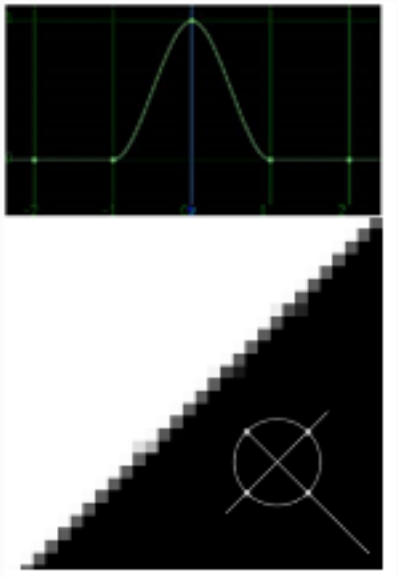
For example, applying Nuke's cubic algorithm to the above rotation, results in a softer, less jagged edge:

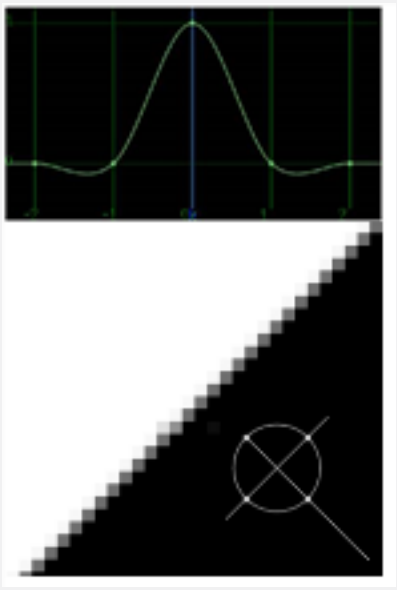
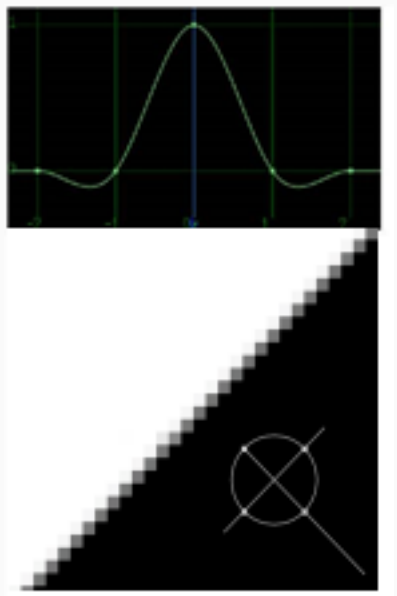


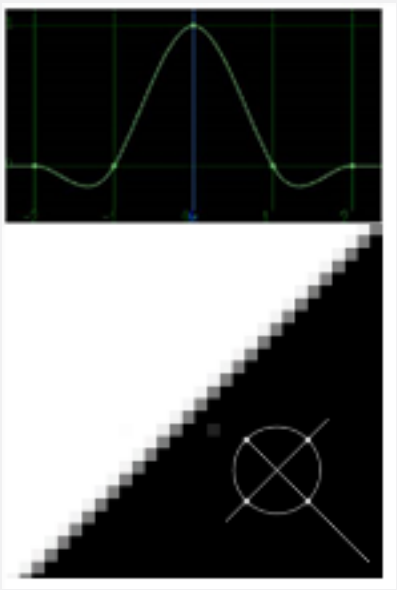
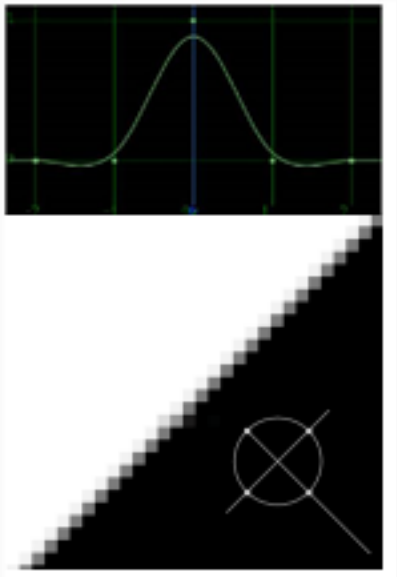
When executing spatial transformations, Nuke lets you select from the filtering algorithms described in the table below.

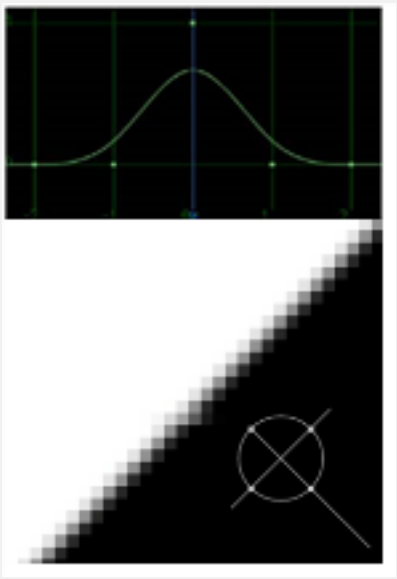
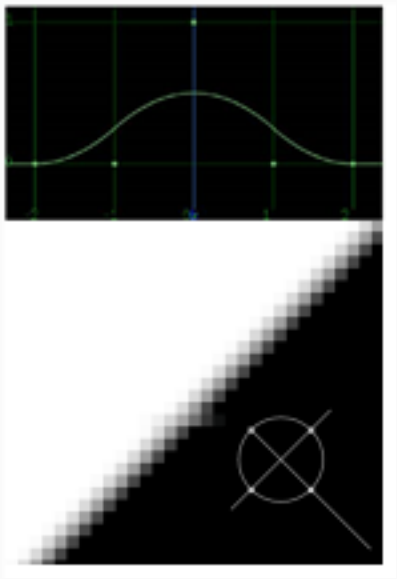
Note that the curves shown in the table plot the manner by which each algorithm samples from neighboring pixels. The center of each curve represents the value of the remapped pixel itself, and the rising and falling portions of each curve represent the amount of sampling that occurs across a five pixel radius.

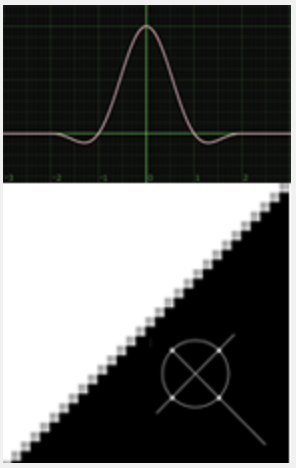
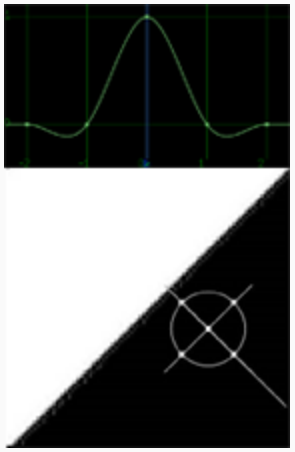
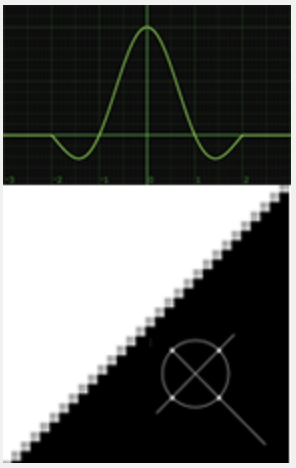
Tip: When using filters that employ sharpening, such as **Rifman** and **Lanczos**, you may see a haloing effect. Some nodes, such as Transform and Tracker, include a **clamp** control to correct this problem.

Filter	Description	Sampling Curve and Output
Impulse	Remapped pixels carry original values.	
Cubic (default)	Remapped pixels receive some smoothing.	

Filter	Description	Sampling Curve and Output
Keys	Remapped pixels receive some smoothing, plus minor sharpening (as shown by the negative -y portions of the curve).	
Simon	Remapped pixels receive some smoothing, plus medium sharpening (as shown by the negative -y portions of the curve).	

Filter	Description	Sampling Curve and Output
Rifman	Remapped pixels receive some smoothing, plus significant sharpening (as shown by the negative -y portions of the curve).	
Mitchell	Remapped pixels receive some smoothing, plus blurring to hide pixelation.	

Filter	Description	Sampling Curve and Output
Parzen	Remapped pixels receive the greatest smoothing of all filters.	
Notch	Remapped pixels receive flat smoothing (which tends to hide *moiré* patterns).	

Filter	Description	Sampling Curve and Output
Lanczos4	Remapped pixels receive minor sharpening (as shown by the negative -y portions of the curve), good for scaling down. The filter number at the end of the filter name denotes the width of the filter (4 pixels).	
Lanczos6	Remapped pixels receive some sharpening (as shown by the negative -y portions of the curve), good for scaling down. The filter number at the end of the filter name denotes the width of the filter (6 pixels).	
Sinc4	Remapped pixels receive a lot of sharpening (as shown by the negative -y portions of the curve), good for scaling down. The filter number at the end of the filter name denotes the width of the filter (4 pixels).	

Note: **Lanczos** and **Sinc** filters exhibit haloing as the filter becomes sharper.



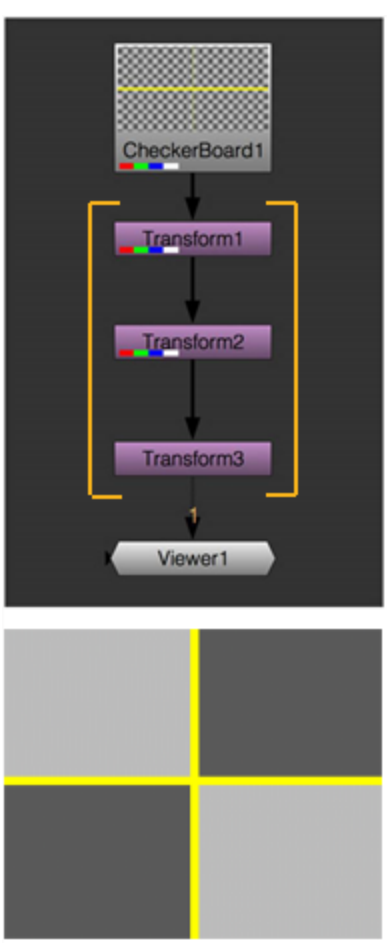
Note: The numbers appended to the filter name denote the width of the filter, similar to the naming scheme used by PRMan. See http://www.renderman.org/RMR/st/PRMan_Filtering/Filtering_In_PRMan.html for more information.

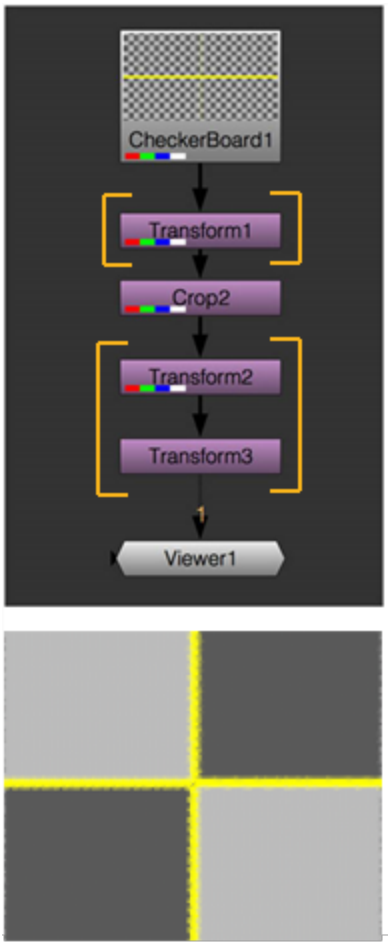
How Nodes Concatenate

Concatenation is behavior that some Nuke nodes perform when you have several nodes transforming your image one after another. When nodes concatenate, they pass on these adjacent transformation operations to the last transforming node in the row, and the last node then performs all the transformations at once.

For example, if you have three Transform nodes in a row, instead of performing each transformation separately and filtering the input image three times, Nuke combines the transformations into one operation and only filters the image once. As filtering unavoidably destroys some image detail, it's important to concatenate as many nodes as possible in order to preserve image quality.

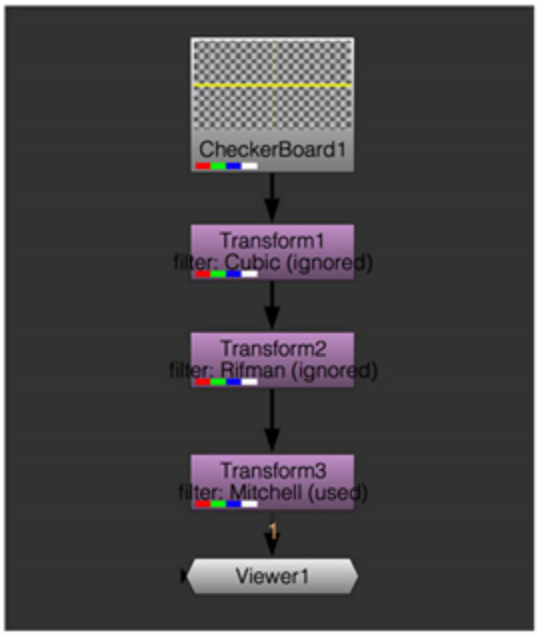
In the example are three Transform nodes that concatenate. The image is only filtered once. Inserting a Crop node between the Transform nodes breaks concatenation. The image is filtered twice, resulting in poor image quality.





In order to concatenate, the concatenating nodes have to be adjacent. So, if you have a node that doesn't concatenate (a Crop node for example) between two concatenating nodes (Transform nodes for example), no concatenation occurs.

If you're using more than one filtering method in the nodes that concatenate, the last filtering method in the series of concatenating nodes is applied on the result.



When nodes concatenate, Nuke only uses the filtering method set on the last concatenating node (in this case, the filtering method used is Mitchell).

As a rule of thumb, the only nodes that concatenate are usually transform nodes. Disabled nodes and a few others, such as Dot and Switch, do not break concatenation.

Color nodes do not concatenate because Nuke works in a 32-bit float, which is enough to avoid banding and visible round-off errors in color.

Please be aware that some transform nodes do not concatenate and should not be interspersed with those that do. There are also transform nodes that receive concatenation, but don't pass anything on.

Nodes that concatenate	Only concatenate upstream	Do not concatenate
AddChannels	BasicMaterial	AdjustBbox
Assert	Diffuse	BlackOutside
Card3D	DisplaceGeo	Crop
CornerPin	Displacement	Mirror
Disabled nodes	Emission	PlanarTracker

Nodes that concatenate	Only concatenate upstream	Do not concatenate
Dot	Environment	PointsTo3D
NoOp	Flare	Position
PointsTo3D	GridWarp	TVIScale
PostageStamp	IDistort	
PrintHash	LensDistortion	
Reconcile3D	MotionBlur2D	
Reformat	Phong	
Remove	Sparks	
Sampler	Specular	
Stabilize	SphericalTransform	
Switch	SplineWarp	
Tracker	STMap	
Transform	Tile	
	TransformMasked	

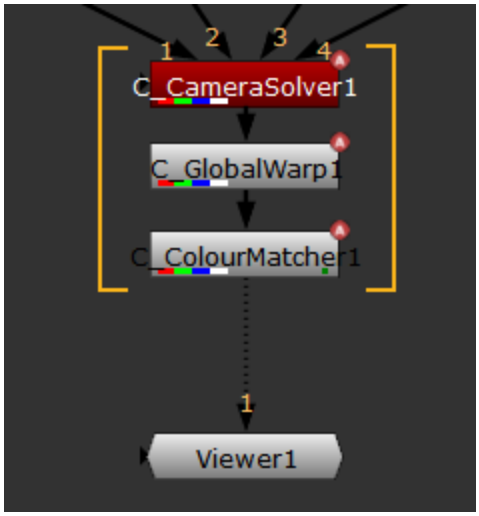
GPU Caching

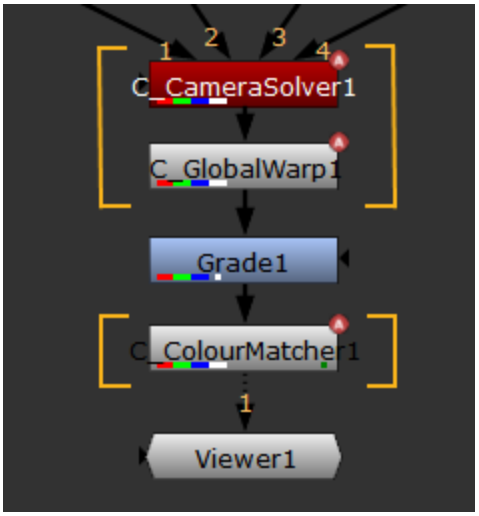
GPU caching cuts down processing overheads when data is passed from the CPU to the GPU and back again by keeping the data on the GPU between cache-enabled nodes. When nodes support GPU caching, they pass on the GPU data to the last node that supports GPU caching and then transfer all the data back to the CPU at once.

GPU caching is supported by all CaraVR nodes, the updated Bilateral2 and SphericalTransform2 nodes, and the BlinkScript node.

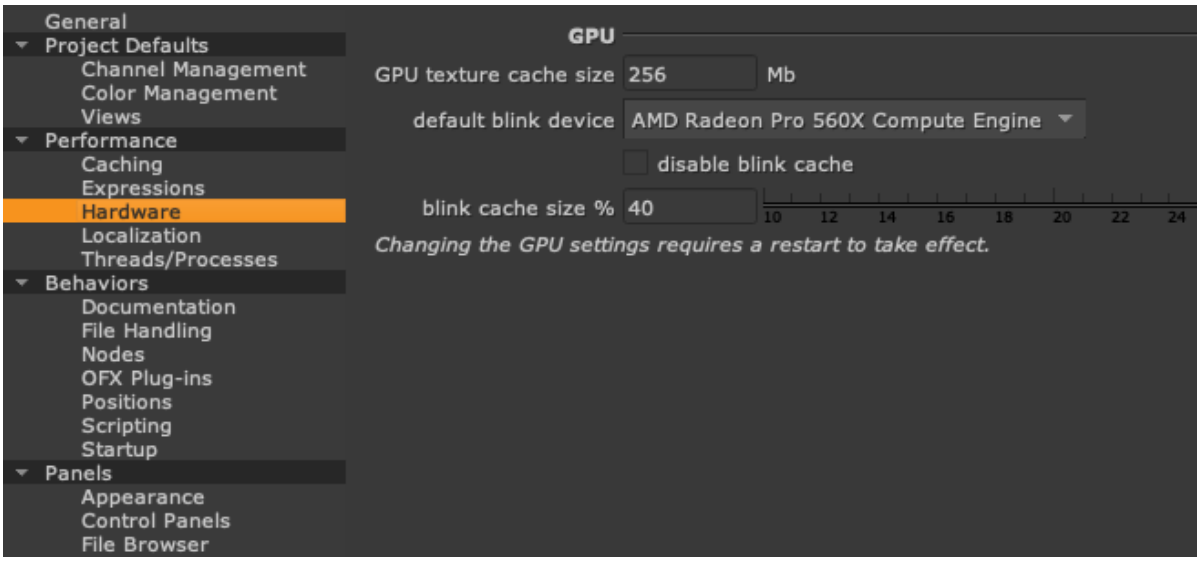
For example, if you have three GPU caching nodes in a row, instead of transferring from the CPU to the GPU and back to the CPU three times, Nuke performs all the computation on the GPU and then transfers the data back to the CPU once, at the end of the node tree.

In the example, the three cached nodes only transfer data back from the GPU once after the C_ ColourMatcher. Inserting a Grade node between the cached nodes breaks the cached sequence. Nuke now transfers data twice for the same operation, once after the C_GlobalWarp and again after the C_ ColourMatcher.



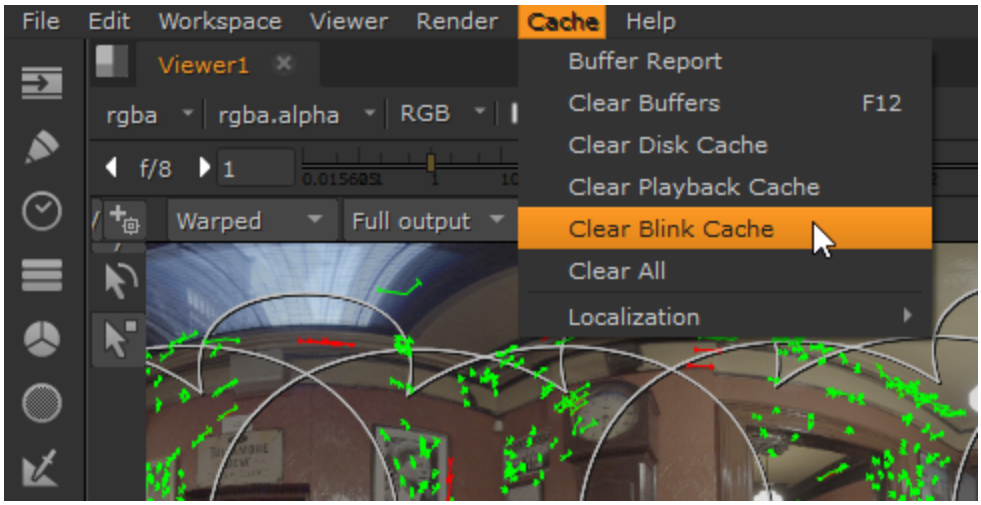


You can control the size of the GPU memory and the percentage of that memory available for caching in Nuke's **Preferences** under **Performance** > **Hardware**. You can also disable GPU caching in the **Preferences**.



Note: Increasing the percentage of memory used for caching does not necessarily decrease processing time, and can in some cases have a negative effect. The default 40% is preferred in most instances.

You can clear the Blink Cache by clicking Nuke's **Cache** menu and selecting **Clear Blink Cache** or **Clear All**.



Applying Core Transformations in 2.5D

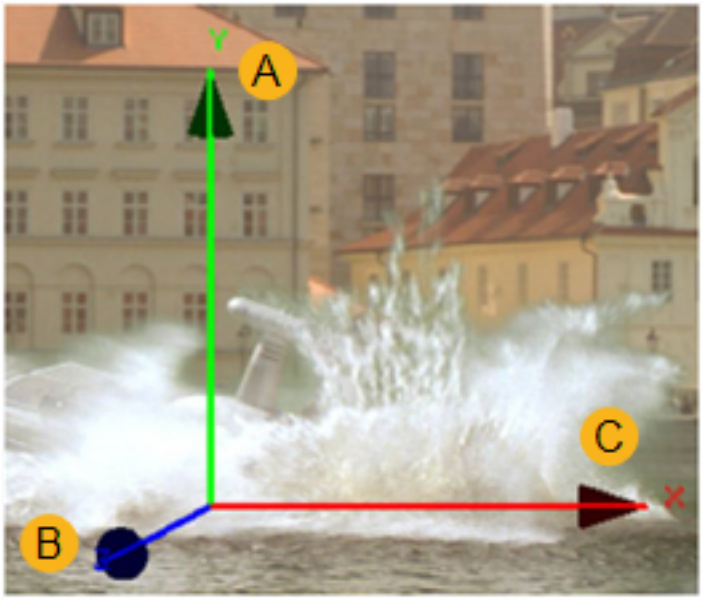
Nuke's Card3D node lets you apply the same geometric transformations possible with the Transform node, but gives you an additional axis of operation, z.

Just to be clear, the Card3D node's transformations are not truly 3D, but rather what is sometimes called "2.5D" - meaning that you can move an element back on the z axis, but doing so does not convey the sense that it is behind or in front of another element. 2.5D transformations are useful for tasks like "cheating" the perspective of an element or "faking" a camera zoom.

Remember, however, that Nuke doesn't limit you to 2.5 dimensions. If you need true 3D capabilities, you can construct a 3D scene. See Classic 3D Compositing for more information.

Using the 3D Transformation Handles

You'll note when viewing the output of a Card3D node that it displays an overlay for executing spatial transformations. This overlay is often a faster alternative to the properties panel. The figure below shows you how to use it.



- A) Drag to translate the frame on the y axis (see Translating Elements).

  Press **Ctrl**/**Cmd** while dragging to rotate the frame on any axis (see Rotating Elements).
- B) Drag to translate the frame on the z axis.

  Press **Ctrl**/**Cmd** while dragging to rotate the frame on any axis.
- C) Drag to translate the frame on the x axis.

  Press **Shift** while dragging to constrain the translation to x.

  Press **Ctrl**/**Cmd** while dragging to rotate the frame on any axis.

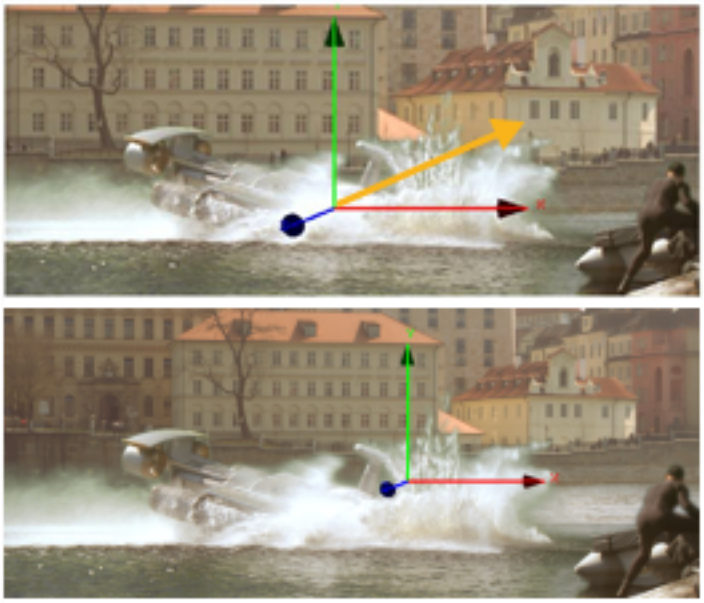
Adding a Card3D Node

To add a Card3D node:

1. Click **Transform** > **Card3D** to insert a Card3D node at the appropriate place in your script.
2. Connect a Viewer to the output of the Card3D node so you can see the effect of your changes.

Translating Elements

When using the Card3D node, you can translate elements on z in addition to the other axes.



To Translate an Element Using the Card3D Node

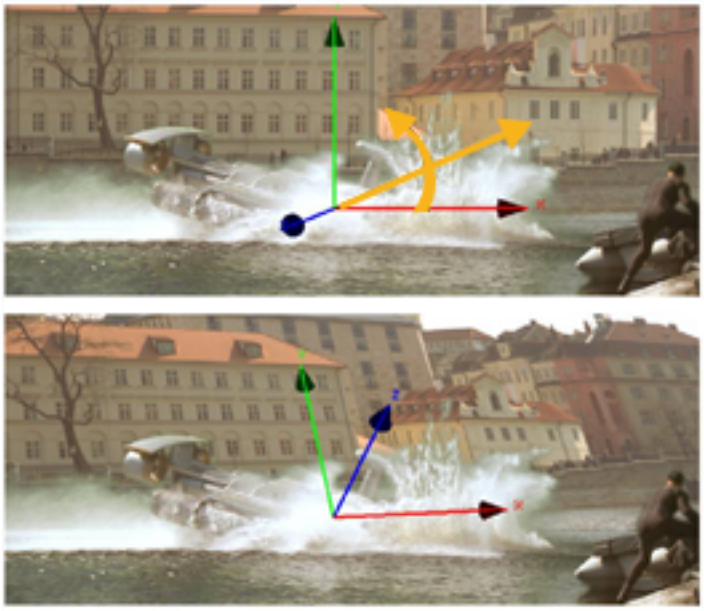
In the Card3D properties panel, increment or decrement the **translate x**, **y**, and **z** fields to slide the element along any axis,

OR

Drag on any axis or transformation overlay.

Rotating Elements

When using the Card3D node, you can rotate elements around the x and y axes, in addition to the z. This is useful for cheating the perspective.



To rotate elements using the Card3D node

1. Position the pivot point as necessary by incrementing or decrementing the **pivot x**, **y**, and **z** fields to move the axis in any direction.

   Alternatively, you can position the pivot point by pressing **Ctrl**/**Cmd**+**Alt** while dragging.
2. Increment or decrement the **rotatex**, **y**, and **z** fields to spin the element around the pivot point.

   Alternatively, you can press **Ctrl/Cmd** while dragging on any axis on the transformation overlay.

Scaling Elements

To scale an element using the Card3D node:

1. Position the pivot point as necessary by incrementing or decrementing the **pivot x**, **y**, and **z** fields to move the axis in any direction.

   Alternatively, you can position the pivot point by pressing **Ctrl**/**Cmd**+**Alt** while dragging.
2. To scale the frame simultaneously on x, y, and z, increment or decrement the **uniform scale** field.
3. To scale the frame asymmetrically, increment or decrement the **scale x**, **y**, and **z** fields.

Skewing Elements

Whereas the Transform node lets you rotate pixel columns only around the z axis, Card3D permits you to so around all three axes.

To Skew an Element Using the Card3D Node

1. Position the pivot point as necessary by incrementing or decrementing the **pivot x**, **y**, and **z** fields to move the axis in any direction.

   Alternatively, you can position the pivot point by pressing **Ctrl**/**Cmd**+**Alt** while dragging.
2. Increment or decrement the **skew x**, **y**, and **z** fields to rotate the pixel columns around the corresponding axes.

Specifying the Order of Operations

The order by which Nuke executes operations can affect the outcome. The Card3D node lets you select the order by which Nuke executes scales, rotations, and translations, as well as the order by which it executes rotation on individual axes.

To Choose the Operation Order for Scales, Rotations, and Translations

In the Card3D properties panel, select an option from the **transform order** dropdown menu, which displays all possible combinations (**S** signifies scale, **R**, rotation; and **T**, translation).

To Choose the Operation Order for Rotations

Select an option from the **rotation order** dropdown menu, which displays all possible axial combinations.

Choosing a Filtering Algorithm

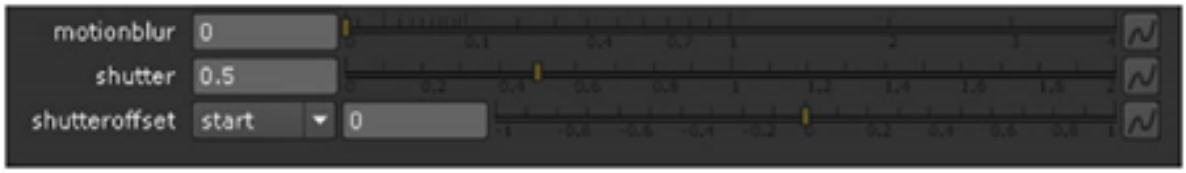
Filtering algorithms let you specify the degree of smoothing and sharpening that remapped pixels receive during transformation. The Card3D node offers the same filter algorithms as the Transform node. See Choosing a Filtering Algorithm for more information.

To choose a filter algorithm, select the desired algorithm from the **filter** dropdown menu.

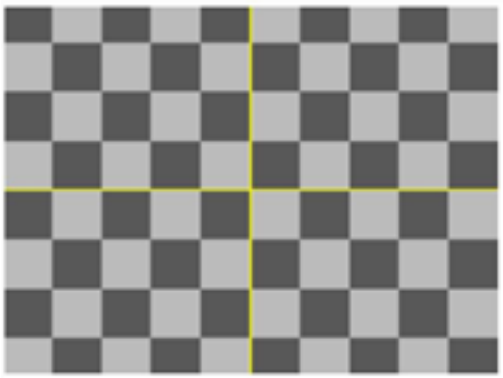
Adding Motion Blur

The following nodes under the **Transform** menu have their own controls for adding motion blur to transformations:

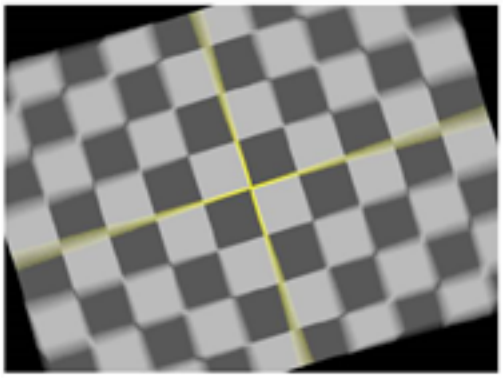
- Transform
- TransformMasked
- Card (3D)
- CornerPin2D
- Reconcile3D
- Tracker
- Stabilize2D



These controls allow you to create motion blur without adding a separate node for it. The output is similar to a TimeBlur node (see Applying a TimeBlur Filter), but rather than averaging the results of several whole images computed at steps over the shutter period, a number of samples are taken at many random times over the shutter period. This effectively gives many more "steps" and thus a smoother-looking result for a smaller total number of computations.



Before rotation and motion blur.



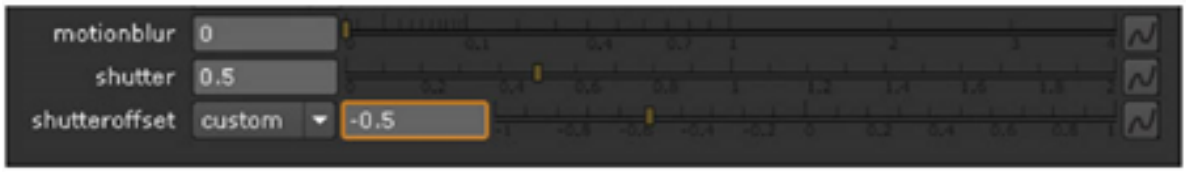
After rotation and motion blur.

When using several of these nodes in a row, the motion blur is concatenated, and the last transform in the chain defines the motion blur applied.

To Add Motion Blur

1. Open the transform node's controls.

2. Create a transform and animate it. For instructions on how to do this, see the Using the Compositing Environment chapter.
3. In the **motionblur** field, enter the sampling rate. This affects the number of times the input is sampled over the shutter time. The higher the rate, the smoother the result. In many cases, a value of 1.0 is enough. Setting the value to 0 produces no motion blur.
4. In the **shutter** field, enter the number of frames the shutter stays open when motion blurring. For example, a value of 0.5 would correspond to half a frame. Increasing the value produces more blur, and decreasing the value less.
5. From the **shutteroffset** dropdown menu, select when the shutter opens and closes in relation to the current frame value:
   - to center the shutter around the current frame, select **centered**. For example, if you set the **shutter** value to 1 and your current frame is 30, the shutter stays open from frame 29.5 to 30.5.
   - to open the shutter at the current frame, select **start**. For example, if you set the **shutter** value to 1 and your current frame is 30, the shutter stays open from frame 30 to 31.
   - to close the shutter at the current frame, select **end**. For example, if you set the **shutter** value to 1 and your current frame is 30, the shutter stays open from frame 29 to 30.
   - to open the shutter at the time you specify, select **custom**. In the field next to the dropdown menu, enter a value (in frames) you want to add to the current frame. To open the shutter before the current frame, enter a negative value. For example, a value of -0.5 would open the shutter half a frame before the current frame.



To Add Motion Blur to an Image Rendered in a Third-party Application

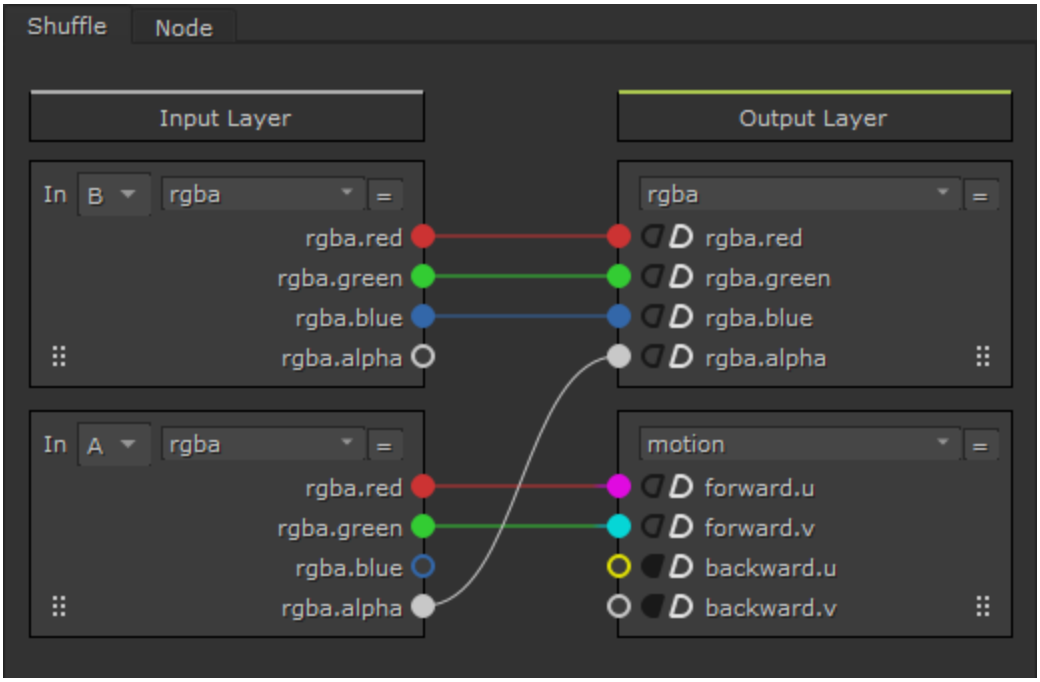
Another way to add motion blur to your image is to use the VectorBlur node. VectorBlur takes each of your image's pixels and blurs them in a straight line, using the u and v channels to determine the blur direction.

VectorBlur expects the values from your input plates to be pixel space screen units, in other words one unit should be equal to one pixel. Nuke uses this information to calculate the distance that one pixel travels between two frames. So, in order to get a working motion blur result, you should make sure Nuke is getting correct values to work with. You might have files using varying values, particularly if you've used a third party application to create your input files. The following is an example of creating motion blur with the VectorBlur node using files written from a third party application.

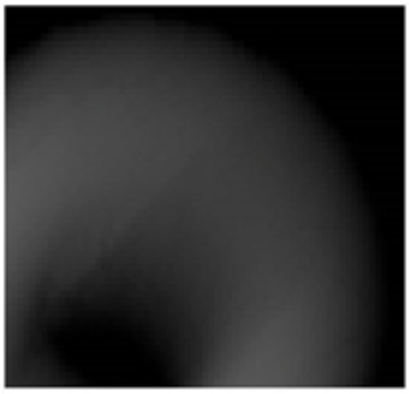
To Create Motion Blur with the VectorBlur Node

1. Read in your footage and motion blur files, for example an **.exr** file with a spinning donut and a **.sgi** file with motion blur vectors that are normalized to have values between 0 and 1.
2. Using the Shuffle node, select which channels VectorBlur should read from your motion vector file (node input A) and color image file (node input B). In this case, you would use the motion vector file's red and green channels as the motion u and v channels, and its alpha channel as the alpha channel. Meanwhile, the image file would output the red, green and blue channels for the main color image.

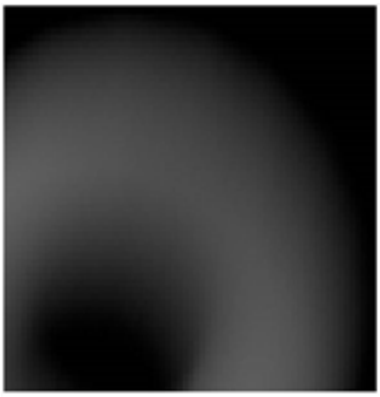
   With this setup, your Shuffle node controls would look like this:



3. Connect the VectorBlur node to the Shuffle node. You also need to tell VectorBlur which motion vector channels to use, so change the **uv channels** control to **motion**.
4. If your motion vectors have been normalized to be between 0 and 1, you can set the u and v values in the **add** control to **-0.5** to offset the motion blur center. This is usually necessary for any motion vectors stored in an integer file format like 16 bit TIFF or TARGA. Vectors that go to negative x or y directions use half the numbers in the range and vectors that go positive use the other half.
5. With the **multiply** and **offset** controls, you can further adjust the amount of motion blur you want to produce. The **offset** value allows you to correct for normalization of your vector values, and the **multiply** value controls the magnitude of them.
6. If the vectors have been premultiplied with the alpha channel, their value is not accurate in places where the alpha is not 1.0. You'll want to check the **alpha** checkbox to use the input image's alpha channel to help VectorBlur to deal with motion vectors that have been premultiplied by this alpha channel.



Alpha disabled.



Alpha enabled.

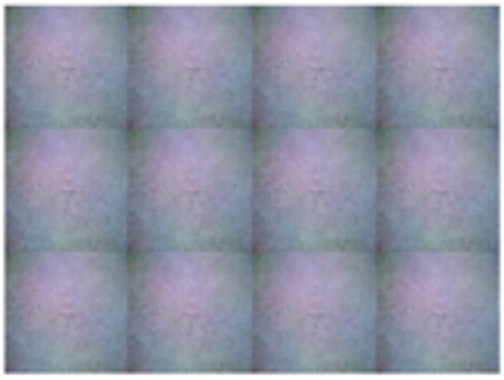
7. Getting a good, even motion blur result largely depends on selecting the right calculation method. In the method dropdown, select:
   - **backward** - backward method is effective and fast but may not be accurate if you don't have motion vector values at all pixels throughout the whole frame
   - **forward** - the forward method is slower, but it gives you a more accurate result, especially in cases where the vectors don't cover the whole frame. In this case, we know the motion vectors are not continuous, so selecting forward is a good option.

Replicating the Input Image Across the Output

The Tile node produces an output image that contains scaled-down, tiled copies of the input image. The output image is the same format as the input.



Before the Tile node.



After the Tile node.

To Use the Tile Node

1. Select the image you want to replicate and select **Transform** > **Tile**.

   A Tile node is inserted in the Node Graph.
2. Attach a Viewer to the Tile node.
3. In the Tile node properties, use the **rows** field to define how many times the image is replicated vertically. Note that the value can be fractional.
4. In the **columns** field, enter the number of times you want to replicate the image horizontally. Note that the value can be fractional.



The original input image.



The output of the Tile node with **rows** set to 4 and **columns** set to 5.

If you want to flip adjacent tiles vertically to form mirror images, check **mirror**.



The output of the Tile node without mirroring.



The same image with vertical mirroring.

If you want to flip adjacent tiles horizontally to form mirror images, check **mirror**.





The output of the Tile node without mirroring.

The same image with horizontal mirroring.

5. From the **filter** menu, select an appropriate filtering algorithm. For more information, see Choosing a Filtering Algorithm.

Warping Images

Nuke's GridWarp and SplineWarp nodes allow you to distort elements in an image and morph one image into another. There are other types of warp available in Nuke, but in this chapter, we focus on the GridWarp and SplineWarp nodes.

Quick Start

Here's a quick overview of the workflow:

1. For both warping and morphing, you can use either GridWarp node (**Transform > GridWarp**) or the SplineWarp (**Transform > SplineWarp**). Working with the GridWarp is sometimes slightly faster, whereas the SplineWarp node allows for more fine adjustment. With both nodes, you first connect your source image to the **src** or **A** input and, if morphing, the destination image to the **dst** or **B** input. GridWarp also allows you to connect an additional background image to the **bg** input.
2. Depending on which node you're using, you can then go on to set your source and destination points, and tweak them until you're happy with your results. For more information, see Warping and Warping an Image Using the SplineWarp Node.
3. If you're looking to morph an image into another, you can do this with both warper nodes as well. For more information, see Morphing and Morphing One Image into Another Using the SplineWarp Node.
4. If you are warping or morphing moving images, rather than stills, have a look at Transforming and Animating Warps and Morphs.

Warping

Warping refers to manipulating an image so that elements in the image are distorted. Unlike many of the transformations described under Transforming Elements, warps are transformations that only affect some of the pixels in an image rather than all of them. For example, you might make an animal's eyes bigger or a person's smile wider without affecting the rest of their features.

This is not to say that the pixels around the area you are moving do not move with the area. They do, because accommodating the change this way often produces more realistic results. However, the distortion

lessens the further you get from the moved pixels. You also have some control over which pixels are moved and which are not, and can isolate the warp to a small area. Still, in an ideal situation, the subject you are going to warp is a subject you can key out or rotoscope to isolate it from its background before you create the warp. This way, you can be sure that the background stays intact.

In addition to performing creative manipulations on the shapes of the subjects in your images, you can also use warping to simulate different types of film or video lenses or to remove unwanted lens distortions.

Below, we discuss how to warp images, first using the GridWarp node and then the SplineWarp node. Finally, we also teach you to animate the warps. Again, we start with the GridWarp node and then show you how to do the same with the SplineWarp node.

Warping Images Using the GridWarp Node

The GridWarp node allows you to warp images by transferring image information from one Bezier grid onto another. When using this node, you first position the source grid, which defines where to warp from. Next, you position the destination grid, which defines where to warp the image to. This grid can be a duplicate of the source grid, or you can define it separately. When you manipulate the destination grid, the corresponding warp is applied to the source image.

The GridWarp node also includes controls for animating the warp and selecting the level of filtering used to remove any artifacts the warp may have caused.

To Warp an Image Using the GridWarp Node

1. Select **Transform** > **GridWarp** to insert a GridWarp node after the image you want to warp.
2. Connect the **src** input and a Viewer to the image.

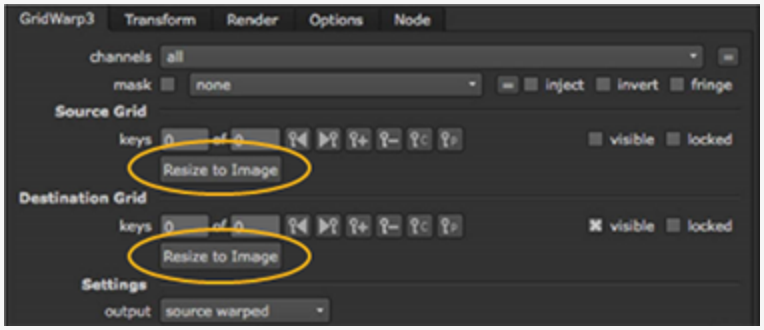


3. When the GridWarp properties panel is open, by default the destination grid overlay appears in the Viewer. You can show or hide the source and destination grids using the and buttons in the Viewer toolbar or the **visible** checkbox in the **GridWarp** tab. Use the destination grid to define the warp areas.



Note: GridWarp automatically attempts to resize the grids to the image size as long as you have not modified any of the control points.

If the grids are not the same size as the input image, click the **Resize to image** buttons under both **Source Grid** and **Destination Grid**.



4. Use the Viewer tools to control the following aspects of the grids:

Note: You can use the copy and paste buttons in the grid controls to copy control point keyframes between the source and destination grids.

Control	What it does
output	Controls the output displayed in the Viewer:

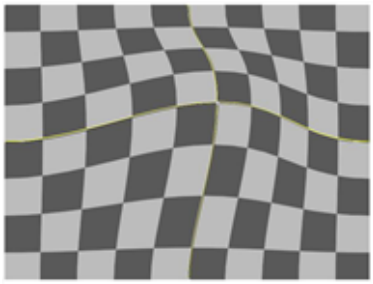
Control	What it does
	• **source** - the source image and source grid. • **sourcewarped** - the source image and destination grid. • **destination** - the destination image and destination grid. • **destinationwarped** - the destination image and source grid. • **morph** - the morphed image, controlled by the **warp** and **mix** parameters, and both grids.
	When enabled, the grids shown in the Viewer depends on the **output** setting. For example, if **output** is set to **source warped**, only the destination grid appears in the Viewer.
	When enabled, the **source** grid is displayed in the Viewer. This control can be overridden by the  button.
	When enabled, the **destination** grid is displayed in the Viewer. This control can be overridden by the  button.
	When enabled, changes to points in a grid are automatically keyed. You can disable this and set your keyframes manually, which is particularly useful if you intend to use the Curve Editor. See Animating Warps.
ripple	Rippling keyframes allows you to adjust the position of a stroke/shape point on one frame and have that same relative adjustment applied to the point across all frames or a specified range of frames. • **off** - ripple edit is disabled. • **all** - ripple all frames in the sequence. • **from start** - ripple frames from the **first** frame to the **current** frame. • **to end** - ripple frames from **current** frame to the **last** frame. • **range** - ripple all frames within the **from** and **to** fields.
label points	When enabled, points selected on the grid are labeled **x,y** measured from the origin.
	When enabled, the transform handle overlays all selected points.
	When enabled, a low resolution preview displays when points are moved on a grid. Once the render is complete, the low-res image is updated.
divisions	Enter the number of divisions required in the field to modify the grid. The number of

Control	What it does
	divisions must be between 3 and 20. You can also click and hold the slider to overlay a preview of the subdivisions. GridWarp attempts to modify the grid based on the current control point distribution. **Note:** When using the preview, accuracy may suffer at lower divisions and additional smoothing may be required at higher divisions.
	Click to enable **Edit** mode. Select individual points, multiple points using shift-click, or marquee groups of points. Edit mode also allows you to adjust the curve between points to produce distortion.
	Click to enable **Insert** mode. Click on a horizontal line to add a vertical line to the grid and vice versa.
	Click to enable **Delete** mode. Click on a grid line to remove it from the Viewer.
	Click to enable **Draw Boundary** mode. The cursor changes to a crosshair and you can drag a marquee in the Viewer to create a custom grid.
	Click to subdivide the grid **columns** across the currently selected area.
	Click to subdivide the grid **rows** across the currently selected area.
	Click to subdivide the grid **columns** and **rows** across the currently selected area.

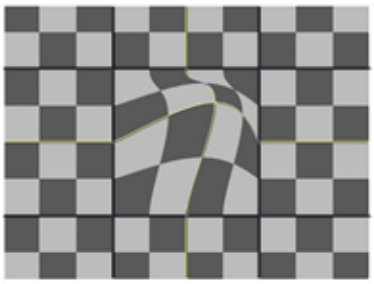
5. Modify the grid around the area you want to warp. Usually, you want the grid to conform to the subject of the source image. For example, if you are warping an animal's eyes, you need to create grid lines that follow the edges of the eyes.

Note: If you have both grids visible when you move a point, and the same point for both grids are on top of each other, both points are moved and you won't see any distortion.

You can use the grid lines to isolate the areas you do not want to warp. You do this by adding grid lines between the area you intend to warp and the area you don't want to change.

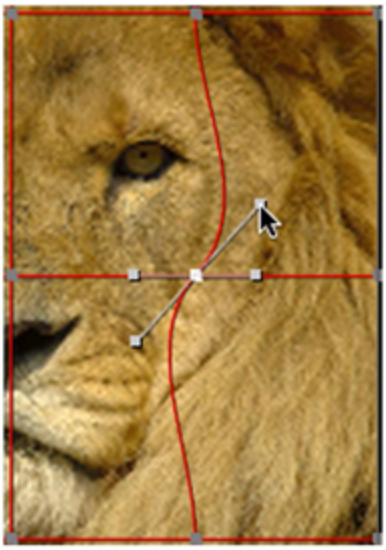


An unlimited warp.



The same warp limited to a small area with grid lines.

When you select a point, four tangent handles appear around it. You can use these handles to modify the curves connecting the points.



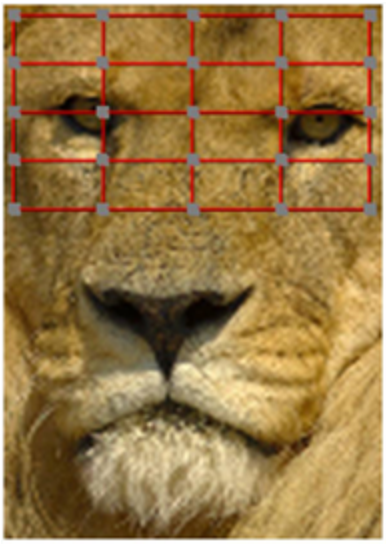
To move several points together, draw a marquee around them and use the transformation overlay that appears.



You can also use the **Draw Boundary** tool in the Viewer to quickly set a user defined grid. Click **Draw Boundary** and draw a marquee over the required area of the image.



The **Draw Boundary** marquee.



The resulting user-defined grid.

Note: You can also use the Curve Editor to edit curves by right-clicking a control point and selecting **Curve Editor** > **points**, **tangents**, or **both**.

The curves appear in the Dope Sheet as long as the GridWarp **Properties** panel is open.

6. In the areas where you want to warp the image, drag the points on the grid to a new position. When you click on a point, the point changes color to indicate whether it's in focus (green by default) and whether

it has an expression set (red by default). You can change the default colors on the **Viewers** tab in the Preferences.



The pixels in these areas are moved in the direction you moved the points. Pixels in the nearby areas are also moved to accommodate the change, but the distortion lessens the further you get from the repositioned points. If you don't want a nearby area distorted, add more grid lines between the area and the points you want to move before you drag the points to a new location.

Tip: You can nudge selected control points by a single pixel using the numeric pad **4** and **6** to nudge left and right, or **8** and **2** to nudge up and down.

Holding down **Shift** and using the numeric pad moves the selected points by 10 pixels, for example, **Shift+6** moves the selected points 10 pixels to the right.

7. To better see what the warped image looks like, press **Q** on the Viewer to toggle the overlay off.



To compare the original and warped images, press **D** on the GridWarp node to disable and enable it. If you see changes in the areas you don't want to warp, go back to modifying the grid.

8. If necessary, animate the grid to match any movement in the source image. For more information on how to do this, see Animating Warps.
9. You can adjust the controls described in the following table to enhance your results.

Control	What it does
GridWarp Tab	
channel	Sets the channels affected by the distortion.
mask	Connect a mask input and set the channel to use as a mask. By default, the mask is limited to the non-black areas of this channel. Use the checkboxes to modify the mask properties: • **inject** - copies the **mask** input to the predefined **mask.a** channel. Injecting the mask allows you to use the same mask further downstream. • **invert** - inverts the use of the mask channel so that the mask is limited to the non-white areas of the mask. • **fringe** - only apply the effect at the edges of the mask.
background	The warped image is rendered on top of an unwarped background. This control sets

Control	What it does
	what to use as that background: • **on black** - render the warped image on top of a constant black image. • **on src** - render the warped image on top of the image connected to the **src** input of the GridWarp node. • **on dst** - render the warped image on top of the image connected to the **dst** input of the GridWarp node. • **on bg** - render the warped image on top of a background image connected to the **bg** input of the GridWarp node.
background mix	Blends between the output of the GridWarp node (at 0) and whatever you have selected from the **background** dropdown menu (at 1).
Set bbox to	Sets the boundary box properties.
Render Tab	
submesh resolution	Sets the number of subdivisions that are created between Bezier curves in the grid. The higher the value, the more accurate the distortion between the grid lines, but rendering time increases.
filter	Select the appropriate filtering algorithm (see Choosing a Filtering Algorithm).
Options Tab	
source color	Sets the source grid color.
destination color	Sets the destination grid color.

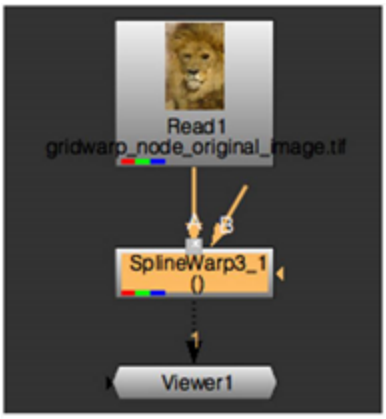
Warping an Image Using the SplineWarp Node

The SplineWarp node deforms an image based on multiple Bezier or B-spline curves that you create. Source curves define where to warp from, while destination curves define where to warp the source image to. Unlike the GridWarp node, you can draw these curves anywhere on the image, rather than only adding points on the existing grid lines, then join them to create the source and destination. The controls for adding and modifying points are similar to the RotoPaint node controls.

To Warp an Image Using the SplineWarp Node

1. Select **Transform** > **SplineWarp** to insert a SplineWarp node after the image you want to warp.

2. Connect the **A** input to the image and attach a Viewer to the SplineWarp node.



3. Use the Viewer tools to control the following:

Control	Name	What it does
	autokey	When enabled, changes to points and shapes are automatically keyed. You can disable this and set your keyframes manually, which is particularly useful if you intend to use the Curve Editor. See Animating Warps for more information.
	label points	When enabled, points selected in the Viewer are numbered sequentially.
	hide curves	When enabled, curve lines are hidden.
	hide points	When enabled, points are hidden.
	hide transform handle	When enabled, the transform handle is hidden when a selection is made.
	hide transform handle when moved	When enabled, the transform handle is hidden when a selection is moved.
	constant selection	When enabled, clicking in the Viewer does not deselect the current shape(s).
	ripple	Rippling keyframes allows you to adjust the position of a stroke/shape point on one frame and have that same relative adjustment applied to the point across all frames or a specified range of frames.

Control	Name	What it does
		• **all** - ripple all frames in the sequence. • **from start** - ripple frames from the **first** frame to the **current** frame. • **to end** - ripple frames from **current** frame to the **last** frame. • **range** - ripple all frames within the **from** and **to** fields.
	add/remove keyframe	Add and remove keyframes at the current playhead position using these buttons.
	output A	When enabled, output the image attached to the **A** input and any shapes in the **curves** list labeled with **A**.
	output A warped	When enabled, output the image attached to the **A** input warped by any shapes in the **curves** list labeled with **A**.
	output AB morph	When enabled, output the morphed **A** and **B** inputs and any shapes in the curves list labeled with **AB**.
	output B	When enabled, output the image attached to the **B** input and any shapes in the **curves** list labeled with **B**.
	output B warped	When enabled, output the image attached to the **B** input warped by any shapes in the **curves** list labeled with **B**.
	show source curves	When enabled, joined **source** curves are displayed in the Viewer.
	show destination curves	When enabled, joined **destination** curves are displayed in the Viewer.
	show correspondence points	When enabled, all **correspondence** points are displayed in the Viewer when a **Correspondence** tool is selected.
	show boundaries	When enabled, all **boundary** curves are displayed in the Viewer.
	show hard boundaries	When enabled, all **hard boundary** curves are displayed in the Viewer.

Control	Name	What it does
		Note: Hard boundary is only useful on closed curves.
	visibility	Controls visibility for the selected shape.
	lock	Controls the lock state of the selected shape. Locked shapes cannot be modified.
	toggle boundary	Toggles the Boundary state of the selected shape(s). Clicking **toggle boundary** on a standard shape defines it as a Boundary and vice versa. See Using Boundary Shapes and Pins to Limit Warp for more information.
	update mode	Sets the Viewer update mode: • **off** - no updates are applied when points are moved. Use this option for complex warps where the Viewer update can be slow. • **on** - the Viewer updates in real-time to help you position points. With complex scripts, this mode maybe prohibitively slow. • **persistent** - similar to **on**, but displays a preview until the render is complete, rather than showing real-time scanlines.
	Select tool	Click to enable **Select** mode allowing you to select points and shapes, or marquee select multiple points and shapes.
	Points tool	Click to enable **Add** mode or toggle between **Add**, **Remove**, **Cusp**, **Smooth**, **Open/Close**, and **Remove Destination** modes. Click on existing shapes to add or modify points.
	Draw tool	Click to enable **Draw** mode or toggle between **Bezier**, **Cusped Bezier**, **B-spline**, **Ellipse**, **Rectangle**, and **Cusped Rectangle** mode. Click on the Viewer to draw the selected shape.
	Correspondence tool	Click to enable **Correspondence** mode or toggle between **Add**, **Modify**, and **Remove** mode:

Control	Name	What it does
		• **Add** - clicking on a joined curve creates a correspondence point. These points are used to improve the warp accuracy in isolated areas, but overuse can cause performance issues. • **Modify** - allows you to move an existing correspondence point along its curve, in either direction, as far as the next point on the shape. • **Remove** - click to remove an existing correspondence point.
	Pin tool	Click to enable **Pin** mode. You can use pins as single point joined curves to create warp or to secure specific points in place to isolate them from warping.
	Join tool	Click to enable **Join** mode or toggle between **Join**, **Reverse**, and **Unjoin** mode.

4. Click  to select source warped mode.
5. Select the **Draw** tool  and draw a source, or warp from, curve as shown below.



Note: If you're creating multiple open splines, draw your first spline and then press **Esc** or select another tool to let SplineWarp know you're done. Reselect **Draw** to begin drawing another spline.

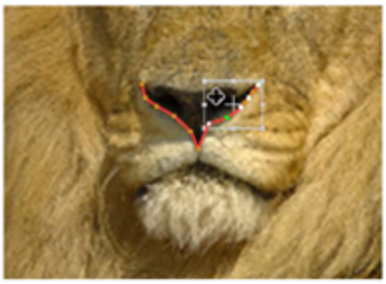


6. You can modify shapes by adding points using the **Add** tool and adjusting the tangent handles that appear around it.



When you mouse over a point, it turns green. Clicking on a point changes its color to indicate it's selected (white by default) and whether it has an expression set (red by default). You can change the default colors on the **Viewers** tab in the Preferences.

To move several points together, draw a marquee around them and use the transformation overlay that appears.



To close an open curve or open a closed curve, right-click on a point and select **open/close curve**.

To remove a point, right-click on the point and select **delete**.

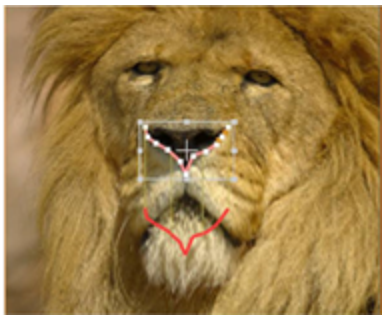
Note: You can also use the Curve Editor to edit curves by right-clicking a control point and selecting **curve editor** > **points** or **points+tangents**.
The curves appear in the Dope Sheet as long as the SplineWarp **Properties** panel is open.

7. Once you're happy with the source curve, either:
   - draw a destination curve and use the Join tool to connect the two curves.

     OR
   - right-click a point on the source curve and select **duplicate in A and join** to produce the destination curve.
8. Select the destination curve and drag points or the entire shape to a new position. The pixels in these areas are warped in the direction of movement.



The image A and warp curves.



The warped image.

9. To better see what the warped image looks like, press **Q** on the Viewer to toggle the overlay off.

   To compare the original and warped images, press **D** on the SplineWarp node to disable and enable it. If you see changes in the areas you don't want to warp, go back to modifying your shapes.

10. Using the **curves** list, you can quickly build up a complex warp hierarchy using the add/remove buttons.



Using the **root**, **layer**, and **pair warp** controls, you can affect different elements as required:

- **root warp** - affects warp globally, irrespective of layer and pairing.
- **layer warp** - affects warp on the selected layers only, irrespective of pairing. This control is disabled if anything other than layers is selected.
- **pair warp** - affects warp on the selected pairings only. This control is disabled if anything other than pairs is selected.

Note: The **pair warp** value is multiplied by the **layer warp** value, which is in turn multiplied by the **root warp** value.

11. If necessary, animate the source warped shapes to match any movement in the source image. For more information on how to do this, see Animating Warps.
12. You can adjust the controls described in the following table to enhance your results.

Control	What it does
SplineWarp tab	
crop to format	When disabled, the input image is not cropped to the **Project Settings** > **full format size**. As a result, warping can introduce pixels from the image outside the format size, rather than black. **Note:** Disabling **crop to format** can affect performance as Nuke has to calculate the warp for the entire image, not just the area inside the format size.
bbox boundary curve	When enabled, a boundary curve is added around the input format, limiting warp at the edges of the image. **Note:** In **AB morph** mode, the format of input **A** is taken as the boundary.
Render Tab	
curve resolution	Adjusts the accuracy of the warp/spline match. Higher values increase accuracy, but sacrifice speed and vice versa. For example, with open splines at low curve resolution, image tearing may occur. You can raise the **curve resolution** value to compensate for tearing, but the render times increase. **Note:** Correspondence points may be used to improve the warping accuracy in a specific part of the curve if turning this value up too high causes performance problems.
boundary curve resolution	Adjusts the number of interpolated points on **boundary** and **hard boundary** curves. Higher values stop the warp from filtering through the boundary curves but sacrifice speed, and vice versa.
preview resolution	Improves the accuracy of the preview at higher values and the rendering speed at lower values.
Classic warping	Set the type of warp function to use: • **disabled** - uses an updated quadratic warp function that copes well with overlapping control points, but has a more local warp effect.

Control	What it does
	• **enabled** - employs a warping function from previous versions of Nuke that has a more global warp effect, but doesn't cope well with overlapping control points.
filter	Select the appropriate filtering algorithm (see Choosing a Filtering Algorithm).

Using Boundary Shapes and Pins to Limit Warp

You may notice that pixels in areas around warping are also moved to accommodate the change, but the distortion lessens the further you get from the repositioned points. If you don't want a nearby area distorted, you can:

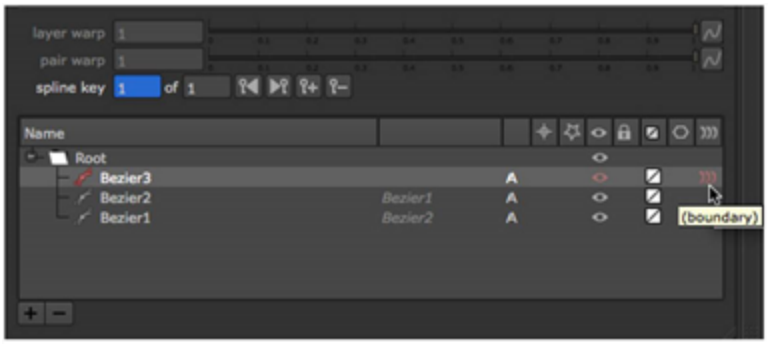
• Designate a curve or shape as a boundary to control or eliminate warp.

• Place pins in the areas you want to limit warp.

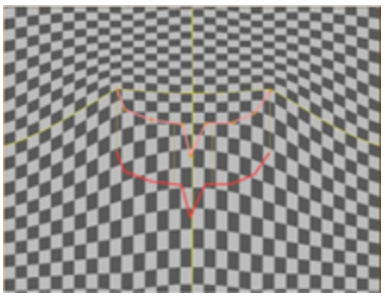
Note: Boundaries only affect the input they are drawn on, **A** or **B**, not both.

• **Boundary** - generally open splines helping to control warp rather than eliminate it.

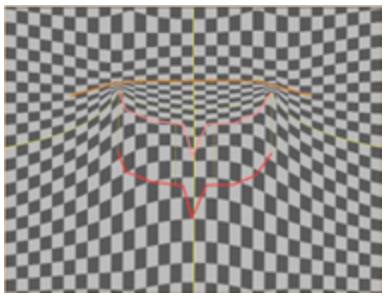
To define a boundary for an existing curve, enable **boundary** in the curves list, as shown below.



Using the example curves from Mr. Lion above, you could control the warp by adding a curve and designating it as a boundary:



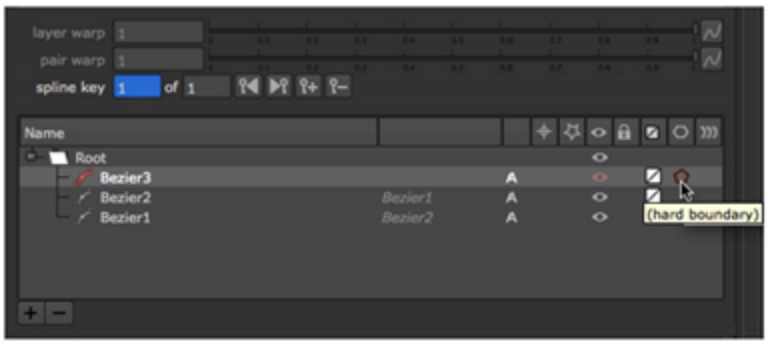
The source and destination curves.



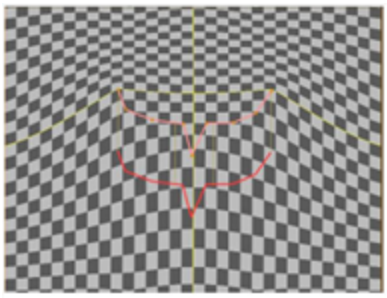
The same curves, but with a boundary above the destination to limit warp.

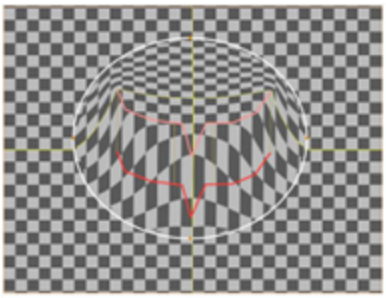
- **Hard Boundary** - closed shapes that eliminate warp outside the area that they enclose.

  To define a hard boundary for an existing shape, enable **hard boundary** in the curves list, as shown below.



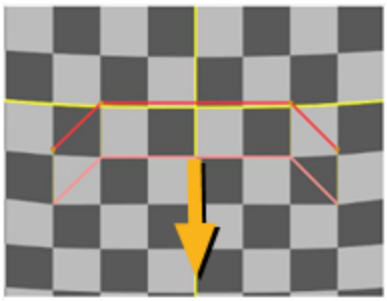
  The effect of a hard boundary on a warp is quite distinct - the hard boundary, colored white by default, limits the warp effect to entirely within the closed shape:

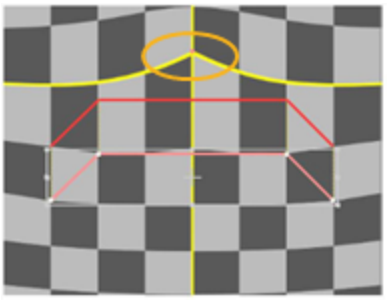




- **Pins** - placing a pin creates a single point source and destination curve, effectively limiting warp in a very localized way.

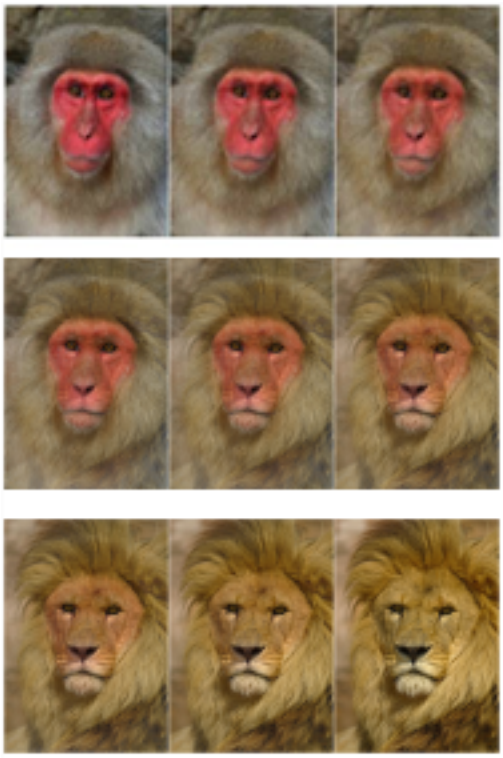
  The effect of a pin on a single joined curve is shown below. The warp direction is shown on the left and the pin's effect on the right.





Morphing

Morphing refers to dissolving two images together so that the subject of one image seems to change shape and turn into the subject of the other through a seamless transition. A morph can be easily noticeable or very subtle. An example of a noticeable morph would be a man turning into a woman or one animal turning into another, whereas a transition from an actor to his stunt man would result in a much more subtle morph.



An image of a monkey turning into an image of a lion.

Morphing can be a time-consuming task, but it can be made easier by good advance planning of the shots. The more similar the characteristics, position, and movement of the subjects you want to morph are, the easier it is to morph them together. If the position and the movement of the subjects do not match, however, you can try to reposition and retime the clips before morphing them together. To do so, use the nodes described in Transforming Elements and Temporal Operations. You can also use the Tracker and PlanarTracker nodes to track the features you want to morph over time or to stabilize your clips before morphing them together. See Transforming Warps for more information.

Below, we first discuss morphing images using the GridWarp node and then using the SplineWarp node.

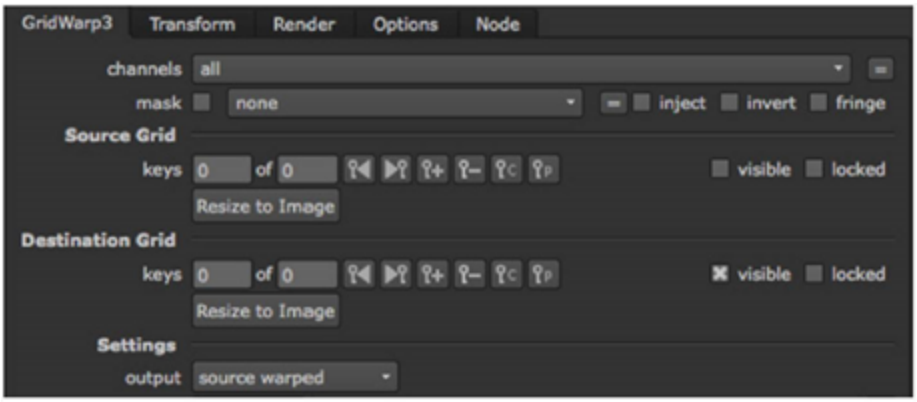
Morphing One Image into Another Using the GridWarp Node

To morph one image into another using the GridWarp node:

1. Select **Image** > **Read** to import the two images you want to morph together.
2. If the images do not have the same resolution, insert Reformat nodes after them. For more information, see Reformatting Image Sequences or Reformatting Elements.
3. Select **Transform** > **GridWarp** to insert a GridWarp node into your script.
4. Connect the source image (the image you want to turn into another image) to the **src** input of the GridWarp node, and the destination image (the image you want the source image turned into) to the **dst** input. Connect a Viewer to the GridWarp node.



5. To make the grids the same size as the input images, click the **Resize to Image** buttons under both **Source Grid** and **Destination Grid**.



6. In the GridWarp **Settings** controls, set **output** to **source** to display the source image and grid.
7. Adjust the grid to follow key features on the **source** image. For example, if you're morphing an animal, you might consider the eyes, nose and mouth.



8. In the GridWarp **Settings** controls, set **output** to **destination** to display the destination image and grid.

Note: If the **source** and **destination** images are similar, you can copy the **source** grid onto the **destination** grid using the copy and paste buttons.

9. Adjust the grid to follow key features on the **destination** image.



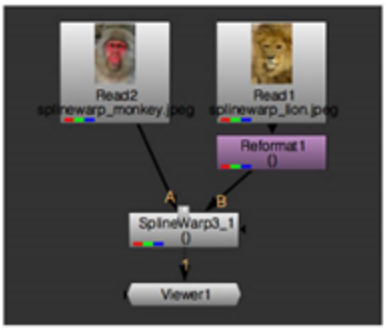
10. In the GridWarp **Settings** controls, set **output** to **morph** to display both grids and activate the **warp** and **mix** sliders.
11. Scrub to the frame where you want the morph to begin. Bring the **warp** slider down to 0 (the source image). Click on the animation button and select **Set key**. Then, scrub to the frame where you want the morph to end and set **warp** to 1 (the destination image).
12. Animate the **mix** control in the same way. Scrub to the frame where the morph starts, and set **mix** to 0 (the source image). Click the animation button and select **Set key**. Scrub to the frame where the morph ends, and enter 1 (the destination image) as the **mix** value.

    Play through the sequence in the Viewer and you'll see that the **source** image is morphed into the **destination** image.

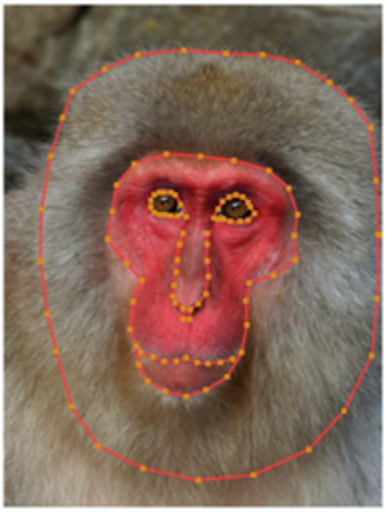
Morphing One Image into Another Using the SplineWarp Node

To morph one image into another using the SplineWarp node:

1. Select **Image** > **Read** to import the two images you want to morph together.
2. If the images do not have the same resolution, insert Reformat nodes as required. For more information, see Reformatting Image Sequences or Reformatting Elements.
3. Select **Transform** > **SplineWarp** to insert a SplineWarp node into your script.
4. Connect the source image (the image you want to turn into another image) to the **A** input of the SplineWarp node, and the destination image (the image you want the source image turned into) to the **B** input. Attach a Viewer to the SplineWarp node.

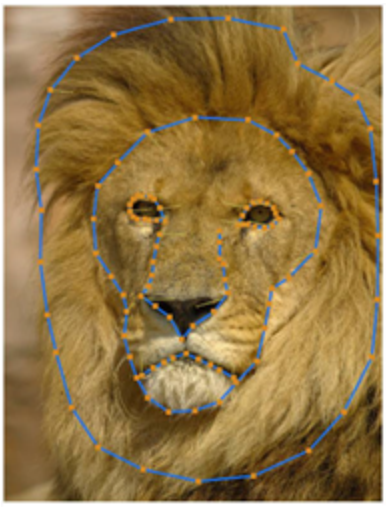


5. In the SplineWarp node's controls, switch **output** between **A** and **B** or click A and B in the Viewer tools. You should see both images in the Viewer.
6. Identify some features that are similar in the source and the destination images. For example, if you are morphing together images of people or animals, these features might include their eyes, noses and mouths as well as the outlines of their faces and heads.
7. Switch to input **A** to draw curves around the key features you identified in the previous step. For more information on how to create the curves, see Warping an Image Using the SplineWarp Node.



You may find it useful to name the entries in the curves list, left_eye, right_eye, etc. Meaningful names can help you later on when you're using the **Join** tool to link the source and destination shapes.

8. Switch to input **B** and draw curves in a similar fashion. Draw the same amount of curves, on similar features, to make the morph as accurate as possible.

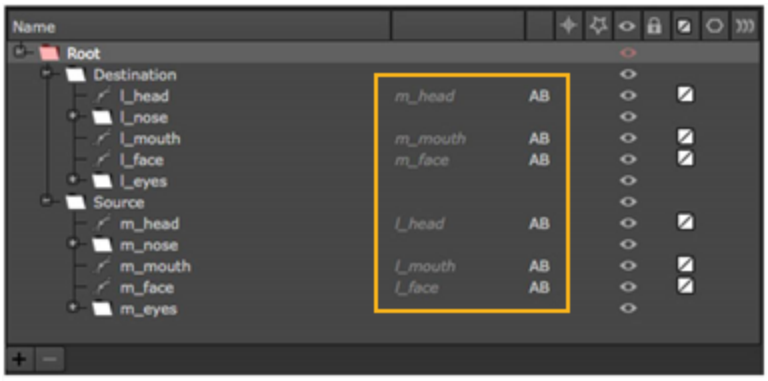


9. Switch the Viewer to output **ABmorph** , which displays both sets of curves, and activate **Join** in the Viewer tools.
10. Work through the **A** input shapes, linking them to their **B** input counterparts. Try to link similar points, such as the corners of the eyes or the side of the mouth.

Note: You can invert the source and destination shapes using the **Reverse** tool.



As you join shapes, the curves list updates showing the linked pairs.



11. Scrub to the frame where you want the morph to begin. Bring the **mix** slider down to **0** (input **A**). Click on the animation button and select **Set key**.
12. Set keyframes for **root warp** in the same way.
13. If you're using stills, as in this example, scrub the playhead to the number of frames required for the morph. Otherwise, scrub to the end of the sequence.
14. Set the **mix** and **root warp** sliders to **1** (input **B**).
15. Press **Q** in the Viewer to disable the spline overlay and press play.

    Input **A** is morphed into input **B**.

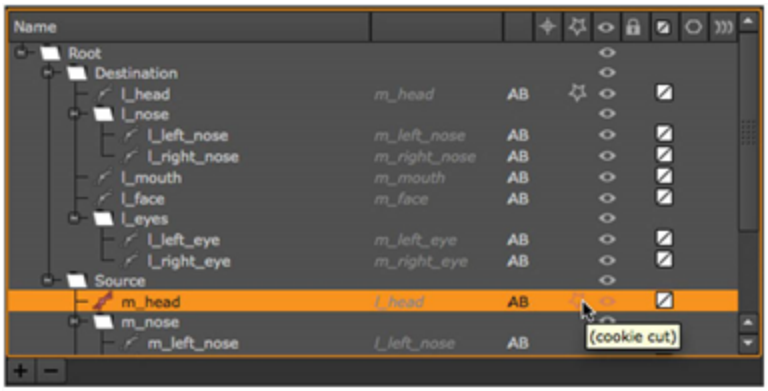
Using the cookie-cutter to create traveling masks

The **curves** list cookie-cutter is primarily designed to overlay a morph between two images, which may not include an alpha channel, on top of a sequence by creating a "traveling mask" automatically between closed cookie-cutter shapes.

For example, the monkey-lion morph demonstrated previously could be merged with a sequence, replacing a face with a monkey face that is morphed into a lion over time.

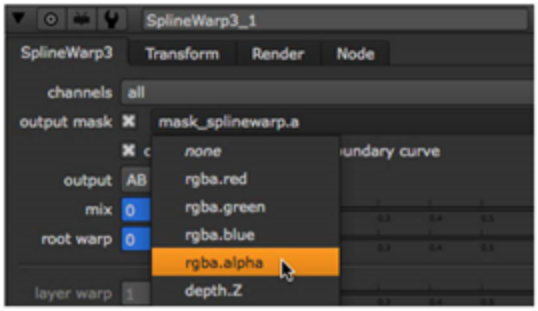
Note: The cookie-cutter only works with closed shapes from the **curves** list.

1. Designate a pair of closed shapes as cookie-cutters by clicking the cookie-cutter icon in the curves list.

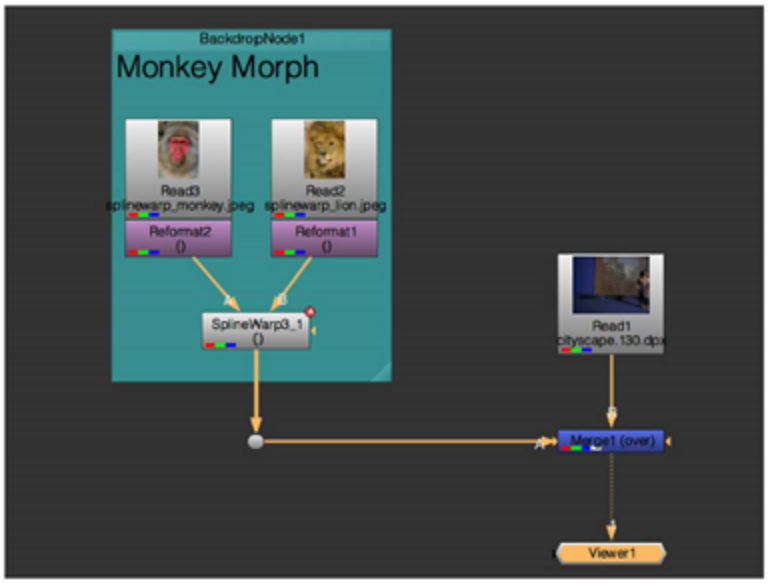


In this instance, enabling cookie-cut for the monkey's head, **m_head**, also designates the lion's head, **l_head**, as a cookie-cutter shape.

2. Use the **output mask** dropdown to select the channel where the traveling mask is stored, or use the **mask_splinewarp.a** default.



3. Animate the morph to follow the required background element as described in Transforming and Animating Warps and Morphs.
4. Finally, merge the output of the SplineWarp node with your sequence. A simple script might appear as follows:



You can confirm the presence of the traveling mask by selecting the **output mask** channel in the Viewer dropdown and then pressing **M** to display the **Mat** channel. You should see an automatically created mask that matches the morphed shape.

Transforming and Animating Warps and Morphs

Unless you are warping or morphing a still image, you'll probably need to transform or animate warp operations. The Tracker and PlanarTracker nodes can help automate transformations, and you can use Expressions and the GridWarp and SplineWarp Properties panel controls as well. See Transforming Warps and Animating Warps.

Transforming Warps

You can transform grids and splines in the same way as any other element in Nuke (see Temporal Operations), but you can also link control points to the Tracker and PlanarTracker nodes to automate any transforms you require. Once you have a suitable track, you can use the animation data to control points on grids and curves.

Using the Tracker Node

1. To generate tracking data, see Tracking and Stabilizing .
2. Make sure both the Tracker and the GridWarp/SplineWarp properties panels are open, then either:
   - **Ctrl/Cmd**+drag the animation button from the Tracker node properties on top of the individual grid/curve point in the Viewer. When you release the mouse, the point follows the animation from the Tracker node,

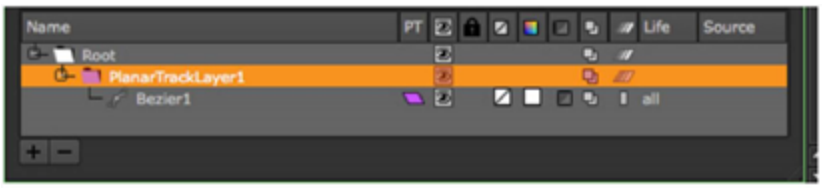
   OR

   - Right-click on the required control point and select **Link to** > **Tracker1** and select the required transform type.

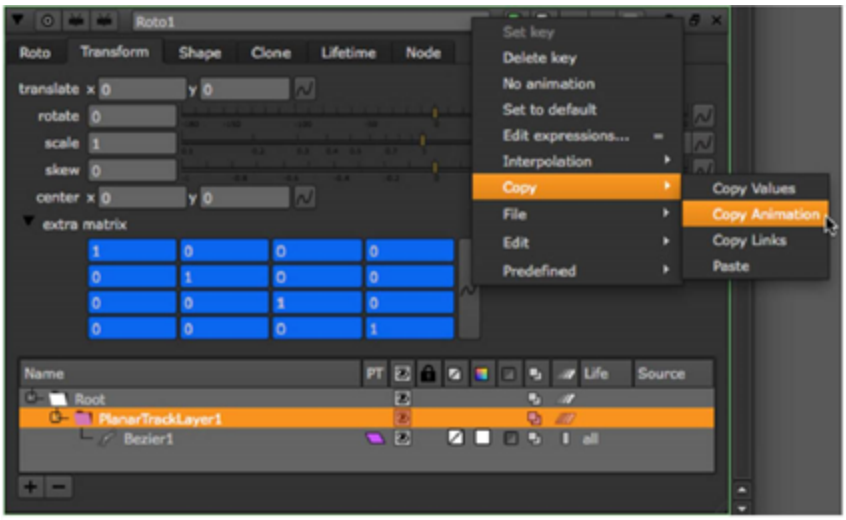
Using the PlanarTracker Node

1. To generate PlanarTracker data, see Tracking with PlanarTracker.

2. Use the **Draw Boundary** button to create a source grid roughly the same shape as the tracked plane, or draw similar splines.
3. On the GridWarp/SplineWarp and RotoPaint **Transform** tabs, open up the **Transform** tab **extra matrix**.
4. On the RotoPaint **Transform** tab, select **PlanarTrackerLayer1** in the curves list.



5. Copy and paste the **extra matrix** animation from RotoPaint to the GridWarp/SplineWarp **extra matrix**.



6. Play through the sequence to check that the grid or splines follow the plane.

Using Expressions with GridWarp

As well as using other nodes to control points, you can do the inverse, using expressions to link control points and their tangents to other operations.

The expressions take the following form:

- GridWarp3_1.destination_grid_col.1.2.pos.x
- GridWarp3_1.destination_grid_col.2.2.tangent.1.x

Note: Control points are indexed starting from the bottom left 1.1, in x.y order, and tangent points are labelled 0, 1, 2, and 3 corresponding to north, south, east, and west respectively.

It's worth mentioning that in order to get the true position of a tangent, not its offset value, you need to concatenate the **pos** and **tangent** expressions. For example:

GridWarp3_1.destination_grid_col.2.2.pos.x + GridWarp3_1.destination_grid_col.2.2.tangent.4.x

Note: You can only modify tangent positions using the Curve Editor.

For more information, refer to Expressions.

Animating Warps

In the GridWarp and SplineWarp node's properties panels, there are controls for animating both the source and the destination grids or shapes. Here, we first look at the GridWarp node and then the SplineWarp node. The instructions assume you know how to warp a still image using these nodes (if not, refer to Warping and Warping an Image Using the SplineWarp Node).

To Animate a Warp Using the GridWarp Node

1. While viewing the source image, adjust the source grid as necessary (see Warping for information on how to do this).

   If **autokey** is enabled, key frames are added every time you adjust the grid. Otherwise, click the **add key** button  under **Source Grid**. This saves the current grid as a key shape.
2. Move to a new frame and adjust the source grid accordingly. A new key shape is set automatically.
3. Repeat the previous step as necessary. If you need to delete a key shape, scrub to the frame where you set it and click **delete key**  under **Source Grid**.
4. Hide the source grid and display the destination grid.
5. While viewing the output of the GridWarp node, adjust the destination grid until you are happy with the warp.

   If **autokey** is enabled, key frames are added every time you adjust the grid. Otherwise, click the **add key** button  under **Destination Grid**. This saves the current grid as a key shape. Click the **set** button under **Destination Grid**. The current grid is saved as a key shape.
6. Move to a new frame and adjust the destination grid again. The modified grid is automatically set as a key shape.
7. Repeat the previous step until you are happy with the animated warp.

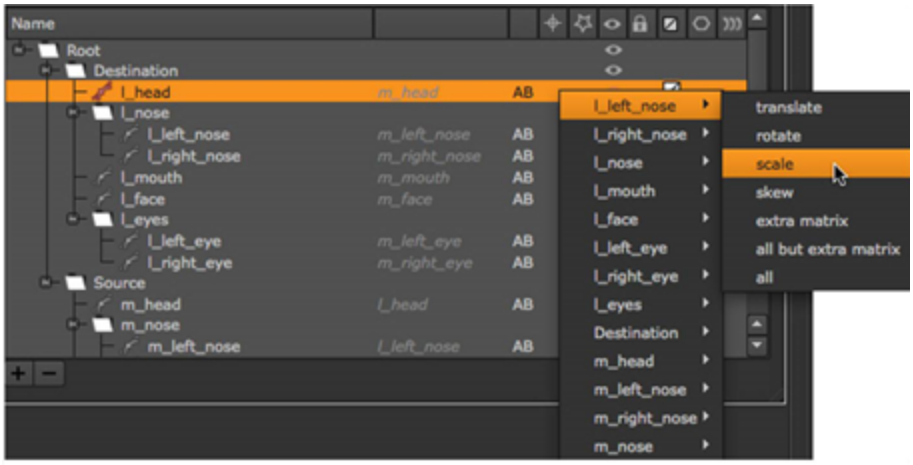
To Animate a Warp Using the SplineWarp Node

1. Create the required curves as described under Warping an Image Using the SplineWarp Node.
   If **autokey** is enabled, key frames are added every time you adjust a shape. Otherwise, click the **add key** button  with **A warped** selected. This saves the current shape as a key frame.
2. Move to a new frame and adjust the shapes accordingly. New key shapes are set automatically.
   Repeat the previous step as necessary. If you need to delete a key shape, scrub to the frame where you set it and click **delete key**  with **A warped** selected.

Linking Transforms Using the SplineWarp Node

Transforms applied in the properties panel **Transform** tab can be expression linked with other shapes in the **curves** list using the **transform linked** toggle.

1. Create the required transform on the source shape.
2. Click the **transform linked** toggle in the curves list.
   A menu displays the available shapes and transform types.



   You can link all transform types associated with the source shape or be selective, such as **all but extra matrix**.
3. In the example, any **scale** keyframes added to the **l_left_nose** shape are applied equally to the **l_head** shape.
4. If you want to remove a link, click the **transform linked** toggle for the required shape, select the link, and click **remove link**.

Temporal Operations

This chapter explains the temporal or time-based operations in Nuke. Learn how to distort time (that is, slow down, speed up, or reverse clips), apply motion blur, and perform editorial operations like slips, cuts, splices, and freeze frames.

Quick Start

Here's a quick overview of the workflow:

1. If you want to speed up or slow down your clip:
   - You can use the Retime node (**Time** > **Retime**). If you want, you can also add a FrameBlend (**Time** > **FrameBlend**) node before your retime to create smoother retiming results. For more information, see Simple Retiming and Interpolation.
   - For more advanced retiming operations and adding motion blur, you can use the OFlow node (**Time** > **OFlow**), or if you have a license for NukeX, the Kronos node (**Time** > **Kronos**). For more information, see OFlow Retiming and Kronos.
   - Sometimes you'll want to create speed-up and slow motion within a clip without altering its actual length. This is called warping and you can use the Retime node (**Time** > **Retime**) to do it. For more information, see Warping Clips.

   After retiming your clip, you may need to adjust the script's global frame range and playback speed (frames-per-second). For more information, see Global Frame Range and Speed .
2. If necessary, you can use the TimeBlur node to apply motion blur to garbage masks that are tracked to a fast moving feature. For more information, see Applying a TimeBlur Filter.
3. If you're looking to perform editorial operations:
   - You can use the TimeOffset (**Time** > **TimeOffset**) and TimeClip (**Time** > **TimeClip**) nodes to slip clips. For more information, see Offsetting and Slipping Clips.
   - The FrameRange (**Time** > **FrameRange**) node allows you to cut clips. For more information, see Cutting Clips.
   - You can use the AppendClip (**Time** > **AppendClip**) node to splice your clips. For more information, see Splicing Clips.
   - Using the FrameHold (**Time** > **FrameHold**) node allows you to set the output of a clip or scene to a particular frame or frame interval. For more information, see Freeze Framing Clips.

Note: Most temporal operations in Nuke can only be applied to 2D compositions, but the FrameRange, TimeOffset, and FrameHold nodes are applicable to 3D nodes, such as geometry.

Distorting Time

Time distortion changes the length of time required to playback a clip in your composite. These operations generally fall under one of two categories: *retiming* and *warping*.

Retiming is the process of slowing playback by adding frames, or accelerating playback by subtracting frames.

Warping is the process of slowing down, speeding up, or even reversing playback on a clip without necessarily altering the overall length.

Tip: When working with temporal operations, it helps to attach a Viewer to the retiming or warping node so you can see the effect of your changes.

Simple Retiming

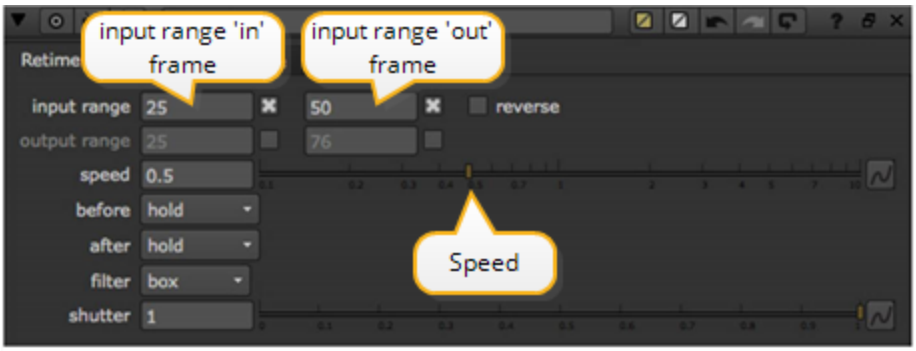
Nuke's Retime node lets you change the playback time for all the frames in a clip or for range of frames within the clip. You can also use it to reverse the clip playback. It does this by dropping or duplicating frames. For higher quality retiming see OFlow Retiming.

To Retime All Frames in a Clip

1. Select **Time** > **Retime** to insert a Retime node into your script.
2. Enter a value in the **speed** parameter. Values higher than 1 increase playback speed; values less than 1 decrease playback speed.
3. Check the **reverse** box if you want to play the clip backwards - making the last frame the first, the first frame the last, and so on.
4. Increase the **shutter** parameter to enable frame-blending (For more information, see Interpolation).

To Retime a Range of Frames in a Clip

1. Select **Time > Retime** to insert a Retime node into your script.
2. Check the boxes for **input range** and enter the “in”and “out” frames.



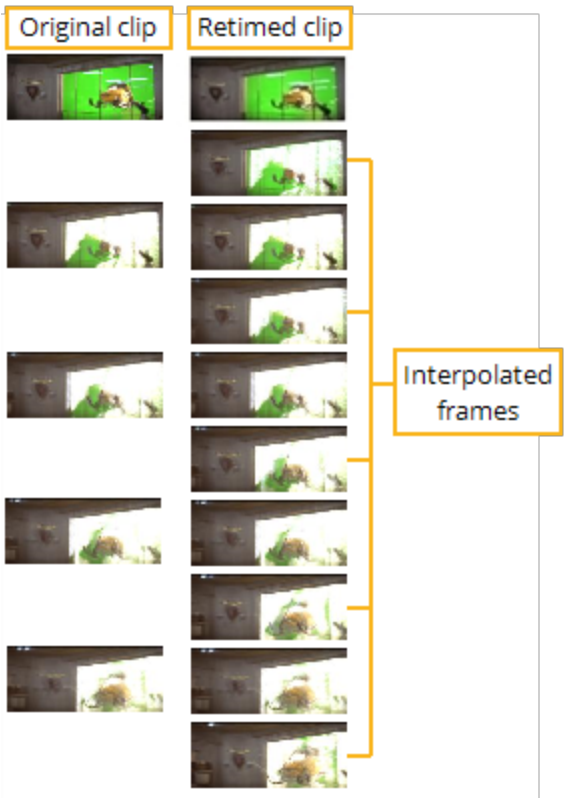
   For example, if your original clip is 50 frames, but you want to only retime the last half, you would input **25** for the in point and leave the out point at **50**.
3. Check the box for **output range** and enter the “in” and “out” frames to retime to a specific clip length.

   or

   Enter a factor in the **speed** parameter and Nuke calculates the **output range** for you. Values higher than 1 increase playback speed; values less than 1 decrease playback speed.
4. Check the **reverse** box to invert the selected frame range.
5. Increase the **shutter** parameter to enable frame-blending.

Interpolation

Time distortions that slow down a clip require the creation of additional, or *interpolated*, frames. For example, suppose you want to slow down the collision, shown in the clip below, by a factor of two. This requires the creation of one interpolated frame for every existing frame in the clip.



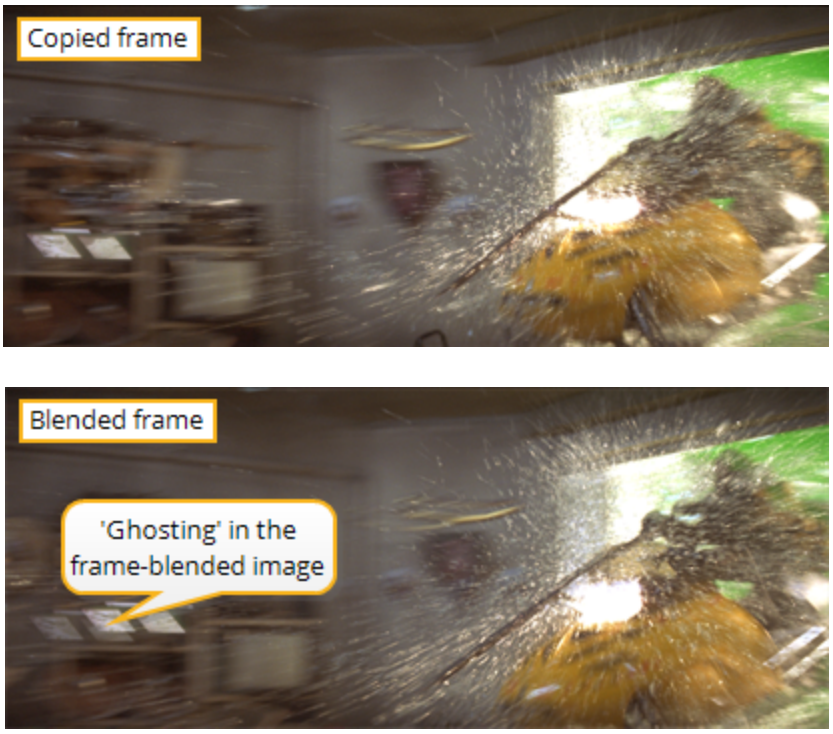
Interpolating frames.

The simplest way for Nuke to interpolate is to duplicate existing frames and increase the length of the clip - this is the default method of interpolation. However, this method can create jittery playback, especially when the image depicts very fast motion and the clip is retimed to be considerably longer than its original length. For such cases, Nuke provides different nodes for smoothly interpolating the frames.

Frame-Blending

The FrameBlend node interpolates frames by generating an additive composite of the frames that precede and follow it, rather than creating mere copies between the existing frames.

Here is an example of the difference between frame-copy and frame-blend interpolation. In the first frame of the figure below, you see a copy of the preceding frame. In the second frame, you see a new image generated by blending the previous and subsequent frames.



Frame-copied versus frame-blended interpolation.

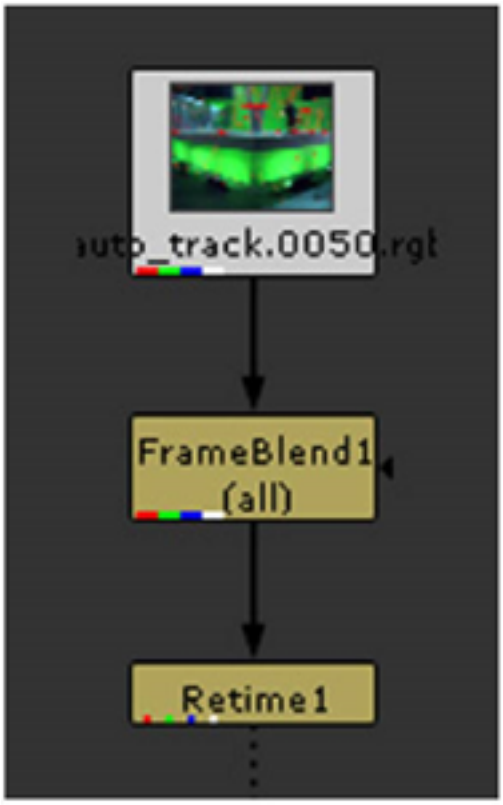
The latter method creates “ghosting” around all fast moving features (the window frame and the pages on the desk, for example). This may look odd when viewed as part of a still frame, but contributes to smoother motion during actual playback.

You can enable frame-blending by manipulating the **shutter** value of a retiming node. Higher shutter values generate more frame-blending. Or, you can insert a FrameBlend node before the temporal effect you want to influence. The below figure shows an example of frame-blending with the Retime node.

To Insert a FrameBlend Node

1. Select **Time** > **FrameBlend** from the toolbar.

   Remember to place it upstream from the temporal effect you want to influence.



2. Enter the number of frames to blend in the **Number of frames** field.

   or

   Check the **Custom** box and enter the starting and ending frames that you want to blend. To use the input range as your custom frame range, click **Input Range**.
3. If necessary, check **Foreground matte** and select the channel to limit the blending effect.

The **output Image count to** option saves a floating point alpha image to a channel you specify; the result indicates the number of images that contributed to each pixel of the matte. To normalize the alpha, divide the number 1 by the number of frames averaged, and then multiply the alpha channel by this result. You can also use the inverse of this matte for additional degraining.

OFlow Retiming

The OFlow node generates high-quality retiming operations analyzing the movement of all pixels in the frames and then rendering new "in-between" images based on that analysis. This node can also add motion blur or enhance the existing motion blur in the image.

Tip: We recommend using NukeX's Kronos node instead, as it includes an improved motion estimation algorithm. For more information, see Kronos.

To Retime with OFlow

1. Select **Time** > **OFlow** to insert an OFlow node after the sequence you want to retime.

   The **Input Range** is set automatically by the source clip when you first create the node. After that, it is only updated if you click **Reset**.
2. If you intend to use **Output Speed** or **Frame** timing, attach a Viewer to the output of the OFlow node,

   OR

   If you intend to use **Input Speed** timing, attach a Viewer before the OFlow node.
3. By default, OFlow is set to perform a linear half-speed slow down using **Timing** > **Output Speed**.
4. Enable **Use GPU if available** to render output on the **Local GPU** specified, if available, rather than the CPU.



Note: For more information on GPU support, see the Release Notes for your version of Nuke.

5. Select the channels you want to apply the retime to using the **channels** dropdown. OFlow retimes **all** channels by default.
6. To adjust the slow down or to speed the sequence up, enter a new value for the **Output** or **Input Speed** control. Values below 1 slow down the clip. Values above 1 speed up movement. The default value, 0.5, created the half-speed slow down. Quarter-speed would be 0.25.
7. Alternatively, you can describe the retiming in terms of 'at frame 100 in the output clip, I want to see frame 50 of the source clip'. To do so, set **Timing** to **Source Frame**. Go to a frame in the timeline, and set **Frame** to the input frame you want to appear at that output position. You'll need to set at least two key frames for this to retime the clip. For example, to slow down a 50 frame clip by half, you can set **Frame** to 1 at frame 1, and to 50 at frame 100. To do a four times slow down, set **Frame** to 1 at frame 1, and to 25 at frame 100.
8. If you'd like to add motion blur to your retimed footage, proceed to Adding Motion Blur.

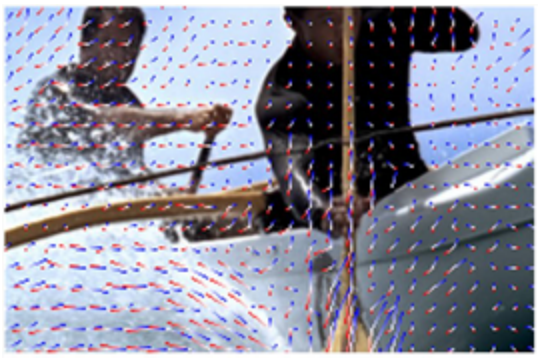
Tip: Some Nuke users may remember nodes called OpticalFlow and OptiFlow. OFlow replaces these nodes. It can be used for retiming and adding motion blur, but it does not have the capability to output motion vectors. To output motion vectors, you could use VectorGenerator (included in NukeX and Nuke).

For retiming and motionblur, we recommend using the Kronos node instead of OFlow. For more information, see Kronos.

Refining the Results

To refine the results:

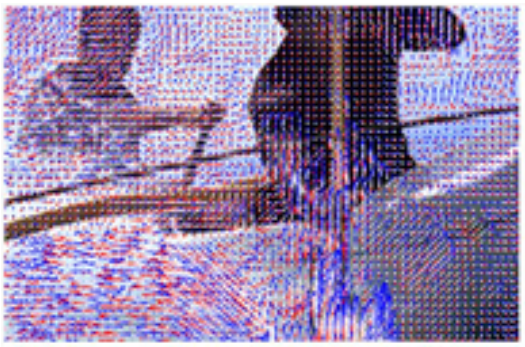
1. To have the motion vectors displayed in the Viewer, expand the **Advanced** control and enable **Overlay Vectors**. Forward motion vectors are drawn in red, and backward motion vectors in blue.



Overlay Vectors enabled.

Note: Motion vectors displayed in the Viewer are added to your output if you don't turn off the overlay before rendering.

2. To set the spacing between motion vectors displayed on the Viewer, adjust **Vector Spacing**. The default value of 20 means every 20th vector is drawn. Note that **Vector Spacing** only affects the Viewer overlay rather than the retimed result.



A **Vector Spacing** value of 4.



A **Vector Spacing** value of 40.

3. Adjust **Vector Detail** to vary the density of the vector field. The larger vector detail is, the greater the processing time, but the more detailed the vectors should be. A value of 1 generates a vector at each pixel. A value of 0.5 generates a vector at every other pixel. For some sequences, a high vector detail near 1 generates too much unwanted local motion detail, and often a low value is more appropriate.

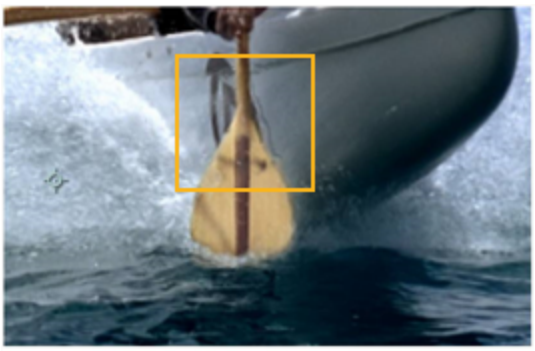


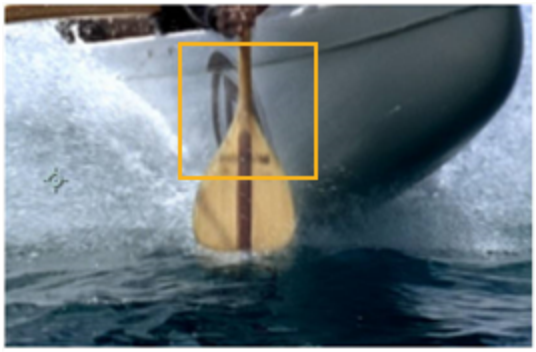
Areas of unwanted local motion detail.



Lower **Vector Detail** is more appropriate in this case.

4. Vector fields usually have two important qualities: they should accurately match similar pixels in one image to another and they should be smooth rather than noisy. Often, it is necessary to trade one of these qualities off against the other. A high **Smoothness** misses lots of local detail, but is less likely to provide you with the odd spurious vector. A low **Smoothness** concentrates on detail matching, even if the resulting field is jagged. The default value of 1.5 should work well for most sequences.





Jagging as a result of a low **Smoothness** value.

Minimal jagging as a result of a high **Smoothness** value.

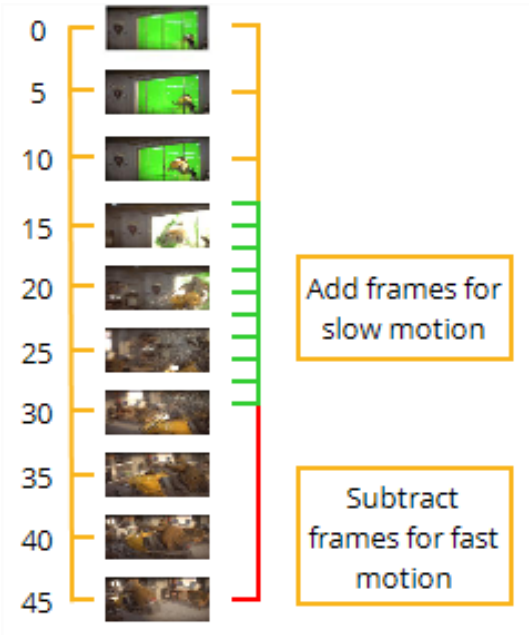
5. Set the type of resampling applied when retiming:
   - **Bilinear** - the default filter. Faster to process, but can produce poor results at higher zoom levels. You can use **Bilinear** to preview a retime before using one of the other resampling types to produce your output.
   - **Lanczos4** and **Lanczos6** - these filters are good for scaling down, and provide some image sharpening, but take longer to process.
6. If the overall brightness in your **Source** footage changes between frames, enable **Flicker Compensation**. This allows OFlow to take into account variations in luminance and overall flickering, which could otherwise cause problems with your output.

   Examples of variable luminance include highlights on metal surfaces, like vehicle bodies, or bodies of water that reflect light in unpredictable ways.

   Note that using **Flicker Compensation** increases rendering time.
7. In order to reduce processing time, much of the motion estimation is done on luminance only - that is, using monochrome images. In most cases this is perfectly acceptable, but the parameters in the **Tolerances** group allow you to concentrate on a particular feature in an image by adding bias to individual colours. You may, for example, wish to increase **Weight Red** to allow the algorithm to concentrate on getting the motion of a primarily red object correct, at the cost of the rest of the objects in a shot.

Warping Clips

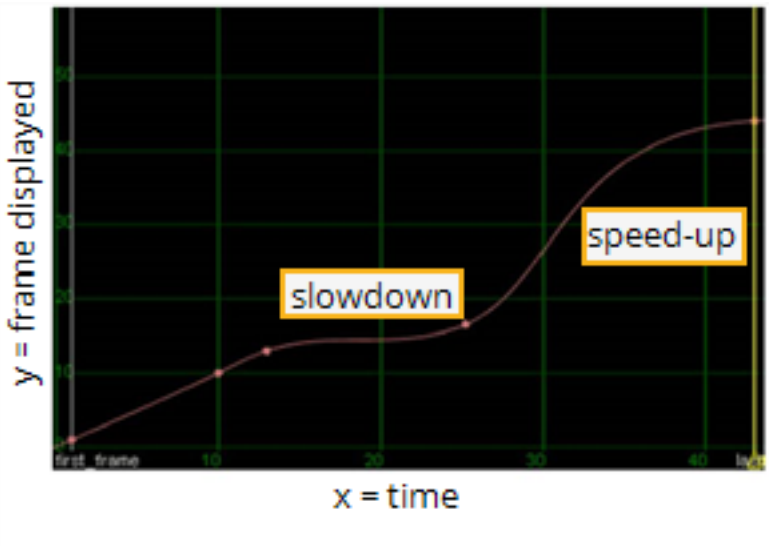
Warping refers to slowing down, speeding up, or even reversing select frames in a clip without necessarily altering its overall length. Otherwise stated, warps add, subtract, or reverse the temporal detail in a range of frames within a clip. For example, the figure below depicts a snowmobile clip (down-sampled to just ten frames for easy representation) that we might want to warp.



Planning the time warp.

One way - in fact, kind of the classic way - to warp this clip would be to play the frames just prior to the collision at their original rate, the frames involving the collision in slow motion, and the frames after the collision in fast motion.

You could achieve such a warp by sculpting the curve in Nuke's TimeWarp curve, which is a part of the Retime node's parameters, to look something like the one depicted below.



Editing the warp curve.

The basic "rules" for editing the warp curve are as follows:

• To slow down motion, decrease the slope of the curve.

- To speed up motion, increase the slope of the curve.
- To reverse motion, create a downward sloping portion on the curve (a dip, in other words).

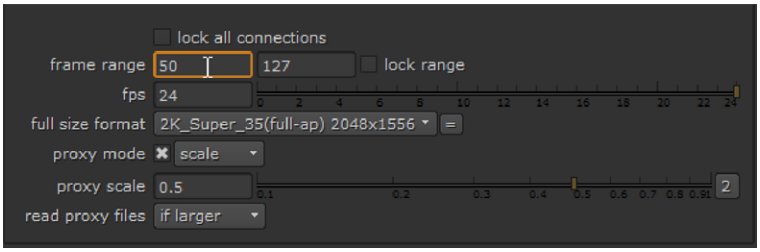
To Warp a Clip

1. Click **Time > Retime** to insert a Retime node into your script.
2. Click the **TimeWarp** tab to reveal the TimeWarp curve.
3. Attach a Viewer to this node, so you can see the effect of your changes.
4. Sculpt the TimeWarp curve according to the rules above. (**Ctrl/Cmd**+**Alt** click to insert keyframe knots on the curve; **Ctrl/Cmd**+drag to reposition keyframe knots; **Ctrl/Cmd**+drag to rotate a keyframe knot control handles.)
5. If you want to enable frame blending on the output, either input a value larger than one in the Retime node's **shutter** parameter, or insert a FrameBlend node prior to the Retime node.

Global Frame Range and Speed

Nuke automatically adjusts the timeline of every Viewer window you open to show the "in" and "out" frames for the clip you're viewing.

After you retime a clip in your compositing script, you may need to adjust the script's global frame range and playback speed (frames-per-second), to account for the retiming operations.



Adjusting the global frame range.

Select **Edit > Project Settings** (or press **S** over the Nuke window) and then enter the new **frame range** and **fps** in the Project settings properties panel.

Applying a TimeBlur Filter

When a fast-moving subject is recorded on film or video, its edges appear to smear as a result of the object's movement while the shutter is open. The longer the shutter remains open at each frame interval, the more obvious this effect. TimeBlur simulates this phenomenon by sampling its input at "divisions" times over "shutter" frames starting at the current frame plus "offset".

Time blur is commonly applied to garbage masks that are tracked to a fast moving feature. The time blur averages the incoming mask image over the shutter period, to better match the motion blur in the original image and creating a more convincing integration.

Tip: You can also use the Roto and RotoPaint **Global Blur** controls to achieve the same results. See Adding Motion Blur for more information on Roto and RotoPaint motion blur.

Applying Motion Blur to a Clip

To apply motion blur to a clip:

1. Click **Time** > **TimeBlur** to insert a TimeBlur node into your script. Place it downstream from the element to which you want to apply motion blur.
2. In the TimeBlur properties panel, set **divisions** to the number of times you want to sample the input over the shutter time. For images with fast-moving content higher values are necessary to eliminate "steppiness" or banding in the output.
3. Set **shutter** to equal the span of time (in frames) over which the input should be sampled. A shutter time of .5 is typical and would correspond with a camera shutter of 180 degrees.
4. Set the **shutteroffset** to control when the sampling of the input starts relative to the frame being rendered, analogous to when the camera shutter opened to capture corresponding film or video footage you might have at the same frame. You may need to adjust this by eye to align, for example, a garbage mask with an underlying feature.

Tip: You may find that using TimeBlur on all the upstream nodes in your composition can be unnecessary and very time consuming. In these cases, you can use NoTimeBlur node to limit the number of nodes to which you're applying TimeBlur. Just insert the NoTimeBlur node in your node tree above the TimeBlur and any nodes you want the TimeBlur node to process.

Editing Clips

As a node-based system, Nuke doesn't have a timeline. Nevertheless, you can still perform editorial operations that you might associate with a timeline. You can slip clips (move them forward or backward in time), cut them, or splice them to other clips.

Offsetting and Slipping Clips

Offsetting a clip refers to moving the clip backwards or forwards in time. In Nuke, you can offset clips using the TimeOffset and TimeClip nodes.

Unlike TimeClip, the TimeOffset node can also be used with 3D nodes, for example, if you want to offset camera times.

To Offset a Clip Using the TimeOffset Node

1. Click **Time** > **TimeOffset** to insert a TimeOffset node into your script. (Place it downstream from the element to which you want to offset.)
2. Attach a Viewer to this node, so you can see the effect of your changes.
3. In the TimeOffset properties panel, check **reverse input** if you want to invert the clip (make the last frame the first, and so on).
4. In the **time offset (frames)** field, type the number of frames by which you want to offset the clip. Enter a negative value to subtract frames from the head of the clip. Enter a positive value to add frames to the head of the clip.
5. Adjust the script length for the new output range. Select **Edit** > **Project Settings**, and enter **frame range** values that match the output range you specified.



Note: It's not mandatory that you adjust the script's frame range after slipping the clip. If you don't, the Viewer fills the empty frames at the tail of the clip by holding on the last frame.

Tip: If you are using 3D nodes, such as cards, spheres, or cubes, together with a TimeOffset node, you can view the object in just a specified frame range by setting the **frame range** control for the object, and attaching a Viewer node to the TimeOffset node. Be sure to set the timeline range to **Input** to view the offset.

Tip: Using TimeOffset, you can also offset clips directly in the Dope Sheet. See Animating Parameters for more information.

Slipping a clip refers to changing the content of the clip that is seen, but not the position or duration. In Nuke, you can slip clips using the TimeClip node.

Just like TimeOffset, the TimeClip node also lets you move clips forwards or backwards in time and reverse the order of frames in a clip. In addition to this basic functionality, you can set the frame range for the clip, set what happens to frames outside of this frame range, fade the clip to or from black, and set expressions to adjust the node's behavior. Using TimeClip, you can also slip the clip directly in the DopeSheet, which can be handy if you prefer editing clips on a timeline.

To Slip a Clip Using the TimeClip Node

1. Click **Time** > **TimeClip** to insert a TimeClip node into your script. Place it downstream from the element that you want to adjust.
2. Attach a Viewer to the TimeClip node to see the effect of your changes.
3. In the **Fade In** and **Fade Out** fields of the TimeClip properties panel, type the number of frames, if any, you want to fade to or from black.
4. Set the new **frame range** for your clip by entering the first and last frame numbers.
5. In the **before** dropdown menu, set how Nuke should treat the frames before the first frame in the range (examples refer to a 20-frame sequence with a first frame value of 5):
   - **hold** - the first frame in the sequence is held until the first frame is reached. Example: 1, 1, 1, 1, 1, 2, 3, 4, etc.
   - **loop** - substitutes an equal number of frames, effectively creating a clip loop. Example: 17, 18, 19, 20, 1, 2, 3, 4, etc.
   - **bounce** - substitutes a reversed equal number of frames, creating a clip bounce. Example: 5, 4, 3, 2, 1, 2, 3, 4, etc.
   - **black** - frames are black until the first frame is reached.
6. In the **after** dropdown menu, set how Nuke should treat the frames after the **last** frame (examples refer to a 20 frame sequence with a last frame value of 5):
   - **hold** - the last frame in the sequence is held until the end of the sequence is reached. Example: 16, 17, 18, 19, 20, 20, 20, 20, etc.

- **loop** - substitutes an equal number of frames, effectively creating a clip loop. Example: 16, 17, 18, 19, 20, 1, 2, 3, 4, etc.
- **bounce** - substitutes a reversed equal number of frames, creating a clip bounce. Example: 16, 17, 18, 19, 20, 19, 18, 17, etc.
- **black** - frames are black from last until the end of the sequence is reached.

7. Use the **frame** dropdown to set the frame mode:
   - **expression** - Lets you enter an expression in the field on the right. The expression changes the relation between the current frame and the frame read in.

     For example, if your clip begins from image.0500.rgb and you want to place this first frame at frame 1 rather than frame 500, you can use the expression **frame+499**. This way, 499 frames are added to the current frame to get the number of the frame that's read in. At frame 1, image.0500.rgb is read in; at frame 2, image.0501.rgb is read in; and so on.

     Another example of an expression is **frame*2**. This expression multiplies the current frame by two to get the number of the frame that's read in. This way, only every other frame in the clip is used. At frame 1, image.0002.rgb is read in; at frame 2, image.0004.rgb is read in; at frame 3, image.0006.rgb is read in; and so on.
   - **start at** - Lets you enter a start frame number in the field on the right. This specifies the frame where the first frame in the sequence is read in. In other words, all frames are offset so that the clip starts at the specified frame.

     For example, if your sequence begins from image.0500.rgb and you enter **1** in the field, image0500.rgb is read in at frame 1. Similarly, if you enter **100** in the field, image0500.rgb is read in at frame 100.
   - **offset** - Lets you enter a constant offset in the field on the right. This constant value is added to the current frame to get the number of the frame that's read in.

     For example, if your clip begins from image.0500.rgb and you want to place this first frame at frame 1 rather than frame 500, you can use **499** as the constant offset. This way, 499 is added to the current frame to get the frame that's read in. At frame 1, image.0500.rgb is read in; at frame 2, image.0501 is read in, and so on.

     You can also use negative values as the constant offset. For example, if you use the value **-10**, Nuke subtracts ten from the current frame to get the frame that's read in. At frame 20, image.0010.rgb is read in; at frame 21, image.0011.rgb is read in; and so on.
8. To reverse the clip within the specified **frame range**, check **reverse**.
9. While the **original range** fields don't actually affect the output of the node, they help make things clear in the Dope Sheet. Open the Dope Sheet and enter some values here and observe how they affect the TimeClip lines.
10. If necessary, adjust the script length for the new output range. Select **Edit** > **Project Settings**, and enter **frame range** values that match the output range you specified.

Tip: Using TimeClip, you can also offset, trim, and slip clips directly in the Dope Sheet. See Animating Parameters for more information.

Cutting Clips

The FrameRange node allows you to set a frame range for a clip. This controls which frames a downstream AppendClip node uses from the input, which frames are displayed in the Viewer when the timeline range dropdown menu is set to **Input**, and which frames are sent to the flipbook.

To make an edit, first use this node to cut portions out of input sequences and then append the results together using the AppendClip node.

To Cut a Clip

1. Click **Time** > **FrameRange** to insert a FrameRange node into your script.
2. Attach a Viewer to this node, so you can see the effect of your changes.
3. In the **frame range** fields, then enter the appropriate in and out point values.

   For example, if your original clip is 50 frames but you want to use only the last 25 frames in your composite, you would enter **25** as the In point and leave the Out point at **50**.
4. Adjust the script length for the new output range. Select **Edit** > **Project Settings**, and enter **frame range** values that match the output range you specified.

Note: It's not mandatory that you adjust the script's frame range after cutting the clip.

Tip: Using FrameRange, you can also set the frame range for a clip directly in the Dope Sheet. See Animating Parameters for more information.

Splicing Clips

Splicing refers to joining clips head-to-tail, thus allowing action to flow from one shot to the next. When you splice clips, you have options for:

- Fading to or from black.
- Dissolving from the first to second clip.
- Slipping the combined clip in time.

To Splice Slips

1. Click **Time** > **AppendClip** to insert an AppendClip node into your script.
2. Attach its **1** and **2** inputs to the clips you want to join. (The clip attached to input 1 precedes the one attached to input 2.)
3. Attach a Viewer to this node, so you can see the effect of your changes.
4. If necessary, expand the script length to accommodate the total length of the newly merged clip:
   • Click **Edit > Project Settings**. The project settings properties panel appears.
   • Enter **frame range** values that matches the total length.
5. In the **Fade In** and **Fade Out** fields of the AppendClip properties panel, type the number of frames, if any, you want to fade to or from black.
   For example, typing a **5** in the **Fade In** field would result in the following effect at the head of the merged clip.



   (The inverse of this effect would occur at the *tail* of the merged clip were you type **5** in the **Fade Out** field.)
6. In the **Cross Dissolve** field, type the number of frames, if any, of overlap you want between the first and second clip.
   For example, leaving **Cross Dissolve** at the default of **0** creates a simple cut - the transition from the first to second clip is instantaneous. Typing in a **5** creates a time span of five frames in which the first clip's gain ramps downward to zero, while the second's ramps upward to 100%.



Dissolve.



Cut.

7. In the **First Frame** field, type the number of frames, if any, by which you want to slip the clip. Enter a negative value to subtract frames from the head of the merged clip. Enter a positive value to add frames to the head of the clip.

8. Slipping the merged clips may create empty black frames at its head or tail. As appropriate, select **First frame** or **Last frame** if you want these empty frames to appear as copies of the first or last frame.

Freeze Framing Clips

Using the FrameHold node, you can set the output of a clip to a particular frame or frame interval, rather than the entire clip.

To Output a Freeze Frame

1. Click **Time** > **FrameHold** to insert a FrameHold node into your script.
2. Attach its input to the clip you want to affect.
3. Attach a Viewer to the FrameHold node, so you can see the effect of your changes.
4. Enter the **first frame** number to freeze as the output of the FrameHold node.



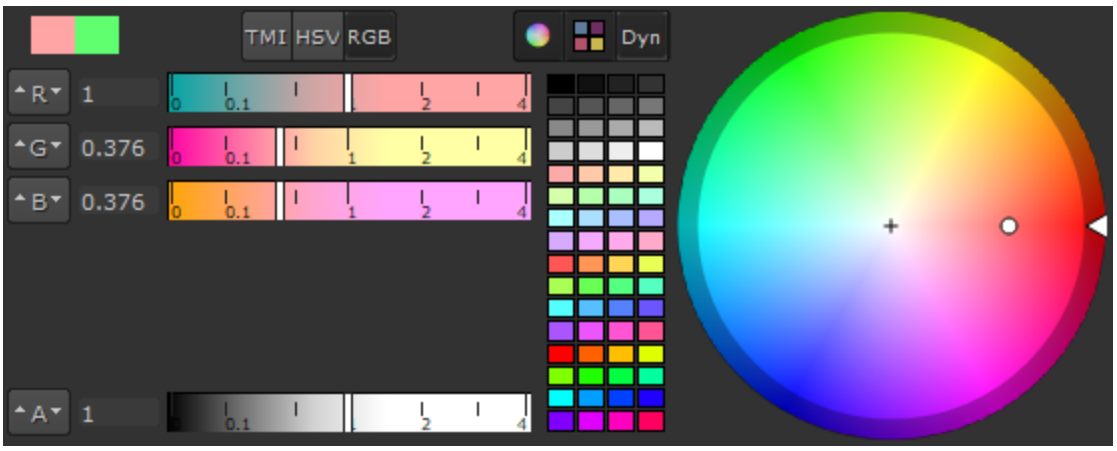
Note: The first frame control defaults to **1**, setting the freeze frame as the first frame of the clip, regardless of the actual frame number. Click **Set to Current Frame** to use the current Viewer frame as the first frame.

To Output a Frame Interval

1. Click **Time** > **FrameHold** to insert a FrameHold node into your script.
2. Attach its input to the clip you want to affect.
3. Attach a Viewer to the FrameHold node, so you can see the effect of your changes.
4. Enter the **increment** frames to skip at output. For example, with a **first frame** of **1** and an **increment** value of **5**, the frames passed down the node tree would be 1, 6, 11, 16, etc.

Working with Color

These pages explain how to use Nuke's color correction nodes and tools to adjust the appearance of the images in your composites. When starting out, this information provides a good overview of Nuke's scopes and color-correction nodes, however not all options are covered here.



Using Scopes

Nuke provides scopes to help you evaluate your footage. Scopes are accessed from the contents menu, **Windows** > **New Scope** sub-menu.

There are a number of global controls (**Preferences** > **Panels** > **Scopes**) that affect how the Scopes display information:

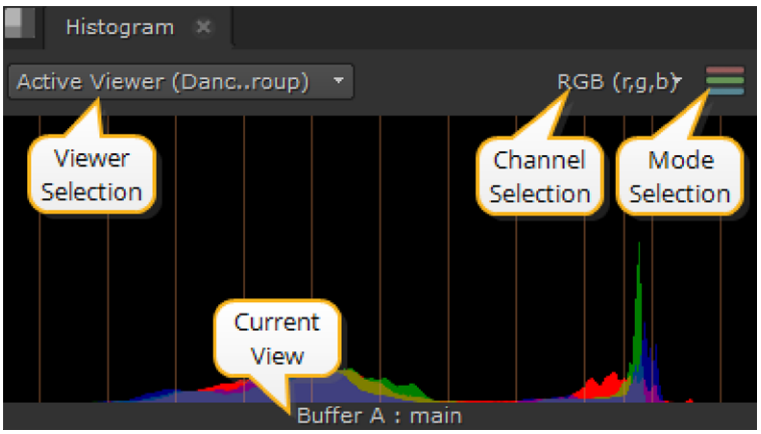
- **black point** - sets the black out of range warning level.
- **white point** - sets the white out of range warning level.
- **luma/chroma encoding** - sets the video standard to use when converting RGB to luma or chroma values in the scope displays, either **REC601** or **REC709**.
- **Include viewer color transforms** - when enabled, scope data includes the applied Viewer color transforms (gain, gamma, and LUT). When disabled, scope data does not include the applied Viewer color transforms. This may slow down rendering, as it may require image calculation.

- **Force full frame** - When enabled, scopes display data for the full frame, regardless of what portion of that frame is displayed in the Viewer. When disabled, scopes only display data for the current area requested by the Viewer rather than the full frame.

Histogram

The content menu **Windows > New Scope > Histogram** provides three color channel and luma channel information that describes the distribution of red, green, blue, and luma pixels throughout the current frame.

The Histogram graphs the number of pixels at each brightness level, and from left to right, the areas of the Histogram represent shadow, mid tones, and highlights.



Tip: You can pan the view area by holding **Alt**, or the middle mouse button, and dragging in the panel.

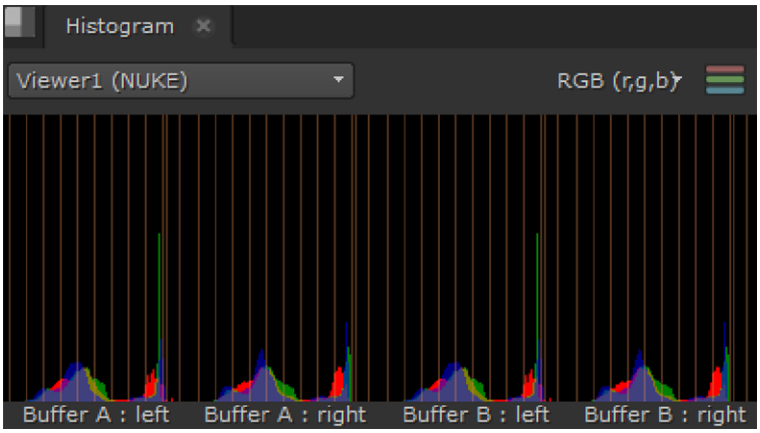
There are a number of view controls on the **Histogram** tab:

- **Viewerselection** - if you have multiple Viewers open, use the dropdown menu to associate Histogram output to the required Viewer.

  The default value, **ActiveViewer**, automatically displays details on the last Viewer you selected.
- **Channel selection** - select the channels to output. The default setting displays RGB, but you can also view channels separately.
- **Mode selection** - select the mode to output. The default setting displays ganged RGB, but you can also view the channels separately by checking **Parade**.
- **Current View** - describes the view currently displayed in the scope, whether it's the A or B buffer and the view. The view defaults to **main**, unless **main** has been replaced in multi-view scripts or projects.

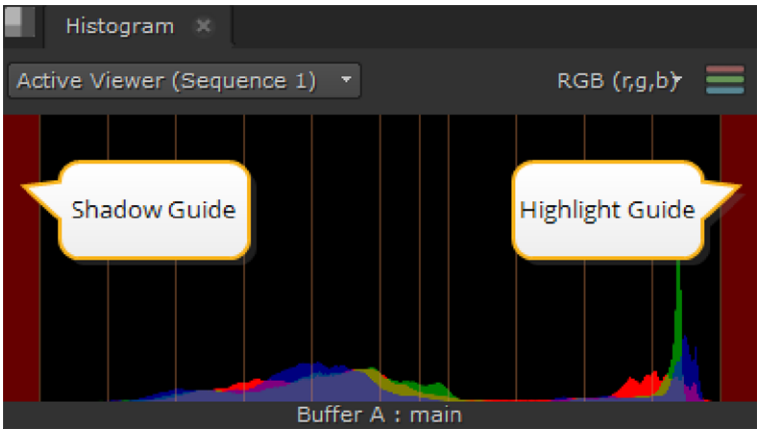
Depending on which Viewer tools and views you have active, you can have up to four scopes displayed at once.

For example, with two stereo Read nodes, one in each input buffer, and **wipe** and **Anaglyph** active, the scopes display something like this:



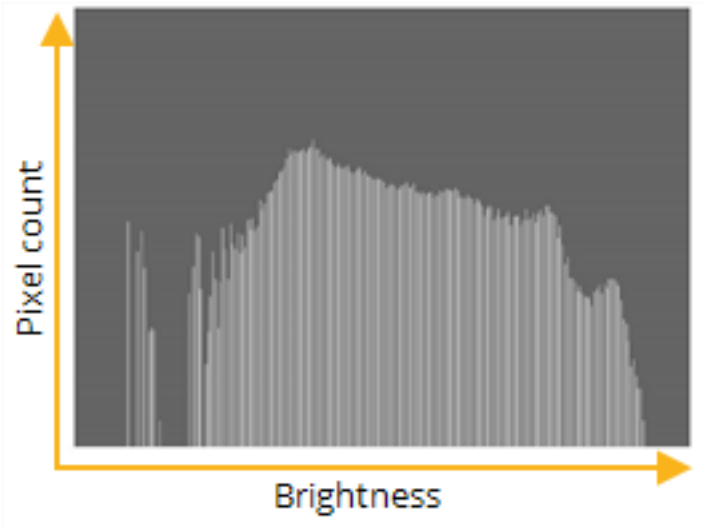
The scopes feature global customizable guides to help you view data. Navigate to **Preferences** > **Panels** > **Scopes** and enter values between **0** and **1** for **Blackpoint** and **Whitepoint**.

The guides at the edges of the Histogram turn red to warn you when the distribution is out of range:



Using the Histogram Node

The properties panel for the Histogram node includes a window that graphs the number of pixels at each brightness level. This is a useful gauge to see whether an image has a good distribution of shadows, midtones, and highlights.



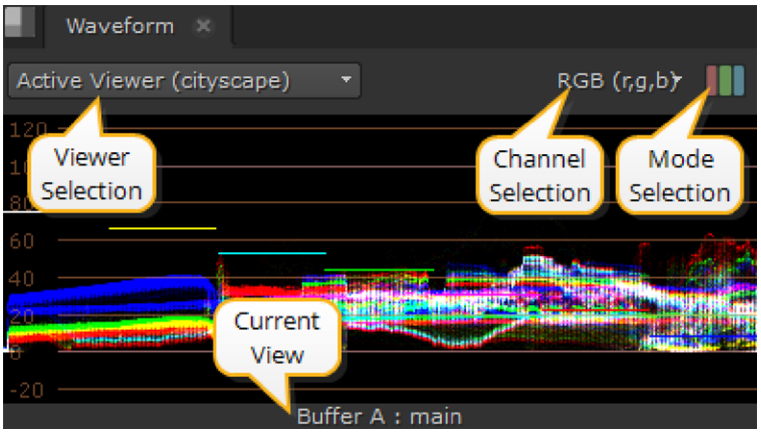
The histogram maps the distribution of shadows, midtones, and highlights.

To Define Tonal Range with the Histogram Node

1. Click **Color** > **Histogram** to insert a Histogram node at the appropriate place in your script.
2. Connect a Viewer to the output of the Histogram node so you can see the effect of your changes.
3. Drag the leftmost **input range** slider till it roughly lines up with the initial boundary of the histogram.
4. Drag the rightmost **input range** slider till it roughly lines up with the final boundary of the histogram.
5. Drag the middle input range slider to define the midtone, or neutral, value.

Waveform

The content menu **Windows** > **New Scope** > **Waveform** scope provides information on luminance, or brightness, which you can use to decide whether the footage is over or under exposed. The information represents luminance values from 0 - 100% (black through the spectrum to white). The higher the waveform, the brighter the image in the Viewer.



Tip: You can pan the view area by holding **Alt**, or the middle mouse button, and dragging in the panel.

The upper white marker is used to measure when over exposure could be a problem. If your waveform has a lot of traces over the white marker, you could consider reducing the brightness of the image. The opposite is true of the lower black marker.

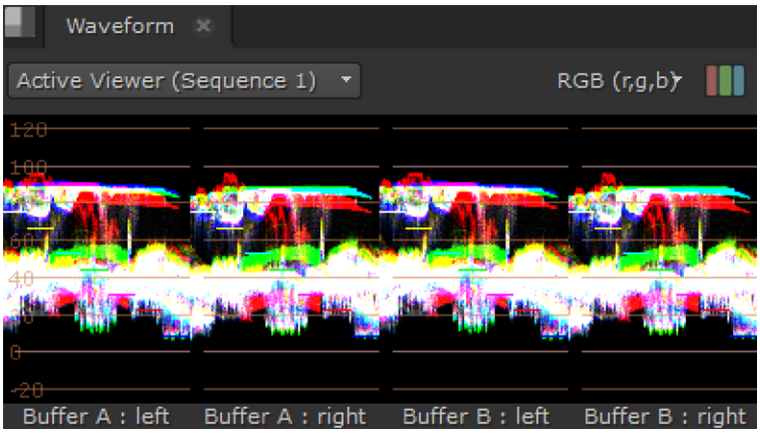
There are also **Viewer** and **Channel** selection controls on the **Waveform** tab:

- **Viewer selection** - if you have multiple Viewers open, use the dropdown menu to associate Waveform output to the required Viewer.

  The default value, **Active Viewer**, automatically displays details on the last Viewer you selected.
- **Channel selection** - select the channels to output. The default setting displays RGB, but you can also view channels separately.
- **Mode selection** - select the mode to output. The default setting displays ganged RGB, but you can also view the channels separately by checking **Parade**.
- **Current View** - describes the view currently displayed in the scope, whether it's the A or B buffer and the view. The view defaults to **main**, unless **main** has been replaced in multi-view scripts or projects.

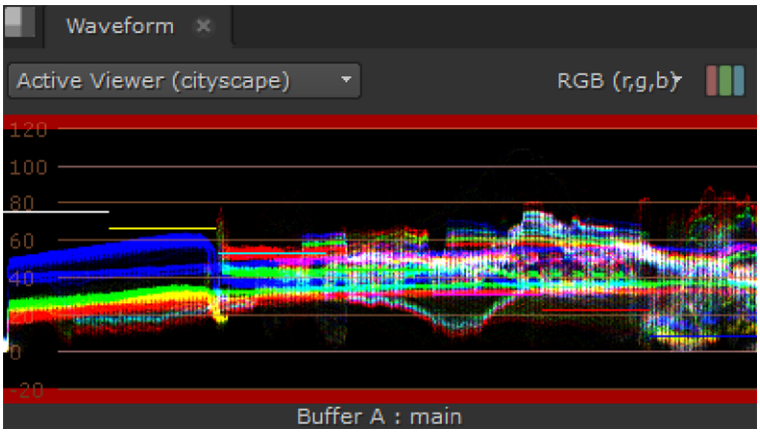
  Depending on which Viewer tools and views you have active, you can have up to four scopes displayed at once.

  For example, with two stereo Read nodes, one in each input buffer, and **wipe** and **Anaglyph** active, the scopes display something like this:



The scopes feature global customizable guides to help you view your data. Navigate to **Preferences** > **Panels** > **Scopes** and enter values between **0** and **1** for **Blackpoint** and **Whitepoint**.

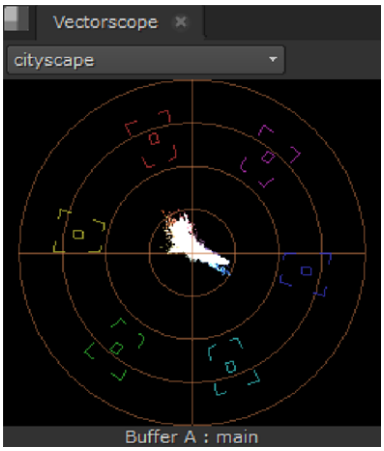
The guides at the top and bottom of the Waveform turn red to warn you when the distribution is out of range:



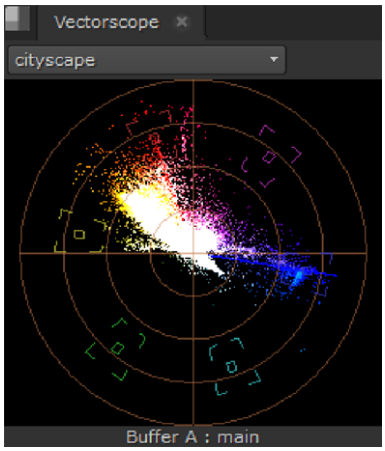
Vectorscope

The content menu **Windows > New Scope > Vector** display color, saturation, and hue information for the current frame. Similar to color wheels, vector scopes display information radially, from the center outward. The farther from the center the data spans, the more saturation is represented.

In the image on the left, you can see that the frame represented contains mostly yellows and reds, but the values are not oversaturated. The image on the right represents a badly saturated frame. Notice the spill of red traces distributed toward the edge of the scope pass the target (the highlighted square).



Normal saturation.



High Saturation.

Tip: You can pan the view area by holding **Alt**, or the middle mouse button, and dragging in the panel.

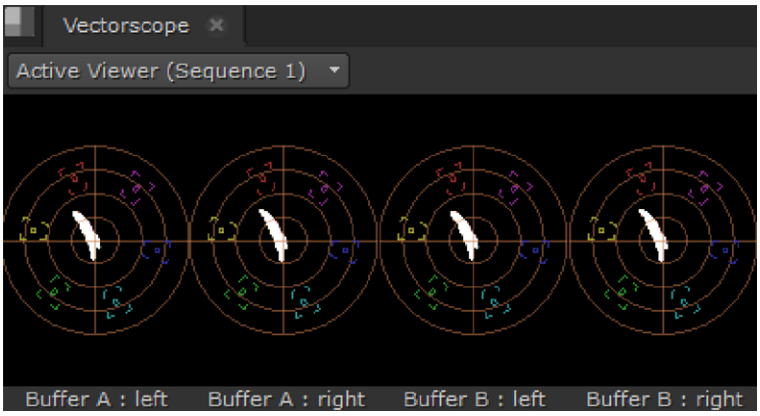
There is also a **Viewer** selection control on the **Vectorscope** tab:

- **Viewerselection** - if you have multiple Viewers open, use the dropdown menu to associate vector scope output to the required Viewer.

  The default value, **ActiveViewer**, automatically displays details on the last Viewer you selected.
- **Current View** - describes the view currently displayed in the scope, whether it's the A or B buffer and the view. The view defaults to **main**, unless **main** has been replaced in multi-view scripts or projects.

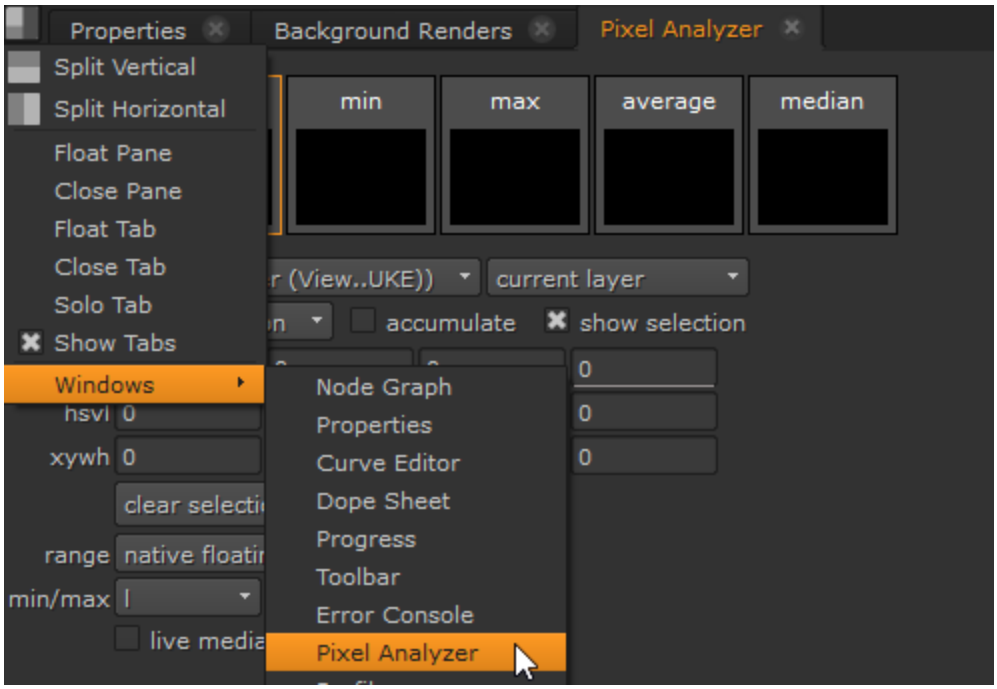
  Depending on which Viewer tools and views you have active, you can have up to four scopes displayed at once.

  For example, with two stereo Read nodes, one in each input buffer, and **wipe** and **Side by Side** active, the scopes display something like this:



Using the Pixel Analyzer

Nuke's Pixel Analyzer enables you to analyze single and multiple pixels, or the entire image, and compare color values between Viewers. The analyzer stores current, minimum and maximum, average, and median values which can then be copied by value to controls on other nodes. For example, you might use the minimum and maximum values from an analysis to set the black and white points on another image. The analyzer is accessed from the contents menu **Windows** sub-menu.



The Pixel Analyzer has two modes:

- **pixel selection** - the default mode, allows you to make single or multiple pixel selections in the Viewer for analysis. You can also analyze an area of the Viewer using a Region of Interest tool.
- **full frame** - analyzes the contents of the current frame, regardless of any selections you've made in the Viewer.

  By default, **full frame** samples the visible region of the Viewer. As a result, actions that change the visible area, such as zooming in and out, alter the available color values.

Note: The **Preferences** >**Panels** > **Scopes** > **Include Viewer color transforms** control does not affect the Pixel Analyzer.

Analyzing Pixel Selections

The analyzer's **pixel selection** mode allows you to select pixels in the Viewer, individually or as a group, and display the color information in swatches. To make selections in the Viewer:

1. Connect a Viewer to the output you intend to analyze. You can connect multiple Viewers to a single output, and vice versa, for comparison.
2. Click the contents menu and select **Windows** > **Pixel Analyzer**.

   The **Pixel Analyzer** panel displays.
3. Use the **sample** dropdowns to control which Viewer and layer provides the swatch information:
   - **viewer** - the default, **current viewer**, always samples the active Viewer if there is more than one. Selecting pixels in a Viewer or clicking on a Viewer's tab causes the Viewer to become the active or current Viewer. Alternatively, you can select a named Viewer from the dropdown when more than one is open.
   - **layer** - the default, **current layer**, displays the layer(s) specified in the Viewer **channels** dropdown.



Note: If **rgba** channels are not present, the first four available channel values are written into the Pixel Analyzer's **rgba** controls.



Note: The color swatches may not update immediately if you select a layer from the **sample** dropdown that is not currently visible in the Viewer, because the new layer must be rendered in the background before the swatch is calculated.

Alternatively, you can select a named layer from the dropdown. When switching between Viewers, the Pixel Analyzer attempts to match the selected layer in the new Viewer. If that layer doesn't exist, the default **current layer** is displayed.

Tip: You can use the **layer** dropdown to sample a layer that is not displayed in the Viewer, allowing you to compare position and color values between layers.

4. Ensure that the mode dropdown is set to **pixel selection**.
5. If you want to make multiple selections, without losing your previous selections, enable **accumulate**.
6. There are three methods of selecting pixels for analysis:
   - **single** - hold **Ctrl**/**Cmd** and click in the Viewer to select a pixel to add it to the swatches.

- **multiple** - hold **Ctrl**/**Cmd** and click-and-drag in the Viewer to select multiple pixels as you move the pointer.
- **region** - hold **Ctrl**/**Cmd**+**Shift** and click-and-drag in the Viewer to create a Region of Interest. Adjust or move the ROI using the Viewer handles.

Tip: To deselect all pixels, hold **Ctrl**/**Cmd** and right-click in the Viewer.

With multiple selections in the Viewer, the color swatches provide the following information:

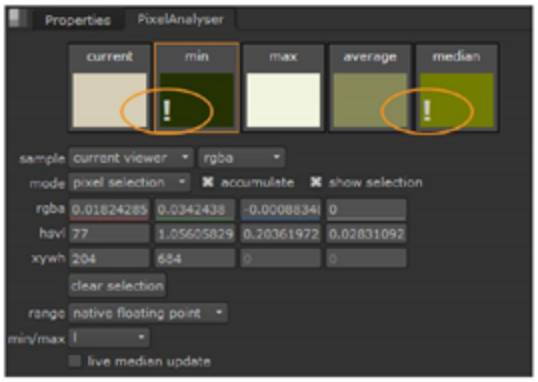
- **current** - the color value of the last pixel selected.
- **min**/**max** - the darkest and brightest color values from the selection.
- **average** - the mean average color value from the selection.
- **median** - the mid-point color between the darkest and brightest colors in the selection.

  You can enable **live median update** to calculate the **median** swatch dynamically, rather than at pen-up.

Note: In **single** mode, all swatches show identical information and in **region** mode, the **current** swatch is disabled.

7. Click on the swatches to display **rgba** and **hsvl** color values for the selected swatch.

   The Pixel Analyzer also detects **inf** (infinite) and **nan** (not a number) color values. The **!** point in the swatch indicates inf/nan values.



8. You can reset the swatches at any time by clicking **clear selection**, except when **full frame** mode is enabled, in which case the button is grayed out.
9. Use the **range** dropdown to select the color bit-depth you intend to display. For example, selecting **8-bit** limits your color values to 255.
10. Set the **min/max** channel you want to use to calculate the min/max color swatches.

This control defaults to luminance (l), but if you wanted to display the minimum values in only the red channel, for example, you could set **min/max** to red (r).

Analyzing Full Frames

The analyzer's **full frame** mode samples the visible region of the Viewer by default. As a result, actions that change the visible area, such as zooming in and out, alter the available color values.

Note: Enabling **full frame processing** forces the Pixel Analyzer to analyze the entire frame, regardless of the portion of the Viewer currently visible.

1. Connect a Viewer to the output you intend to analyze. You can connect multiple Viewers to a single output, and vice-versa, for comparison.
2. Click the contents menu and select **Pixel Analyzer**.

   The **Pixel Analyzer** panel displays.
3. Use the **sample** dropdowns to control which Viewer and layer provides the swatch information:
   - **viewer** - the default, **current viewer**, always samples the active Viewer if there is more than one. Selecting pixels in a Viewer or clicking on a Viewer's tab causes the Viewer to become the active or current Viewer. Alternatively, you can select a named Viewer from the dropdown when more than one is open.
   - **layer** - the default, **current layer**, displays the layer(s) specified in the Viewer **channels** dropdown.

Note: If **rgba** channels are not present, the first four available channel values are written into the Pixel Analyzer's **rgba** controls.

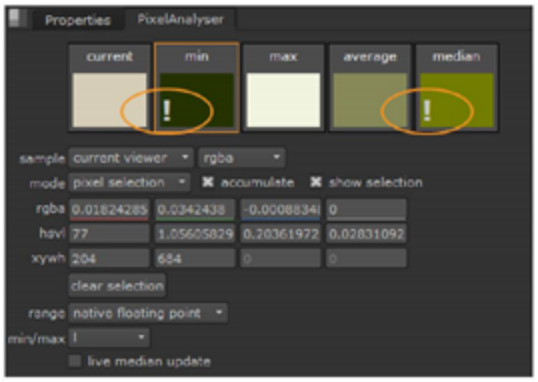
Note: The color swatches may not update immediately if you select a layer from the **sample** dropdown that is not currently visible in the Viewer, because the new layer must be rendered in the background before the swatch is calculated.

Alternatively, you can select a named layer from the dropdown. When switching between Viewers, the Pixel Analyzer attempts to match the selected layer in the new Viewer. If that layer doesn't exist, the default **current layer** is displayed.

Tip: You can use the **layer** dropdown to sample a layer that is not displayed in the Viewer, allowing you to compare position and color values between layers.

4. Ensure that the mode dropdown is set to **full frame**.
5. Click on the swatches to display **rgba** and **hsvl** color values for the selected swatch.

   The Pixel Analyzer also detects **inf** (infinite) and **nan** (not a number) color values. The **!** point in the swatch indicates inf/nan values.

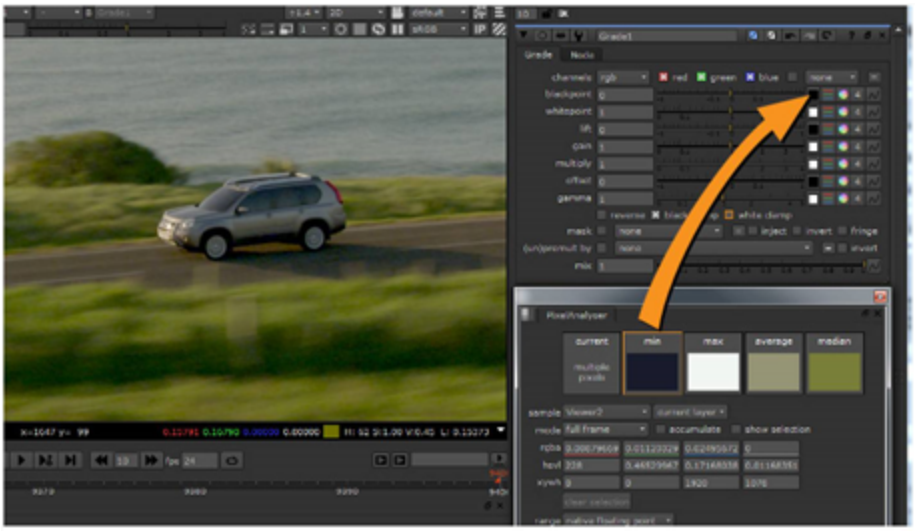


6. Use the **range** dropdown to select the color bit-depth you intend to display. For example, selecting **8-bit** limits your color values to 255.
7. Set the **min/max** channel you want to use to calculate the min/max color swatches.

   This control defaults to luminance (l), but if you wanted to display the minimum values in only the red channel, for example, you could set **min/max** to red (r).

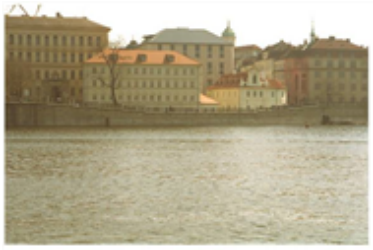
Applying Analysis Data

You can use color information stored in the Pixel Analyzer swatches in any other control that includes a swatch. For example, **min**/**max** color swatches analyzed in another shot can be applied to the current shot by dragging the swatch to the Grade nodes's **blackpoint** and **whitepoint** controls.



Making Tonal Adjustments

Defining tonal range (the black point, white point, and neutral value) is typically the first step in color correcting a clip. Tonal range adjustments often improve contrast, but more importantly, they set the stage for subsequent color corrections by properly dividing the colorspace into shadow, midtone, and highlight regions.



Before tonal adjustment.



After tonal adjustment.

Several of Nuke's color correction effects offer tonal adjustment tools. Of these, Grade and Histogram are probably the most intuitive to operate.

Sampling White and Black Points

The Grade node lets you define white and black points by sampling pixels from a Viewer frame.

To Define Tonal Range with the Grade Node

1. Click **Color** > **Grade** to insert a Grade node at the appropriate place in your script.
2. Connect a Viewer to the output of the Grade node so you can see the effect of your changes.
3. In the Grade properties panel, use the **channels** dropdown menu to select the channels you wish to process.
4. Click the **blackpoint** parameter's color swatch.

   The eye dropper icon appears.
5. In the Viewer, press **Ctrl/Cmd**+**Shift** while clicking on the pixel you want to define as the black point (typically the darkest pixel).
6. Click the **whitepoint** parameter's color swatch. The eye dropper icon appears.
7. In the Viewer, press **Ctrl/Cmd**+**Shift** while clicking on the pixel you want to define as the white point (typically the lightest pixel).



Tip: You can discard sampled pixels by **Ctrl**/**Cmd**+right-clicking in the Viewer.

Making Basic Corrections

Adjustments to contrast, gamma, gain, and offset often comprise the bulk of the work in color correction. Some artists prefer to makes these adjustments via sliders; others prefer curves (which represent the range of color values in an image.) Nuke's ColorCorrect and ColorLookup nodes offer tools to suit either preference.

See Using ColorCorrect Sliders or Using ColorLookup Curves



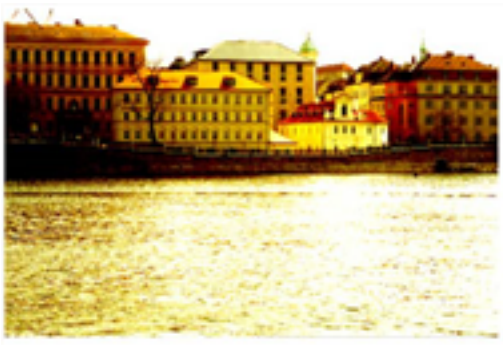
Original.



Contrast boost.



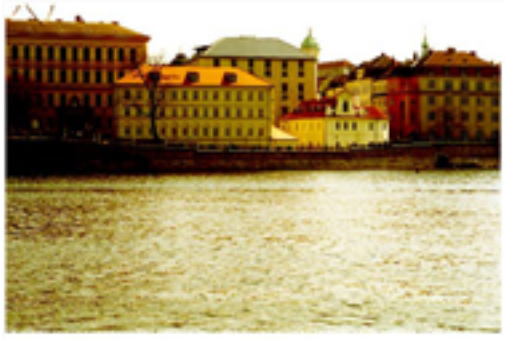
Original.



Gain boost.



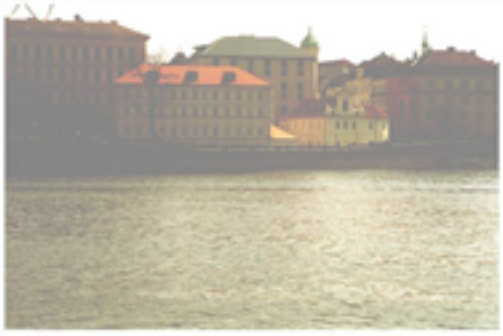
Original.



Gamma boost.



Original.



Offset boost.

Using ColorCorrect Sliders

The ColorCorrect node is particularly convenient for making quick adjustments to contrast, gamma, gain, and offset. A single window houses sliders for all these basic corrections and allows you to apply these to a clip's master (entire tonal range), shadows, midtones, or highlights.

To Adjust Contrast, Gain, Gamma or Offset with the ColorCorrect Node

1. Click **Color** > **ColorCorrect** (or press **C**) to insert a ColorCorrect node at the appropriate place in your script.
2. Connect a Viewer to the output of the ColorCorrect node so you can see the effect of your changes.
3. In the ColorCorrect properties panel, use the **channels** dropdown menu to select the channels you wish to process.
4. Drag the slider appropriate to the region you want to affect an operation you want to apply.

   For example, to brighten the images highlights, you would drag on the **highlights gain** slider.



   Remember too that you can use the color sliders to apply any of the corrections on a per channel basis.
5. On the **Ranges** tab, enable **test** to show what is considered to be in the shadows, midtones, or highlights.

   This overlays the output with black (for shadows), gray (for midtones), or white (for highlights). Green and magenta indicate a mixture of ranges.



Original image.



With **test** enabled.

6. Still on the **Ranges** tab, you can use the **shadow** and **highlight** lookup curves to edit the range of the image that is considered to be in the shadows or highlights. You can also look up color information for the current pixel in the Viewer.

   To return a curve to its default values, select it and click **reset**.

> **Warning:** Do not adjust the **midtone** curve. Midtones are always equal to 1 minus the other two curves.

7. To control how much of the original luminance is preserved after the color correction, enable and adjust **mix luminance**. A value of 0 means the altered luminance is used in the output image. A value of 1 produces a luminance value close to that of the original input image.



Original image.



Mix luminance set to 0.



Mix luminance set to 1.

Note: When **mix luminance** is set to 1, the resulting luminance value is close to the original luminance, but not exactly the same. The difference may vary depending on the color corrections applied to the source image.

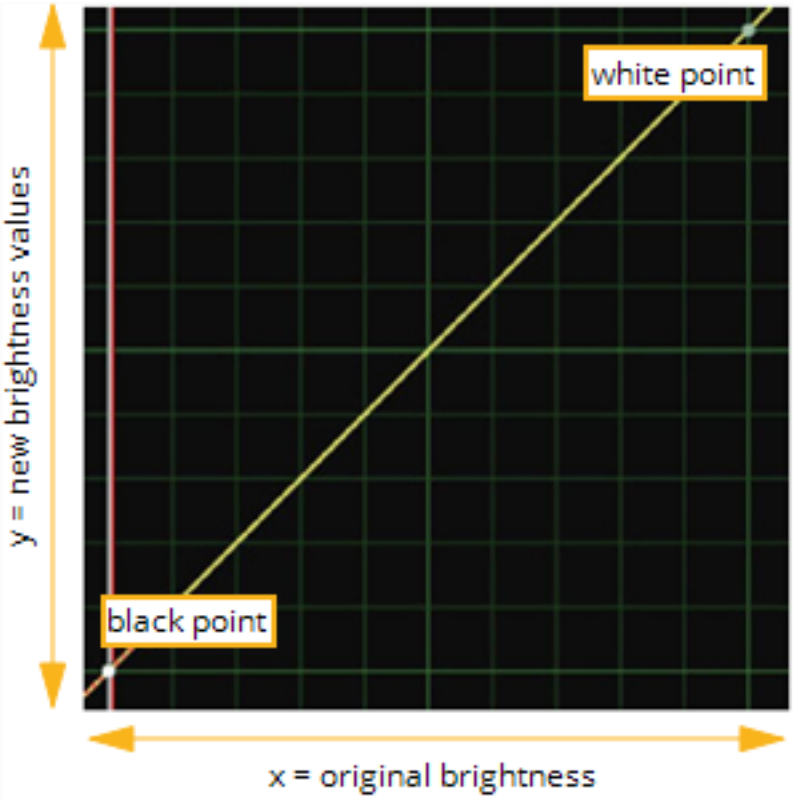
Adjusting Black Levels with the Toe Node

Toe lifts the black level, in a similar way to **gain** controls, but with a roll-off so that whites are mostly not affected.

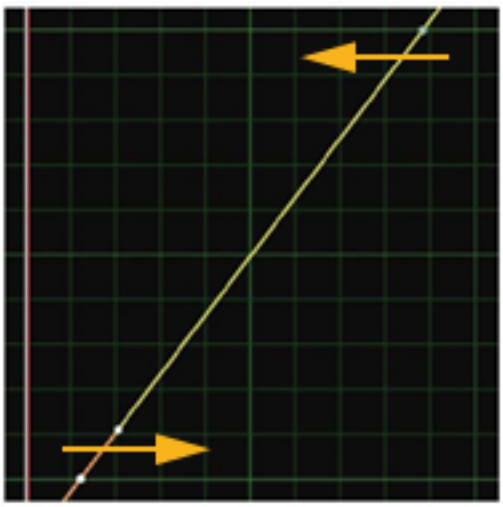
1. Click **Color** > **Toe** to create a Toe node. Connect it to the image whose black levels need adjusting.
2. Adjust the **lift** slider to change the black values to the specified gray value, without affecting any original white values of the image.
3. If necessary, you can limit the effect to a particular channel with the **channels** controls.
4. If you need to, you can pick a channel in the **(un)premultby** dropdown to divide the image first with that channel and then multiply it again afterward. Doing this can sometimes improve your color correction results on anti-aliased egdes.
5. You can also use the **mix** control to dissolve between the original input (value 0) and the full effect of the Toe node (value 1). If you only want to use one channel for mixing, you can specify that using the **mask** control.

Using ColorLookup Curves

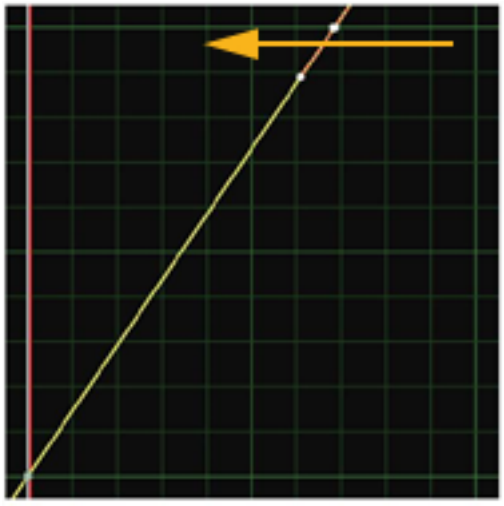
If you prefer to work with color curves, you can use the ColorLookup node to make contrast, gamma, gain, and offset adjustments (and, in fact, many others). *Color curves* refer to line graphs of a given color channel's brightness. The horizontal axis represents the channel's original, or input, values, and the vertical axis represents the channel's new, or output values.



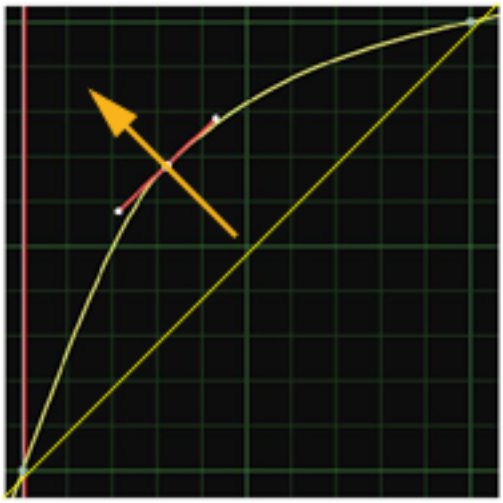
As Figure shows, you can edit the ColorLookup node's color curves to make all of the types of corrections that are possible through the ColorCorrect node - and you can generally make these corrections with more flexibility and precision than is possible with sliders.



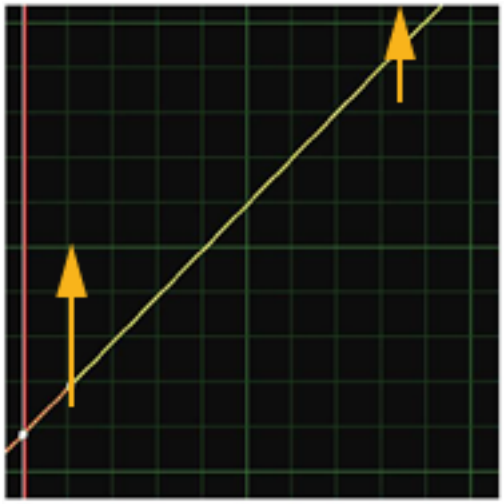
Corrections through the ColorLookup node's color curves: Contrast boost.



Corrections through the ColorLookup node's color curves: Gain boost.



Corrections through the ColorLookup node's color curves: Gamma boost.



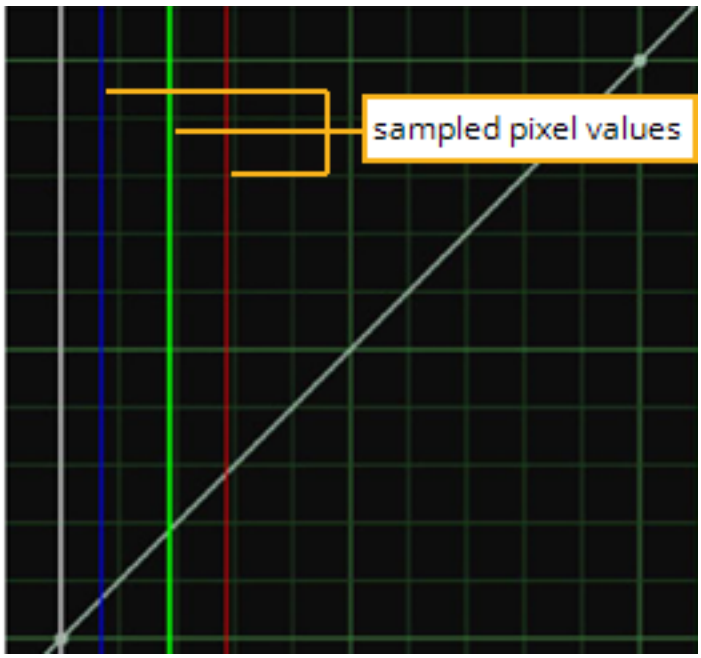
Corrections through the ColorLookup node's color curves: Offset boost.

To Make Basic Corrections with the ColorLookup Node

1. Click **Color** > **ColorLookup** to insert a ColorLookup node at the appropriate place in your script.
2. Connect a Viewer to the output of the ColorLookup node so you can see the effect of your changes.
3. In the ColorLookup properties panel, click **red**, **green**, **blue**, or **alpha** if you want to limit the subsequent operations to a particular channel.

   You can select multiple curves in order to edit one curve with reference to another. Otherwise, select the **master** curve (which represents all channels).

4. To speed up the color calculation, the ColorLookup node uses a precomputed lookup table between 0 and the value specified in the **range** field. You can adjust the **range** value, or uncheck the **use precomputed table** box if necessary to tell ColorLookup to not use a precomputed table.
5. In the Viewer, drag the cursor over the pixels you want to sample for the correction. In the ColorLookup properties panel, press **Ctrl**+**Alt** (**Cmd**+**Alt** on a Mac) while clicking on the curve to set points at the places where the red, green, and blue lines intersect with the color curve.



Viewing values from a sampled color.

6. Edit the position of the points and adjust the tangent handles to adjust the curve shape for the color correction.

   As an alternative to steps 4 and 5, you can use the **source** control to pick a source color for adding points. Then, use **target** to pick a destination color. Finally, do one of the following:

   • Click **Set RGB** to add points on the red, green, and blue curves, mapping **source** to **target**.

   • Click **Set RGBA** to add points on the red, green, blue, and alpha curves, mapping **source** to **target**.

   • Click **Set A** to add points on the alpha curve, mapping **source** to **target**.

   You can use these controls to match shadow, midtone, and highlights on two plates, for example. Set **source** to shadow rgb in one, **target** to shadow rgb in the other, then press **Set RGB**. Same for midtone and highlight areas.

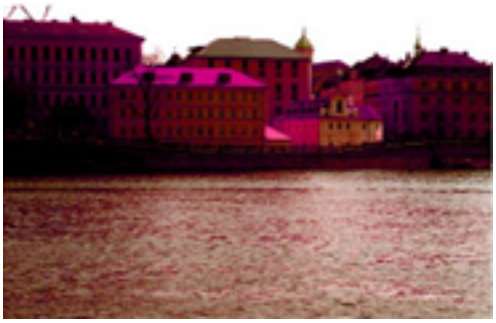
Making Hue, Saturation, and Value Adjustments

For certain color correction tasks like spill suppression, you ideally want to influence only a very narrow range of color values. For such tasks, it's often helpful to use effects that employ the Hue, Saturation, and Value (HSV) color model. As its name indicates, the HSV color model breaks color into three components:

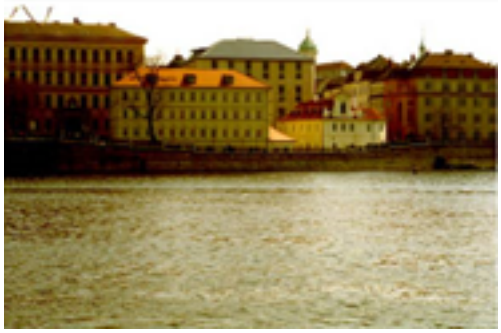
- *Hue*, which refers to the color's location on the traditional color wheel.
- *Saturation*, which refers to the extent to which the color has "soaked up" its hue.
- *Value*, which refers to the brightness of the color.



Original.



Hue shift.



Original.



Saturation decrease.



Original.



Value decrease.

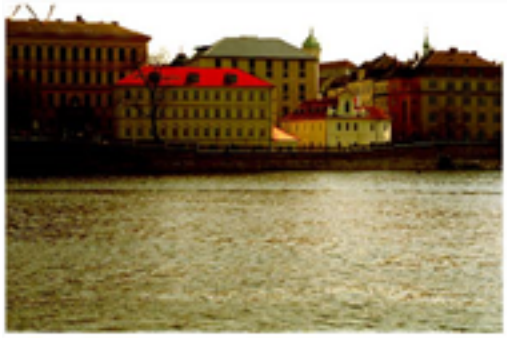
Nuke offers effects that allow you to correct the hue, saturation, and value components individually or collectively.

Correcting HSV

Nuke's HSVTool node lets you simultaneously adjust hue, saturation, and value components from a single properties panel. It also features a color replacement tool. The main strength of this node is the precision it offers in limiting corrections to a narrow swath of colors.



Adjusting color within a specific range of pixel values.



Adjusting color within a specific range of pixel values.

For example, suppose you wanted to add a bit more punch to the waterfront scene by diversifying the rooftop hues. To do so, you could limit the correction to the rooftop's ochre-colored hues by sampling a few pixels, then shift their values. Because you limited the color range, the surrounding image would be generally unaffected by the shift.

To make HSV corrections with the HSVTool node

1. Click **Color** > **HSVTool** to insert an HSVTool node at the appropriate place in your script.
2. Connect a Viewer to the output of the HSVTool node so you can see the effect of your changes.

3. Limit, as appropriate, the range of colors you want subsequent corrections to influence:
   - In the HSVTool properties panel, click the **srccolor** color swatch. **Ctrl**/**Cmd**+click on the Viewer to sample a single color displayed, or **Ctrl**/**Cmd**+**Shift**+drag to sample a range of colors. To sample a single color from the node's input while viewing its output, **Ctrl**/**Cmd**+**Alt**+click on the Viewer. To sample a region from the input, **Ctrl**/**Cmd**+**Alt**+**Shift**+drag on the Viewer.



Tip: You can discard sampled pixels by **Ctrl**/**Cmd**+right-clicking in the Viewer.

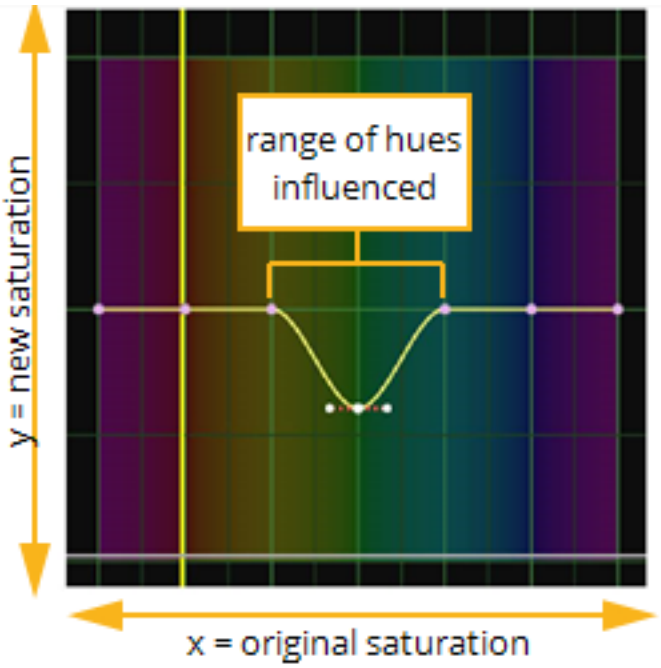
   - The **Range** sliders on **Hue**, **Saturation**, and **Brightness** clamp to the sampled range.
   - For any color component, drag on the **Range** sliders to expand the color range as necessary.
   - For any color component, drag on the **Range Rolloff** slider to fine tune the color range. Doing so, adjusts the amount of falloff allowed past the limits defined by the Range sliders.
4. Make the necessary HSV corrections:
   - For hue corrections, drag on the **Rotation** slider to input color wheel value between 0 and 360.
   - For saturation corrections, drag on the **Saturation Adjustment** slider to input values between **-1** (completely desaturated to some shade of gray) and **1** (completely saturated).
   - For value corrections, drag on the **Brightness Adjustment** slider to input values between **-1** (black) and **1** (white).

You can also make color replacements using the srccolor and dstcolor parameters: First sample the color you wish to replace with the srccolor color swath, then sample the color which you wish to use as the replacement with the dstcolor color swath. The color in dstcolor replaces the color in srccolor throughout the image.

Also, keep in mind that the HSVTool node makes an excellent keyer. You can use its Hue, Saturation, and Brightness range sliders to precisely select a range of colors, then use the channel output dropdown at the bottom of the dialog to output this selection as a matte channel. This dropdown lets you specify which color components (hue, saturation, value, etc.) are added to the matte.

Correcting Hue Only

Nuke's HueCorrect node lets you make precision adjustments to the levels of saturation in a range of hues. You do so via edits to a series of suppression curves.



Editing the suppression curve.

By selecting which curve you edit and how much of that curve you alter, you can precisely limit the influence of the effect.

Suppressing spill

For the compositor, HueCorrect is obviously of greatest use in diminishing green, blue, or redscreen spill.

To suppress spill with the HueCorrect node:

1. Click **Color** > **HueCorrect** to insert a node at the appropriate place in your script.
2. Connect a Viewer to the output of the HueCorrect node so you can see the effect of your changes.
3. In the HueCorrect properties panel, select the channels you want to influence:
   - Click **sat** to influence all channels (red, green, blue, and alpha) equally.
   - Click **lum** to influence all channels, but with luminance weighting in effect (meaning that the red channel receives approximately 30% of the effect; the green, 60%; and the blue, 10%).
   - Click **red** to apply the curve as a lookup on the red channel only, looking up the pixel's hue on the curve and then multiplying the red value in the pixel by the lookup result.
   - Click **green** to apply the curve as a lookup on the green channel only, looking up the pixel's hue on the curve and then multiplying the green value in the pixel by the lookup result.

- Click **blue** to apply the curve as a lookup on the blue channel only, looking up the pixel's hue on the curve and then multiplying the blue value in the pixel by the lookup result.
- Click **r_sup** to apply a suppression function to reduce the level of the red channel. While the red curve is used to directly multiply the red channel by the curve value, the r_sup curve is used to control the amount that the red channel is suppressed.
- Click **g_sup** to apply a suppression function to reduce the level of the green channel. While the green curve is used to directly multiply the green channel by the curve value, the g_sup curve is used to control the amount that the green channel is suppressed.
- Click **b_sup** apply a suppression function to reduce the level of the blue channel. While the blue curve is used to directly multiply the blue channel by the curve value, the b_sup curve is used to control the amount that the blue channel is suppressed.

  Note that you can select multiple curves in order to edit one curve with reference to another.

4. If necessary, drag the cursor over the Viewer to sample the image pixels that are representative of the part of the image you want to correct. Then, in the HueCorrect properties panel, press **Ctrl**+**Alt** (**Cmd**+**Alt** on a Mac) while clicking on the curve to plot a particular pixel's value on the curve. This lets you see what portion of the curve you want to edit.
5. Edit the curve as necessary - typically this means dragging down on control points in the hue region that you wish to suppress.
6. To control how much of the original luminance is preserved after the color correction, enable and adjust **mix luminance**. A value of 0 means the altered luminance is used in the output image. A value of 1 produces a luminance value close to that of the original input image.



Original image.



Mix luminance set to 0.



Mix luminance set to 1.

Note: When **mix luminance** is set to 1, the resulting luminance value is close to the original luminance, but not exactly the same. The difference may vary depending on the color corrections applied to the source image.

Correcting Saturation Only

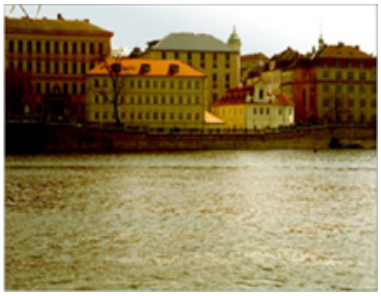
For the times when you just want to correct the saturation component and don't require limiting the correction to any particular channel, you can use Nuke's Saturation node. Its controls are bare bones - basically, just a **saturation** slider.

To make saturation corrections with the Saturation node

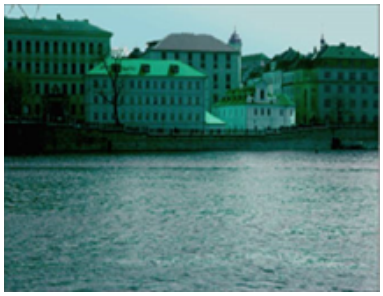
1. Click **Color** > **Saturation** to insert a Saturation node at the appropriate place in your script.
2. Connect a Viewer to the output of the Saturation node so you can see the effect of your changes.
3. Drag the **saturation** slider to make the necessary corrections.

Masking Color Corrections

Virtually all the color-correction effects in Nuke include **mask** parameters that lets you limit the correction to the non-black pixel values of a matte image. For example, suppose you want to add a blue cast to the following scene without affecting the buildings.



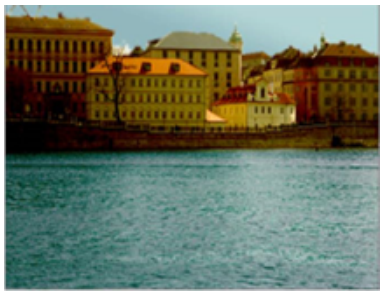
Original image.



Color-corrected image.



Mask image.



Color-corrected image.

You could create a garbage mask that covers the river, then boost the red channel's gamma in the area of the frame that underlies the mask.

Typically, mask controls are located toward the bottom of the properties panel. However, in the case of multi-purpose effects like HSVTool, there may be multiple mask controls, so that you can limit each type of correction with a different mask.



Selecting a mask channel.

To Mask a Color Correction

1. Open the node's properties panel and locate the **mask** controls.
2. Select the channel you wish to use as the mask from the dropdown menu.
3. If you check **inject** in the **mask** controls, the mask from the **mask** input is copied into the predefined **mask.a** channel. This way, you can use the last mask input again downstream. You can also set a stream of nodes to use **mask.a** as the mask, and then change the masking of all of them by simply connecting a new mask into the mask input connector of the first node.
4. If necessary, check the **invert** box to reverse the mask.
5. To only apply the effect at the edges of the mask, check **fringe**.
6. If the overall effect of the node is too harsh, you can blend back in some of the input image by dragging on the **mix** slider.
7. If you want to output only the portion of the frame underlying the mask, check the **(un)premult by** box.

Applying Grain

Grain matching - ensuring that all of the elements in a composite, including those which were digitally generated, look like they were shot on the same film stock - is often one of the final steps in achieving a convincing integration of all of a composite's elements. Nuke offers effects for synthetically creating grain and for reading in practically-created grain (grain derived from actual film stock).



An example of applying grain to an image: Grainless image.



An example of applying grain to an image: Grained image.

Using Synthetic Grain

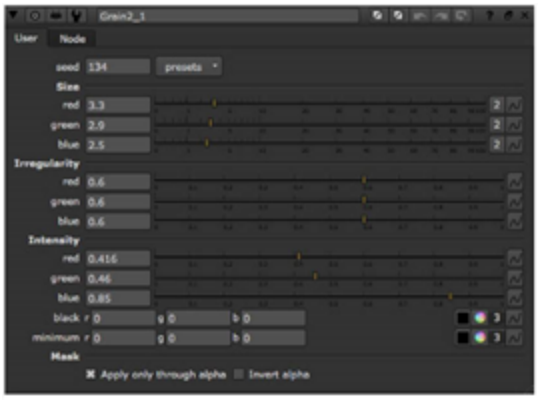
Nuke offers several nodes for creating synthetic grain: Dither, Grain, ScannedGrain, and if you have a NukeX license, F_ReGrain. Of these, Dither is the crudest - it basically lets you specify the amount of noise per channel.

Grain includes presets for matching film stock and a means for controlling the mix between the generated grain and the backplate. ScannedGrain offers film stock presets, plus synthetic grain controls for applying practical grain.

F_ReGrain, for NukeX, is designed to sample an area of grain from one image and then to generate unlimited amounts of this grain with exactly the same statistics as the original. This new grain can then be applied to another image. See Using F_ReGrain for more information.

To add synthetic grain with the Grain node

1. Click **Draw** > **Grain** to insert a Grain node at the appropriate place in your script.
2. Connect a Viewer to the output of the Grain node so you can see the effect of your changes.



3. From the **presets** dropdown menu, select one of the film stock you want to match.

Tip: The Grain node's presets include a variety of Kodak film stock with different grain values, so you can find the most suitable for your scene.

4. Adjust the **Size** sliders for the red, green, and blue channels to shrink or enlarge the granules.
5. Adjust the **Irregularity** sliders to increase or decrease the random quality of the grain, according to the different channels.
6. Adjust the **Intensity** sliders to increase or decrease the contrast of the grain against the original image.

Using Practical Grain

Although Nuke's ScannedGrain node offers controls for creating synthetic grain (ones comparable to those just discussed), its main use is for reading in and applying scanned grain - that is, grain derived from actual film stock. If your facility has such sequences available, you can read them in and apply them using the ScannedGrain node. Creating grain files is described below, as well as using the resulting grain files with the ScannedGrain node.

To Create Film Stock Sequences

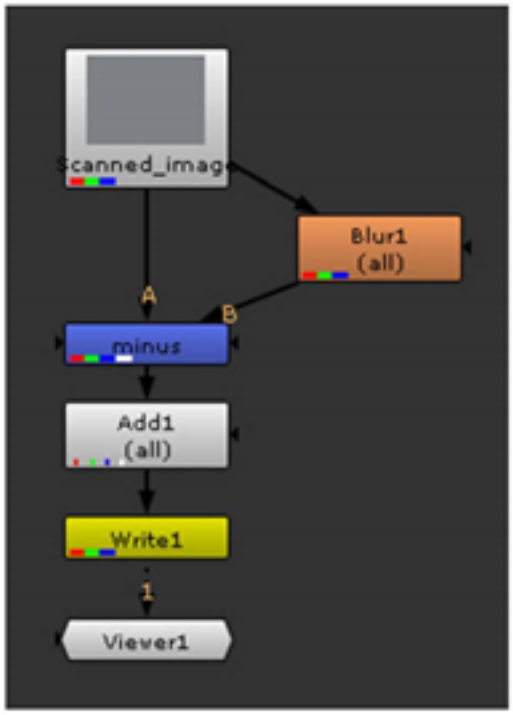
1. Film a gray card. Only about 50 frames are needed.
2. Scan the film in.
3. Select **Image** > **Read** to load the scanned image into Nuke.
4. Add a Blur node (**Filter** > **Blur**) after the image to blur the image until you cannot see any grain. Then, blur the image a bit more.
5. Select **Merge** > **Merge** to insert a Merge node in your script. Connect the A input of the Merge node into the original image, and the B input into the Blur node. Then, open the Merge controls and select **minus** from the **operation** dropdown menu. The blurred image is subtracted from the original image.

The purpose of this and the previous step is to subtract any overall gray level from the grain so that only the grain is left.

6. Select **Color** > **Math** > **Add** to insert an Add node after the minus node. In the Add node controls, enter 0.5 in the **value** field. This adds a value of 0.5 to all channels.

   This step is necessary, because the ScannedGrain node subtracts 0.5 from the channels when it reads the grain file (the subtraction is needed to store negative numbers in most file formats).

7. Select **Image** > **Write** to insert a Write node after the Add node. Render the output. Any file format will do (for example, we have used the .rgb extension in the grain files on our website).



To Add Scanned Grain with the ScannedGrain Node

1. Click **Draw** > **ScannedGrain** to insert a ScannedGrain node at the appropriate place in your script.
2. Connect a Viewer to the output of the ScannedGrain node so you can see the effect of your changes.
3. Click the folder icon of the **grain** field and navigate to the appropriate film stock sequence. Select **Open**.
4. If necessary, check the **resize** box to scale the grain sequence up or down to match your working resolution.
5. In the **min. width** field, define a minimum width (in pixels) that images have to have in order to receive grain.
6. Enter values into the red, green, and blue **amount** fields to increase or decrease on a per-channel basis the density of granules. (This is accomplished, crudely speaking, by boosting or reducing the gain of the grain sequence.)

Now you're ready to fine-tune the blending between the grain and backplate.

To Mix the Grain and Backplate

1. Drag on the **saturation** slider to increase or decrease the intensity of the grain's hue across all channels.
2. If necessary, you can also use the supplied curve editor to edit the grain sequence's color curves. In this manner, you can alter gain, gamma, contrast, etc. on a per channel basis. (These curves function in the same manner as those described in Using ColorLookup Curves).
3. To set a low threshold, based on the input image, below which the grain is not subtracted, adjust the **low_clip** slider.

Applying Mathematical Operations

Nuke's **Color** icon in the Toolbar houses a number of nodes which are designed to apply common mathematical operations to channels. These operations include clamps, offsets, inversions, multiplications, and expressions.

Clamping Channel Values

To *clamp* a channel's values is to ensure that its blackest blacks and whitest whites are visible on an intended display device. Nuke's Clamp node lets you assign "legal" values to colors that are either too light or dark for the intended display device.



Clamping black and white pixels to "legal" values.



Clamping black and white pixels to "legal" values.

For this effect, you use Nuke's Clamp node.

1. Click **Color** > **Clamp** to insert a Clamp node at the appropriate point in your script.

2. Connect a Viewer to the output of the Clamp node so you can see the effect of your changes.
3. In the Clamp properties panel, use the **channels** field to select the channel you wish to clamp.
4. Drag the **minimum** slider to the legal value. (This has the effect of causing black values to go gray.)
5. Drag the **maximum** slider to the legal value. (This has the effect of causing white values to go gray.)

Offsetting Channel Values

To *offset* a channel's values is to add a fixed value to them, which, in effect lightens the whole channel. You can also add a negative value to a channel, in which case the channel gets darker.



Offsetting channel values.



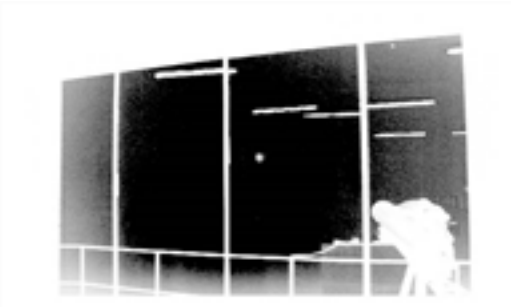
Offsetting channel values.

For this effect, you use Nuke's Add node.

1. Click **Color** > **Math** > **Add** to insert a Add node at the appropriate point in your script.
2. Connect a Viewer to the output of the Add node so you can see the effect of your changes.
3. In the Add properties panel, use the **channels** field to select the channel you wish to offset.
4. Use the **value** slider to input the value you wish to add to the channel's values.
5. If you are using premultiplied input images, you may want to check **(un)premult by** and select **rgba.alpha** from the dropdown menu. This simulates doing the addition before the premultiplication was done.

Inverting Channel Values

To *invert* a channel is to subtract its values from one, which causes its blacks to become white and its whites to become black. In the course of building a script, you'll have frequent need to invert mattes in particular.



Inverting channel values.



Inverting channel values.

To invert channels you use Nuke's Invert node.

1. Click **Color** > **Invert** to insert an Invert node at the appropriate point in your script.
2. Connect a Viewer to the output of the Invert node so you can see the effect of your changes.
3. In the Invert properties panel, use the **channels** field to select the channel you wish to invert.

Multiplying Channel Values

To *multiply* a channel's values is to times them by a given factor, which has the effect of lightening the channel while preserving the black point. (This operation is also knows as gain.)



Multiplying channel values.



Multiplying channel values.

For this effect, you use Nuke's Multiply node.

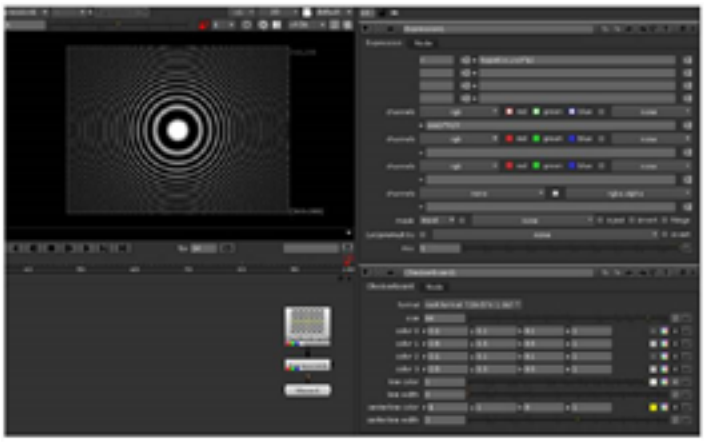
1. Click **Color** > **Math** > **Multiply** to insert a Multiply node at the appropriate point in your script.
2. Connect a Viewer to the output of the Multiply node so you can see the effect of your changes.
3. In the Multiply properties panel, use the **channels** field to select the channel whose values you wish to multiply.
4. Use the **value** slider to input the factor by which to you want to times the channel's values.

Applying Expressions to Channel Values

Up till now, the discussion has focused on how to apply simple mathematical formulae - additions, subtractions, multiplications, etc. - to a channel's values. Nuke's Expression node, however allows you to apply complex formulae to a channel's values. The actual syntax for expressions is rather complex, and thus must be deferred to Expressions. For now, you can read about the basics of how to operate the Expression node.

1. Click **Color** > **Math** > **Expression** to insert an Expression node at the appropriate point in your script.
2. Connect a Viewer to the output of the Expression node so you can see the effect of your changes.
3. In the Expression properties panel, use the channel dropdown menus and buttons to select the channel to which you wish to apply an expression.
4. Type the actual expression in the **=** field next to the channel.

   For example, to assign noise to the red channel, then boost the gain of that result by 20 you would type **(random*r)*20**.
5. If necessary, you can apply different expressions to different sets of channels by repeating the above steps for the other channel dropdown menus and buttons.
6. If you need to use a long expression in several fields, you can use the fields on top of the properties panel for assigning the expression temporarily to a variable. Enter your variable on the left side of the = sign, and the expression on the right. You can then use the variable to represent the expression in the = fields next to the channels.



A checkerboard modified using an Expression node.

Transforming the Color Space

Whenever you read a clip into a script, it is automatically converted to Nuke's native color space, which is 32-bit per channel RGB, a linear format. This conversion takes place even if the clip you read in is in the Kodak Cineon format, which is a logarithmic format.

The reverse of this conversion, called a lin-to-log conversion, also automatically takes place when you write the processed element back out of the script - that is, Nuke automatically converts it back into a Cineon element.



Article: See Knowledge Base article Q100319 for more information on colorspace in Nuke.

Overriding the Default Cineon Conversion

Nuke uses the Kodak-recommended settings when making Cineon conversions in either direction. It's rare that you would want to override these settings, but if it becomes necessary you can use Nuke's Log2Lin or PLogLin nodes.

To Override the Default Cineon Conversions Using Log2Lin

1. Double-click on the Read node of the Cineon element whose conversion you wish to override.
2. In the Read properties panel, set the **Input Transform** dropdown menu to **linear**. This halts the automatic log-to-lin conversion.
3. Click **Color**> **Log2Lin**to insert a log2lin node directly after the Read node.
4. In the Log2Lin properties panel, set the **operation** dropdown menu to **log2lin**.
5. Set **black**, **white**, and **gamma** to the appropriate values.
6. Copy the Log2Lin node and insert it just before the element's Write node.
7. Open up the properties panel of the second Log2Lin node and set the **operation** dropdown menu to **lin2log**. This gives you the reverse of the conversion you created above.
8. Double click on the element's Write node.
9. In the Write properties panel, set the **output transform** dropdown menu to **linear**. This halts the automatic lin-to-log conversion and lets the one you create above have priority.

To Convert Between Logarithmic and Linear Color Spaces

You can also convert between logarithmic and linear color spaces with the PLogLin node. This alternative method is better known as the Josh Pines log conversion, and it's based on using a single gray value, rather than a black and a white one, like in the Log2Lin node. To use the PLogLin node:

1. Much like with the Log2Lin node, when you're using the PLogLin node you need to first make sure your Read or Write nodes aren't automatically converting your color space. To do this, click on your Read or Write node, and select **linear** in the **Input Transform** or **output Transform** dropdowns.
2. Create a PLogLin node by clicking **Color** > **PLogLin**. Connect it to either your Read node's output or the Write node's input, depending on which one you're converting the color space for.
3. In the operation dropdown, select the operation you want PLogLin to perform. Select **log to lin** to convert from logarithmic to linear, and **lin to log** to do the reverse.
4. Adjust the **linear reference value** slider to the linear value that corresponds with the logarithmic reference value and the **log reference value** slider to the value that corresponds with the linear reference value.
5. You can also adjust the film response gamma value in the **negativegamma** field, and use the **densitypercodevalue** field to tell PLogLin what type of change occurs in the negative gamma for each log space code value when converting.
6. If you need to, you can pick a channel in the **(un)premult by** dropdown to divide the image first with that channel and then multiply it again afterward. Doing this can sometimes improve your color correction results on anti-aliased edges.
7. You can also use the **mix** control to dissolve between the original input (value 0) and the full effect of the PLogLin node (value 1). If you only want to use one channel for mixing, you can specify that using the **mask** control.

To Create a 3D LUT in Log Color Space (for example Cineon)

For good visual fidelity when using 3D LUTs, it is recommended that a log color space is used for 3D LUT generation. To create a 3D LUT in a log colorspace (Cineon in this example):

1. Create a CMSTestPattern node at a required density.
2. Connect this to a Colorspace node to convert **input: Cineon** to **output: Linear**. This is a reverse log conversion, and gives a higher density of samples in the shadows of your grading curve, with reduced density in the highlights.
3. Connect the Colorspace node to your grading nodes.
4. Connect the grading nodes to another Colorspace node to convert **input: Linear** to **output: Cineon**. This is the log conversion, and converts the grade samples back to the normalized range (0-1) for 3D LUTs.

It is important to distribute the samples appropriately due to the low resolution of 3D LUTs.

5. Connect a GenerateLUT node to the last Colorspace node.

Tip: To reproduce the grade, set a Vectorfield node to **Cineon** for both the **colorspace in** and **colorspace out** controls.

Making Other Colorspace Conversions

You can also convert elements from Nuke's native color space to other color spaces more appropriate to a given process or intended display device. For conversions such as this, use Nuke's Colorspace node, which supports RGB, HSV, YUV, CIE, and CMS formats (and various sub-formats).

To Convert an Element in Nuke's Native Color Space into Another Color Space

1. Click **Color** > **Colorspace** to insert a Colorspace node into the appropriate place in your script.
2. In the Colorspace properties panel, set the rightmost dropdown menu in the **output** controls to the appropriate standard.
3. Set the dropdown menu in the middle of the **output** controls to the appropriate standard.
4. Set the leftmost dropdown menu in the **output** controls to the color space of your choice.
5. If you wish to reverse this conversion later in the script:
   - Copy the Colorspace node and insert it at the appropriate point in your script.
   - Set the **output** controls to **sRGB**, **D55**, and **RGB**.
   - Set the **input** controls to match the values you entered in steps 2, 3, and 4 above.
6. If you wish write out the element in the new color space:
   - Double-click on the element's Write node.
   - In the Write properties panel, set the **colorspace** dropdown menu to **linear**. This halts the automatic conversion and lets the one you create above have priority.

Converting Color Spaces with the OCIOColorSpace Node

Much like the ColorSpace node, you can use the OCIOColorSpace node for converting an image sequence from one colorspace to another.

Conversions with the OCIOColorSpace node are based on the OpenColorIO library (for more information, see https://opencolorio.org). Using OCIOColorSpace is pretty simple:

1. Select **Color** > **OCIO** > **OCIO ColorSpace** and connect it to your image sequence.
2. Select the channel or layer you want the conversion to affect using the **channels** controls.
3. In the **input** dropdown, select the colorspace of your input image.
4. In the **output** dropdown, select the colorspace you want to convert the image to.

For more information on the OCIOColorSpace node, and other nodes based on the OpenColorIO library, see Color Nodes in the Nuke Online Help for more information.

Note: Nuke is bundled with a pre-compiled version of the OpenColorIO library and a suite of OCIO nodes. If you already have a pre-built version of the OCIO library on your system and your environment is not set up correctly for Nuke, you may encounter problems. For OCIO to work correctly, Nuke requires the compiled versions of PyOpenColorIO and libOpenColorIO to match. To override Nuke's pre-packaged version with your own custom version, follow the steps below:

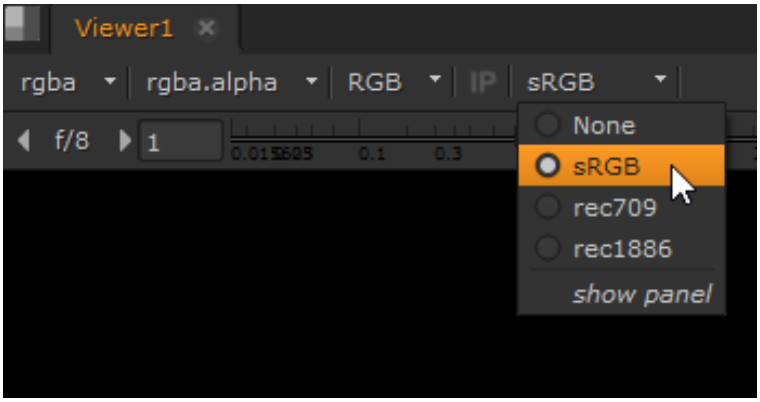
- **Linux:**
  Manually replace the Nuke packaged versions of **libOpenColorIO.so** and **PyOpenColorIO.so** with your custom versions of these files. These can be found in the **<NukeInstall>** and **<NukeInstall>/plugins** directories respectively.
- **Mac:**
  Set your **NUKE_PATH** to the location of **PyOpenColorIO.so** and your **DYLD_LIBRARY_PATH** to the location of **OpenColorIO.so**. For example:
  export NUKE_PATH="/myOCIOLibraryLocation/"
  export DYLD_LIBRARY_PATH="/myOCIOLibraryLocation/"
- **Windows:**
  Both **OpenColorIO.dll** and **PyOpenColorIO.pyd** must be in the same directory. You then need to set your **NUKE_PATH** to this directory. For example:
  set NUKE_PATH=\myOCIOLibraryLocation\
- **All Platforms:**
  In addition to the steps above, you need to set the OCIO environment variable to point to your OCIO configuration file. This overrides the configuration file specified in Nuke's preferences (see

Setting Preferences). For example:
export OCIO="/myOCIOConfigLocation/config.ocio"
The OCIO nodes in Nuke are compiled against a specific version of the OCIO libraries (for the current version, see Third-Party Libraries and Fonts). If you're using your own custom libraries, recompile the OCIO nodes against your versions of the libraries. Failure to follow these steps may result in errors when adding OCIO nodes or modifying OCIO options in the preferences.

Changing the Viewer Color Space

By default, a script's Viewers display images in Nuke's native color space. You can, however, set a script's Viewers to display images in non-native color spaces. Changing the display color space in no way affects your rendered output. You are applying a display-only lookup table.

Select the desired color space from the viewer process dropdown menu.



Adding Context Viewer Processes

You can add variables to register certain viewer processes in OCIODisplay nodes by creating a custom **config.ocio** file and then specifying the variable in the **to_reference** file transform.

In this example, the variable is called SHOT. If you intend to use the same name, make sure that SHOT is not assigned as an environment variable.

To edit the **ocio.config** file:

1. Open your **ocio.config** file in a text editor and review the **colorspaces** entries. The example shows the sRGB entry.

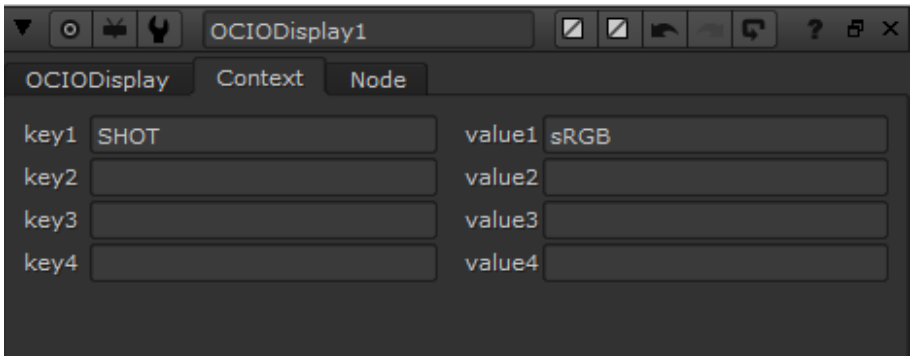
```
- !<ColorSpace>
name: sRGB
```

```
family: ""
  equalitygroup: ""
  bitdepth: 32f
  description: |
  Standard RGB Display Space
  isdata: false
  allocation: uniform
  allocationvars: [-0.125, 1.125]
  to_reference: !<FileTransform> {src: "sRGB.spi1d", interpolation: linear}
```

2. Edit the **to_reference** line to include the variable you intend to create.
   `to_reference: !<FileTransform> {src: "$SHOT.spi1d", interpolation: linear}`
3. Save the file.

To register the viewer process:

1. Launch Nuke.
2. Press **S** over the Node Graph to open the **Project Settings**, or navigate to **Edit** > **Project Settings**.
3. Switch to the **Color** tab and use the **color management** dropdown to select **OCIO**.
4. Set **OCIO config** to **custom** and enter the file path to your **ocio.config** file.
5. Click the viewer process dropdown and select the process you want to register, in this case **sRGB (default)**.
   An error message displays in the Viewer because Nuke can't find the specified reference.
6. Click the viewer process dropdown and select **show panel** to open the OCIODisplay node properties.
7. Click the **Context** tab and enter the **key1** and **value1** pair as SHOT/sRGB.



The LUT specified by the variable is applied to the Viewer.

Note: Viewer processes are display-only LUTs and do not affect rendered output.

Filtering and Spatial Effects

This chapter explains how to create custom filter effects using the Convolve node (see Applying Convolves) and how to simulate depth-of-field (DOF) blurring using the ZDefocus node (see Simulating Depth-of-Field Blurring).

Applying Convolves

The Convolve node lets you create custom filter effects by supplying your own filter image. This image is used as the convolution matrix. In other words, the new value of a pixel is calculated by centering the filter image on the pixel, examining its neighbors, multiplying each pixel value by the corresponding pixel values in the filter image, and then adding the results together. This allows you to defocus a clip and create lens blur effects (bokeh) in the shape of the filter image (also see Simulating Depth-of-Field Blurring).



The **image** input.



The **filter** input.



The result.

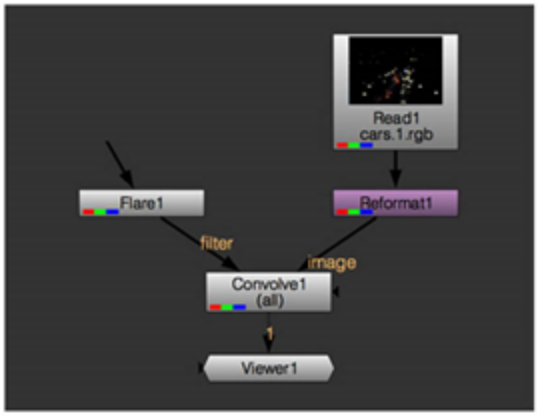
Using the Convolve Node

To use the Convolve node:

1. Click **Filter** > **Convolve** to insert a Convolve node after the image you want to receive the convolution filter effect (the **image** input).
2. Attach a Viewer to the output of the Convolve node.

   An error appears in the Viewer because we haven't connected the **filter** input yet.
3. Connect a filter image to the **filter** input. This image represents the shape and size of the camera aperture used to shoot the **image** input. It can be any shape you like, for example, a pentagon or a hexagon.

   If you don't have a filter image, you can create one using the Roto node (**Draw** > **Roto**) or the Flare node (**Draw** > **Flare**).



A simple Convolve setup.

   Note that you don't necessarily need to crop the filter image to a smaller size, as fast Fourier transforms are used to speed up convolutions with large filter images.
4. In the Convolve node controls, set **channels** to the channels of the **Source** image that you want to affect. By default, the convolve effect is applied to all channels.
5. To select what channel to use from the **filter** input, do one of the following:
   - Set the **filter channel** dropdown menu to the channel you want to use. By default, this menu is set to **rgba.alpha**.
   - To use the same channels from the **filter** input as the **image** input (that is, whatever **channels** is set to), check **use input channels**.
6. In most cases, you can leave **normalize** checked. This means the **filter** input is divided by the sum of all the pixels in it before using it, and ensures that the convolution doesn't change the overall brightness of the image.

7. Use the **filter bounds** dropdown menu in the Convolve controls to select whether you want to limit the filter image to:
   - **shape** - the **filter** input's bounding box. In this case, Convolve only uses the bounding box area, and the center of the filter is the center of the bounding box. This is the default value. You may want to use it, for example, if your **filter** input is a roto shape with a small bounding box that doesn't fill the entire format area.
   - **format** - the **filter** input's format. In this case, Convolve uses the entire format area, allowing you to offset the filter image within the format.
8. If you want to mask the convolve effect, check **mask** and select a mask channel using the controls on the right. For example, you can select a depth channel as a mask to simulate depth-of-field blurring.



A simple Ramp node used as the mask.



The result.

Note that you can also use the ZDefocus node for more accurate depth-of-field blurring. For more information, see Simulating Depth-of-Field Blurring.

9. To dissolve between the original **image** input and the full convolve effect, adjust the **mix** slider.

Tip: You can check **Use GPU if available** to have the node run on the graphics processing unit (GPU) rather than the central processing unit (CPU).

For more information on GPU support, see the Release Notes for your version of Nuke.

You can select the GPU to use in the Preferences. Press **Shift**+**S** to open the **Preferences** dialog, make sure you're viewing the **Preferences** > **Performance** > **Hardware** tab, and set **default blink device** to the device you want to use. You must restart Nuke for the change to take effect.

If you are using a render license of Nuke, you need to add **--gpu** on the command line.

Simulating Depth-of-Field Blurring

The ZDefocus node blurs the image according to a depth map channel. This allows you to simulate depth-of-field (DOF) blurring.



The original image.



Using ZDefocus to simulate a more narrow depth of field. In this case, areas under the green sign remain in focus, whereas areas in front of and behind it are blurred. Any defocused highlights in the image bloom in the shape of the filter image, creating a bokeh effect.

In order to defocus the image, ZDefocus splits the image up into layers, each of which is assigned the same depth value everywhere and processed with a single blur size. After ZDefocus has processed all the layers, it blends them together from the back to the front of the image, with each new layer going over the top of the previous ones. This allows it to preserve the ordering of objects in the image.

Quick Start

Here's a quick overview of the workflow:

1. Create a ZDefocus node and connect it to your script.
   See Connecting ZDefocus.
2. Adjust the blur settings.
   See Adjusting Blur Settings.
3. Adjust the shape of any out-of-focus highlights.
   See Adjusting the Shape of Out-of-Focus Highlights.
4. If necessary, enhance the highlights to make lens shape effects more visible.
   See Enhancing Out-of-Focus Highlights.
5. If necessary, mask the blur effect.
   See Masking Blur Effects.

Tip: You can check **Use GPU if available** to have the node run on the graphics processing unit (GPU) rather than the central processing unit (CPU). For more information on GPU support, see the Release Notes for your version of Nuke.

You can select the GPU to use in the Preferences. Press **Shift**+**S** to open the **Preferences** dialog, make sure you're viewing the **Preferences** > **Performance** > **Hardware** tab, and set **default blink device** to the device you want to use. You must restart Nuke for the change to take effect.

If you are using a render license of Nuke, you need to add **--gpu** on the command line.

Connecting ZDefocus

To connect ZDefocus:

1. Create a ZDefocus node by clicking **Filter** > **ZDefocus**.
2. Connect the ZDefocus node's **image** input to the image you want to blur.

Note: This image also needs to contain a depth map channel. If your depth channel and rgba channels exist in two different files, you can use a Shuffle node to combine them.

3. Use the **channels** dropdown menu to select the channels you want to blur.
4. Set **depth channel** to the channel in the **image** input that contains the depth map. By default, the depth information is taken from **depth.Z**.

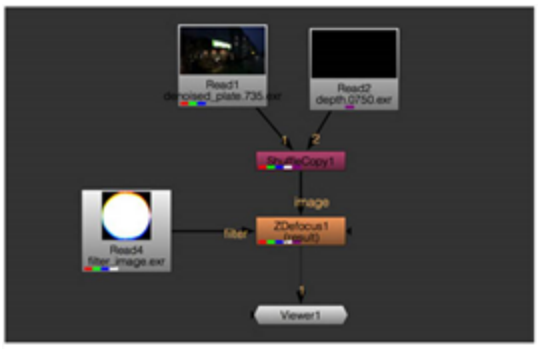
   Note that the depth map should not be anti-aliased. If it is, pixels along an edge between two objects can be assigned a depth that is in-between the depth of the front object and back objects. This looks wrong, as it suggests that those edge pixels are floating somewhere between the objects.
5. If you want to use your own filter image (rather than the predefined **disc** or **bladed** images), connect that to the ZDefocus node's **filter** input and set **filter type** to **image** in the node's properties.

   The filter image represents the shape and size of the camera aperture used to shoot the input footage. As the clip in the **image** input is blurred, any highlights in the clip bloom in the shape of the **filter** image, creating a bokeh effect.

Note: Loading scripts from pre-Nuke 8.0v7 enables the **legacy resize mode** checkbox automatically, for backward compatibilty, and uses the **filter bounds** dropdown to determine how images used in filtering are resized.
Adding new ZDefocus nodes hides the **legacy resize mode** checkbox and allows you to use the **image filter** dropdown to give you more flexibility when calculating blur.
See Using a Custom Filter Image for more information.

   You can create a filter image using the Roto node (**Draw** > **Roto**) or the Flare node (**Draw** > **Flare**), for example.

   Note that you don't necessarily need to crop the filter image to a smaller size, as fast Fourier transforms are used to speed up convolutions with large filter images.
6. Attach a Viewer to the ZDefocus node.



A ZDefocus script.

7. Proceed to Adjusting the Shape of Out-of-Focus Highlights below.

Adjusting Blur Settings

To adjust the Blur settings:

1. Use **math** to specify how you want to use the **depth channel** to calculate the distance between the camera and an object. For example, some programs use higher values to denote further away, while in others they mean closer to the camera:
   - **direct** - The Z value in the **depth channel** directly controls blur. For example, if Z is 0.5, then the blur size is 0.5 times the value of the **size** control (unless this is bigger than **maximum**, in which case it is clamped to **maximum**).
   - **depth** - The Z value in the **depth channel** is the distance between the camera and whatever is in the image at that pixel.
   - **far = 0** - The Z value in the **depth channel** is equal to 1/distance. The values are expected to decrease from large positive values close to the camera to zero at infinity. This is compatible with depth maps generated by Nuke and RenderMan.
   - **far = 1** - Near plane = 0, far plane = 1. This is compatible with depth maps generated by OpenGL.
   - **-direct** - As with the **direct** mode, the Z value in the **depth channel** directly controls blur. In other words, each layer is blurred by the same amount as in the **direct** mode. However, in this mode, the layers are interpreted as being in the opposite order, so a higher depth value places a layer in front of another rather than behind it.
   - **-depth** - The Z value in the **depth channel** is -distance in front of the camera. This is the same as **depth**, but the distances are negative to start with.
   - **far = -0** - The Z value in the **depth channel** is equal to -1/distance. This is compatible with depth maps generated by Maya.
   - **far = -1** - Near plane = 0, far plane = -1.
2. In the Viewer, drag the **focal point** widget on top of the area that you want to be entirely in focus.

   This automatically updates the **focal point** coordinates in the ZDefocus properties and sets the **focus plane** control to the Z depth value at those coordinates. Any layers with this Z depth value are left in focus. You can also set a depth slice around the focus plane that is in focus (as described in step 4), but any other areas of the image are blurred according to the depth map.



Focal point widget under the green sign.



Focal point widget near the cars in the distance.

Tip: The **fill foreground** control, enabled by default, attempts to compensate for missing information by filling regions in the foreground which are revealed when the foreground goes out of focus. However, because the true image information isn't available in these regions, **fill foreground** can sometimes introduce undesirable artefacts by adding things which aren't there. If you see blurry artefacts in the foreground, rather than sharp edge artefacts, try disabling this control.

To better see the effect of your changes, you can also set **output** to **focal plane setup**. ZDefocus now displays depth-of-field (DOF) info in the **rgb** channels:

- **red** - Less than DOF (in front of the area that's in focus).
- **green** - Inside DOF (in focus). Note that if **depth of field** is set to **0**, nothing is displayed in green.
- **blue** - Greater than DOF (behind the area that's in focus).

When **show image** is enabled, this information is overlaid upon the input image.



Focal point widget under the green sign.



Focal point widget near the cars in the distance.

3. To widen the area that is entirely in focus, increase **depth of field**. This sets a depth slice around the focus plane that is entirely in focus (and shown in green whenever **output** is set to **focal plane setup**). True theoretical depth of field would set this to zero.



Depth of field set to 0.



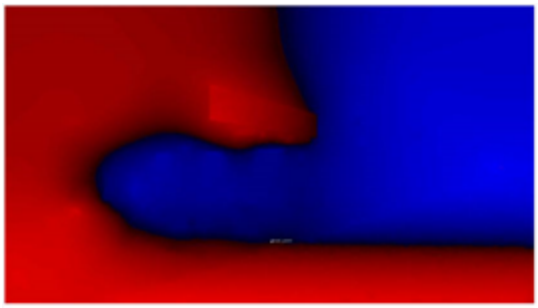
Depth of field set to 1.

4. To apply a small amount of blur to the in-focus region, make sure **blur inside** is enabled. This gives a smoother transition between the in-focus region and the out-of-focus regions around it.
5. Set **output** back to **result** and adjust the amount of blur at infinite depth by setting the **size** value. Note that the amount of blur nearer the camera than the focus plane may be larger.

   If you have set **math** to **direct**, the size is multiplied by the depth to give the blur size at that depth. Setting **size** to 1 allows you to use the values in the depth map as the blur size directly.
6. If you increased the **size** value in step 5, it's a good idea to also increase the **maximum** value. No blurring greater than this value is generated no matter where the object is in relation to the camera.

   For maximum processing speed, you may want to keep this value as low as possible.
7. By default, **automatic layer spacing** is enabled, which means ZDefocus automatically works out how many depth layers to use, based on the maximum blur size (**maximum**). In this mode, the layers are closer together near to the focal plane, where a small change in the blur amount is more obvious, and increasingly more widely-spaced further away (this is equivalent to setting **layer curve** to a value of 1 when controlling the layers manually; see step 8).

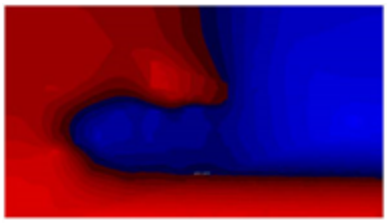
   To visualize the layers, you can set **output** to **layer setup**. This is like **focal plane setup**, but displays depth-of-field (DOF) information after the depth has been divided into layers.



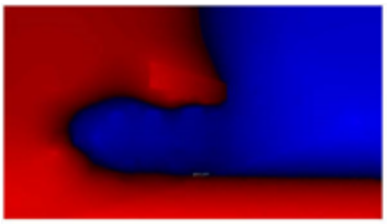
Output set to **layer setup**.

The maximum number of blur sizes that are used between 0 and **maximum** is 256. This means you can have up to 256 layers behind the focal plane, and up to 256 in front of it as well.

8. If you uncheck **automatic layer** spacing, you can control the number of layers manually using **depth layers**. Note that the more layers you use, the longer it takes to process the blur.

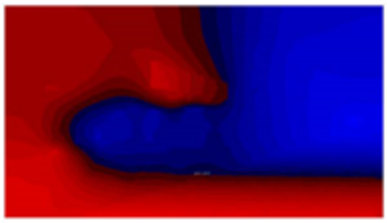


Depth layers set to 10.

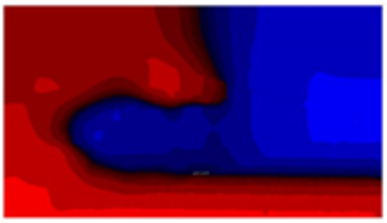


Depth layers set to 50.

If necessary, you can also increase **layer curve** to concentrate the layers around the focal plane. The default value of 0 produces evenly spaced layers.



Layer curve set to 0. This produces evenly spaced layers.



Increasing the **layer curve** value concentrates the layers around the focal plane.

9. Proceed to Adjusting the Shape of Out-of-Focus Highlights.

Adjusting the Shape of Out-of-Focus Highlights

As the clip in the **image** input is blurred, any out-of-focus highlights (bokeh) in the clip assume the shape of the **filter** image (of the **filter type** selected).



The original image.



Out-of-focus highlights (bokeh) in the shape of the **filter** image.

How you create the filter image is up to you. You can:

- Use a predefined disc shape as the filter image. See Using a Predefined Disc Image.
- Use a predefined bladed image (polygon) as the filter image. See Using a Predefined Bladed Image.
- Use your own custom image in the **filter** input as the filter image. See Using a Custom Filter Image.

Using a Predefined Disc Image

1. To see the effect of your changes, set **output** in the ZDefocus controls to **filter shape setup**.
2. Set **filter type** to **disc**.
3. Use the **filter shape** control to dissolve the filter shape between Gaussian at 0 and disc at 1.



Filter shape set to 0 (Gaussian).



Filter shape set to 1 (disc). This is the default value.

4. Use the **aspect ratio** control to set the filter aspect ratio, which is 1:1 by default. Values less than 1 squeeze the filter on the x axis, and values larger than 1 squeeze it on the y axis.

   This allows you to simulate the cat's eye effect, caused by vignetting inherent within some lens designs.



Aspect ratio set to 0.5.



Aspect ratio set to 1.5.

5. Proceed to Enhancing Out-of-Focus Highlights.

Using a Predefined Bladed Image

1. To see the effect of your changes, set **output** in the ZDefocus controls to **filter shape setup**.
2. Set **filter type** to **bladed**.
3. Use the **blades** control to set the number of iris blades that make up the camera's diaphragm. A value of 3 produces a triangle, 4 a square, 5 a pentagon, 6 a hexagon, and so on.

   This field only accepts integers larger than 1.



Blades set to 5. This is the default value.



Blades set to 8.

4. Adjust **roundness** to control the rounding of the filter polygon's sides. A value of 0 equals no rounding.



Roundness set to 0.



Roundness set to 8.

5. Use **rotation** to rotate the filter image in degrees. Positive values produce counter-clockwise rotation, and vice-versa.

6. Use the **aspect ratio** control to set the filter aspect ratio, which is 1:1 by default. Values less than 1 squeeze the filter on the x axis, and values larger than 1 squeeze it on the y axis.

   This allows you to simulate the cat's eye effect, caused by vignetting inherent within some lens designs.

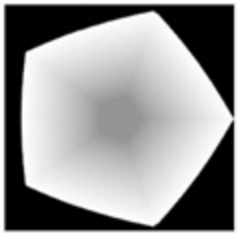


Aspect ratio set to 0.



Aspect ratio set to 2.

7. To adjust the distribution of light inside the out-of-focus highlights:
   - Adjust **inner size** to control the size of the inner polygon, as a percentage of the outer polygon.

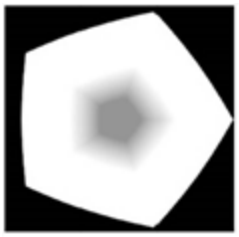


Inner size set to 0.2.



Inner size set to 0.8.

   - Use **inner feather** to add outward or inward feathering around the inner polygon. With values larger than 0.5, your feather effect is outward and, respectively, if your values are smaller than 0.5, the feather effect is inward. A value of 0.5 produces no feathering.



Inner feather set to 0.2.

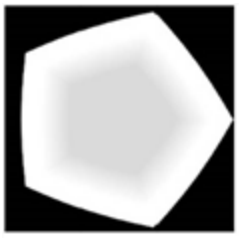


Inner feather set to 0.8.

   - Adjust inner brightness to control the brightness of the inner polygon, where 0 is equal to black and 1 to white.



Inner brightness set to 0.3.



Inner brightness set to 0.7.

- If you want to simulate catadioptric lenses, check **catadioptric**. When using catadioptric lenses, the defocused areas of the image are annular, producing donut-shaped bokeh. You can use **catadioptric size** to adjust the size of the catadioptric hole in the filter.



Catadioptric size set to 0.3.



Catadioptric size set to 0.7.

8. Proceed to Enhancing Out-of-Focus Highlights.

Using a Custom Filter Image

1. Set **filter type** in the ZDefocus controls to **image**.

   This tells ZDefocus to use the **filter** input rather than a predefined shape as the filter image.

   Note that the **filter** image can be a color image. This can be useful, for example, if you want to add color fringing to your out-of-focus highlights to simulate chromatic aberration.



The **filter** image.

2. If you want to display the **filter** image in the Viewer, set **output** to **filter shape setup**.

Note: Loading scripts from pre-Nuke 9.0v2 enables the **legacy resize mode** checkbox automatically, for backward compatibilty, and uses the **filter bounds** dropdown to determine how images used in filtering are resized.
Adding new ZDefocus nodes hides the **legacy resize mode** checkbox and allows you to use the **image filter** dropdown to give you more flexibility when calculating blur.

3. To select what channel to use from the **filter** input, do one of the following:
   - Set the **filter channel** dropdown menu to the channel you want to use. By default, this menu is set to **rgba.alpha**.
   - To use the same channels from the **filter** input as the **image** input (that is, whatever **channels** is set to), check **use input channels**.
4. In newer Nuke scripts, use the **image filter** dropdown to select the required filter. See Choosing a Filtering Algorithm for more information on the available filters.

Note: When using filters that employ sharpening, such as **Rifman** and **Lanczos**, you may see a haloing effect. If necessary, check **clamp image filter** to correct this problem.

In older Nuke scripts with **legacy resize mode** enabled, use the **filter bounds** dropdown to select whether you want to limit the filter image to:

- **shape** - the **filter** input's bounding box. In this case, ZDefocus only uses the bounding box area, and the center of the filter is the center of the bounding box. This is the default value. You may want to use it, for example, if your **filter** input is a roto shape with a small bounding box that doesn't fill the entire format area.
- **format** - the **filter** input's format. In this case, ZDefocus uses the entire format area, allowing you to offset the filter image within the format.

5. Proceed to Enhancing Out-of-Focus Highlights.

Enhancing Out-of-Focus Highlights

To enhance the out-of-focus highlights:

1. To make bokeh lens shape effects warmer and more visible, check **gamma correction**.
   This means a gamma lookup curve of 2.2 is applied before blurring and then reversed for the final output.



Without **gamma correction**.



With **gamma correction**.

2. You can also make the lens shape effects more visible by enabling **bloom**.

   Any highlights above the **bloom threshold** value are multiplied by the **bloom gain** value. Highlights below the **bloom threshold** are not affected.

   This allows you more control over the highlights than **gamma correction**; however, **gamma correction** may bring out some of the highlights better.



Without **bloom**.



With **bloom**.

3. Proceed to Masking Blur Effects.

Masking Blur Effects

To mask the blur effect:

1. Do one of the following:
   - Make sure there is a mask channel in the ZDefocus node's **image** input and nothing is connected to the **mask** input.
   - Connect a mask to the **mask** input of the ZDefocus node. If you cannot see the **mask** input, open the node's controls and make sure **mask** is set to **none**.

     If you want the mask from the **mask** input copied into the predefined **mask.a** channel, also check **inject**. This way, you can use the **mask** input again downstream.
2. If you don't want to use the alpha channel as the matte, select the channel you want to use from the **mask** dropdown menu.

   By default, the blur is limited to the non-black areas of this channel.
3. If necessary, check **invert** to reverse the mask, so that the blur is limited to the non-white areas of the mask.
4. To blur only the edges of the mask, check **fringe**.

5. To dissolve between the original image (at 0) and the full ZDefocus effect (at 1), adjust the **mix** slider. A small light-gray square appears on the node in the Node Graph to indicate that the full effect is not used.

Removing Elements Using Inpaint

Nuke's Inpaint is a time saving node for removing unwanted elements, such as tracking markers, blemishes, or wires. Inpaint uses surrounding pixels to fill an area marked in the alpha channel of the source image or **Matte** input. Inpaint also benefits from GPU acceleration to provide fast results.

See Simple Element Removal Using Inpaint for more information.

For more complex textures, the **Stretch** controls bias the inpainting in a defined direction, allowing you to replicate textures in the inpainted area.



Source clip.



Inpainted texture.

You can also use the **Detail** controls to pull high frequency textures from another part of the **Source** image, or even from a different image using the **Detail** input to improve results. For example, you can apply grain to an inpainted area using a Grain node attached to the **Detail** input.



Source clip.



Grain texture.

Note: The Viewer **gain** and **gamma** controls have been adjusted to show the effect more clearly.

See Inpainting Textures and Detail for more information.

Simple Element Removal Using Inpaint

Inpaint employs contextual paint strokes or alpha shapes to quickly remove unwanted elements in shots. This means that the correction is updated at each frame automatically, so differences in color or light are taken into account.



Source clip.



Wires removed using Inpaint.

You can drive paint and roto shapes using a Tracker node or GridWarpTracker node to propagate your changes throughout a sequence.



Source clip.



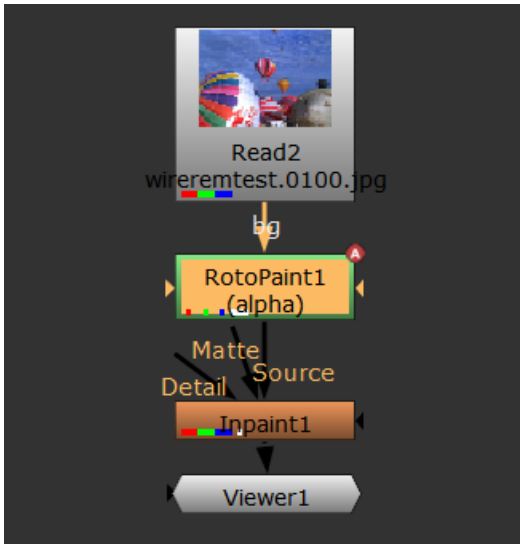
Element removed using Inpaint.

Cleanup Using Inpaint

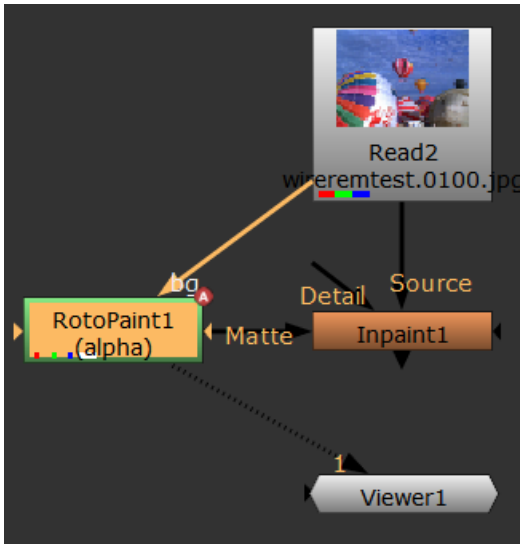
RotoPaint strokes are a quick way to clean up tracking markers or rigging using Inpaint.

Tip: You can animate paint strokes using tracking data in the same way as roto shapes. See Animating Roto Shapes with Inpaint for more information.

1. Add a RotoPaint node to the node tree.
2. Add an Inpaint node from the **Filter** menu and set the **Fill Region** to either **Source Alpha** or **Matte Alpha**, depending on how your RotoPaint node is connected.



Source Alpha.



Matte Alpha.

3. Open the RotoPaint **Properties** panel and set the **output** control to **alpha**.

Note: If your background plate has an alpha channel, you can enable **replace** in the RotoPaint **Properties** so that the correction strokes replace the plate's alpha channel.

4. Select the Brush tool from the RotoPaint controls on the left of the Viewer and paint your correction strokes.

Tip: The **Mat** Viewer display style (press **M** in the Viewer) is active in the image to show the wires and the paint at the same time.



Inpaint pulls pixels from around the strokes to fill the painted area.



Source clip.



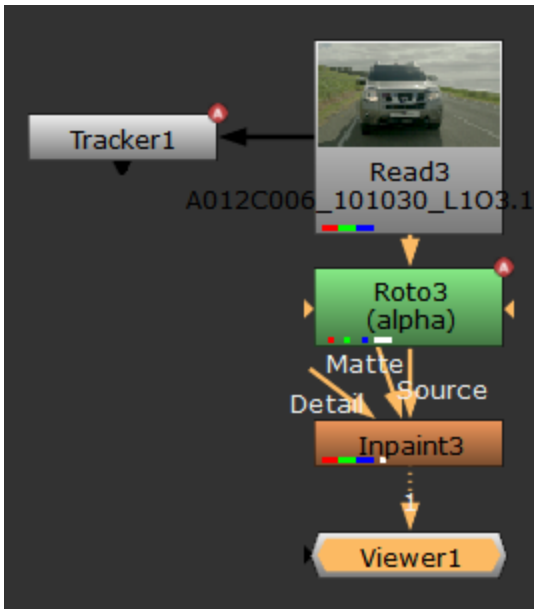
Wires removed using Inpaint.

See Inpainting Textures and Detail for more information on improving inpainting results.

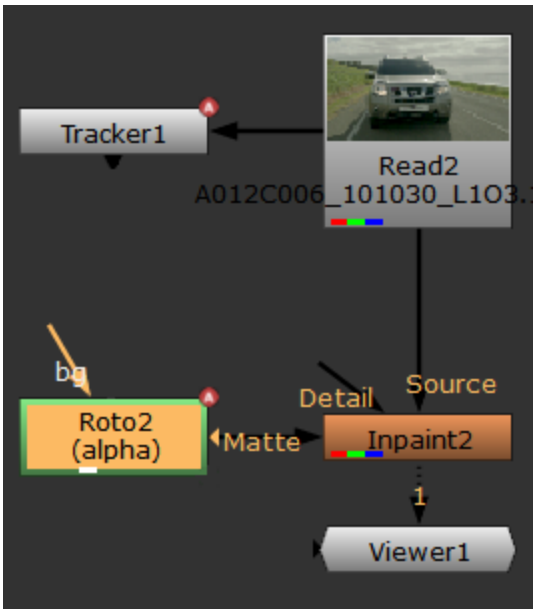
Animating Roto Shapes with Inpaint

Driving Roto shapes using tracking data is a quick way to remove an element throughout a shot using Inpaint.

1. Track the elements you want to remove throughout your sequence using a Tracker node. See Tracking and Stabilizing for more information.
2. Add a Roto or RotoPaint node to the node tree.
3. Add an Inpaint node and set the **Fill Region** to either **Source Alpha** or **Matte Alpha**, depending on how your Roto node is connected.



Source Alpha.



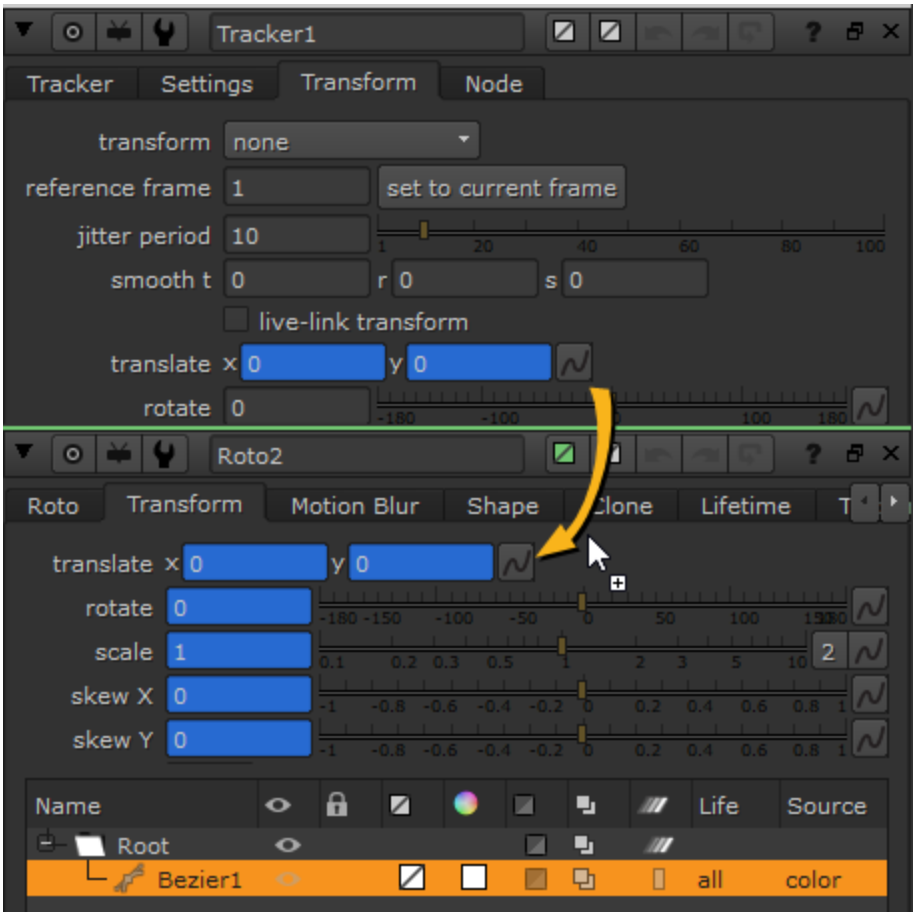
Matte Alpha.

4. Use the Roto node **Bezier** tool to draw an alpha shape around the element you intend to remove.

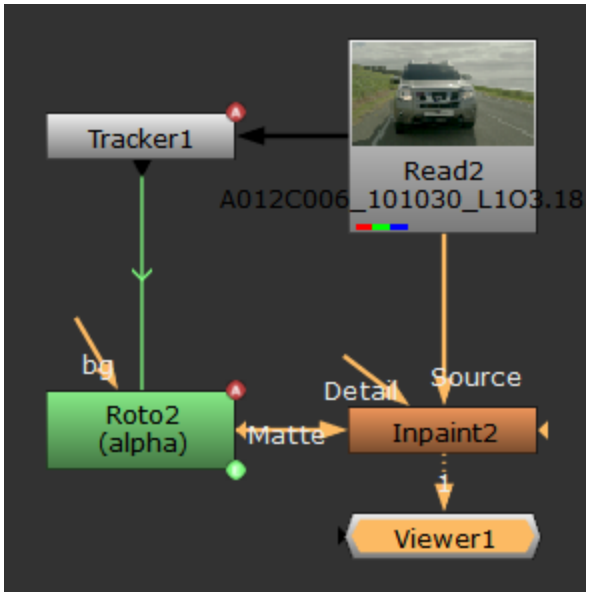


Inpaint pulls pixels from around the shape to fill the area enclosed by the shape.

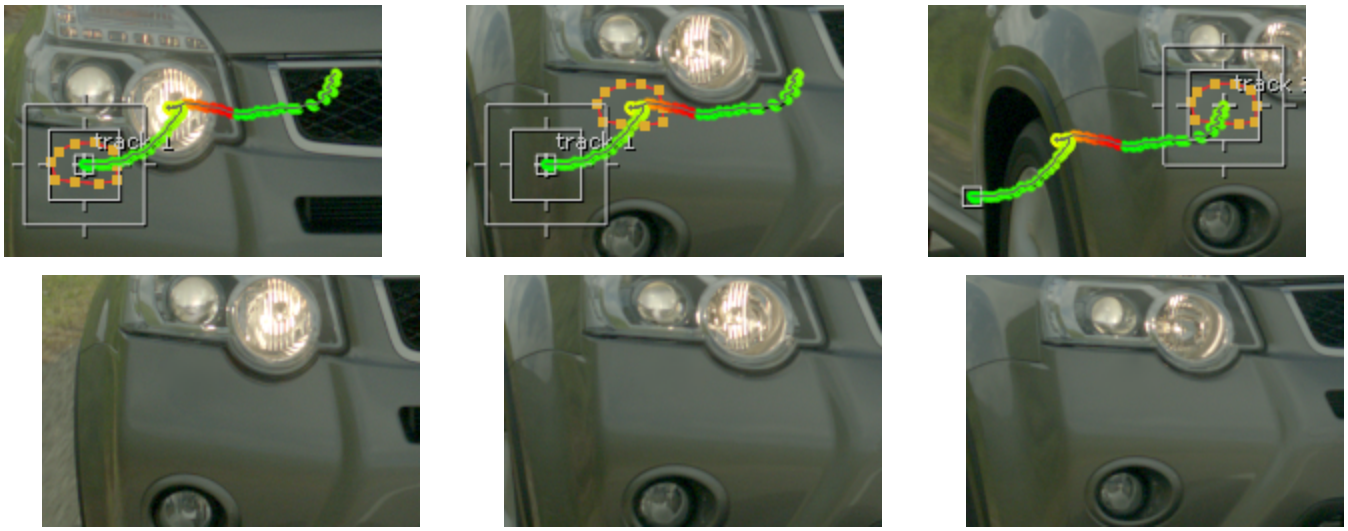
5. Open the Tracker node's **Properties** panel and select the track in the **Tracker** tab.
6. Click the **Transform** tab and **Ctrl/Cmd** drag the Tracker **translate** animation control to the corresponding control in the Roto **Properties**.



An expression link, represented by the green arrow in the Node Graph, is created between the controls so that the Tracker's **translate** keyframes drive the Roto shape automatically.



7. Play through the sequence to check that the shape covers the element you want to remove throughout the shot.



Tip: In some sequences, the inpainted texture can flicker between frames. Try adjusting the **Smoothness** control to reduce the effect. Higher values can help to reduce flickering between frames, but at the expense of local detail.

See Inpainting Textures and Detail for more information on improving inpainting results.

Inpainting Textures and Detail

Removing elements with Inpaint is easy and fast, but the results on more complex textures and shots are not always perfect. Inpaint's properties include **Stretch** and **Detail** controls to improve the results.

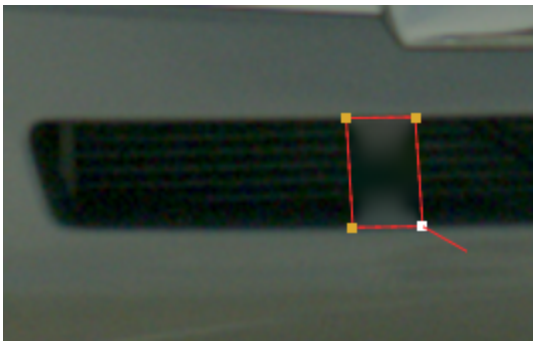
Improving Textures

Inpaint's **Stretch** controls allow you to manipulate the extent and direction of the painted pixels.

1. Create your paint stokes or roto shapes as described in Simple Element Removal Using Inpaint.
   In this example, we're using a roto shape in the alpha channel.



Source clip.



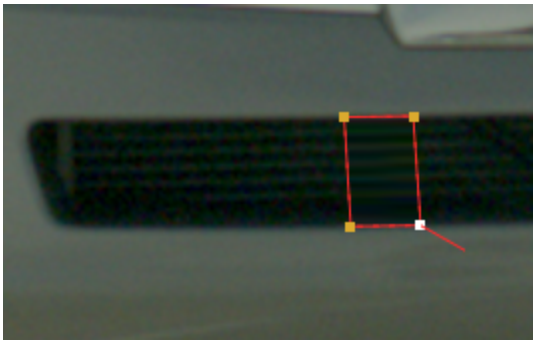
Initial roto shape.

The inpainting in the example is influenced by the car body above and below the grill.

2. Increase the **Amount** to increase the stretch applied to edge pixels and adjust the **Direction** to control the origin of the pixels.
   In this example, the texture we're interested in is mostly horizontal, so changing the direction to 90° pulls in pixels from the top and bottom of the roto shape.
3. Set the **Direction** to 0 or 360 to pull pixels from the left and right of the shape.
4. Increase the **Amount** to stretch the inpainted pixels toward the center of the shape.
   The grill textures are stretched from the left and right to fill the inpainted area correctly.



Source clip.



Inpainted texture.

Inpainting Detail

Inpaint's **Detail** controls allow you to replace image data lost due to inpainting.

1. Create your paint strokes or roto shapes as described in Simple Element Removal Using Inpaint.
   In this example, we're using a roto shape in the alpha channel.

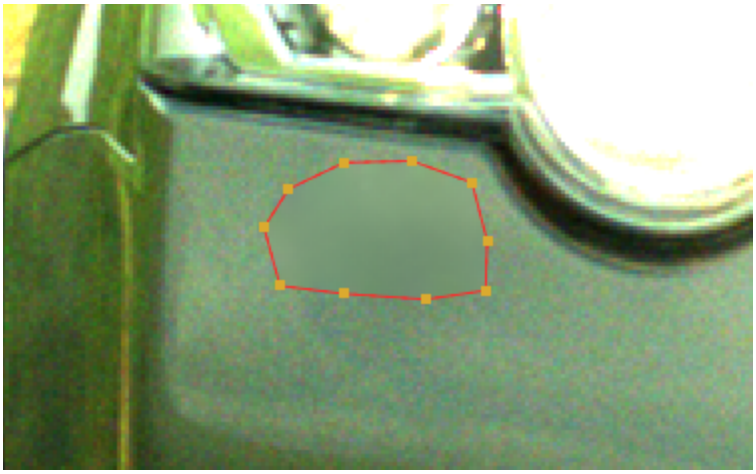


Source clip.



Initial roto shape.

Zooming into the Viewer shows that the inpainting has smoothed the grain in the source image.

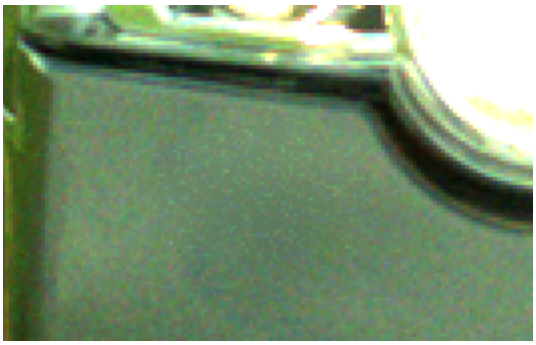


Note: The Viewer **gain** and **gamma** controls have been adjusted to show the effect more clearly.

2. Set the Inpaint **Source** dropdown to either **Source** or **Detail** to determine the origin of the image detail.
   - **Source** - the detail is pulled from the **Source** image.
   - **Detail** - the detail is pulled from the **Detail** input.
3. Use the **Amount** and transform controls to set how much detail is added and where in the image the detail originates.
   In the example, you could pull grain detail from another part of the car body in the **Source** image or you could attach a Grain node to the **Detail** input and simulate the required grain.



Inpainted texture.



Grain added from the specified Source.

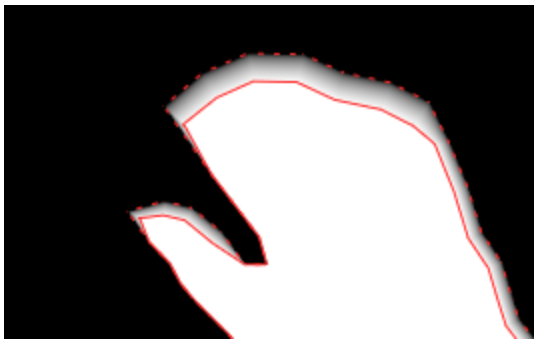
Note: The Viewer **gain** and **gamma** controls have been adjusted to show the effect more clearly.

Correcting Foreground Color Using EdgeExtend

Nuke's EdgeExtend node allows you to correct unpremultiplied foreground color at the edge of soft mattes. The mattes can be hand-drawn roto shapes or keyed mattes created by nodes like Keylight and Primatte.

See Using EdgeExtend for more information.

You can also output an edge mask to allow you to work on the edges separate from the rest of the image. A typical use case is adding grain to the edges of a matte without affecting the source image.



Dilated matte.



Edge mask.

See Working with Edge Masks for more information.

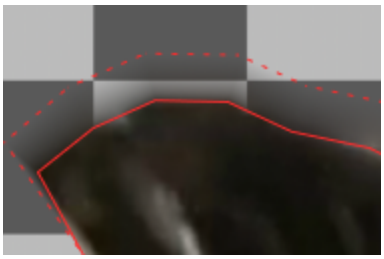
Note: Images illustrating EdgeExtend courtesy of Little Dragon Studios. All rights reserved in the United States and/or other countries.

Using EdgeExtend

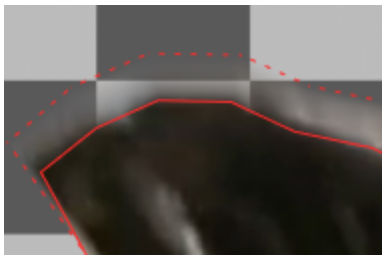
EdgeExtend allows you to correct unpremultiplied foreground color at the edge of soft mattes by pulling pixels from deeper inside or outside your matte.



Source image.



Merged matte.

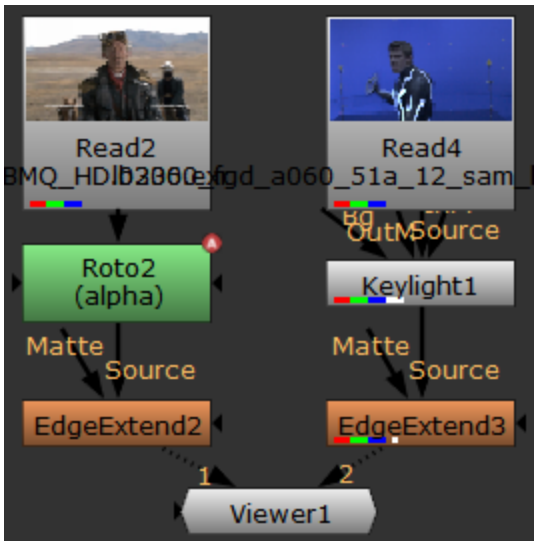


Matte dilation correcting foreground motion blur color.

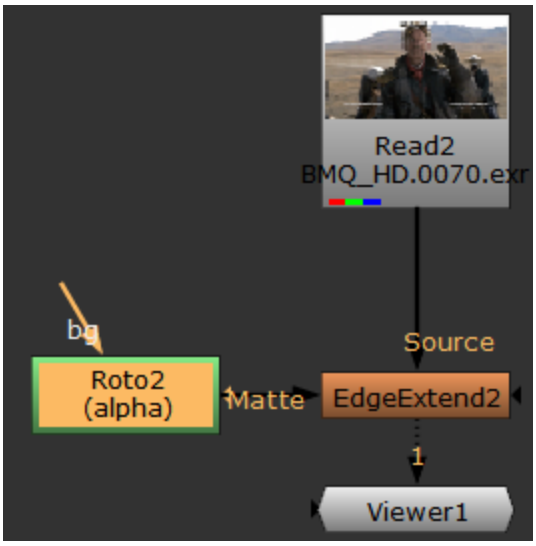
1. Add a Roto node to the node tree and use the **Bezier** tool in the Viewer to create the matte. You don't need this step if you're using the image's alpha channel or a keyed matte from nodes such as Keylight or Ultimatte.

Tip: See Using the Bezier and Cusped Bezier Tools for more information on using Roto and RotoPaint.

2. Add an EdgeExtend node and set the **Matte** control to either **Source Alpha** or **Matte Alpha** in the **Properties** panel, depending where the alpha channel was generated.



Source or keyed alpha.



Matte alpha.

3. Adjust the **Erode** control to erode or dilate the matte. Positive values decrease the extent of the matte and negative values increase it.



Matte erosion.

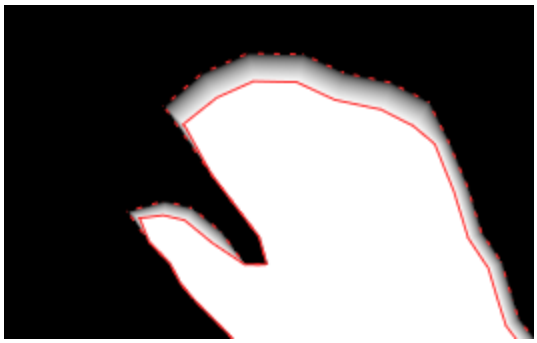


Matte dilation.

In the example, dilation recovers more color from the motion blur. You can then output the matte to an **Edge Mask** channel so that you can work on the edges in isolation, without affecting the rest of the image. See Working with Edge Masks for more information.

Working with Edge Masks

Edge masks allow you to work on the edges separate from the rest of the image. You can write edge masks to an existing channel preset or create a new custom layer to hold the mask.



Dilated matte.



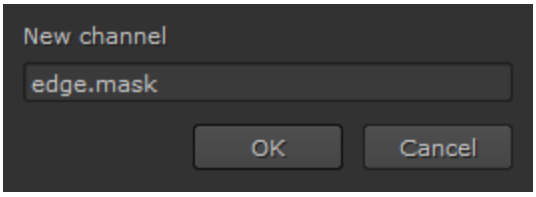
Edge mask.

1. Dilate or erode your matte as described in Using EdgeExtend.
2. In the EdgeExtend node **Properties** panel, click the **Edge Mask** dropdown and select a channel to store the mask,

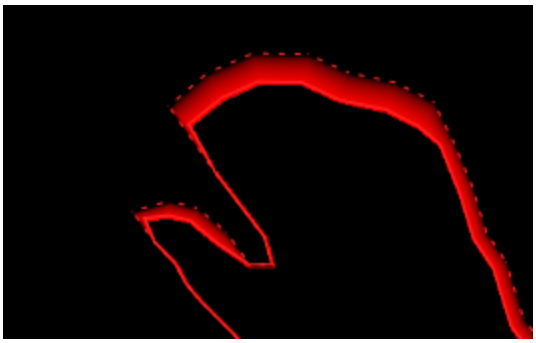
   OR

   Click **new** and enter the name of a new layer to hold the mask channel.

   The layer is the first part of the name and the channel is the second part. For example, **edge.mask** creates an **edge** layer containing a **mask** channel.



3. Click **OK** to create the mask channel.
4. Enable **Premultiply** and select the **edge** layer in the Viewer to display the edge mask.



5. To see the edge mask and the image together, switch the Viewer layer dropdown to **rgb** and the channel dropdown to **edge.mask**, and then press **M** in the Viewer to enable the **Mat** Viewer display style.



You can now work on the edge mask in isolation in the **edge.mask** channel without affecting the rest of the image.

Bilateral Filtering

The Bilateral node employs a smoothing filter that operates by mixing nearby source pixels according to their spatial distance and color similarity. Unlike the Blur node, this filter is particularly good at preserving edges, though it can be computationally expensive.

The Bilateral node includes GPU acceleration and an optional **guide** input to compute color similarity while filtering, which can be useful when scaling images.

Preserving Edges

The standard blurring employed by Nuke can destroy edges in images, particularly as the size of the blur increases. Bilateral features a number of controls that can help retain edges by giving more 'weight' to pixels at a given position or color value.

The default, low values for the **Positional** and **Color Sigma** only blend pixels that are close to each other and of a similar color, which can help preserve edges.

For example, adjusting the controls of the Bilateral node can help preserve edges in dense point clouds.



Blending in the 3D Viewer with no filter.

Blending using the Bilateral node.

Using the Guide Input

Bilateral's optional **guide** input can be connected to a depth map to improve filtering further using color similarity. The depth in the image helps Nuke to preserve edges using the extra information in the **guide** image.



The source image with no filter applied.



Blur **Size** 10 with a high **Positional Sigma**.

The same blur with **Color Sigma** adjusted.

Connecting the **guide** input converts the output from the filter to the same format as the guide image, resulting in a joint bilateral resampling filter. If the **src** and **guide** images have the same format, the node acts as a cross bilateral filter.

If the **guide** input is not connected, the output format is equal to that of the **src** input and acts as a standard bilateral filter.

Creating Effects

Several nodes in Nuke let you create various effects on your input images. In this chapter, we describe three of these nodes: LightWrap, Glint, and Text.

Quick Start

Here's a quick overview of the workflow:

1. To adjust the soft edges and light spills that occur in the border between your foreground and background, you can use the LightWrap node (**Draw** > **LightWrap**). For more information, see Background Reflections on Foreground Elements.
2. With the Glint node you can add star-shaped glints on your image. See Creating Star Filter Effects on Image Highlights.
3. Using the Text node, you can add text elements, such as credits, to your footage. For more information, see Creating Text Overlays.

Background Reflections on Foreground Elements

You can use the LightWrap node to create background reflections on foreground elements. The node creates a reflection of light around the edges of your foreground element by blending in whatever is in the background.



The composite without the LightWrap effect.



The composite with the LightWrap effect.

You may have noticed that objects filmed in front of very bright backgrounds have the appearance of softened edges as the light spills round from the background. When adding foreground layers onto a background plate, using the LightWrap node can dramatically improve the quality of your composite.



The composite without the LightWrap effect. Note the hard edges around the trailing hand and the bottom of the skirt.



The composite with the LightWrap effect. Note how the foreground fits better into the background.

If you want to use LightWrap, you should apply it on your foreground element before you composite the foreground over the background with the Merge node.

Using the LightWrap Node

To use the LightWrap node:

1. Select **Draw** > **LightWrap** to add a LightWrap node after your foreground and background images.

2. Connect your foreground element to input **A** of the LightWrap node, and the background image to input **B**.
3. Connect a Viewer to the output of the LightWrap node so you can see the effect of your changes.
4. Adjust the **Diffuse** and **Intensity** sliders to control both the spread and brightness of the reflections on the foreground element. These sliders need to be balanced out together. You may want to start by bringing Diffuse all the way down to better see what you are blending in from the background. Then, adjust Intensity before going back to the Diffuse slider and, if necessary, Intensity again until you are happy with the result.



Low **Diffuse**.



High **Diffuse**.



Low **Intensity**.



High **Intensity**.

5. If you want to create a uniform effect around the edges of the foreground rather than have the effect adjust itself according to the background, check **Disable luminance based wrap** on the **LightWrap** tab.
6. In case you don't want to merge the LightWrap effect with the foreground element in order to keep the LightWrap effect as a separate element, check **Generate wrap only** on the **LightWrap** tab.



7. By default, the LightWrap effect is only applied inside the foreground element's alpha. If you want to extend the effect outside it, making the element seem to glow, check **Enable Glow**.



8. On the **Tweaks** tab, you can also adjust the following controls:
   - **FGBlur** to determine how much the foreground matte is blurred. The more blur, the more of the background is added to the foreground.
   - **BGBlur** to control how much the background is blurred before it is merged with the foreground element.
   - **Saturation** to adjust the saturation of the effect.
   - **Luma Tolerance** to increase or decrease the luminance values of the effect.
   - **Highlight Merge** to control how the foreground element is merged with the background. The default merge operation, called **plus**, adds the elements together, producing a glow effect.
   - Check **Use constant highlight** to use a constant color of your choice rather than the background in the LightWrap effect. Select the color using the controls next to **Constant Highlights Color**.
9. On the **CCorrect** tab, you can color correct the LightWrap effect produced.

Creating Star Filter Effects on Image Highlights

With the Glint node, you can create star-shaped rays around all the bright points in an image.



The original image with bright points.



The image after using the Glint node.

Tip: This effect can be helpful for adding realism to scenes with water and bright sunlight.

Using the Glint Node

To use the Glint node:

1. Select **Draw** > **Glint** to add a Glint node after the image you want to add star-shaped rays to.
2. From the **channels** dropdown menu and checkboxes, select the channels to which you want to apply the effect.
3. In the **no. of rays** field, enter the number of rays you want coming out of the bright points in your image. For example, if you want to create five-pointed stars, enter 5.
4. To change the threshold for how bright the highlights in the input image need to be to cause the glint effect, adjust the **tolerance** slider. Only the pixels above the threshold bloom with the effect.



Low **tolerance** value.



High **tolerance** value.

5. To determine the length of the rays, adjust the **length** slider. To give every other ray a different length and determine that length, adjust the **odd ray length** slider.
6. To determine how many steps the rays are formed of, enter a value in the **steps** field. The more steps you use and the shorter the rays are, the smoother the rays become.

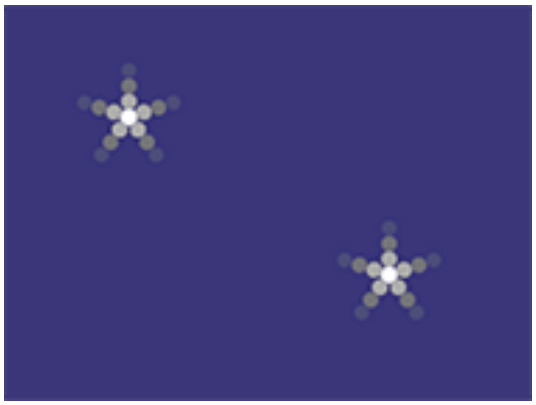


The **steps** value set to 3.

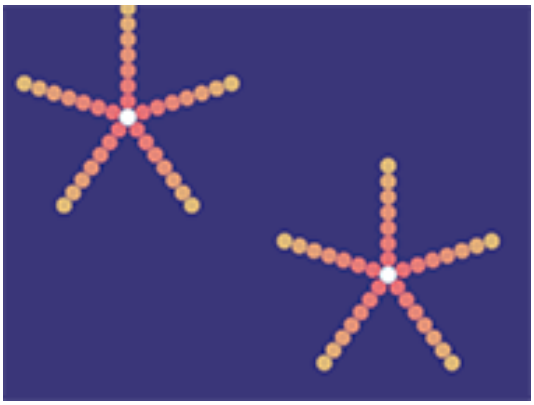


The **steps** value set to 5.

7. To rotate the star-shapes, adjust the **rotation** slider. Increasing the value rotates the rays clockwise, whereas decreasing the value rotates them counter-clockwise.
8. To change the color in the beginning of the rays near the center point of the stars, adjust the **from color** slider. To change the color in the end of the rays, adjust **to color**. By default the from color is set to white, and the to color to black.



From color set to white
and **to color** set to black.



From color set to pink
and **to color** set to yellow.

9. If needed, you can also make the following adjustments:
   - If you want to change the aspect ratio of the stars, adjust the **aspect ratio** slider.
   - By default, the brightest image on the rays is used as the center point for the star. However, if you prefer the images forming the rays to be added up in forming the center point, uncheck **max**.
   - To only output the Glint effect without merging it into the original input image used to create it, check **effect only**.



Effect only disabled.



Effect only enabled.

   - To mask the shape that is used to create the rays, check **w** and select the mask channel from the dropdown menu.
   - To perform a gamma correction on the highlights that cause glint before the glint effect is applied, adjust the **gamma** slider.
   - To mask the glint effect, check **mask** and select a mask channel using the controls on the right.
   - To dissolve between the original input image and the full glint effect, adjust the **mix** slider.

Creating Text Overlays

Using Nuke's Text node, you can add text overlays on your images. You can simply type in the text you want to have displayed, or use Tcl expressions (such as **[metadata values]**) or Tcl variables to create a text overlay.

See Entering Text for a list of Tcl expressions, Tcl variables, HMTL named entities, hex entities, and decimal entities you can use in the Text node.

Text overlays can also be animated using animation layers so that their properties, such as position, size, and color, change over time. These features make the Text node useful, for example, for creating slates or scrolling credits.

Preparing a Text Overlay

Creating text overlays involves some preparation, particularly which channels hold the text and the positioning of the initial text box.

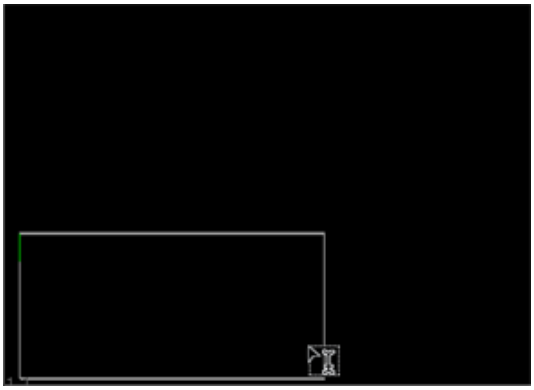
1. Select **Draw** > **Text** to create a Text node and connect it to a Viewer.
2. In the Text node properties, select the channels you want the text to appear in from the **output** controls.
3. If you want to multiply any channels by the drawn text so that they are set to black outside the text shape, select those channels using the **premult** controls.
4. From the **clip to** dropdown menu, select how you want to restrict the output image.
   - **no clip** - do not restrict the output image.
   - **bbox** - restrict the output image to the incoming bounding box.
   - **format** - restrict the output image to the incoming format area.
   - **union bbox+format** - restrict the output image to the combination of the incoming bounding box and format area.
   - **intersect bbox+format** - restrict the output image to the intersection of the incoming bounding box and format area.
5. If you want to clear the affected channels to black before drawing on them, check **replace**.

   By default, **replace** is not enabled and the text is drawn on top of the input image.
6. If you're happy with the text initially appearing in the top left corner of the format area, you can start Entering Text using the **message** field.
7. If you'd like to have more control over the text's initial position or draw a text box to constrain your text, ensure that **edittext** is enabled above the Viewer and then either:
   - Click in the Viewer to place the cursor where you want the text to appear,

OR

- Draw a box in the Viewer to contain your text.

Tip: The cursor appears as in the Viewer when you initially add the Text node.



Text entered in a custom box is limited by the **boxx**, **r**, and **t** boundaries on the **Transform** tab, but the **y** boundary can be overstepped.

Tip: You can adjust the box at any time using the **boxxyrt** controls or by dragging the Viewer handles.

8. If you want to mask out part of the image, see Masking Regions of the Viewer
9. Once you're happy with the cursor position, proceed to Entering Text.

Entering Text

The **message** field in the properties panel is used to enter the text, Tcl expressions, Tcl variables, or a combination of these that you want to display in your output. For some examples of these formats, see the table below.

Text entry behaves much the same as regular text editors, but there are a few rules to observe:

- The **edit text** control above the Viewer must be enabled if you want to type directly in the Viewer, though you can enter text in the message field at any time.
- Press **Return** to begin a new line in both the Viewer and the **message** field.
- Navigate around the Viewer or **message** field using the arrow keys.
- Tcl expressions must be placed in square brackets, such as **[date]**.

- To display special Unicode characters, such as foreign language characters and copyright signs, you can:
  - Use HTML named entities, such as **<** to display <, or **©** to display ©.
  - Use hex entities, such as **<** to display <, or **©** to display ©.
  - Use decimal entities, such as **<** to display <, or **©** to display ©.
  - Type Unicode characters, such as < or ©, on your keyboard or cut and paste them from other applications. UTF-8 character encoding is used to store them in the control's value and in the saved Nuke script.

These special characters only work if the font you are using supports the required character.



Note: We recommend using the above entities rather than typing <, for example. This is because future versions of the Text node may interpret HTML mark-up. In HTML, some characters, such as the greater than and less than signs, are reserved. If you used these signs within your text now, future versions could mistake them for HTML mark-up.

Example Variables and Entities

The following table gives examples of Tcl expressions, Tcl variables, HMTL named entities, hex entities, and decimal entities you can use in the **message** field of the Text node.

Message	Prints
Tcl expressions	
[date]	Week day, day, month, hh:mm:ss, and time zone. For example, **Thu Jan 15 14:22:20 GMT**.
[date %a]	Abbreviated week day name. For example, **Thu**.
[date %A]	Full week day name. For example, **Thursday**.
[date %b]	Abbreviated month name. For example, **Jan**.
[date %B]	Full month name. For example, **January**.
[date %d]	Day (01-31).
[date %D]	Date (dd/mm/yy). For example, **15/01/10**.

Message	Prints
[date %H]	Hour (00-23).
[date %I]	Hour (01-12).
[date %m]	Month (01-12).
[date %M]	Minutes (00-59).
[date %p]	AM or PM.
[date %r]	Time (12-hour clock). For example, **11:04:07 AM**.
[date %S]	Seconds (00-59).
[date %T]	Time (24-hour clock). For example, **14:06:54**.
[date %y]	Abbreviated year (00-99). For example, **10**.
[date %Y]	Full year. For example, **2010**.
[date %z]	Numeric time zone. For example, **-0800**.
[date %Z]	Time zone. For example, **GMT**.
[frame]	Frame number. For example, **23**.
[metadata]	List of all the keys in the incoming metadata.
[metadata values]	List all of the keys and values in the incoming metadata.
[metadata *key*]	Value of the key in the incoming metadata. Replace *key* with the name of the key whose value you want to display. For example, you may be able to use **[metadata input/filename]** to display the name and location of the image file, or **[metadata input/ctime]** to display the timestamp for an input file.
[value root.name]	Script directory path and script name. For example, **Users/john/Nuke_scripts/myscript.nk**.
Tcl variables	
$env(*ENVIRONMENT_VARIABLE*)	The value of the environment variable specified. Replace *ENVIRONMENT_VARIABLE* with an environment variable you have set. For example, you can use **$env(USER)** to display the user name (for example, **john**) on Mac and Linux, or **$env(USERNAME)** to display it on Windows and Linux.

Message	Prints
	For a list of environment variables specific to Nuke, see Environment Variables.
$version_long	The full version number of Nuke. For example, 17.0v3.
$threads	Number of render threads used to calculate images. This is in addition to a main thread used to update the graphical user interface (GUI).
HTML named entities	
&	&
'	'
Å	Å
Á	Á
Â	Â
Æ	Æ
À	À
Ç	Ç
©	©
é	é
ê	ê
è	è
ë	ë
€	€
>	>
<	<
Ñ	Ñ
ø	ø

Message	Prints
õ	õ
Ö	Ö
ö	ö
"	"
®	®
ß	ß
Ü	Ü
ü	ü
hex entities	
#	#
%	%
&	&
*	*
@	@
™	™
œ	œ
š	š
<	<
>	>
©	©
é	é
decimal entities	
£	£

Message	Prints
©	©
®	®
¿	¿
ê	ê
ß	ß
à	à

Tip: To get a list of all the Tcl expressions you can use with **date**, type **x** on the Node Graph, set the script command dialog that opens to **Tcl**, enter **date -h**, and click **OK**.

Fonts and Font Properties

The FreeType library used by the Text node supports a large number of fonts, including TrueType, OpenType, and PostScript fonts. In order to display the available fonts faster in the **font** dropdown menu, these fonts are cached as an XML file when you open a Text node's properties panel. The font cache XML file is called **fontmapping.fcache** and its location is specified by the environment variable NUKE_TEMP_DIR. See Updating the Font Cache for more information.



Note: See Environment Variables for more information on setting environment variables.



Note: The font cache file is not used when rendering and is not required on a render farm.

Nuke retrieves the fonts available in the Text node from various locations in a set order, before caching them:

1. The location specified in the **Project Settings** > **Font** > **project font path** control.
2. The **.nuke/fonts** directory and all other plug-in folders.
3. The location specified by the NUKE_FONT_PATH environment variable.

Note: Locations 1 and 3 allow you to specify multiple paths using the OS standard syntax. On Windows, for example, you could specify "c:\windows\font1;c:\windows\font2;..."

4. The local fonts folder.
5. The system fonts folder, assuming that **Project Settings** > **Font** > **include system fonts** is enabled. For example, on Windows C:\Windows\Fonts\.

Note: If several locations contain the same font, Nuke uses the font from the first directory it finds it in.

In general, there is no guarantee that a font family and style, such as {Arial : Regular}, will render identically cross-platform. To avoid this, Nuke ships with several fonts common to all platforms. If you disable **Project Settings** > **Font** > **include system fonts**, only the fonts that ship with Nuke are available in the font controls.

See Third-Party Libraries and Fonts for a list of fonts that ship with Nuke.

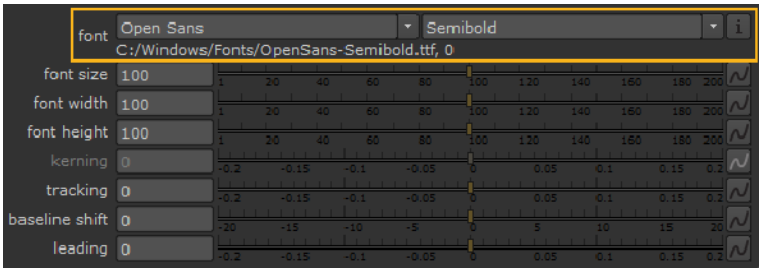
Note: Only fonts that are physically stored in a file are available in the **font** controls. As a result, the list of system fonts available in Nuke may be different to that available in other software applications on the same machine.

Selecting a Font

1. Highlight the text you want to affect in the **message** field or Viewer.
2. In the Text properties panel, click the **font family** dropdown to select the required family.
3. Select the required **font style** from the dropdown.

Note: The styles available depend on the family selected.

4. Click the information icon to display the location of the selected font. For example:



The font family and style used are saved in the Nuke script, along with some extra information, in order to make sure the same family and style are used if the script is opened on a different machine or operating system. If the font family and style are not the same, Nuke displays a warning message.

Adjusting Font Size and Spacing

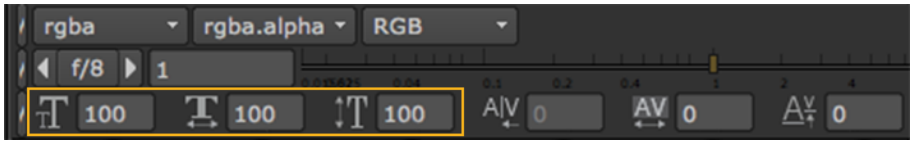
1. Highlight the text you want to affect in the **message** field or Viewer.
2. To adjust the overall size of the font, use the **font size** slider. When **leading** is set to 0, this parameter also controls the spacing between each line of text.

   When rendering the font, **font size** controls the font hinting used. Font hinting adjusts which pixels are interpolated to more clearly render a font. At small sizes and on low resolution output devices, it has a big impact on the legibility of the font. For best results, you should use the **font size** parameter (rather than the **scale** control on the **Group** tab) to control the size of the font and keep **scale** set to 1.

Tip: Use the **global font scale** control to adjust the scale multiplier for all characters in the current Text node, regardless of the **font size** specified for individual characters.

3. You can adjust fonts asymmetrically using the **font width** and **font height** controls.

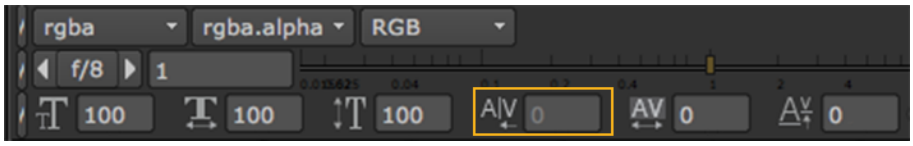
   You can also use the controls above the Viewer to adjust font size.



4. To increase or decrease the spacing between individual characters, adjust the **kerning** slider. By using negative values, you can make the letters overlap.

Note: If you select more than one character or the last character in a group, the **kerning** control is disabled.

You can also use the control above the Viewer to adjust kerning.



5. To increase or decrease the space between each character and the previous character adjust the **tracking** control. Negative values move characters towards the preceding character.

   You can also use the control above the Viewer to adjust tracking.



6. To adjust the selected text's height above the baseline, in screen space, use the **baseline shift** control. The baseline is the imaginary line upon which most letters rest. This control allows you to reliably line up the baseline of different fonts.

   You can also use the control above the Viewer to adjust baseline.



7. If your text overlay contains several lines of text, you can adjust the spacing between each line by using the **leading** slider. By using negative values, you can make the letters overlap.

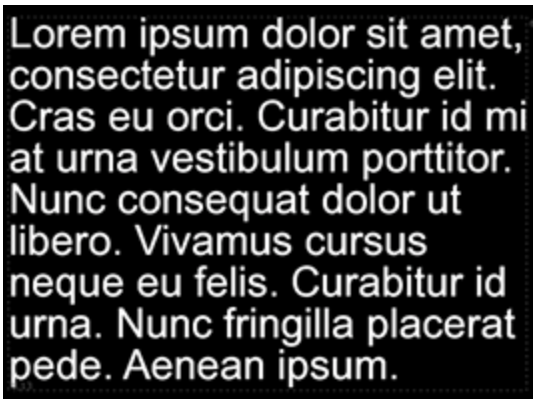


Note: Unlike the **baseline shift** control, **leading** affects all text within the bounding box regardless of selection.

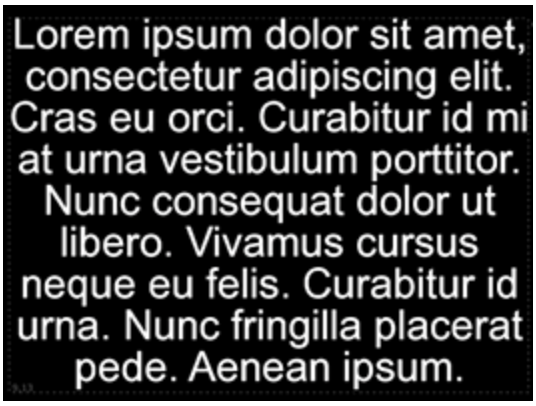
Justifying Fonts

The justify controls affect the positioning of the text within the bounding box on the x and y axes.

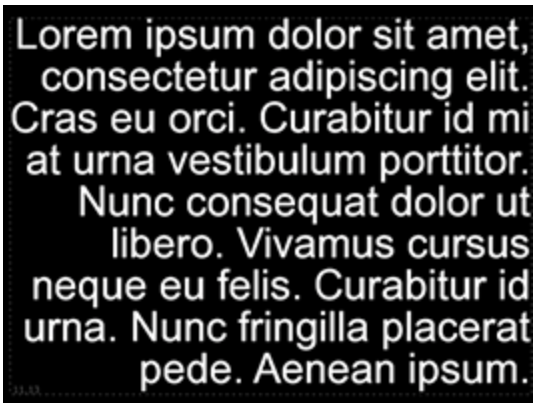
1. Use the **xjustify** dropdown to control how you want to align the text horizontally:
   - **left** - align the text along the left edge of the on-screen text box. This leaves the right edge of the text ragged.



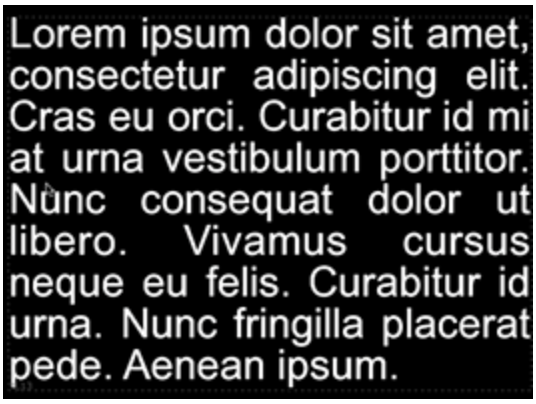
- **center** - align the text from the center of the on-screen text box. This leaves both edges of the text ragged.



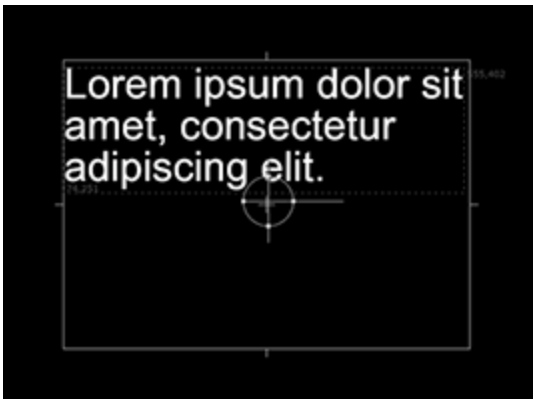
- **right** - align the text along the right edge of the on-screen text box. This leaves the left edge of the text ragged.



- **justify** - align the text both along the left and the right edge of the on-screen text box. This leaves no ragged edges. The justification is done by expanding the spaces between letters. If there are no spaces or the spaces get more than about three times wider than they were, letters are expanded.



2. Use the **yjustify** dropdown to control how you want to align the text vertically:
   - **top** - align the text against the top edge of the on-screen text box.



   - **center** - align the text from the center of the on-screen text box.



   - **bottom** - align the text against the bottom edge of the on-screen text box.



Updating the Font Cache

Once the font cache has been generated, Nuke accepts the content of the cache and no longer generates it from scratch every time it is requested by the font controls. If you add or remove fonts, the cache is not automatically updated and you need to update it manually. There are several ways to do so:

- Navigate to **Project Settings** > **Font** and click **Rescan font paths**.
- Manually remove the **fontmapping.fcache.xml** font cache file from NUKE_TEMP_DIR and restart Nuke.
- Open Nuke's Script Editor and execute **nuke.rescanFontFolders()**

Go to **Help** > **Documentation** > **Python Developer's Guide** for more information on Python commands relating the font cache.

Transforming Text

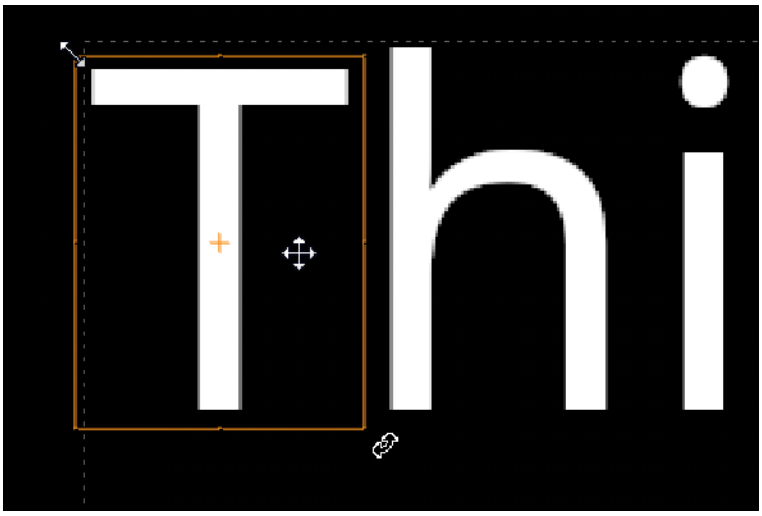
The Text node allows you to transform selections in the Viewer or create groups of selections in the **animation layers** table and transform them together. Both methods support animation using keyframes.

Transforming Selections

Selections in the Viewer are transformed by hand, that is, by manipulating handles drawn on your selections.

1. Enable **transform** in the controls above the Viewer.
2. Select the required character(s) in the Viewer to enable the transform handles.

3. If you hover the cursor over a selection, the cursor changes to show the handle's function depending on position.



4. Translate, rotate, and scale your selections by dragging the required handles.

   If you want to animate your changes, see Animating Transforms for more information.

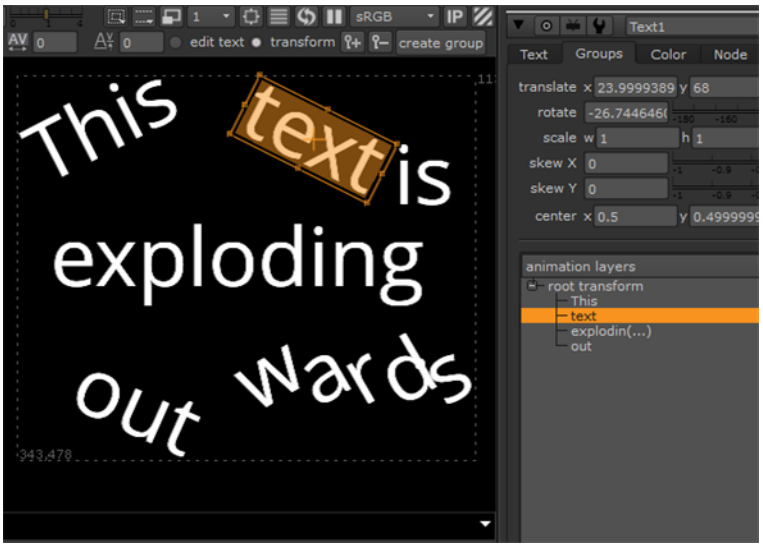
Tip: By default, scaling is done uniformly around the center of your selections. To scale relative to the opposite corner or side instead, hold down **Ctrl**/**Cmd** while dragging the transform handles.

Transforming Groups

The **Group** tab saves group selections in the **animation layer** table allowing you to affect layers separately using the transform controls.

1. Open the properties and click the **Group** tab to display the transform controls and **animation layers** table.
2. Select the required character(s) in the Viewer, or **message** field, and then click the **+** button to create an animation layer, or click **create group** above the Viewer.

   You can create as many animation layers as required, each one is named according to the selections you made in the Viewer.
3. Select an animation layer and use the transform controls on the **Group** tab to transform your selection.



If you want to animate your changes, see Animating Transforms for more information.

4. Click the **-** button to remove an animation layer.

Animating Transforms

The Text node supports animation in a similar way to most other Nuke nodes, in that you can animate individual controls in the properties panel, but the **Group** tab **animation layer** table enables you to visualize your animations more easily.

Animating Selections

1. Set the starting position for the selection(s) using the Viewer handles as described in Transforming Text.
2. Click the  button above the Viewer to add a starting keyframe.
3. Move the playhead to a new frame and transform the selection using the Viewer handles.

   A keyframe is added automatically on the new frame.
4. Repeat the process to produce the required animation.
5. Click  to remove a keyframe at the current frame.

Animating Groups

1. Set the starting position for the groups using the transform controls as described in Transforming Text.

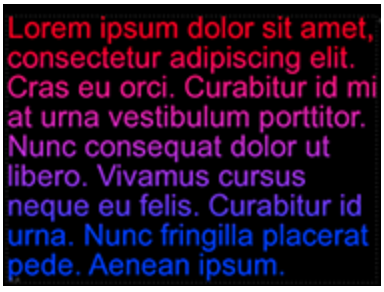
2. Select a group in the animation layers table and then click the button above the Viewer to add a starting keyframe.
3. Move the playhead to a new frame and transform the group using the transform controls or the handles in the Viewer.

   A keyframe is added automatically on the new frame.
4. Repeat the process to produce the required animation.
5. Click to remove a keyframe at the current frame.

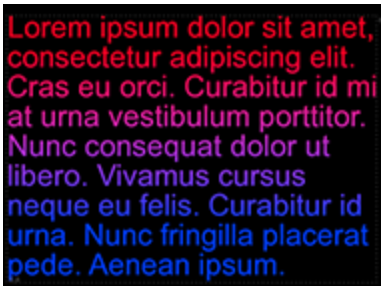
Changing the Text Color

To change the text color:

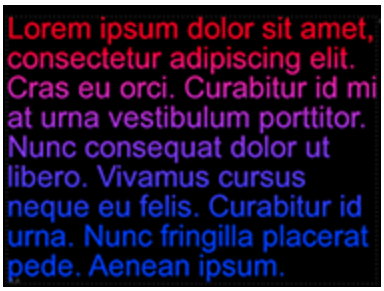
1. In the Text node properties, click the **Color** tab.
2. Adjust the **color** parameter or click on the color picker button to select a color for the text.
3. If you want to create a color gradient across the text, select anything other than **none** from the **ramp**:
   - **linear** - the ramp changes linearly from one color into another.
   - **smooth0** - the ramp color gradually eases into the point 0 end. This means colors in the point 0 end are spread wider than colors in the point 1 end.
   - **smooth1** - the ramp color eases into the point 1 end. This means colors in the point 1 end are spread wider than colors in the point 0 end.
   - **smooth** - the ramp color gradually eases into both ends. This means colors in the point 0 and point 1 ends are spread wider than colors in the center of the ramp.



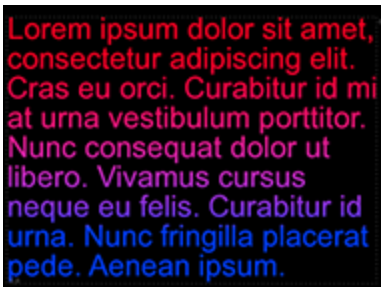
Linear ramp.



smooth ramp.



smooth0 ramp.



smooth1 ramp.

4. Use the **color** control to select a color for the ramp at the point 1 end (by default, the top end). Then, use **color 0** to select the color for the ramp at the point 0 end (by default, the bottom end).
5. To adjust the spread and angle of the ramp, drag the **p0** and **p1** Viewer handles to a new location. You may need to press **O** twice on the Viewer to make these controls appear.

   Alternatively, you can enter new **x** and **y** values for the **point 1** and **point 0** controls in the Text node properties.

Masking Regions of the Viewer

If you want to mask the effect of the Text operation, the easiest way is to use a Roto node connected to the **mask** input:

1. After connecting the Roto node, draw the mask in the Viewer. You can use any channel, but Roto defaults to the **alpha** channel.
2. Select the mask channel from the Text node's **mask** dropdown menu, in this case **rgba.alpha**.
3. To invert the mask as shown, check **invert**.



Standard masking.

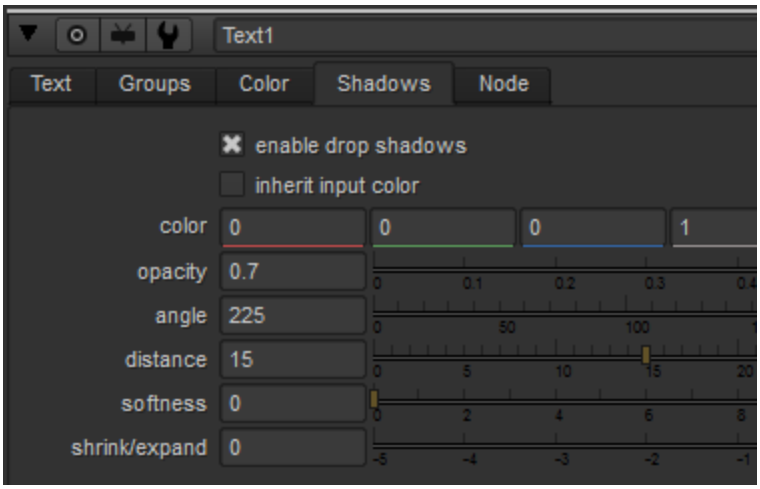


Inverted masking.

Adding Shadows to Text

You can add shadows to your text using the Shadows tab in the Text node's properties.

To add a shadow to your text, simply select the **Shadows** tab and check **enable drop shadows**.



If you want the drop shadow to be the same color as the input image, enable **inherit input color**. Otherwise, use the **color** controls to select a color for the drop shadow.

Use **opacity** to adjust the opacity of the drop shadow relative to the input's alpha channel.



Opacity set to 0.4.



Opacity set to 0.8.

To adjust the direction of the shadow, use the **angle** slider. A value of 0 or 360 equals left direction.



Angle set to 0 (or 360).



Angle set to 225.

Use **distance** to set the distance of the shadow from the input content.



Distance set to 15.



Distance set to 45.

If you want to blur the shadow, increase the **softness** value.



Softness set to 0.



Softness set to 20.

If necessary, set the **shrink/expand** slider to a negative value to erode the shadow or a positive value to dilate it.



Shrink/expand set to -5.



Shrink/expand set to 5.

Adding a Drop Shadow

The DropShadow gizmo allows you to create a drop shadow for any input image that has an alpha channel with values greater than 0. To add a drop shadow an image:

1. Select **Filter** > **DropShadow** to insert a DropShadow gizmo after your image Read node.
2. Connect your background image to the DropShadow gizmo's **bg** input. Your image and drop shadow are merged over the background image.
3. In the DropShadow gizmo properties, check **enable drop shadow** to enable the drop shadow effect.



with drop shadow disabled



with drop shadow enabled

4. If you want the drop shadow to be the same color as the input image, enable **inherit input color**. Otherwise, use the **color** controls to select a color for the drop shadow.



with black selected



with the inherited input color

5. Use **opacity** to adjust the opacity of the drop shadow relative to the input's alpha channel.



Opacity set to 0.4



Opacity set to 0.8

6. To adjust the direction of the shadow, use the **angle** slider. A value of 0 or 360 equals left direction.



Angle set to 0 (or 360)



Angle set to 225

7. Use **distance** to set the distance of the shadow from the input content.



Distance set to 5



Distance set to 15

8. If you want to blur the shadow, increase the **softness** value.



Softness set to 0



Softness set to 10

9. If necessary, set the **shrink/expand** slider to a negative value to erode the shadow or a positive value to dilate it.



Shrink/expand
set to 0



Shrink/expand
set to 3.5

10. Use the **input [operation] bg** dropdown menu to select how you want to combine the **input** and **bg** inputs.

Analyzing and Matching Clips

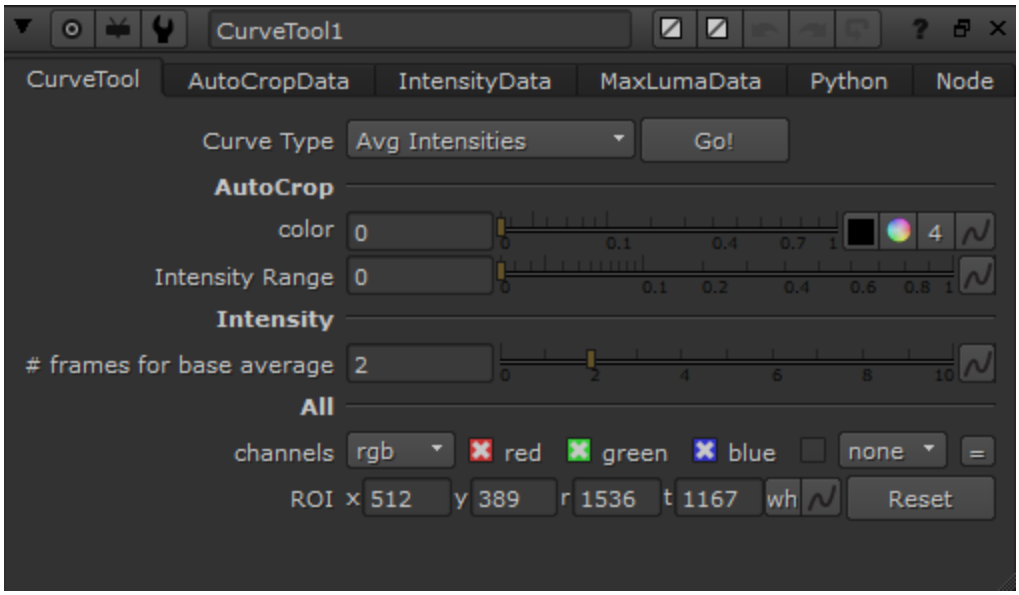
This chapter concentrates on the CurveTool node. The node analyzes an aspect of a frame sequence and creates an animation curve based on the analysis. You can then use the curve data to drive effects elsewhere.

For instance, you can add matching flicker to a CG render.

Introduction

You can use the CurveTool node to analyze four different aspects of your frame sequence, depending on which curve type you select in the node controls:

- **AutoCrop** finds black regions (or any color you pick) around the edges of the frame sequence and tracks their size and position over time. This is useful for running a Crop node to remove unnecessary outer pixels and speed up the calculation. For more information, see Cropping Black Edges .
- **Avg Intensities** is useful for obtaining the average pixel values in a frame sequence and then matching that intensity elsewhere. It takes the first value in the frame range and the next value selected, adds them together and divides by two, returning the average between the two. You might want to use it to match the background plate's fire flicker in the smoke in the foreground plate, for example. For more information, see Analyzing the Intensity of a Frame Sequence and Removing Flicker .
- **Exposure Difference** analyzes the exposure changes in the frame sequence. It takes the first value in the frame range and the next value selected, and returns the difference between the two. You can use the results to match the same exposure elsewhere. For more information, see Analyzing Exposure Differences.
- **Max Luma Pixel** tracks the brightest and dimmest pixels in the frame sequence. This can be useful in the following case, for example. Let's say you have a night-time sequence depicting a person moving inside a dark house holding a flashlight, and want to add lens flare on the moving flashlight. Knowing where the brightest pixel is located over time allows you to match-move the lens flare and position it correctly without having to manually animate it. For more information, see Tracking the Brightest and Darkest Pixels .



Tip: If you are familiar with Shake, you may have used the PixelAnalyzer node. The CurveTool node is the Nuke equivalent of PixelAnalyzer.

Cropping Black Edges

You can crop black edges (or any color you choose) from your footage to eliminate unnecessary computation:

1. Select **Image** > **CurveTool** to insert a CurveTool node after the image sequence you want to analyze.
2. Make sure a Viewer is connected to the CurveTool node.
3. In the CurveTool controls, select **AutoCrop** from the **Curve Type** dropdown menu.
4. Using the **color** parameters, select the color you want to track.
5. To control how far the color can deviate from the selected color and still be cropped off, use the **Intensity Range** slider.
6. From the **channels** dropdown menu and checkboxes, select the channels you want to analyze.
7. If you want to analyze an area in the frames rather than entire frames, define a region of interest either by dragging the edges of the frames to a new position in the Viewer, or by defining the area using parameters labeled **ROI**.
8. Click **Go!** to analyze the frames. This opens the **Frames to Execute** dialog.
9. In the dialog, define the frames to analyze. Enter the first frame, followed by a comma and the last frame. Click **OK**. Nuke starts analyzing the frame sequence.

10. You'll find the results of the analysis on the **AutoCropData** tab where the parameter values have turned blue to indicate they are animated over time. To see the animation curve, right-click on a parameter field and select **Curve editor**.



Once Nuke has created the animation curve, you can copy the animation or any of its values into a Crop node, for example, to match the analyzed crop area there. **Ctrl**/**Cmd**+click on the animation button and drag and drop it to another parameter to create an expression linking the two.

Analyzing the Intensity of a Frame Sequence

You can analyze your footage to find out the average intensity values in it:

1. Select **Image** > **CurveTool** to add a CurveTool node in an appropriate place after the image sequence you want to analyze and match.
2. Connect a Viewer to the CurveTool.
3. In the node's controls, select **Avg Intensities** from the **Curve Type** dropdown menu.
4. Select the channels you want to analyze from the **channels** dropdown menu and checkboxes.
5. By default, the region of interest that is to be analyzed covers the entire frame. If you want to analyze a smaller area, resize and reposition the region of interest in the Viewer by dragging its edges to a new position. You can also resize the region of interest using the **ROI** parameters in the properties panel.
6. In the **# frames for base average field**, enter the range of frames that each frame being analyzed is compared against. The frames are compared onwards from each frame analyzed. Thus, a value of 1 would compare each frame to the frame following it, whereas a value of 5 would compare each frame to the following 5 frames.

   The higher frame range you use, the more accurate and time-consuming the calculation becomes. However, a high frame range is not always needed. For analyzing and matching fire flicker, you'd probably want to go frame by frame, whereas removing flicker would require a wider frame range to ensure a good average is obtained as the result.
7. To analyze the sequence, click **Go!**. This opens the **Frames to execute** dialog.
8. In the dialog, specify the frame range you want to analyze and match. Enter the first frame, followed by a comma and the last frame. Click **OK**. Nuke now analyzes the frame sequence.

9. Move to the **IntensityData** tab where you'll find the results of the analysis. You'll notice that the parameter input fields have turned blue. This indicates that they are animated. To see the animation curve, right-click on the values and select **Curve editor**.

Once Nuke has created the animation curve, you can copy the animation or any of its values into a color correction node, for example, to match the analyzed intensity there. **Ctrl**/**Cmd**+click on the animation button and drag and drop it to another parameter to create an expression linking the two.

Removing Flicker

You can also use the CurveTool to stabilize flickering in your footage. To do this:

1. Connect a CurveTool node to your footage.
2. In the **Curve Type** dropdown, select **Avg Intensities**.
3. If necessary, select the channels you want analyzed in the channels controls, and adjust the region of interest (ROI) box in the Viewer to cover your analysis area.
4. When you're ready, click **Go!** and specify a frame range in the **Frames to Execute** dialog that opens. CurveTool analyzes your footage, and the resulting values appear on the **IntensityData** tab.
5. Create a Grade node (**Color > Grade**) and connect it to your footage.
6. From the CurveTool's **IntensityData** tab, **Ctrl/Cmd**+drag the analysis result to the **multiply** field in the Grade node controls.
7. Right-click on the multiply field, and select **Edit expression**, or press the equals (**=**) button with the field selected.
8. In the dialog that appears, add **1/** in front of the **Expression** field entry. This inverts the brightness values detected by the CurveTool node, and enables the Grade tool to stabilize brightness changes causing flickering in your footage.

Analyzing Exposure Differences

You can analyze the differences in exposure in your frame sequence:

1. Select **Image** > **CurveTool** to add a CurveTool node after the image sequence you want to analyze.
2. Add a Viewer after the CurveTool.
3. Under **Curve Type**, select **Exposure Difference**.
4. From the **channels** dropdown menu and checkboxes, select the channels you want to analyze.

5. If you want to analyze an area in the frame rather than the entire frame, define a region of interest either by dragging the edges of the frame box to a new position in the Viewer, or by defining the area using parameters labeled **ROI**.
6. To analyze the sequence, click **Go!**. The **Frames to Execute** dialog opens.
7. Specify the frame range you want to analyze. Enter the first frame, followed by a comma and the last frame. Click **OK**. Nuke now performs the analysis.
8. You can find the results of the analysis on the **IntensityData** tab where the parameter input fields have turned blue to indicate they are animated. To see the animation curve, right-click on one of the input fields and select **Curve editor...**.

Once Nuke has created the animation curve, you can copy the animation or any of its values into a color correction node, for example, to match the analyzed exposure there. **Ctrl**/**Cmd**+click on the animation button and drag and drop it to another parameter to create an expression linking the two.

Tracking the Brightest and Darkest Pixels

You can track the brightest and the darkest pixels in your frame sequence:

1. Select **Image** > **CurveTool** to add a CurveTool node after the image sequence you want to analyze.
2. Connect a Viewer to the CurveTool.
3. From the **Curve Type** dropdown menu, select **Max Luma Pixel**.
4. Click **Go!** to analyze the frame sequence. This opens the **Frames to Execute** dialog.
5. Define the frame range you want to analyze. Enter the first frame, followed by a comma and the last frame. Then, click **OK**. Nuke analyzes the frame sequence, tracking both the position and the values of the brightest and darkest pixels.
6. You can find the results of the analysis on the **MaxLumaData** tab. You'll notice that the input fields have turned blue to indicate that they are animated over time. To see the animation curve, right-click on an input field and select **Curve editor**.

Once Nuke has created the animation curve, you can copy the animation or any of its values into another node to match that node's effect with the brightest or darkest pixels in the frame sequence. **Ctrl**/**Cmd**+click on the animation button and drag and drop it to another parameter to create an expression linking the two.

3D Compositing with the New 3D System

The **New 3D System** in Nuke enables you to efficiently work with modern 3D scenes at scale. Our new architecture brings Nuke's 3D System in line with modern standards and USD-based workflows, with a focus on intuitiveness and efficiency for compositors.

The Story of the New 3D System

The Why

Why do we have two systems?

The **Classic 3D System** works well for many smaller setups. However, when there is a need to bring in denser meshes, larger scale scenes, and play them back in real time while adding projections or animated textures, these larger assets need to work in a more performant manner. The **New 3D System** allows you to bring in the assets you need no matter the size, and allows comp artists to leverage their skills at any part of the production that needs it.



Note: To ensure you don't lose access to any workflows you're used to, the Classic 3D system nodes are still available in Nuke.



Note: For those who are looking to transition from the **Classic 3D System** to the **New 3D System**.This helpful guide provides a comparison of nodes, matching them by equivalent function and purpose - 3D System Nodes: From Classic to New.

Why USD?

The New 3D System's USD architecture embraces open standards and allows for workflows that not only provide better performance and scalability, but also allow for greater collaboration between artists. Having a non destructive workflow that means you can work more seamlessly with the lighting artists, FX TDs, layout

department, will mean less time waiting for asset handovers and enables comp artists to see exactly what artists in other departments see.

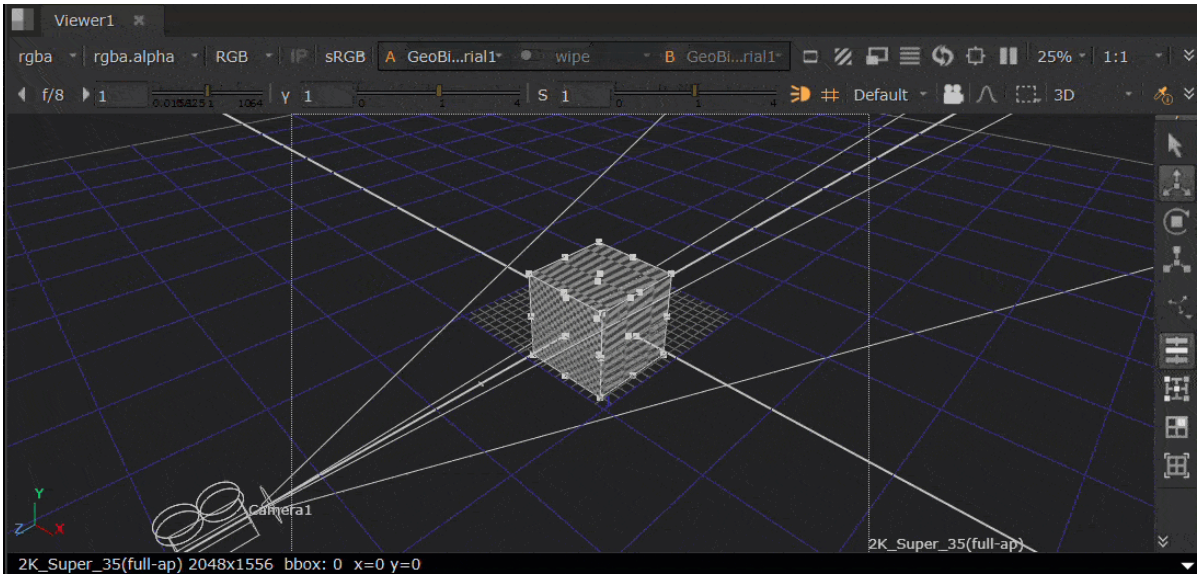
The Philosophy

The driving philosophy has always been that this is a 3D system for comp artists. This means that the New 3D system isn't intended as a replacement for other DCCs, and even though it's based on a USD architecture, it has been designed for compositing workflows first. Whether it's used for projections for cleanup work, setting up matte paintings, additional particle simulations to complement a hero effect from FX, or even just adding small background details to help guide the eye, to better compose the image and tell the story, Nuke's New 3D System creates an intuitive way to work for compositors, adding greater control over your elements and easier management in the scene graph.

For any artist put off by the idea of it being USD, this system should feel like 3D in Nuke always has, and that USD is just the underpinnings of greater performance, new collaborative workflows and better rendering. We have worked hard to make this the most comper-friendly approach to USD, so that you can focus on doing what you've always done, using the same workflows you've always known.

Note: For artists newer to USD, learn more about the terminology at USD Basics: Stage, Prims, and Layers.

Note: See more about how Nuke approaches USD at USD Concepts in the New 3D System.



New 3D System Overview

Learning about the New 3D System

This User Guide explains how to set up a scene in Nuke's **New 3D System**, such as how to add geometry and cameras in the 3D workspace. You'll also see how to texture objects, transform objects and cameras, and render out scenes for use in other areas of your script.

Here are some good places to get started:

New 3D System User Guide

- USD Basics: Stage, Prims, and Layers
- Importing into Your Scene
- Creating Prims in Your Scene
- New 3D Scene Basics
- New 3D Scenes in the Scene Graph
- Using Paths and Masks
- Viewing Your Scene
- Working with Cameras in the New 3D System
- Projections in the New 3D System
- Transforming and Modifying
- Assigning Materials and Shaders
- Light Your Scene
- Rendering in the New 3D System
- Export From Your Scene
- Debugging Your Scene



Note: For those who are newer to using USD, it may be helpful to start with USD Basics: Stage, Prims, and Layers.

New 3D System Nodes

There are two sets of 3D nodes in Nuke - **New 3D System** nodes and **Classic 3D System** nodes.

New 3D System nodes are colored depending on the node type, and Classic 3D System nodes are always green.

Note: For those who are looking to transition from the **Classic 3D System** to the **New 3D System**. See this helpful guide which provides a comparison of nodes, matching them by equivalent function and purpose - 3D System Nodes: From Classic to New.

Thanks to Nuke's new tools, you can pass data easily across your node graph and use the full toolset of nodes and workflows.

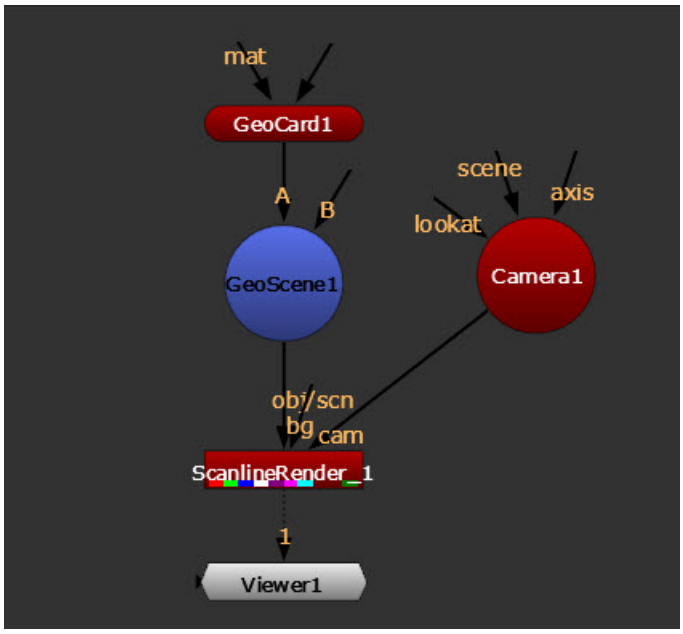
The New 3D System nodes are found in right click node menu under **3D** > **3D**, whereas the Classic 3D System Nodes are under **3D** > **3D Classic**.

Tip: Most **New 3D System** nodes start with the prefix 'Geo'.

Note: For a list of **New 3D System** nodes with links to documentation for each node, go to New 3D System Nodes (Beta) in the Reference Guide.

Note: Some nodes remain in Beta and are marked with a Beta tag in the menu and node Properties. While we welcome feedback on these nodes, they are still under development and are not production ready and should not be used in production.

Tip: You can mix 2D and 3D objects together in the node tree to create scenes. For example, you can texture a 3D object with a 2D clip or generator such as the Checkerboard node, or take the rendered output from a 3D scene and use it as a 2D background. Learn more about constructing a scene at New 3D Scene Basics.



A New 3D System basic scene

USD Basics: Stage, Prims, and Layers

The New 3D System tools have been designed for a painless transition from the Classic system by using familiar approaches where possible. However, as a USD beginner, you'll encounter a number of new concepts and terminology that are useful to grasp to make full use of the enhanced workflows.

The USD Stage

The first USD term to understand is a **Stage**. A stage can basically be thought of as a scene. It holds the entire **Scene Graph** and associated data. When working in the New 3D System, every node network is essentially a stage and if you have two node networks with their own 3D data each network will have their own stage. If you then merge those node networks together you merge them to create one stage.

USD Prims

A **prim** (short for primitive) is the fundamental building block in a 3D scene. It represents an individual object or entity within the scene hierarchy. A piece of geometry is a prim, a camera is a prim, really anything in the scene graph is likely a prim.

In the Scene Graph you will also see columns for Type, Kind and Purpose. These are **Primitive Attributes**.



See more about Primitive Types and Primitive Attributes or learn how to create prims in the New 3D System at Creating Prims in Your Scene.

USD Layers

Layers are a crucial concept in USD as they facilitate non-destructive editing and collaboration. A layer is a self-contained file that stores a part of the scene, supporting edits and enabling the creation of alternate versions without impacting the original source.

A single **.usd** file is a layer. A layer can represent a "piece" of a scene such as a prop, a character, part of a rig, a lighting setup, and so on. It can also represent "defaults" that you can reference in, to create a baseline, for example a layer containing settings for a shot sequence, which you reference in the files representing each shot in the sequence.

Sublayering overlays the contents of the imported file (or layer) over the existing content. So essentially you are combining layers together, some on top, or some underneath to build up your stage.

When combining layers, you might sometimes combine layers which have the same prim in both layers, but with different information, say a Camera with different position or animation data. To decide which layer's data we pull from, USD has a concept known as **opinion strength**.

As you might expect, the layer with the strongest **opinion strength** will determine the data that is used for this prim that exists in both layers.

This is really powerful and at the heart of why USD can be so collaborative. It allows you to non-destructively make changes, as your changes can live on their own sublayer, while allowing other artists' work to live on a different layer and when you combine these layers together to build your stage you get the result based on the strongest opinion with no data loss.



Learn more about opinion strength in Nuke at USD Concepts in the New 3D System.

USD has two different approaches to "**importing**" the contents of another USD file: **Sub-layering** overlays the tree of the imported file over the current tree, while **Referencing** attaches the contents of the imported tree as a branch on the existing tree.

You can think of sublayering as composing different versions of the whole scene (for example, overlaying the lighting department's version of the scene with final lighting over the layout department's version of the

scene with scratch lighting), and referencing as adding a part to the scene (for example, importing props to begin layout).



Learn more about importing at Importing into Your Scene.

When importing you might also notice the option to load or not load **payloads**. In USD, a payload is essentially a reference you can choose not to load, making it easier to just bring in the parts of the scene you care most about.



Learn more about payloads at Payloads and Deferred Loading.

By separating the scene in this way, each layer can be worked on independently and non-destructively.

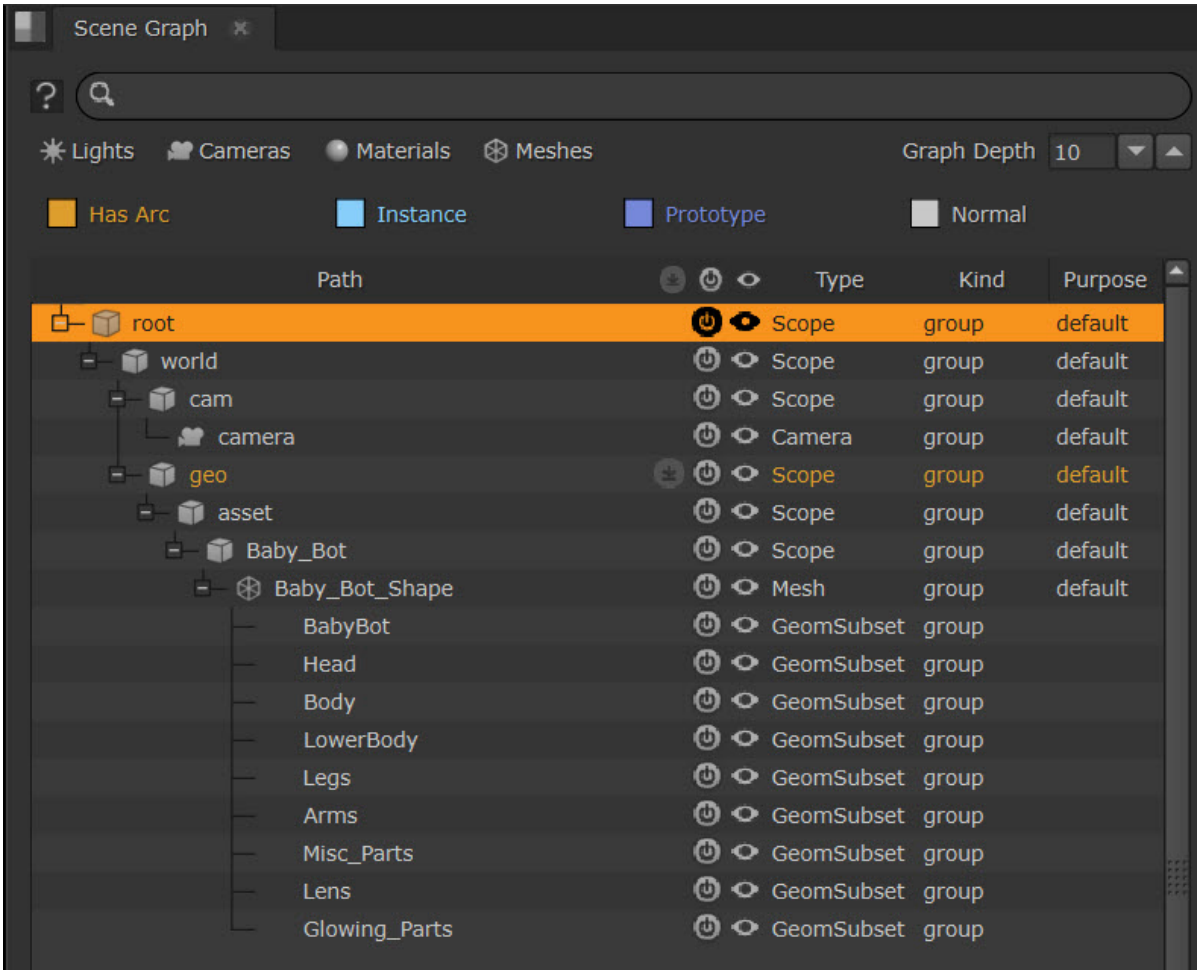
USD in Nuke

In Nuke, each of the new 3D nodes creates a USD layer. Layers can also be imported as a **.usd** file. These layers can be composed together to form a stage.



Note: The order of assembling layers is important due to the effect of layer overrides in the stack. A different order may apply different overrides. Keep this in mind when merging the outputs of the 3D nodes.

The image below shows the layers of a simple scene as shown in the Scene Graph, with multiple layers and sublayers:



Learn more about the Scene Graph at New 3D Scenes in the Scene Graph.

Note: For a glossary of USD terminology, see USD Terms and Concepts from Pixar.

Primitive Types and Primitive Attributes

Primitive Types

There are several different types of prims in the USD framework: **Xforms**, **Scopes**, **Meshes**, **Materials**, **Shaders**, **Cameras**, and **Lights**. In Nuke, these types come with a specific default behavior, but essentially these are just containers which you can set properties on.

Type	Description	Example Nuke Nodes	Read more
Xform	An Xform prim is a container that you can put transformation data on.	GeoXformPrim (USD native)	Creating Scope and Xform Prims
Scope	A Scope prim is a container that you can't put transformation data on.	GeoScopePrim (USD native)	Creating Scope and Xform Prims
Meshes	A Mesh prim is a geometric surface.	GeoCube, GeoCylinder, GeoSphere, GeoCard (USD native)	Creating Geometry Prims
Materials	A Material prim defines the appearance of a surface.	No USD native material prim nodes in Nuke - but materials can be assigned via GeoBindMaterial	Assigning Materials and Shaders
Shaders	A Shader prim is used for defining shading networks.	No USD native shader nodes in Nuke - but shaders can be assigned via GeoBindMaterial The PreviewSurface node is a representation of the "USDPreviewSurface" shader which is meant to model a "modern" physically based surface with both specular and metalness workflows.	Shading Geometry
Cameras	A Camera prim is a viewpoint onto the scene.	Camera (Not USD native but can be converted to USD) GeoEditCamera, GeoCamera	Working with Cameras in the New 3D System

		(USD native)	
Lights	A Light prim represents a light source in the scene.	GeoDomeLight, GeoDiskLight, GeoDistantLight, GeoSphereLight (USD native) PointLight, SpotLight, DirectLight, EnvironmentLight (Not USD native but can be converted to USD)	Light Your Scene

Primitive Attributes

Primitive Attributes give you access to edit USD attributes in some of the New 3D System nodes in Nuke. You don't necessarily need to use these, but if you want some extra control, these knobs can be edited. On nodes such as GeoCard, you can find and expand the **Primitive Attributes** section in the **Properties** panel.

Kind	**Kind** lets you determine the selection category for this card, this can be used to refine your selection based on kinds in the 3D viewer selection controls. • Assembly: an important group model, often a published asset or reference to a published asset • Group: models that simply group other models. • Component: a "leaf model" that can contain no other models • Model: base class for all model kinds. "model" is considered an abstract type and should not be assigned as any prim's kind. • Subcomponent: an identified, important "sub part" of a component model.
Purpose	**Purpose** sets how your prims are displayed in the viewer allowing you to define the viewer render quality. • Default: Assigns the purpose based on the Layer settings, otherwise defaults to Render • Render: Assign to prims intended for final high-quality rendering. • Proxy: Assign to prims intended for lightweight proxy rendering. Guide: Assign to prims intended as markings, lines, arrows or interaction guides.
Visibility	**Visibility** sets if your geometry and any descendants are displayed as hidden in the Viewer. • Inherited: When enabled the visibility of the selected prim is the same as the prim's parent.

	• Invisible: the selected prim and its descendants display as hidden in the viewport.
Mesh Attributes	Under **Mesh Attributes** you also have the following controls: • Subdivision: sets how meshes are rendered in the viewer based on different subdivision models • Boundary: specifies where the boundary conditions needed for interpolation is defined • Double Sided: lets you shade both sides of your geo
Display	**Display** concerns itself with how your card is displayed in your 3D viewer. • Display Color allows you to colour your geometry. • Display Opacity sets how transparent it is, but these are just for viewer display not rendering.

Tip: Full descriptions of the controls can be found in each node's Reference Guide page - see New 3D System Nodes (Beta).

In the Scene Graph

The Prim **Type**, **Kind** and **Purpose** that has been set also appears in the Scene Graph for easy reference, but they cannot be edited from the Scene Graph.

See more about the Scene Graph at New 3D Scenes in the Scene Graph.

USD Concepts in the New 3D System

Nuke's Approach to USD

Nuke's **New 3D System** is designed for all artists and meant to be intuitive for artists who are not familiar with USD. In the New 3D System, you have the choice of a range of nodes, some of which allow you to create objects (prims) natively in USD in Nuke and some that import or convert data, which allow you to work

flexibly with a range of assets. Nuke intends to use USD in a way that is useful for compositors, such as by having additional controls, like the strength ordering options as outlined below.

- Having some basic USD knowledge may help your understanding of the New 3D System. It will be helpful to understand terminology like prims, layers and stages and these terms will be used throughout this User Guide. See USD Basics: Stage, Prims, and Layers to learn about these.
- When compositing in Nuke, you may want to use USD to bring in assets, cameras, and lights (prims) and then manipulate these in your scene. To learn more about prims, see Primitive Types and Primitive Attributes.

Nuke's Controls

In USD typically, you author by setting the value on an object's (prim's) attribute and this is considered 'authoring an opinion'. The USD framework works by arranging layers and objects (prims) in a hierarchy. Therefore some of the layers, based on USD composition arc rules and strength ordering, have a higher opinion than others and may override others. However, the way Nuke approaches this is to let artists have full control over this so they can change the rules to support an efficient workflow for them.

You'll see that certain nodes have additional settings, such as GeoBindMaterial node's **Binding Strength** parameter, which helps make sure your materials are being assigned regardless of other layers which may be present. See more at Using GeoBindMaterial to Assign Materials.

We have introduced **Authoring Modes** to give greater control of specific knobs.

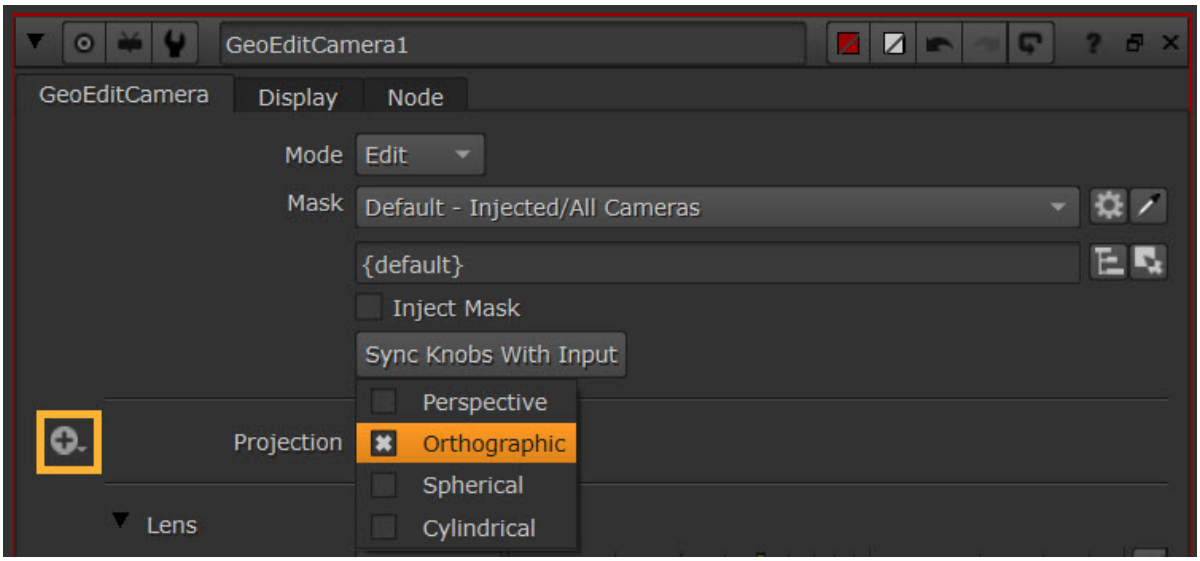
Authoring Modes

Across some of the New 3D System nodes, there are knobs to the left hand side  that let you select from two options: **Set Value** and **No Change**. Directly interacting with the knob automatically updates this to **Set Value** mode, making changes faster. You can then easily revert back to **No Change** if you want to restore the USD default value.

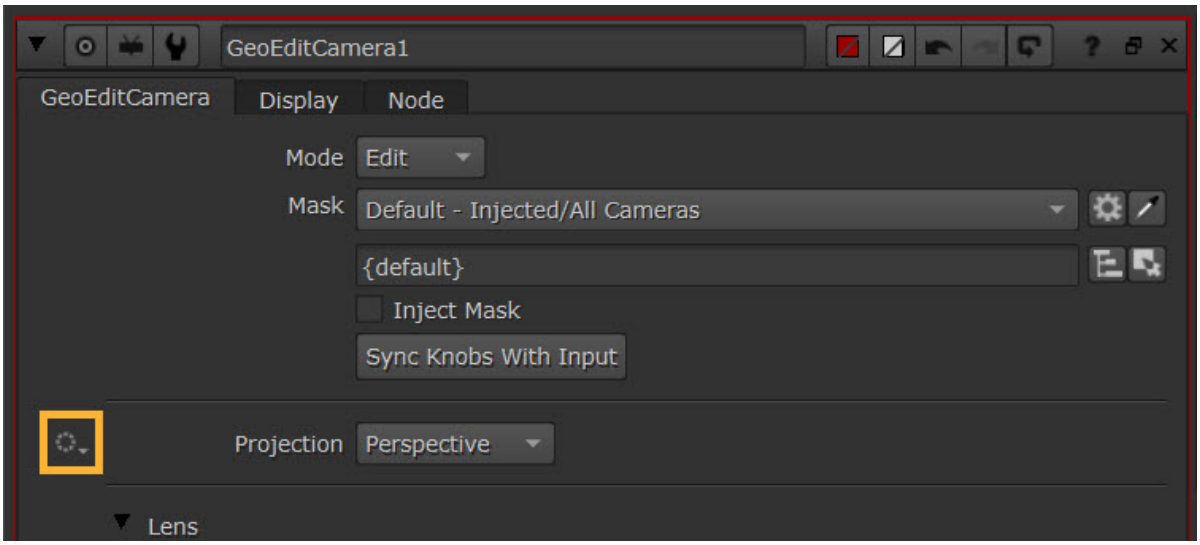
Learn more about Authoring Modes

Because the value you set is an opinion, there may be stronger opinions set elsewhere in your stage that modify this attribute value when the final stage is composed. These added authoring mode options allow you greater control over this:

Set Value - Directly interacting with the knob has it automatically update to 'Set Value' mode. Use this option if you want to set the knob's current value, regardless of whether it's the knob's default or not. This will be the USD prim attribute, or 'authored'. Depending on whether the knob's value matches the USD prim attribute's default schema value will determine if the attribute is actually 'overridden'.



No Change - use this option if you do not want to author any value on the USD prim attribute, or if you want to reset the knob's value back to the knob's default (not the USD schema default).



Note: On the GeoEditCamera and GeoEditLight nodes you'll also find the **Sync Knobs With Input** button. When clicked this button gets the values from the prim at the current UI time and sets the knobs to these values.

USD Configuration

USD Versions

Refer to Third-Party Libraries and Fonts to find out about which USD version is supported by Nuke. To change or modify the USD version within Nuke, it can be accessed in its own subfolder, which has the same structure as a USD build.

Tip: You can set an environment variable to make swapping between different USD versions with Nuke easier.

For setting up your own Custom USD version with Nuke, please reference the **Readme.md** in the following Nuke directory:

/source/FnUsdShim/Readme.md which is found next to the Nuke executable.

USD Python Bindings

Nuke exposes Pixar's USD Python API allowing you to manipulate USD data directly through Python, giving you greater control over how you work with USD data in Nuke and more flexibility when building your USD-based pipeline tools. The API can be used in any Python script and in Nuke's Script Editor directly by calling:

```
import pxr
```

For more information on how to use Pixar's USD Python bindings, please refer to their official USD documentation: https://openusd.org/release/index.html

3D System Nodes: From Classic to New

This guide is a quick reference for users transitioning from the **Classic 3D System** to the **New 3D System**. It provides a comparison of nodes, matching them by equivalent function and purpose.

To learn more about each node's controls, click the node name in the table to access its Reference Guide page. Additionally, this table includes links to further guides containing basic implementation examples which may help in learning how to integrate the nodes into your workflows.

Geometry | Lights | Shader | Modify | Cameras & Tracking | Utility | Rendering

Classic 3D System Node	Equivalent New 3D System Node	Notes	New 3D System Guides
Geometry			
Card	GeoCard		Creating Geometry Prims
Cube	GeoCube		Creating Geometry Prims
Cylinder	GeoCylinder		Creating Geometry Prims
Sphere	GeoSphere		Creating Geometry Prims
DepthToPoints	GeoPoints		Creating Points Clouds
ModelBuilder	No equivalent node		
PointCloudGenerator	GeoGeneratePoints		
PositionToPoints	GeoPoints		Creating Points Clouds
PoissonMesh	GeoPointsToMesh		
ReadGeo	GeoImport, GeoReference	To learn more about the difference between GeoImport and GeoReference, see Importing into Your Scene.	Importing with GeoImport Referencing with GeoReference
WriteGeo	GeoExport		Export From Your Scene
Lights			
Light	GeoEditLight	GeoEditLight allows you to	Light Your Scene

		create or edit USD lights. To work with nodes more similar to the Classic System, see the nodes in the rows below.	
Point	PointLight	PointLight's USD equivalent is SphereLight. In the New 3D System, you can also use the native USD node for this, GeoSphereLight.	Using Nuke Lights Using USD Lights
Direct	DirectLight	DirectLight's USD equivalent is DistantLight. In the New 3D System, you can also use the native USD node for this, GeoDistantLight.	Using Nuke Lights Using USD Lights
Spot	SpotLight	SpotLight's USD equivalent is DiskLight. In the New 3D System, you can also use the native USD node for this, GeoDiskLight.	Using Nuke Lights Using USD Lights
Environment	EnvironmentLight	EnvironmentLight's USD equivalent is DomeLight. In the New 3D System, you can also use the native USD node for this, GeoDomeLight.	Using Nuke Lights Using USD Lights
Relight	No equivalent node		
Shader			
AmbientOcclusion	ScanlineRender	In ScanlineRender's **Advanced** tab, under **Integrator** you can switch into **Occlusion** mode. Make sure you are using the correct ScanlineRender node. The New 3D System's	

		ScanlineRender node has the same name as the Classic node, but works differently. In the node menu select **3D** > **3D**, not **3D** > **3D Classic**. The New node will appear in your Node Graph as **ScanlineRender2**.	
ApplyMaterial	GeoBindMaterial		Using GeoBindMaterial to Assign Materials
BasicMaterial	BasicSurface		BasicSurface: Shader Usage
FillMat	FillShader		FillShader: Cut Out
MergeMat	MergeLayerShader		MergeLayerShader: Merging Shaders
BlendMat	No equivalent node		
Project3D	Project3DShader		Project3DShader: Projecting Textures
Diffuse	BasicSurface, MtlXStandardSurface		BasicSurface: Shader Usage MtlXStandardSurface: MaterialX Shader
Emission	BasicSurface, MtlXStandardSurface		BasicSurface: Shader Usage MtlXStandardSurface: MaterialX Shader
Phong	No equivalent node	There is no specific Phong equivalent node, but similar effects can be achieved using BasicSurface and MtlXStandardSurface.	BasicSurface: Shader Usage MtlXStandardSurface: MaterialX Shader

Specular	ReflectiveSurface		ReflectiveSurface: Generating Reflections
Displacement	No equivalent node		
UVTile	No equivalent node	There is no specific UVTile equivalent node, but there are UV options on ScanlineRender.	Rendering in the New 3D System
Wireframe	WireframeShader		WireframeShader: Applying Wireframe Overlay
Transmission	ReflectiveSurface, MtlXStandardSurface		ReflectiveSurface: Generating Reflections MtlXStandardSurface: MaterialX Shader
Modify			
TransformGeo	GeoTransform		Transformations in 3D Scenes
MergeGeo	GeoMerge		Merging 3D Data
CrosstalkGeo	No equivalent node		
DisplaceGeo	GeoDisplace		Manipulating Geometry Data
EditGeo	No equivalent node		
GeoSelect	GeoSelection		Using Geometry Data
LookupGeo	No equivalent node		
LogGeo	No equivalent node		
Normals	GeoNormals		Using Geometry Data
ProceduralNoise	GeoNoise		Manipulating Geometry Data

RadialDistort	GeoRadialWarp		Manipulating Geometry Data
Trilinear	GeoTrilinearWarp		Manipulating Geometry Data
UVProject	GeoProjectUV		Projections in the New 3D System
Cameras & Tracking			
Camera	Camera, GeoCamera, GeoEditCamera	How you are using USD will affect your workflow. Read the Working with Cameras in the New 3D System guide to understand the difference between these nodes. The New 3D System's Camera node has the same name as the Classic node, but works differently. In the node menu make sure to select **3D** > **3D**, not **3D** > **3D Classic**.	Working with Cameras in the New 3D System
CameraTracker	CameraTracker	This node is shared between the Classic and New 3D Systems, but make sure to uncheck the **Classic 3D** checkbox in the Properties panel for use with the New 3D System.	
DepthGenerator	DepthGenerator	This node is shared between the Classic and New 3D Systems.	
DepthToPosition	DepthToPosition		
Utility			
Axis	Axis	The New 3D System's Axis	Using Axis Node in

		node has the same name as the Classic node, but works differently. In the node menu make sure to select **3D** > **3D**, not **3D** > **3D Classic**.	Your Scene
Scene	GeoScene		New 3D Scene Basics
Rendering			
ScanlineRender	ScanlineRender (for most use cases), SplatRender (for Gaussian Splats)	Make sure you are using the correct ScanlineRender node. The New 3D System's ScanlineRender node has the same name as the Classic node, but works differently. In the node menu make sure to select **3D** > **3D**, not **3D** > **3D Classic**. The New node will appear in your Node Graph as **ScanlineRender2**.	Rendering in the New 3D System
RayRender	ScanlineRender	The ScanlineRender node now has ray-tracing. Make sure you are using the correct ScanlineRender node. The New 3D System's ScanlineRender node has the same name as the Classic node, but works differently. In the node menu select **3D** > **3D**, not **3D** > **3D Classic**. The New node will appear in your Node Graph as **ScanlineRender2**.	Ray-tracing with ScanlineRender
(Renderman) PrmanRender	No equivalent node		
(Renderman)	ReflectiveSurface +	The Reflection controls are no	Ray-tracing with

Reflection	ScanlineRender	longer Renderman specific.	ScanlineRender
(Renderman) Refraction	ReflectiveSurface + ScanlineRender	The Refraction controls are no longer Renderman specific.	Ray-tracing with ScanlineRender
(Renderman) ModifyRIB	No equivalent node		

Importing into Your Scene

GeoImport and GeoReference allow you to import into your scene in Nuke.

This can be especially helpful when working with USD. Using USD files in your pipeline allows you to organize large scenes more effectively and edit objects non-destructively, so you never lose upstream work. USD is also a versatile format that most VFX software packages support, making it easy to work between stages in production, from texture painting, through look development and into compositing.

Nuke imports the contents of your USD file using the GeoImport node. You can control the payloads from the initial import to avoid loading everything in one go. Or you can reference part of the file using the GeoReference node if you know the object's path you want to import.

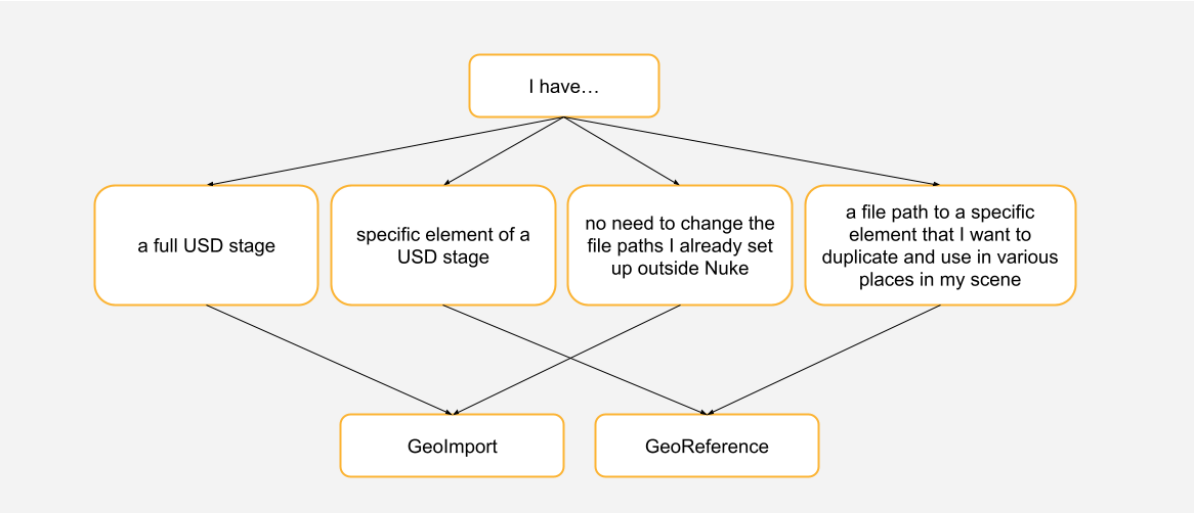


Note: Alternatively, you can create geometry such as cards, cubes, and spheres directly in the New 3D System, avoiding the need to import. To read about how to do this, see Creating Prims in Your Scene.

Importing with GeoImport vs. GeoReference: Which Node to Use?

GeoImport brings an entire scene into Nuke, including any lights, cameras, and materials in the scene. GeoImport does not allow you to change the path of the object in the scene graph so be aware that conflicts can easily occur when importing the same file more than once - to avoid these entirely, use the GeoReference node to create an instance, or light-weight copy, of the required objects.

To summarise:



Learn more:

To see more about how to use GeoImport, see Importing with GeoImport.

To see more about how to use GeoReference, see Referencing with GeoReference.

Importing with GeoImport

GeoImport brings an entire scene into Nuke, including any lights, cameras, and materials in the scene. It supports the import of **.usd**, **.usda**, **.usdc**, **.abc**, **.mltx**, **.ply** and **.splat** files.

In Nuke's USD-based 3D system, importing the same file twice does not duplicate the objects because USD only allows objects with unique paths to exist in the scene.

Note: If you do want to import another object, use the GeoReference node, see Referencing with GeoReference.

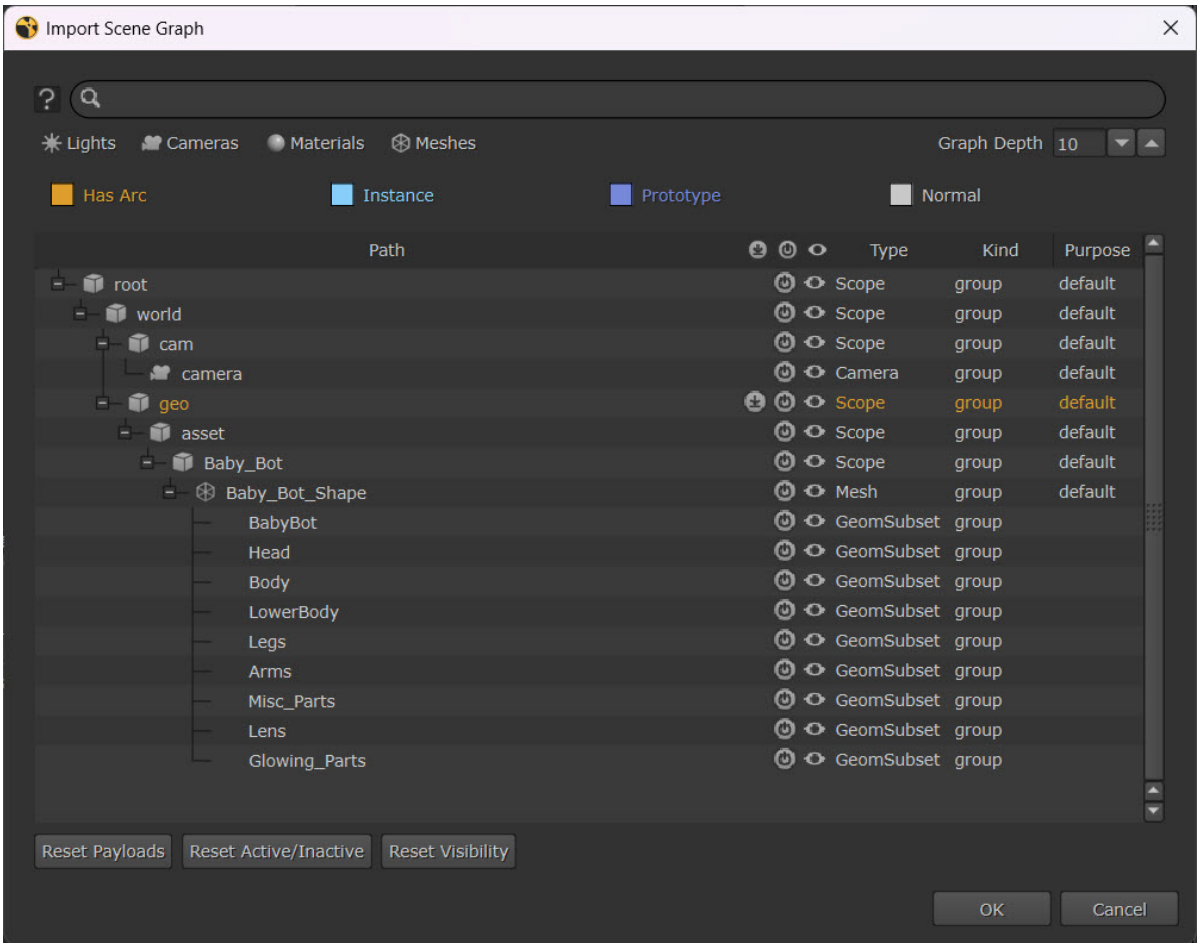
With GeoImport, there is the option to use payloads and activation to improve the loading performance, as explained below.

Note: Nuke doesn't support the import of instanced prims. In USD typically, an instanced prim would be resistant to certain processes, for example, binding a material wouldn't work on those instances. In Nuke, instanced prims behave like non-instanced prims and you would need to set up instances in Nuke separately. See Instancing Geometry to learn more.

To import a USD file:

1. In the top menu bar, click **Workspace** > **3D** to switch to Nuke's dedicated 3D workspace.
2. Create a GeoImport node by clicking **3D** > **Create** > **GeoImport** in the left-hand toolbar, OR press **Tab** in the Node Graph to display the node selector, type **GeoImport**, and then press **Enter**.

   A GeoImport node is added to the Node Graph and its Properties are displayed.
3. In the **File** field, enter the path on disk of the USD file you want to import. GeoImport supports **.usd** and **.usda**, and **.usdc** for USD (as well as **.abc**, **.mltx**, **.ply**, and **.splat** files).
4. Once you've selected a file, an **Import Scene Graph** dialog will appear.



Tip: You can also use the **R** keyboard shortcut to open the Import Scene Graph Dialog.

On this dialog, you can select what payloads are loaded/unloaded and what is active/deactivated via the buttons.

Payloads: By default all payloads will be loaded - you'll see the light grey icon  when payloads are loaded on an item. You can toggle this button on and off for individual items, or use the button at the top of the column to unload all the items at once. When the button is dark grey , the payload is unloaded.

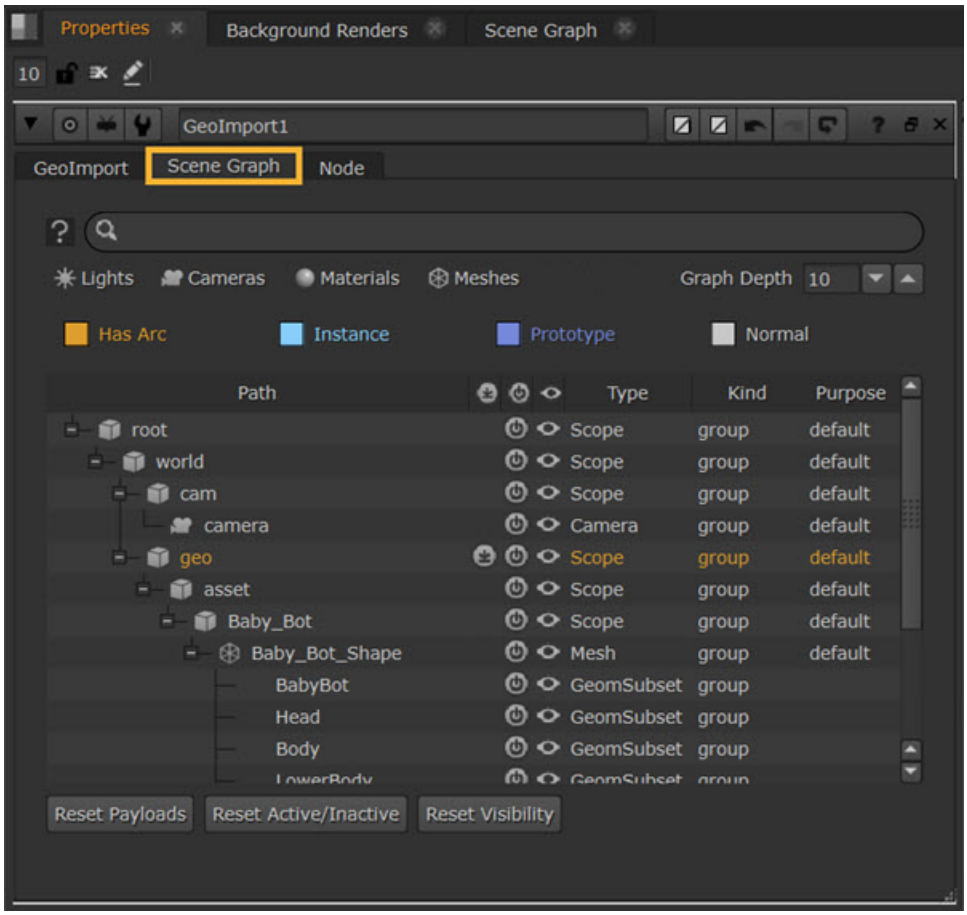
Activation: By default all items will be activated - you'll see the light grey icon  when an item is active. You can toggle this button on and off for individual items, or use the button at the top of the column to deactivate all the items at once. When the button is dark grey, the item is deactivated.

Visibility: By default all items will be visible - you'll see the light grey icon  when an item is visible. You can toggle this button on and off for individual items, or use the button at the top of the column to deactivate all the items at once. When the button is dark grey, the item is invisible.

Note: To see a full explanation of the controls on the GeoImport dialog, see Import Scene Graph Dialog below.

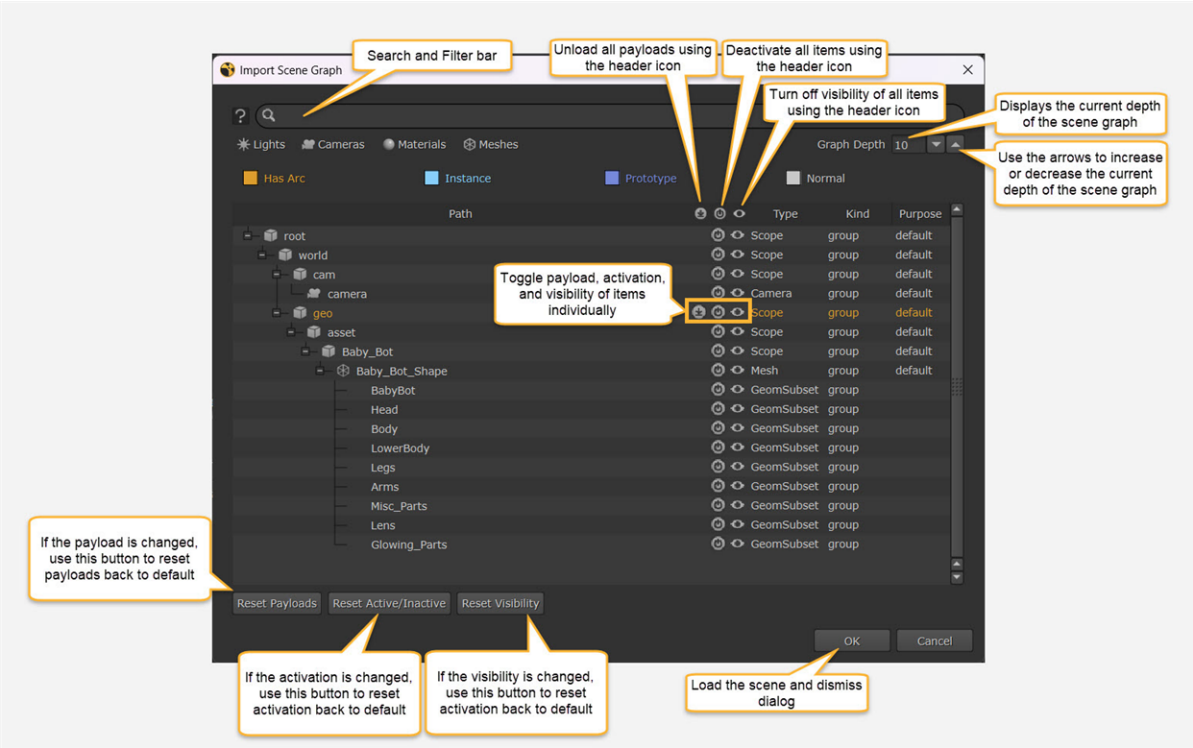
Note: To learn more about the concepts of payloads in USD, see Payloads and Deferred Loading.

5. Once you have made your selections you can hit **OK**, and GeoImport will load your stage into Nuke with payloads and activation that you have enabled.
6. You may wish to move sublayer metadata to the root layer of the stage so that this is not lost. If so, you can use the **Promote layer metadata to stage** checkbox on the GeoImport node.
7. Connect a **Viewer** to the GeoImport node to display the scene in the Viewer.
8. If you wish to amend the payloads, activation, and visibility after the initial import, you can do so by accessing the **Scene Graph tab** inside the GeoImport node.



Import Scene Graph Dialog

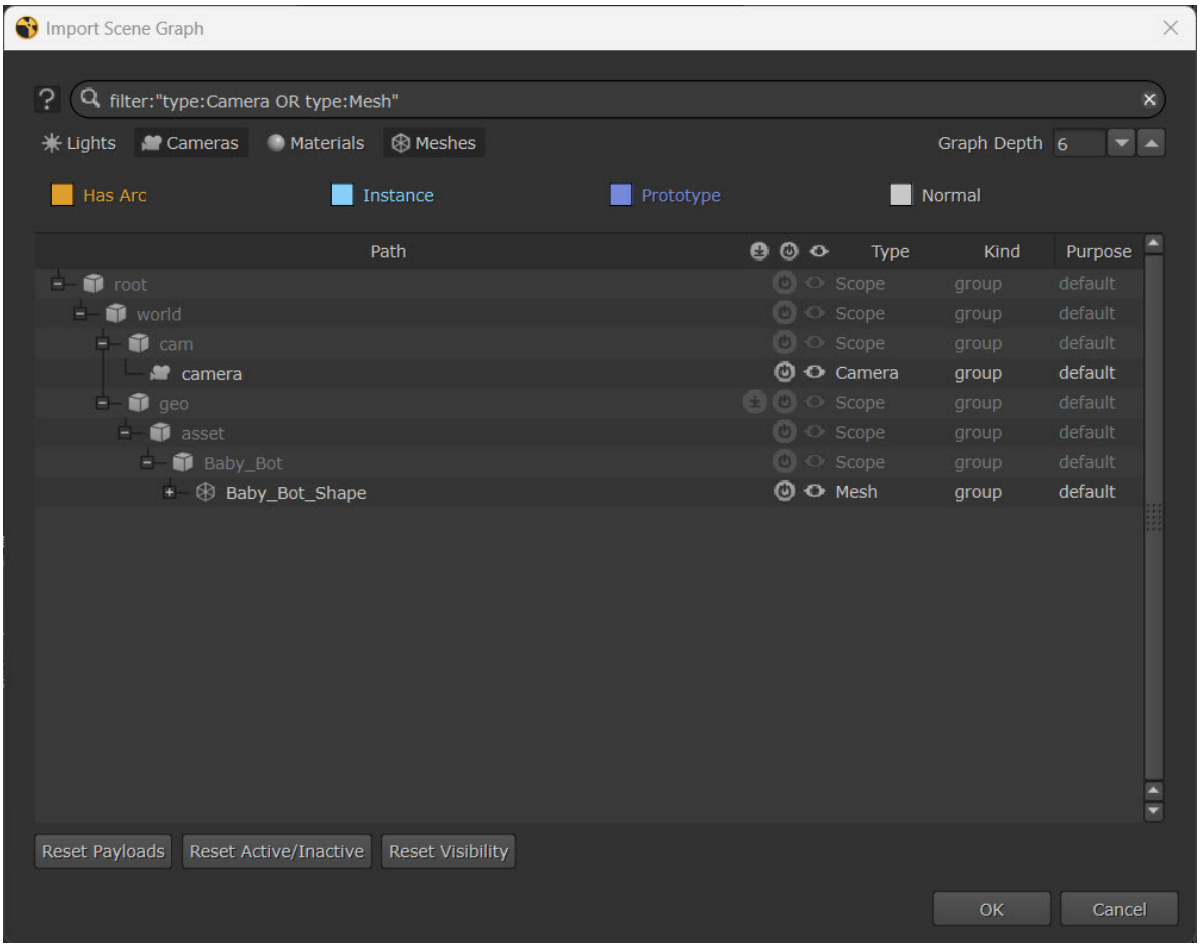
This dialog appears the first time you use the **File** knob of GeoImport. After that point you can access the same options via the **Scene Graph** tab inside the node.



Using the Search and Filter bar, you can either type the name of the prims to search for, or use the preset type filter buttons below (Lights, Cameras, Materials, Meshes) which will show you the prims of that type, with other prims greyed out.

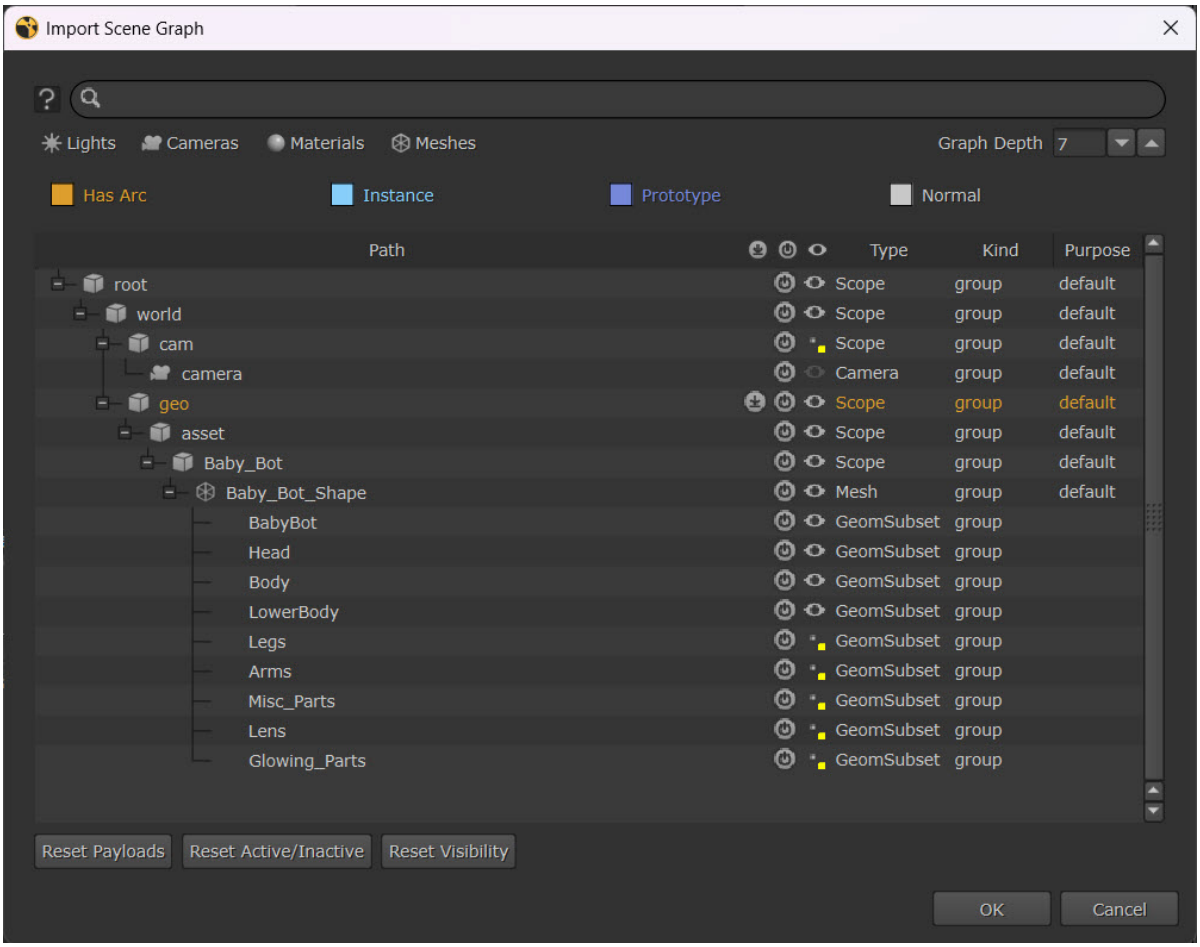
Tip: You can apply multiple filters at once.

In this example, Cameras and Meshes have been applied:



Note: If you use the filters, the Graph Depth will automatically adjust based on what you've searched for.

If you make changes to the payloads, activation, or visibility, any changes will appear with a yellow override icon.



Referencing with GeoReference

In Nuke's USD-based 3D system, importing the same file twice does not duplicate the objects inside because USD only allows objects with unique paths to exist in the scene. This is where the GeoReference node can help you because it allows you to change the path of an object when you bring it into Nuke.

GeoReference doesn't duplicate objects, it creates a light-weight reference to an object often referred to as an instance in other applications, such as Katana. You can use the File Path control to reference a file on disk or the Prim Path control to reference an object already in the scene graph:

- **File Path** references an object from an external file and you can give the object a different path so that you can import instances of the same object several times. If you use GeoImport, the path to the object can't be changed, so you would overwrite the object each time you import it.

- **Prim Path** creates an instance of an existing object in the scene and gives it a new path. This is an easy way to create instances of objects in the 3D system. Bear in mind that GeoReference also applies the transforms on the referenced object to the instance, so it appears in the scene in the same place as the original object.



Note: See Instancing Geometry to learn more about how to copy objects already in the scene graph via instancing.

Reference an Object Using a File Path

Referencing an object using a file path creates an instance from a specified USD file and gives it a new path, effectively duplicating the object. The object does not have to exist in the scene because you specify a file from which the object is referenced.

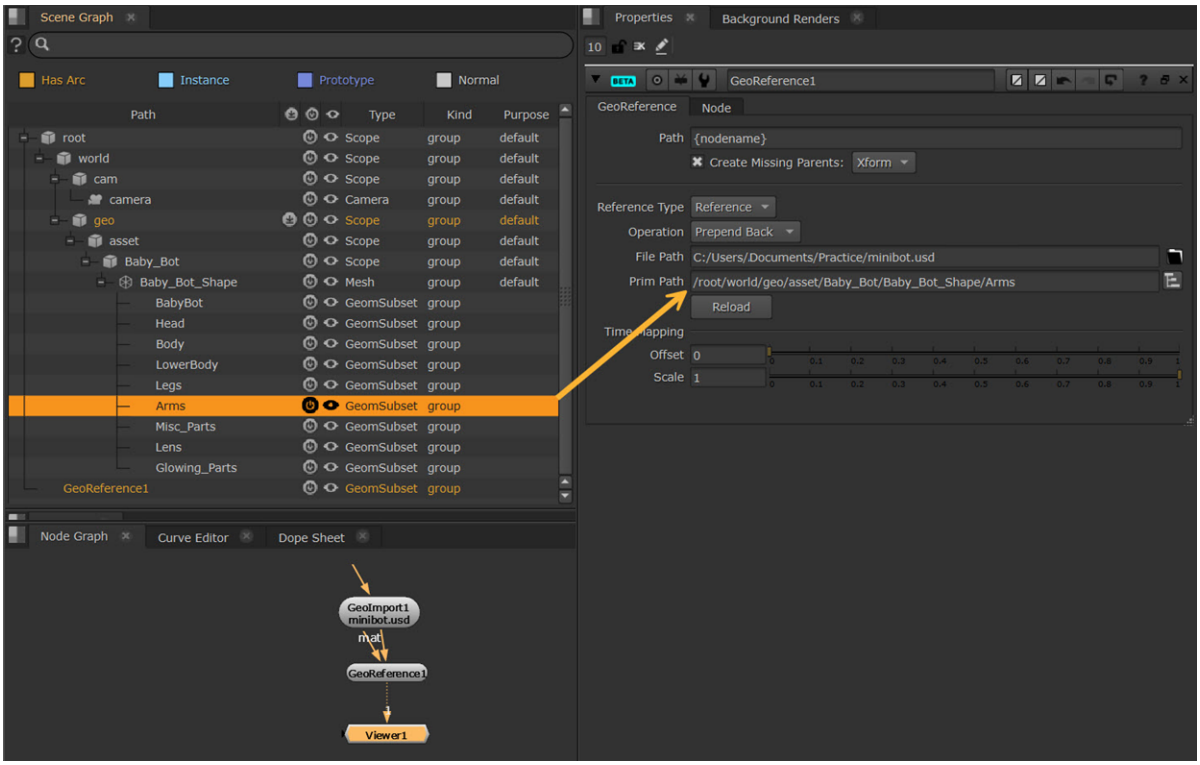
To reference an object using the **File Path**:

1. Add a GeoReference node by clicking **3D** > **Create** > **GeoReference** in the left-hand toolbar,

   OR

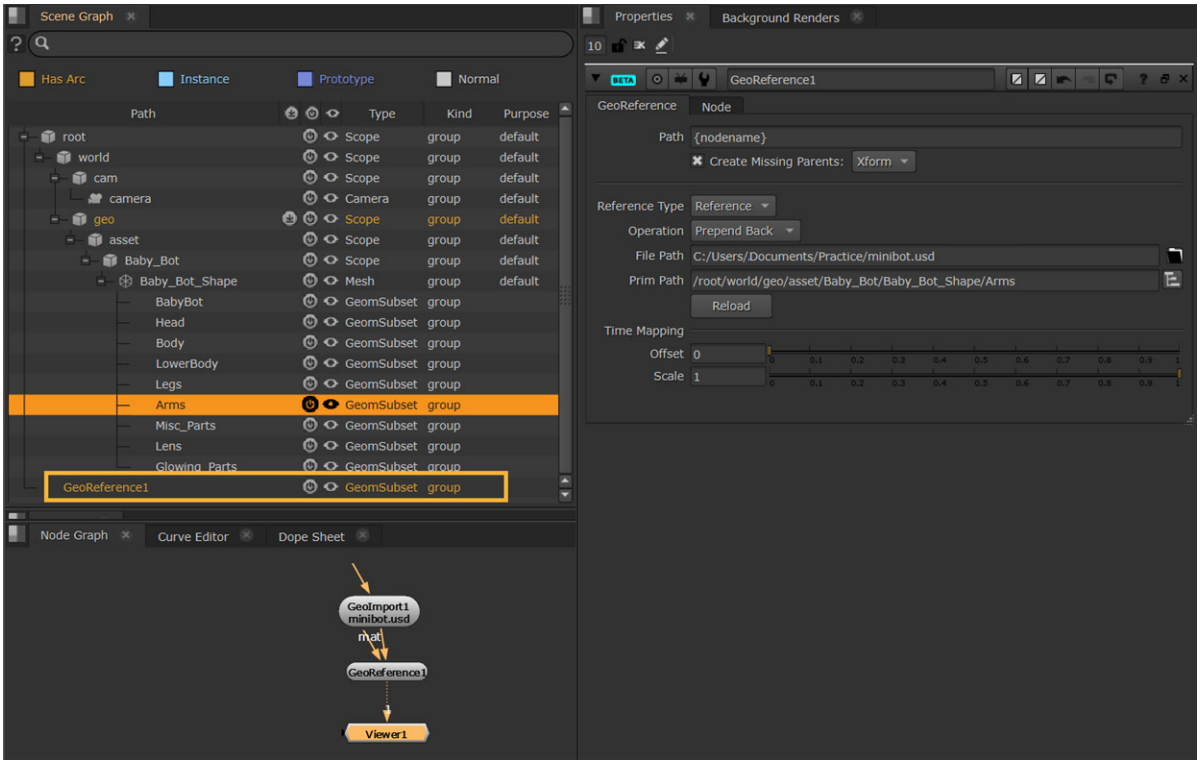
   Press **Tab** in the Node Graph to display the node selector, type **GeoReference**, and then press **Enter**.

   A GeoReference node is added to the Node Graph and its **Properties** are displayed.
2. Select the **Reference Type** you want to add:
   - **Reference** - load the object immediately.
   - **Payload** - payloads can be delayed to stop heavy geometry loading until you're ready.
3. Enter the **File Path** to the USD file on disk that contains the object you want to reference.
4. Enter the **Prim Path** to the object you want to reference. You can enter the path manually or, if you're not sure what the path is, you can load the file using a GeoImport node and then locate the source object in the **Scene Graph** tab. Drag the location to the **Prim Path** control to add it automatically.



The new reference object is added to the bottom of the scene graph by default using the **{nodename}** variable in the **Path** control. In this case, **{nodename}** is GeoReference1. **Create Missing Parents** sets the type of prim to create if there are missing ancestors in the path.

For more information on paths and masks, see Using Paths and Masks.



Reference an Object Using a Prim Path

Referencing an object using only a prim path creates an instance from an existing object and gives it a new path, effectively duplicating the object.

To reference an object using only the **Prim Path**:

1. Add a GeoReference node downstream of an existing prim by clicking **3D** > **Create** > **GeoReference** in the left-hand toolbar,

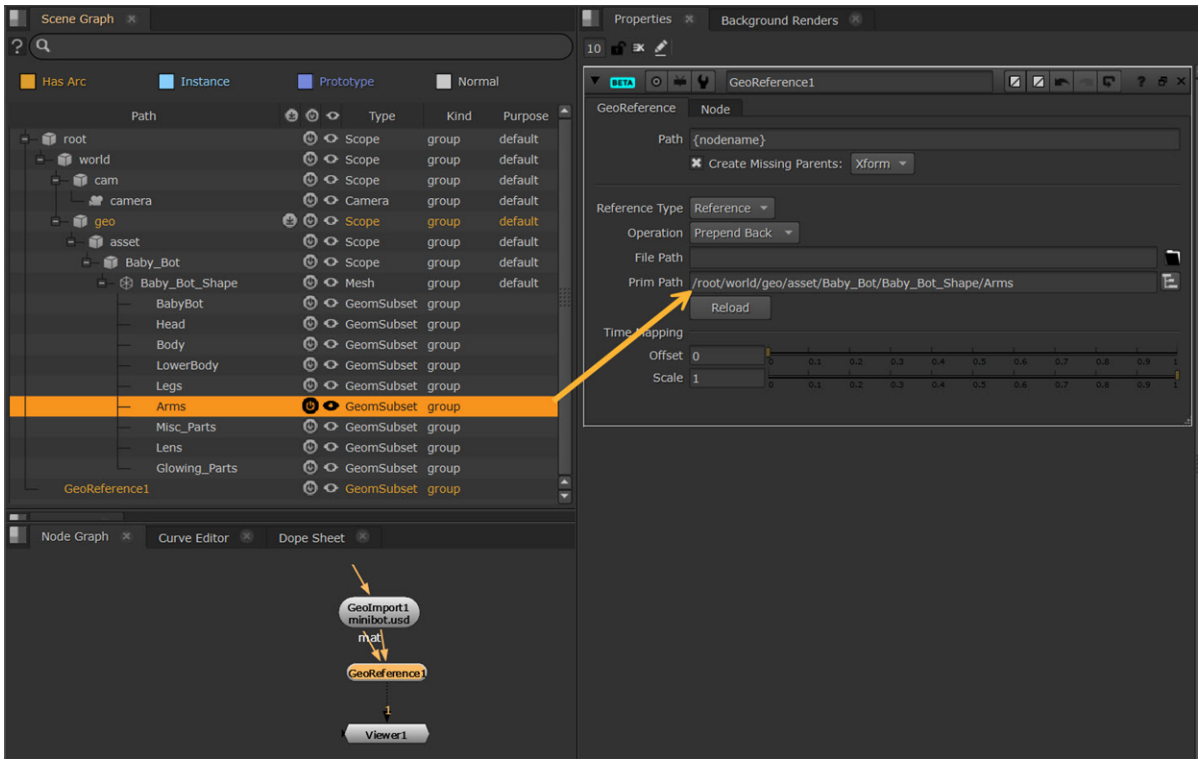
   OR

   Press **Tab** in the Node Graph to display the node selector, type **GeoReference**, and then press **Enter**.

   A GeoReference node is added to the Node Graph and its **Properties** are displayed. In this simple example, the GeoReference node is immediately downstream of the original GeoImport containing the prim we're going to reference. In reality the GeoReference node can be anywhere downstream of the prim it's referencing.

2. Select the **Reference Type** you want to add:
   - **Reference** - load the object immediately.
   - **Payload** - payloads can be delayed to stop heavy geometry loading until you're ready.

3. Enter the **Prim Path** to the object you want to reference. You can enter the path manually or, if you're not sure what the path is, you can locate the source object in the **Scene Graph** tab and drag the location to the **Prim Path** control to add it automatically.



The new reference object is added to the bottom of the scene graph by default using the **{nodename}** variable in the **Path** control. In this case, **{nodename}** is GeoReference1. **Create Missing Parents** sets the type of prim to create if there are missing ancestors in the path.

For more information on paths and masks, see Using Paths and Masks.

Referencing an existing prim copies its transforms as well, so the referenced prim is rendered in exactly the same position as the source object.

Payloads and Deferred Loading

Payloads let you choose which parts of a scene are loaded into memory.

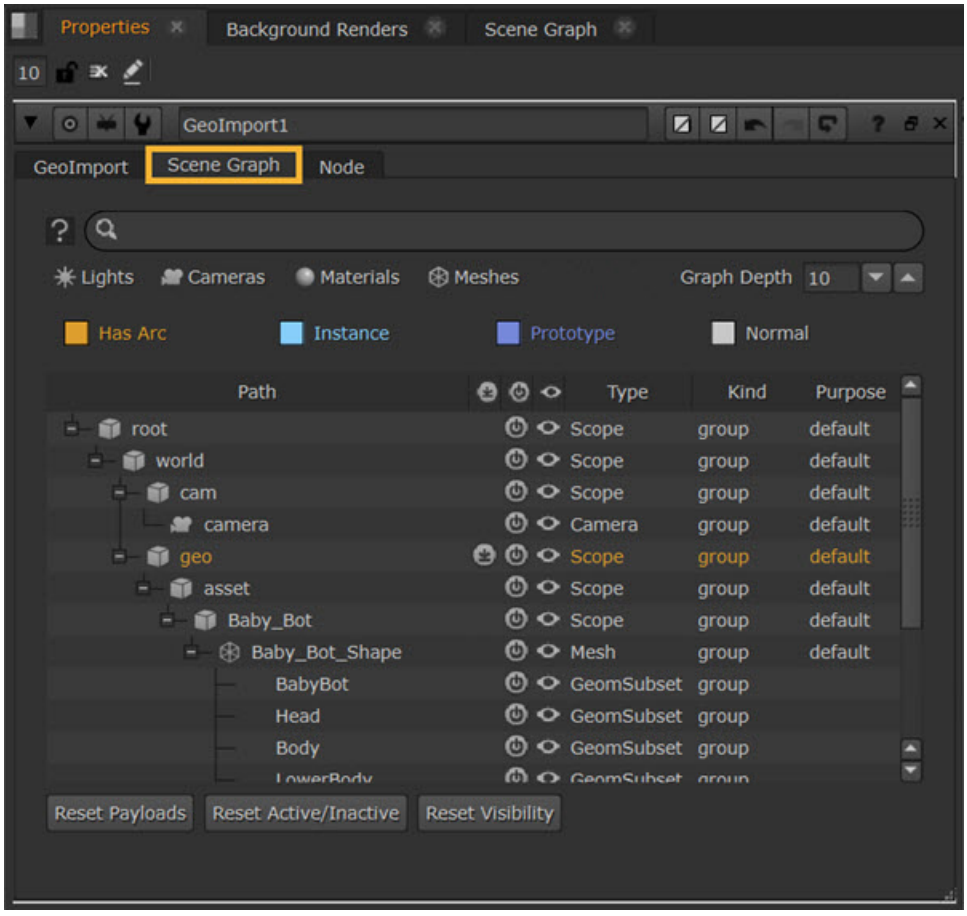
This helps you save on memory and speed up load times, which can be particularly useful for complex scenes with high levels of detail. This is a USD concept which Nuke approaches by adding simple UI controls.

In Nuke, payload controls take place in the GeoImport node, allowing you to control what is loaded in from the initial import.

- When you first insert a GeoImport node, a dialog will appear where you can select your payloads via the payload button: 

Note: To learn about the controls, see Import Scene Graph Dialog.

- Once you have made your payload selections, GeoImport will load your stage into Nuke with payloads you have enabled.
- These controls can then be later accessed and edited from the **Scene Graph tab inside the GeoImport node**, or via the Project's main scene graph.



To see more about how to use the GeoImport node, see Importing with GeoImport.

Note: To learn more about payloads in USD, see OpenUSD's documentation on Packaging assets with payloads.

Creating Prims in Your Scene

As explained on the Primitive Types and Primitive Attributes guide, you can create various types of USD prims directly in Nuke.

Find out how to do this:

For geometry, see Creating Geometry Prims.

For scopes and Xforms, Creating Scope and Xform Prims.

For points clouds, see Creating Points Clouds.

For cameras, see Working with Cameras in the New 3D System.

For lights, see Light Your Scene.

Creating Geometry Prims

Sometimes you just need simple reference geometry or a card for projection workflows and Nuke's New 3D system ships with similar nodes to the Classic 3D System for these tasks. Creating geometry such as cubes, spheres, and cards is the same as the classic 3D workflow. You add nodes, such as GeoCube or GeoCard, to the Node Graph from the left-hand toolbar or by pressing Tab in the Node Graph and typing the node name you need.

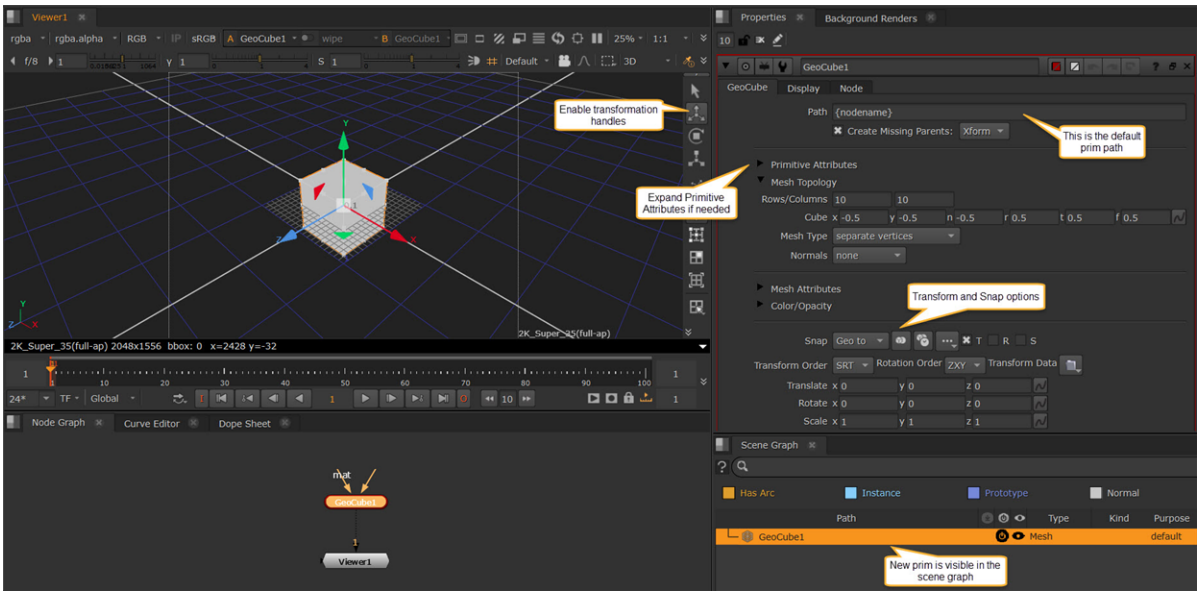
Note: If you are less familiar with USD terminology, see USD Basics: Stage, Prims, and Layers.

To create a Nuke geometry prim:

1. In the node graph, right click and go to **3D** > **3D** > **Create** > **GeoCard**, **GeoCube**, **GeoCylinder**, or **GeoSphere**.

   The specified node is added to the Node Graph and its Properties are displayed.

   Connect a **Viewer** to the geometry node to display the scene in the Viewer, and make sure the Viewer is set to **3D** mode. See Viewing Your Scene for more information.
2. The new geometry is added to the bottom of the scene graph by default using the **{nodename}** variable in the Path control. In this case, **{nodename}** is **GeoCube1**. For more information on paths and masks, see Using Paths and Masks.
3. Use the controls in the **Properties** panel under the **Mesh Topology** section to edit the dimensions of your prim - there are a variety of different options depending on the prim. For example, GeoSphere has a **Radius** knob, and GeoCylinder has a **Height** knob. You can also use the controls under the **Primitive Attributes** section to control the geometry's attributes such as **Kind** and **Purpose**. To learn more about these see Primitive Types and Primitive Attributes.
4. Use the Transform controls in the Properties panel or use the transformation handles to position your geometry in the scene. If you want to transform multiple objects simultaneously you should add a GeoTransform node instead. See Transformations in 3D Scenes and GeoTransform for more information.



Tip: In the node's **Snap** menu, you can now switch between snapping Geo or snapping Pivots. Learn more about the snap menu at Using Transformation Tools.

5. You may then wish to assign materials to your geometry. The **mat** input can be used to add shaders and materials directly to your prim. See more about this at Assigning Materials and Shaders.

Note: Each node has specific attributes which help to edit that particular prim. For example, GeoCard nodes also include LensDistortion and Deform controls. See the Node Reference Guide pages linked below for the full breakdown of all the controls.

Geometry Creation Nodes - Node Reference

GeoCard

Creates a card object in the scene.

GeoCube

Creates a cube object in the scene.

GeoCylinder

Creates a cylinder object in the scene.

GeoSphere

Creates a sphere object in the scene.

Creating Scope and Xform Prims

Scopes and Xforms both allow you to create groups of prims in your scene.

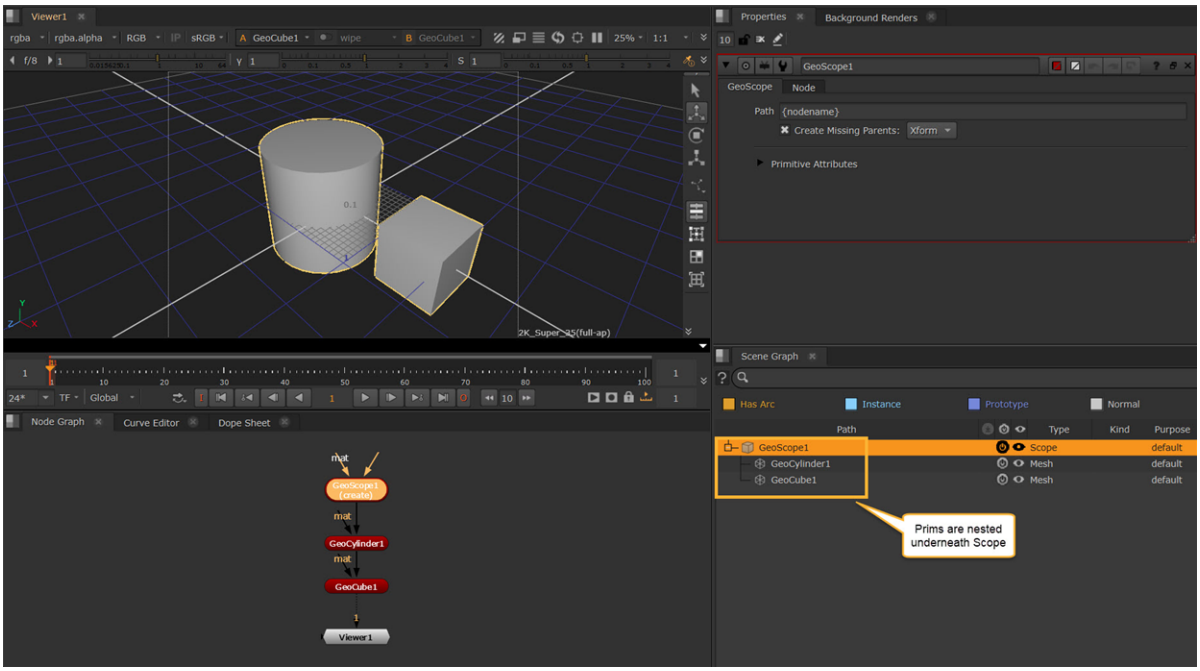
- GeoScopePrim allows you to create the simplest form of grouping, which is called a **Scope**. Scopes are like containers which can be useful for organizing the scene graph in higher up locations. It is similar to an Xform in that it is a container prim. However, the most important distinction between an Xfrom and a Scope to be aware of is that Scope cannot be transformed.

- GeoXformPrim introduces the ability to create **Xforms**. Xform prims are another form of grouping that can also be transformed themselves. This is great if you want to nest geometry under one group and then transform that top level group Xform so that all the geometry that are nested underneath also move, but retain their own unique transform data. The **Prim Transform Order** knob allows you to specify at which point the transform order is specified to.

Using GeoScope

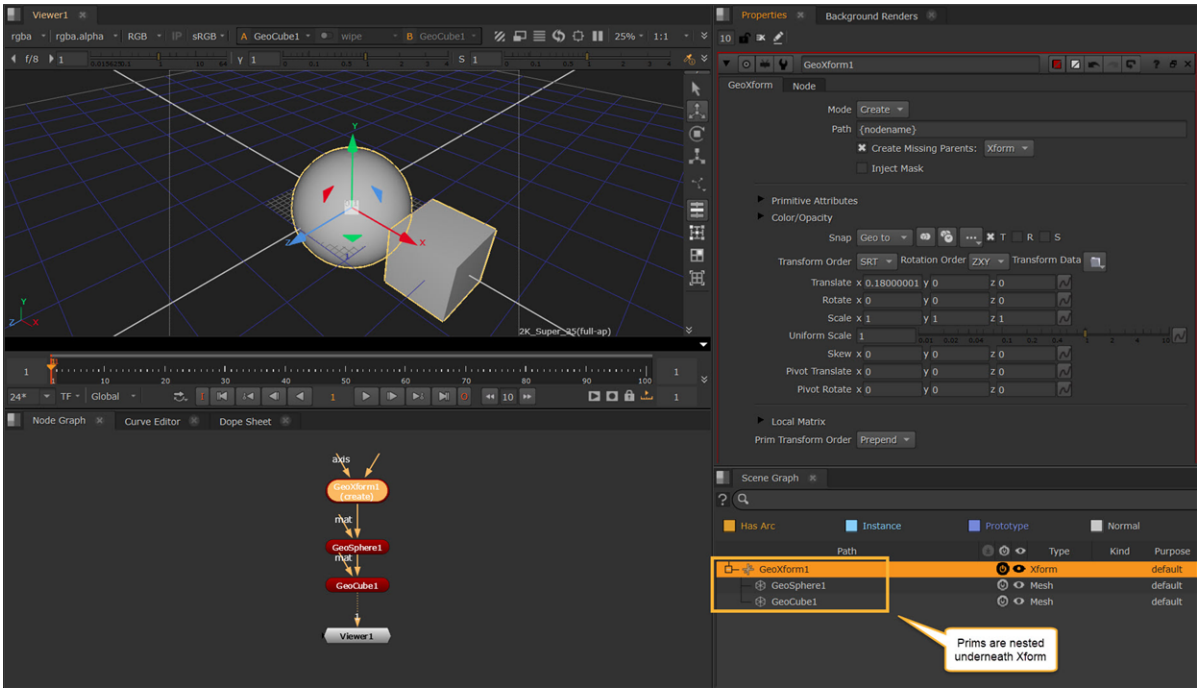
For example, you could use a GeoScopePrim to group a Cube and a Cylinder together. By connecting a GeoScopePrim node upstream of your other geometry prims, they will now be grouped together in the scene graph. Now if you wanted to change the Visibility of both prims, you could change this attribute on the GeoScope instead of changing them both individually.

Note: To read step by step how to create a Cube or Cylinder, see Creating Scope and Xform Prims.



Using GeoXform

To transform multiple prims at once, you can use a GeoXformPrim node. By connecting a GeoXformPrim node upstream of your other geometry prims, they will now be grouped together in the scene graph. Now you can then use the transform handles, snap menu or other transformation tools to move them in one group.



Note: See more about transformations at Using Transformation Tools.

Scope and Xform Nodes - Node Reference

GeoScopePrim

Creates an object used to group other objects together.

GeoXformPrim

Creates and edits Xform prims in your USD scene.

Creating Points Clouds

GeoPoints creates a point cloud prim from its input. This is usually the simplest way to create a point cloud from inputs such as meshes and particles.

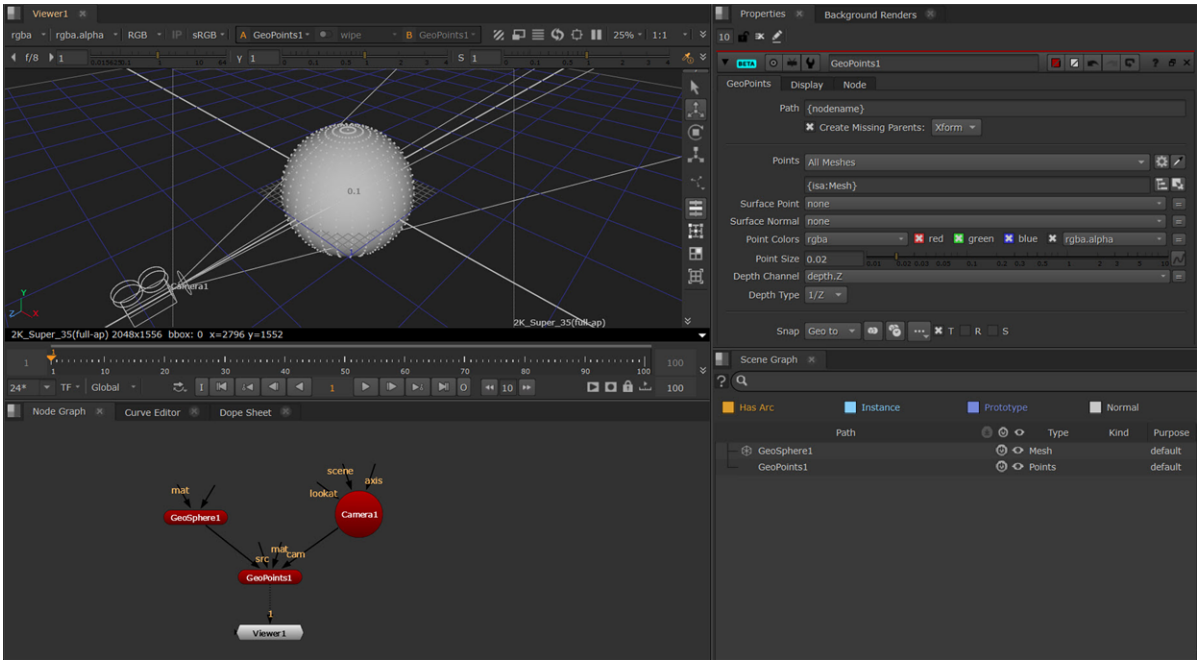
Note: GeoPoints is similar to the Classic 3D system DeepToPoints and PositionToPoints nodes, but wrapped into one.

The **Points** knob is where you can select the prim(s) to use for your point cloud. **All Meshes** may be a convenient option in order to turn your input mesh into a point cloud prim.

The **Surface Point** and **Surface Normal** knobs are for inputting channels that contain position or normal information. You can also populate this information using the two additional input pipes on the node.

You can then adjust the **Point Size** to your liking. Be aware that sometimes the default point scaling can get blocked by the mesh making it look as if no points have been generated so you may need to adjust this setting.

Here is an example of creating a point cloud from a sphere prim:



Tip: You can also use points clouds to generate a mesh (NukeX only). The GeoPointsToMesh node uses geometry normals to generate a mesh from a point cloud using Poisson Surface Reconstruction.

Note: There are also various nodes to use for Points Clouds with CameraTracker, such as PointsGenerator, DepthGenerator, and DepthToPosition.



Note: CameraTracker, DepthGenerator and DepthToPosition can be used for both the Classic 3D System and the New 3D System.

Point Cloud Nodes - Node Reference

GeoPoints

Creates a point cloud from its input.

GeoPointsToMesh

Uses geometry normals to generate a mesh from a point cloud.

Point Cloud with CameraTracker Nodes - Node Reference

PointsGenerator

Create a dense point cloud camera tracker data.

DepthGenerator

Generates a depth map from camera tracker data.

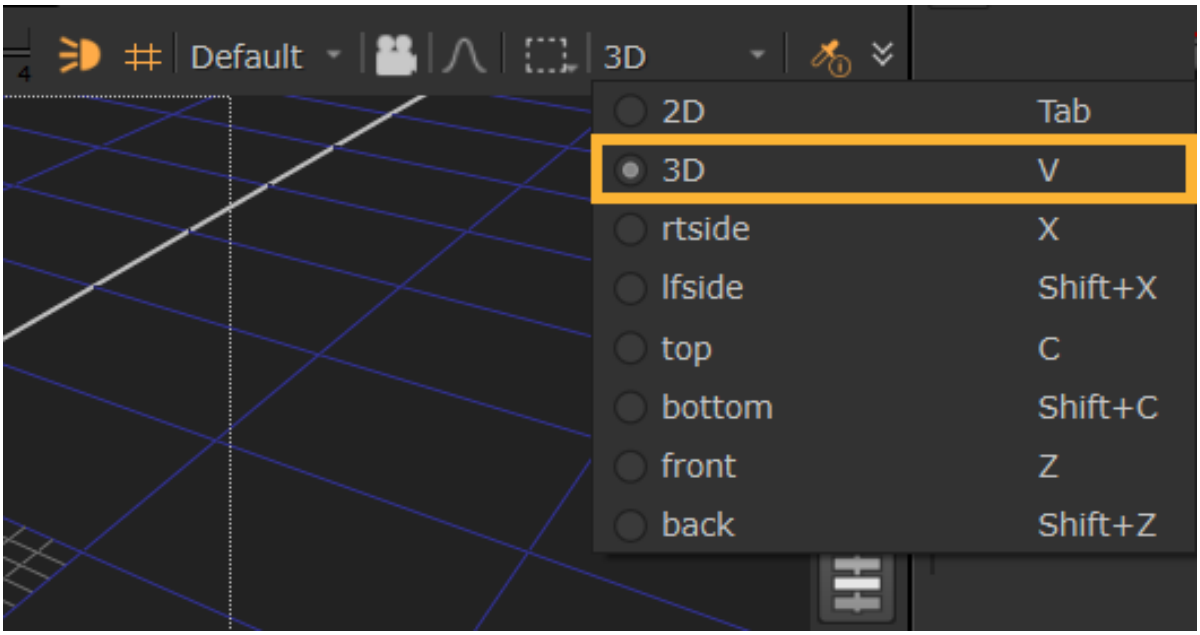
DepthToPosition

Creates a 2D position (xyz) pass from camera tracker data.

Viewing Your Scene

The Viewer's 3D Mode

Open a Viewer and use **V** to switch to **3D** mode - or select the view you want from the dropdown menu at the top right corner of the Viewer window.



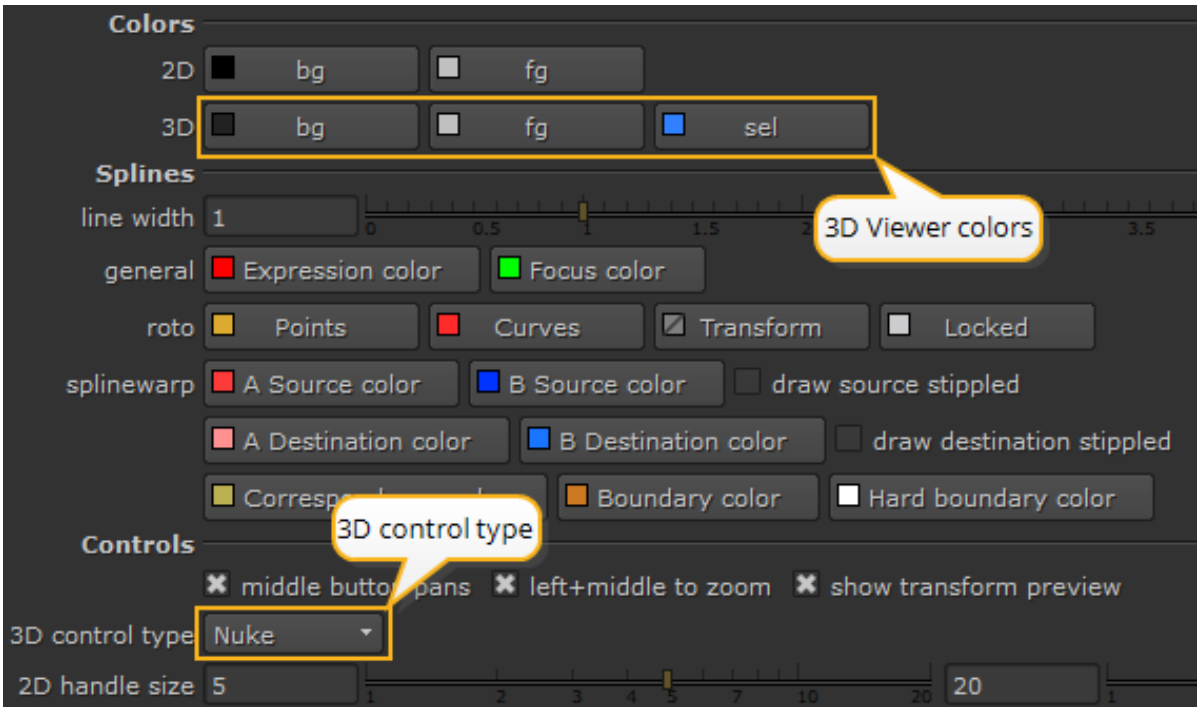
The "built-in" views give you different perspectives on your 3D scene. You can quickly switch between the views by pressing the keyboard shortcuts for right view (**X**), left view (**Shift+X**), top view (**C**), bottom (**Shift+C**), front (**Z**), back (**Shift+Z**), and three-quarter perspective (**V**).

Navigate in the 3D Viewer

- Dolly: Press **Alt** and **middle-mouse-button** drag.
- Pan: Press **Alt** and **left-mouse-button** drag.
- Tilt: Press **Ctrl/Cmd** and **left-mouse-button** drag.
- Spin: Press **Ctrl/Cmd** and **left-mouse-button** drag.
- Roll: Press **Ctrl/Cmd+Shift** and **left-mouse-button** drag.
- Fit the scene: Press **F** to fit the entire 3D scene within the Viewer.

Change the 3D Viewer Display Properties

1. Open the **Preferences** dialog (**Shift+S**), and select **Panels** > **Viewer Handles**.



2. Make the desired changes to the 3D **bg** and **fg** colors.
3. From the **3D control type** dropdown menu, select the navigation control scheme you want to use (Nuke, Maya, Houdini, Lightwave, or Modo).
4. Click **OK**.

Note: The 3D control type also affects the mouse button assignments for panning and zooming in the node graph and 2D Viewers.

Look Through Cameras and Lights

You can also look through Cameras and Lights in your stage to be able to see their angles and reposition them. See Look Through Camera to learn more.

Selection

The 3D Viewer has a variety of control buttons in the 3D toolbar to help with navigating your scene. See Selection in the 3D Viewer.

Hydra

Nuke's **New 3D System** uses a Hydra renderer, and nodes in the **Classic 3D System** use a "Nuke" renderer.

As such, the Viewer's **renderer** knob only has one option: **GL**, which is both the Hydra and Nuke renderers, depending on if you are viewing New or Classic 3D nodes.

For the New 3D System, this currently renders using the Hydra renderer, **HDStorm**.

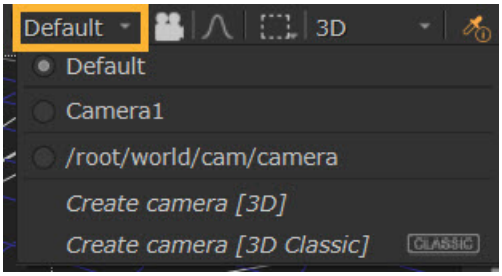
To learn more about Hydra, see Hydra in the 3D Viewer.

Look Through Camera

Looking directly through the cameras and lights in your stage can be helpful for positioning. Via the 3D Viewer dropdown, you can select cameras and lights that are active in the scene and the Viewer will switch to their view.

Cameras supported: You can look through any cameras created with the Camera or GeoCamera node, or any imported cameras that are being edited via GeoEditCamera.

1. Press **V** to make sure you are looking through the 3D perspective view, and not one of the orthographic views.
2. From the 3D Viewer window, select the camera from the dropdown menu in the top right corner. Here you can see both a Nuke camera (created by the Camera node), and an imported USD camera (that is being edited by GeoEditCamera node).



Tip: This dropdown also provides shortcuts to creating new cameras.

3. Select your desired camera, and the Viewer will switch to view through your camera.

Tip: Select a camera object and press **H** to look through it.

Note: This selection does not change the camera used for rendering. This changes only the camera to "look through" for the current 3D Viewer.

3D Camera View

You can lock the 3D view to the selected camera or light. You can toggle between the unlocked and locked modes by clicking the **3D view lock** button (next to the camera dropdown) , or by pressing **Ctrl/Cmd+L**.

•  **unlocked** - move freely in the 3D view without restrictions. The 3D view lock button is gray.

•  **locked** - lock your movement to the camera or light you've selected in the dropdown menu on the right side of the 3D view lock button. The 3D view lock button is red.

To Use the Interactive 3D Camera View Mode

With the interactive 3D camera view mode you can change the camera or light values according to your movement in the Viewer. You can activate the interactive mode by **Ctrl/Cmd**+clicking the 3D view lock button. When the interactive mode is on, the **3D view lock** button turns green: 

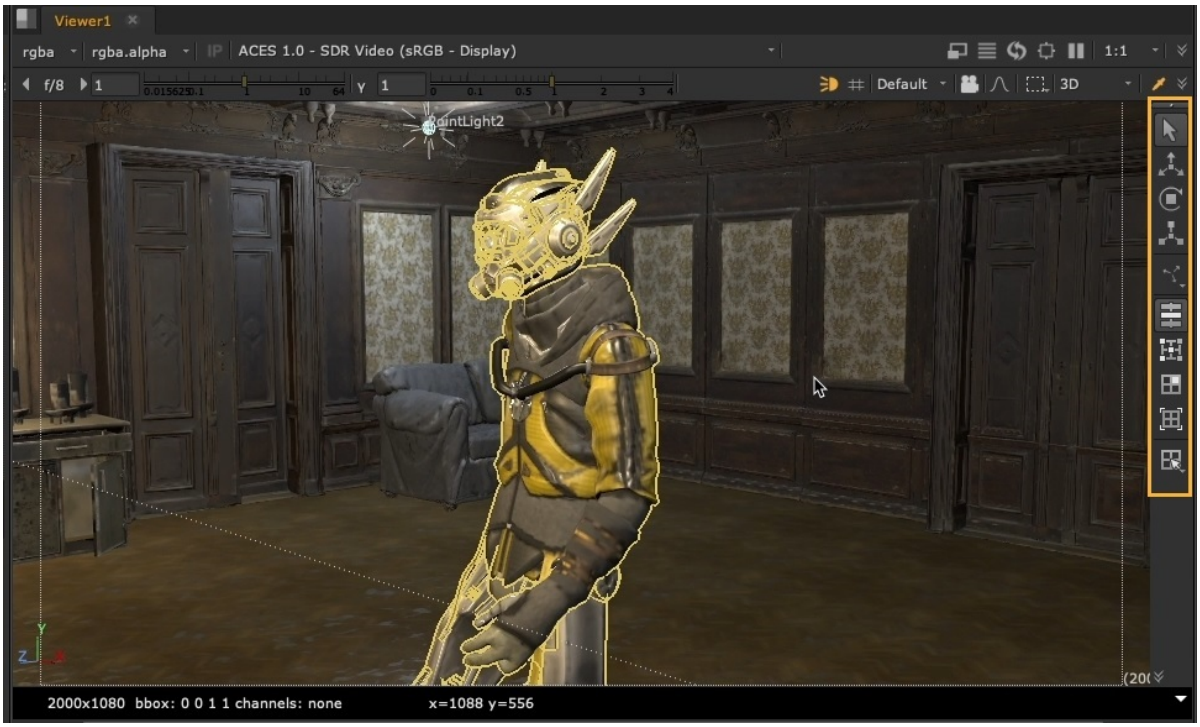
In order to activate the interactive mode, you need to have a Camera or a Light node selected in the dropdown on the right side of the **3D view lock** button.

When the interactive mode is on, you can use the plus (**+**) and minus (**-**) keys to change the translate values of the camera or light you've selected. When the interactive mode is off, these keys zoom your view in and out.

Selection in the 3D Viewer

The new 3D system includes an enhanced selection system, designed to provide artists with increased flexibility and control when navigating large scenes using the latest 3D architecture.

The selection control buttons are in the 3D toolbar to the right of the 3D viewer panel.

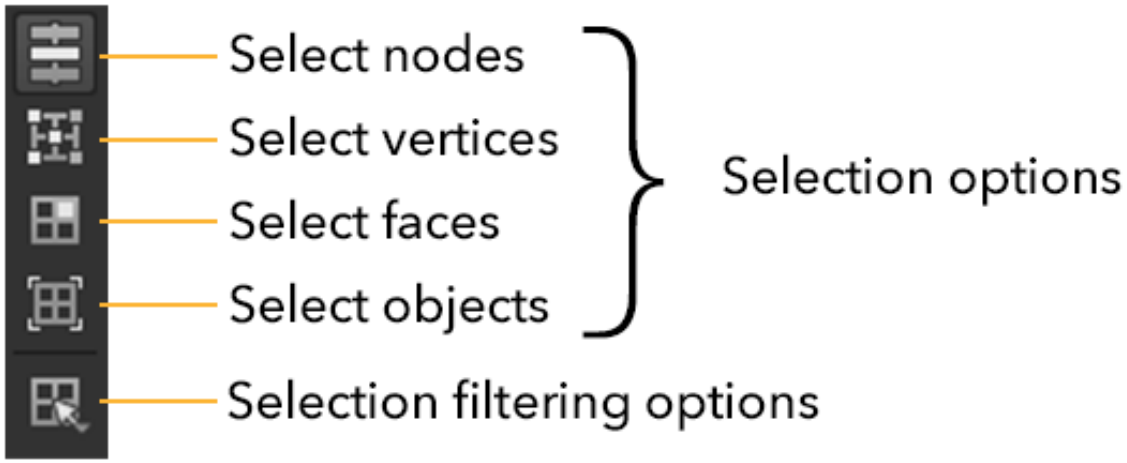


The 3D Toolbar on the right of the Viewer.

This toolbar offers essential controls for transforming and making selections in the 3D space.

Note: Any selection you make in the 3D viewer is also shown in the Scene Graph (see New 3D Scenes in the Scene Graph). Conversely, you can select prims in the scene graph and see the corresponding selection in the 3D viewer.

3D Toolbar Selection Buttons - Quick Reference



Note: The 3D toolbar also includes 3D transform buttons. To learn more about these, see Using the Transform Handles

Using the Selection Tools

Selection is a simple process:

- Choose the objects and type of item that you'd like to select from the filtering options menu . For example, if you want to select all USD components, choose **Component**.
- Choose a selection mode for the type of element that you want to select (node, vertex, face, or object).
- Click, or click and drag a marquee to select items in the viewer.

Simply put, when you click or drag in the 3D viewer, you select the items based on the chosen filtering options and see the elements according to the selection mode.



Here, the selection is filtered by mesh in face selection mode. The mesh is the torso part of the figure.

Once you've made a selection, you can change the selection mode at any time to switch element types.

To focus your selection on a part of the item, choose a selection mode and select the part. We call this two-tier selection.

Two-Tier Selection

Sometimes, you may want to locate an object in a scene quickly and then focus your operation on a particular part of that object. For example, finding a character in a busy scene and then modifying part of its body or costume. We've streamlined this by providing a "two-tier" selection workflow.

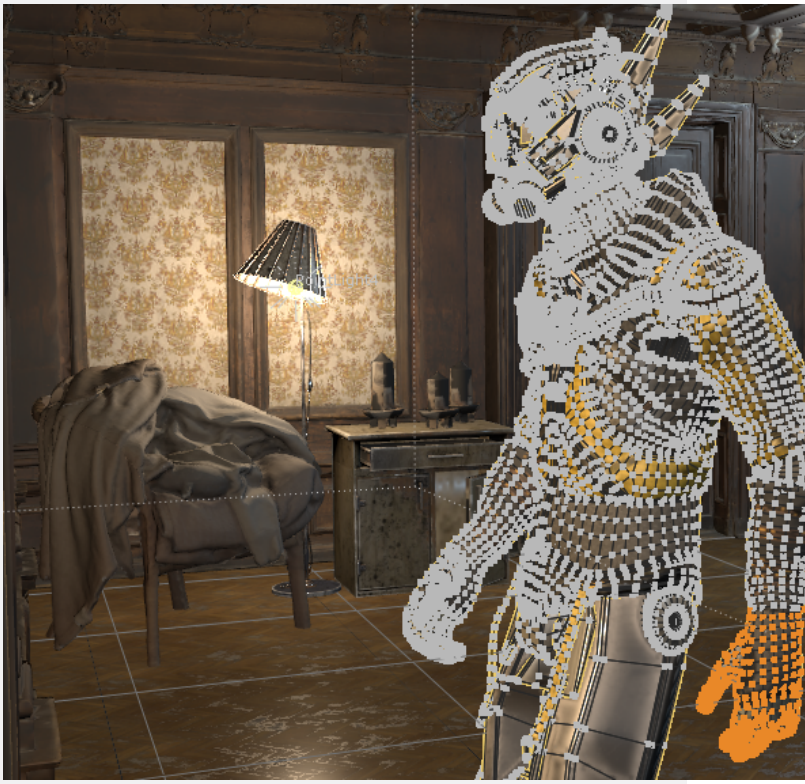
Two-tier selection is the process of first selecting an item and then using the selection modes to sub-select part of the item as follows:

- *Tier 1*: Select the item based on the filtering option.
- *Tier 2*: Sub-select part of the item based on the selection mode (node, vertex, face, or object).

For example, you might filter by **component** to select an entire component type part of the scene. Once selected, you could then activate vertex selection mode to limit your selection to a set of vertices within that component. You can see an example of this below.



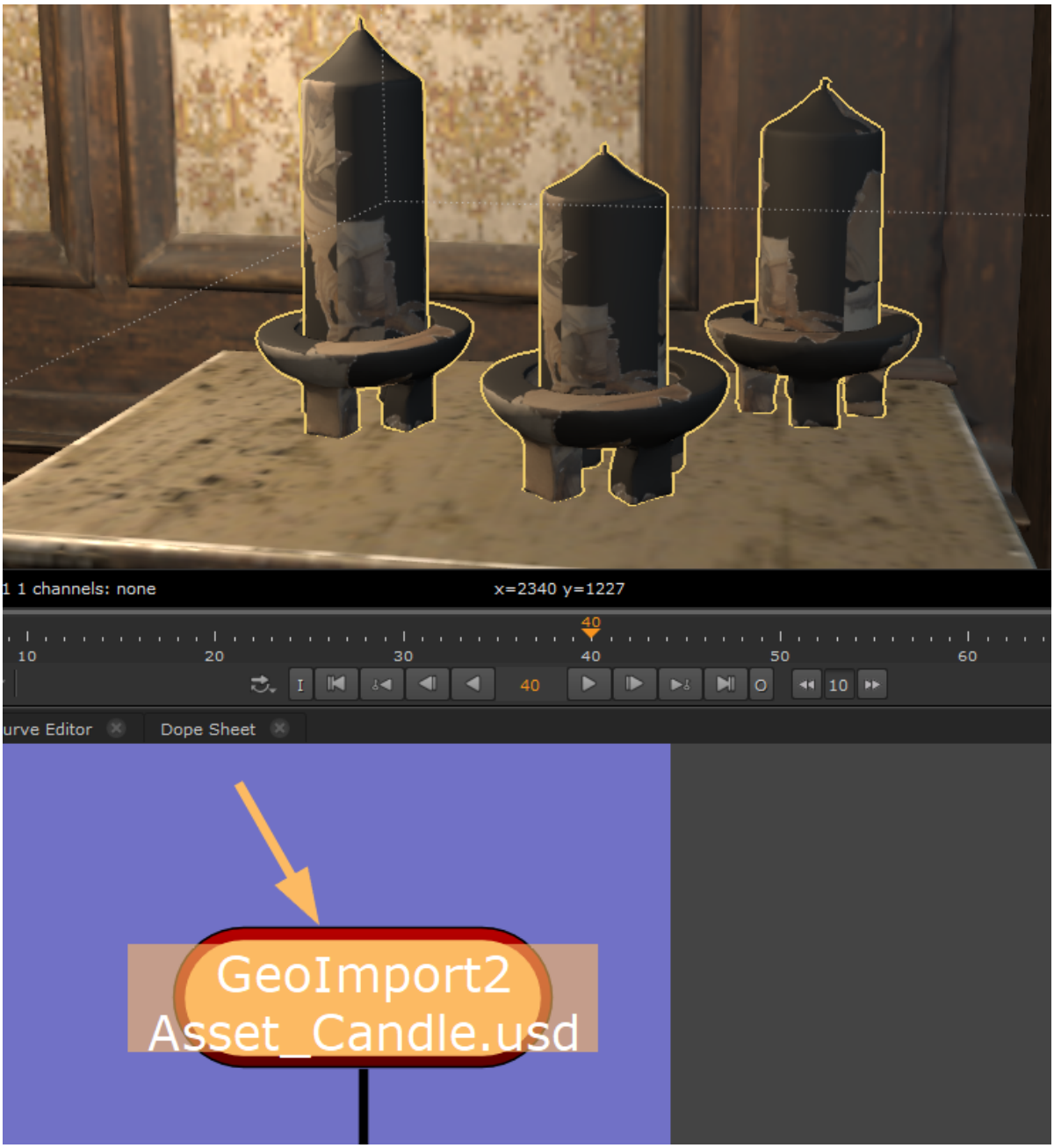
First-tier selection - A **component** type filter is active to select the figure. This is done in vertex selection mode.



Second-tier selection - the glove has been selected by drag selecting.

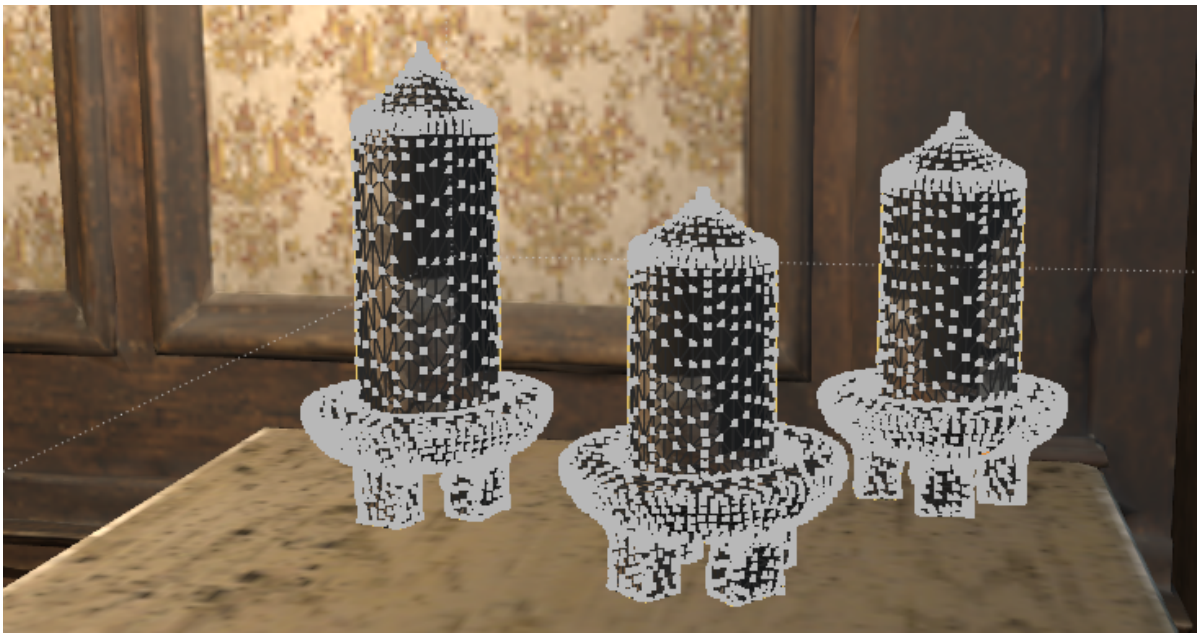
Selection Modes

Nuke offers four selection modes: node, vertex, face, or object. The table below shows examples using a component selection. The candles were imported in a separate USD file.



Node selection

The node that generates the geometry is highlighted on the node graph. You can double-click in the 3D viewer to open the node properties.



Vertex selection



Face selection



Object selection

Adding to Selections

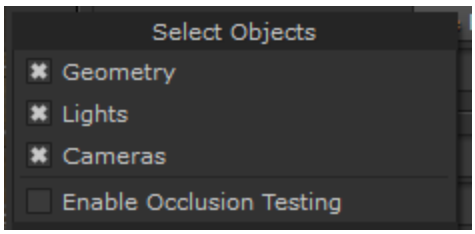
You can add to any selection using **Shift** +click or **Shift** + drag.

Filtering the Selection Mode

The selection filter determines the type of objects and items that are selected in the 3D viewer.

Click the filtering selection button to open the filtering options.

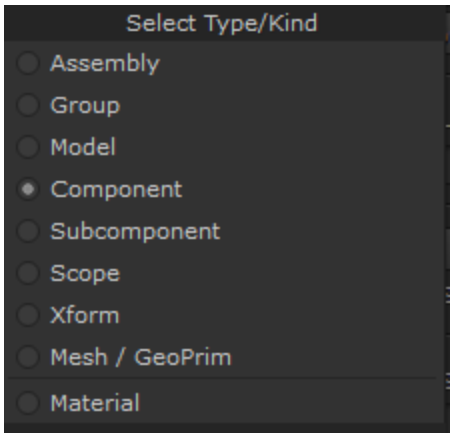
Filter by Object Type and |Toggle Occlusion Testing



Check the boxes to select the objects that you want to include in a selection. You can choose from **Geometry**, **Lights**, and **Cameras**.

Check **Enable Occlusion Testing** to select only objects that are immediately visible from your camera position, and not hidden behind surfaces. See Selection Based on Camera Occlusion .

Filtering by Type/Kind



The Type/Kind corresponds to the **Type** and **Kind** columns in the Scene Graph (seeNew 3D Scenes in the Scene Graph).

"Type" refers to the data category of an element. For example, **Mesh** represents meshes, and **Xform** represents a transform.

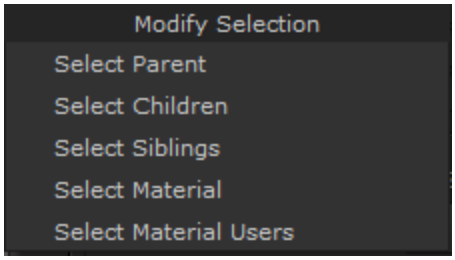
"Kind" refers to the organizational role of an element in the hierarchy. It is categorized as values like **component** for individual objects or **assembly** for a collection of components.

Nuke's selection filtering menu offers the following types:

- **Assembly** - A high-level organizational unit that combines multiple groups or components, typically representing a complete object or scene, such as a character and its accessories.
- **Group** - This container acts as a cluster for grouping related components or subcomponents, facilitating efficient organization and hierarchy. For example, it enables grouping all engine parts of a vehicle together.
- **Model** - A self-contained entity or object within the scene, such as a character, vehicle, or other significant entity, consisting of multiple components and subcomponents.
- **Component** - A fundamental building block in a USD scene graph is an individual object or model component, such as a door in a building or a wheel on a car.

- **Subcomponent** - A smaller, individual part within a component in a USD scene graph, such as the knob of a door or the spokes of a wheel.
- **Scope** - A non-renderable organizational element that hierarchically groups related items without applying any transformations or attributes to the contained elements.
- **Xform** - Short for transform, this is a transformation schema applied to an element, specifying its position, rotation, and scale in 3D space, determining the object's placement and orientation in the scene.
- **Mesh** - A geometric primitive comprised of vertices, edges, and faces, serving as the visual structure of objects such as the surface of a 3D car model or character.
- **Material** - Surface properties of an object, such as color, texture, and reflectivity, interact with light, and rendering. These attributes contribute to realistic visual effects on the applied mesh.

Modify Selections by Hierarchy or Materials



After making a selection, you can use this menu to change the selection to a relative of the selected item as defined by the USD scene graph.

- **Parent** - Select the higher-level entity or node in the scene graph hierarchy directly above the selected item.
- **Children** - Select the nodes or entities in a USD scene graph that are directly under the selected item.
- **Siblings** - Select the nodes or entities that share the same parent node as the selection.
- **Material** - Select only the material(s) that are in the current selection.
- **Material Users** - Select all nodes or entities in the scene graph that share the same material(s) as the selection.

Selection Based on Camera Occlusion

In the selection filtering options,  you can toggle **Enable Occlusion Testing** to choose to select all items throughout the depth of the scene or only those visible to you at the current camera position:

Enable Occlusion Testing off: The selection includes all items behind the selection marquee throughout the entire depth of the scene. This means that even if you can't see them (they're occluded), they'll be selected if they're within the bounds of the marquee.

Enable Occlusion Testing on: The selection is limited to only items that are visible to you from the current camera position. Items behind the selection but within the marquee aren't included.



A selection of the top faces of a glove.



The back of the glove is unselected as **Enable Occlusion Testing** is on.



The back of glove following the same selection but with occlusion testing off.

Occlude the Selection Outline

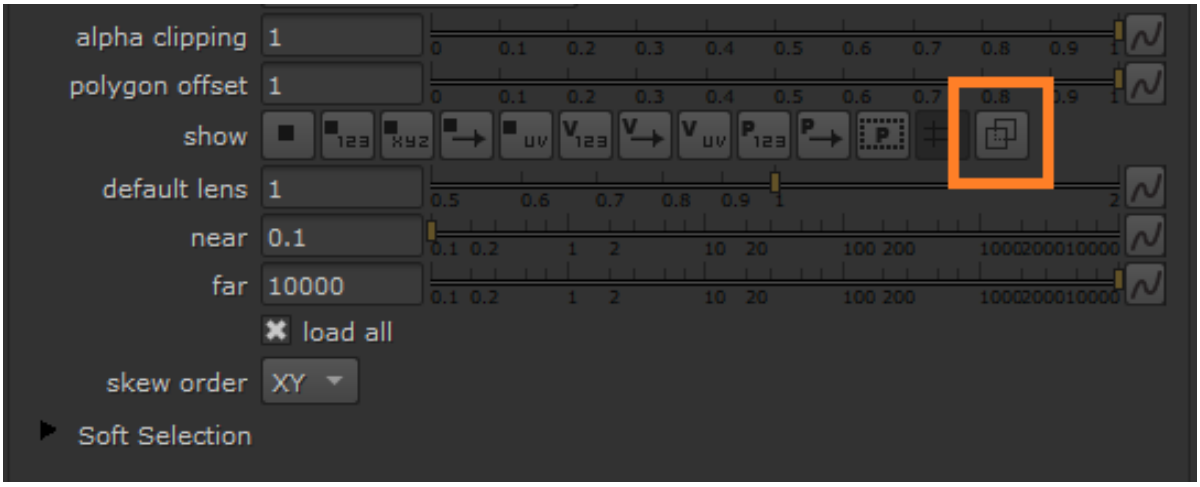
When selecting prims, you may need to select something behind another prim. In which case, Nuke provides two ways of drawing the selection outline:

- Maintain the complete selection outline for a visual reference of the selected prim, or
- prevent the selection outline from overlapping with the foreground prims, and occlude it behind them.

Selection occlusion allows you to customize selections for different scenarios, providing greater control over how you view the scene.

You can toggle the occlusion of the selection outline in the Viewer settings in the **3D** tab using the **object_selection_depth_tested** button.

Note: To open the Viewer settings, simply locate the Viewer node in the scene graph and double-click it.



The Object Selection Depth Testing button. Toggle this to occlude or reveal the selection border around prims.



Object selection depth testing **off**.

Part of the sofa selection appears over the figure in front.



Object selection depth testing **on**.

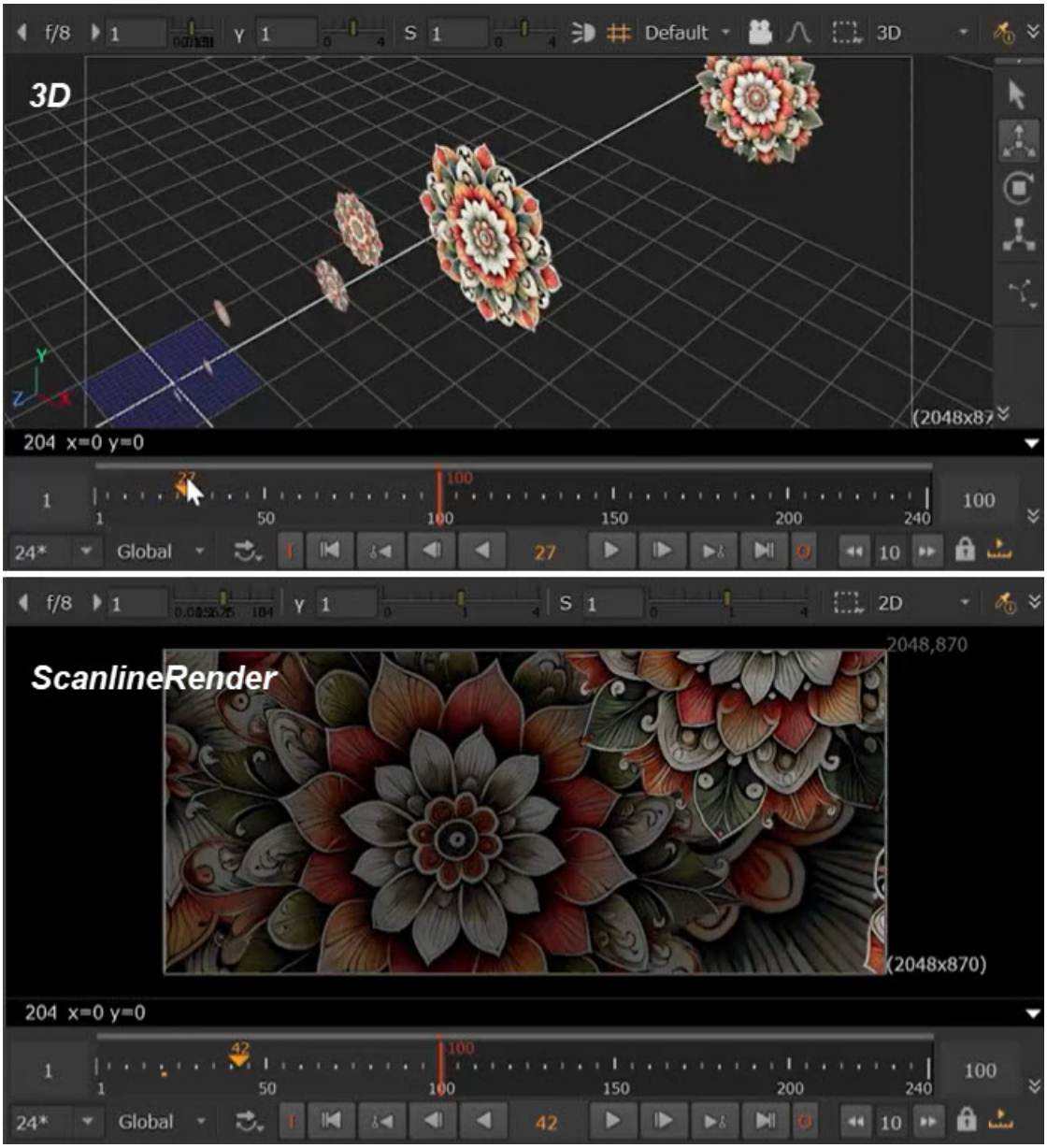
The sofa selection is occluded by the figure in front.

Hydra in the 3D Viewer

What is Hydra?

Hydra is rendering architecture, capable of processing large scenes with many changes. Hydra is used by Pixar for asset preview in many of its tools alongside USD.

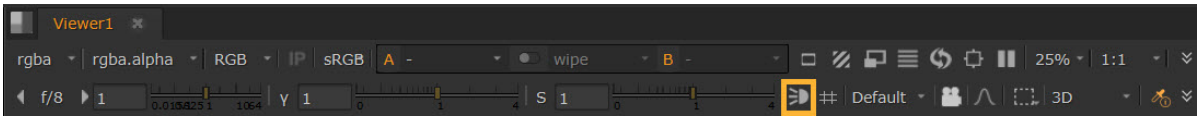
Nuke's New 3D System uses a Hydra-based 3D viewport, which uses **HDStorm** as a renderer. This means the 3D Viewer is consistent with other applications in your pipeline, such as Katana, Solaris, or USDView, as well as providing output that allows you to visualise more from your 3D viewer helping you make more choices before going to render.



Hydra and Lighting

One important thing to note is that Hydra needs a **light source** to display materials, so if you want to see materials then you can to enable the Viewer **headlamp**, or add lights to your scene. If you don't do either of these, Nuke has a backup default dome light that will allow you to see the materials until you add your own lighting.

Toggle the Viewer headlamp with this button:



Tip: You can also enable or disable the headlamp by opening your **Viewer** node **Properties**, selecting the 3D tab and using the **headlamp** checkbox. This panel also contains a range of options for controlling the way the Hydra viewer displays your renders.

To learn more about how to use the 3D Viewer, see Viewing Your Scene.

Render Delegates

If you have questions about concepts such as *scene delegates*, the *render index*, and *render delegates*, refer to OpenUSD's documentation on Hydra.

Note: In future Nuke releases, we'll be looking to update the Viewer to support additional render delegates so you can pick the right renderer for your viewport experience.

Configuring Your Scene

- Adding nodes to the node graph is the first step to setting up a 3D Scene.

See New 3D Scene Basics.

- The scene graph is one of the best ways to visualise your USD stage hierarchy.

See New 3D Scenes in the Scene Graph to learn about the various tools in the scene graph.

- A key part of setting up a 3D scene is to use paths and masks. Paths set where a prim lives inside of the scene graph hierarchy which is achieved using the Path knob. Masks define which specific prim or selection of prims to affect with your node using the Mask knob.

See Using Paths and Masks.

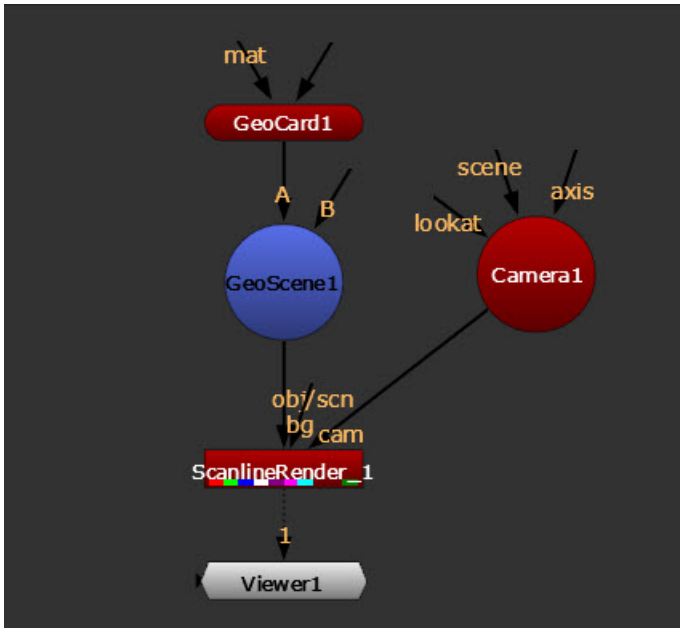
• Merging works slightly differently in New 3D scenes.

See Merging 3D Data.

New 3D Scene Basics

Similar to the **Classic 3D System**, there are some essential concepts to setting up your scene in the node graph. In the **New 3D System**, GeoScene node receives the output from the geometry nodes and sends the composite of those objects to the ScanlineRender2 node, where the output is converted back to 2D.

An example of a simple scene:



To bring in geometry, you can import it - see Importing into Your Scene. Or you may prefer to create geometry directly in Nuke - see Creating Geometry Prims.

Tip: For those who are looking to transition from the **Classic 3D System** to the **New 3D System**. See this helpful guide which provides a comparison of nodes, matching them by equivalent function and purpose - 3D System Nodes: From Classic to New.

The GeoScene Node

The GeoScene node combines all your elements in one 3D workspace - the geometry, cameras, lights, and materials, bringing them together into the scene. GeoScene accepts “axis op” inputs and is the only node that can do this. Essentially any circular shaped node is an “axis op” as it is based on an Axis - this includes Camera, Axis, DirectLight, SpotLight, EnvironmentLight, PointLight.

While GeoScene is the node where you combine all your scene elements together in one place, GeoMerge is what you can use to combine all your geo node data together and define how your layers are composed.



Note: GeoMerge has a variety of merging controls, but be aware that there are some USD behaviors which can affect how you use this node for merging. See more at Merging 3D Data.

The ScanlineRender Node

The ScanlineRender node allows you to toggle between a 2D and 3D view of the scene, and accepts **cam** inputs directly from the Camera.



Note: See more about this node at Rendering in the New 3D System.



Note: ScanlineRender's Classic 3D system equivalent is also named ScanlineRender. For New 3D System scenes, make sure you are adding the New 3D System node (found in right click node menu under **3D** > **3D**), not the Classic one (**3D** > **3D Classic**).

To Add a ScanlineRender Node

Connect the ScanlineRender **obj/scn** input to a GeoScene or geometry node.

Connect the **cam** input to the main camera.

Connect the optional **bg** input to composite a background image into the scene.

Press **Ctrl+I** (**Cmd+I** on a Mac) to open a new Viewer to display the output of the ScanlineRender node.

When an image is connected to the **bg** input, its resolution becomes the output resolution for the ScanlineRender node.

Camera nodes

Camera nodes may be connected to the GeoScene node or the ScanlineRender node. The camera connected to the ScanlineRender node is the camera used for rendering.

Drag an output connector from the Camera node to a GeoScene node or connect the Camera node to a ScanlineRender node's **cam** input.



Note: See Working with Cameras in the New 3D System to learn more about using cameras in your scene, as well as learn more about other USD Camera nodes.

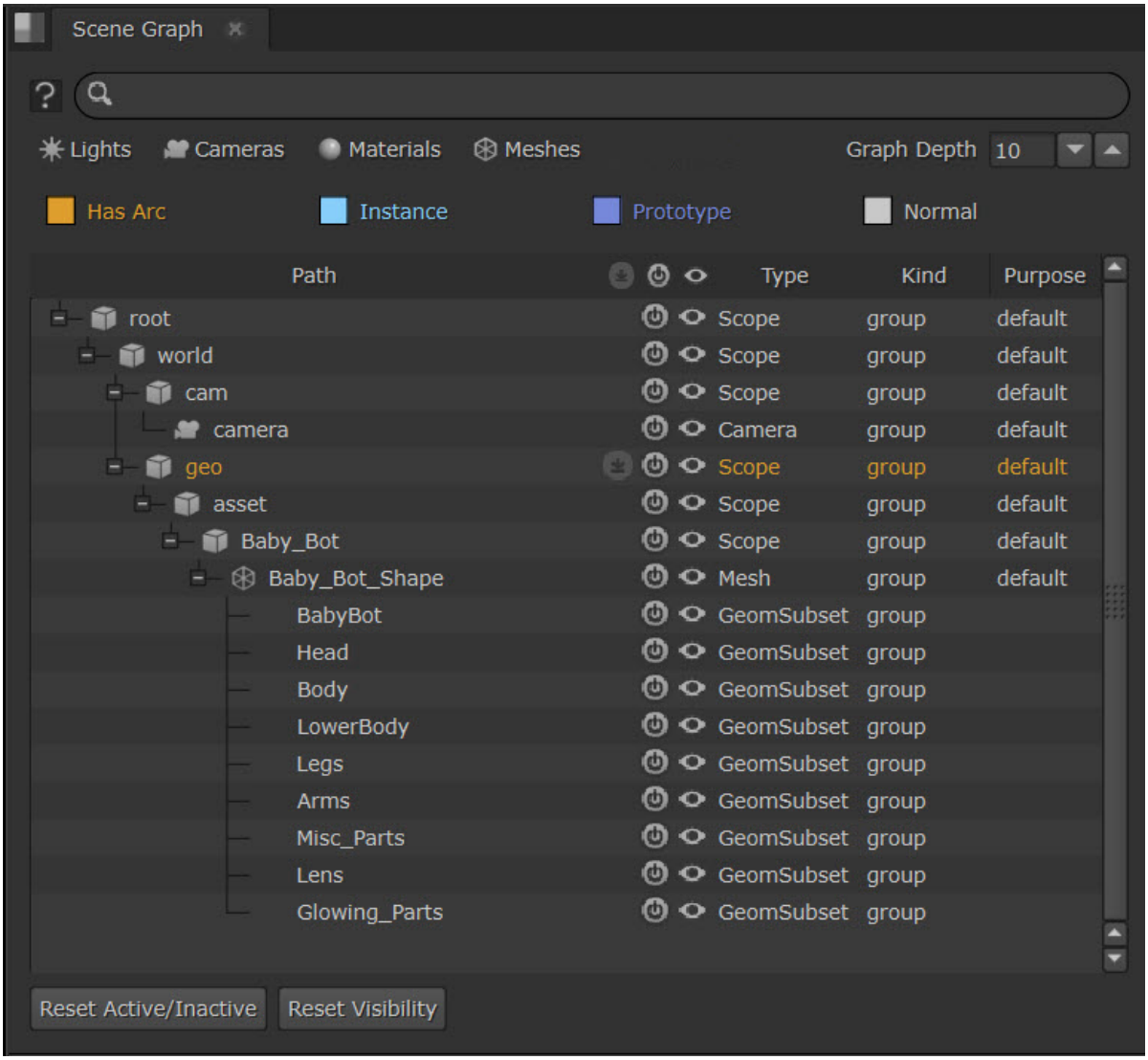
Any additional cameras should be connected to the GeoScene node. When you have multiple cameras associated with a 3D scene, you can switch between them by selecting the viewing camera from the dropdown menu at the top of the Viewer. See Look Through Camera to learn more.

New 3D Scenes in the Scene Graph

The **Scene Graph** offers an overview of the scene, and is a tool to view, navigate and manage large 3D scenes, allowing artists to work with more flexibility. Essentially, the Scene Graph is a list of everything within the 3D stage at the point where the Viewer is connected. Selecting entries in the Scene Graph highlights the associated objects in the Viewer.

Accessing the Scene Graph

In the menu select **Workspace** > **3D** to open 3D workspace where the **Scene Graph** will appear. Or go to > **Windows**> **Scene Graph** to open it directly.



Scene Graph Features

Hierarchical Structure

The Scene Graph uses a hierarchical representation of the USD stage, in contrast to Nuke's classic flat structure. This hierarchy covers prims such as geometry, lights, cameras, and materials, each with a unique ID or path, promoting a well-organized structure. The Scene Graph is a list of everything within the 3D stage at the point where the Viewer is connected.

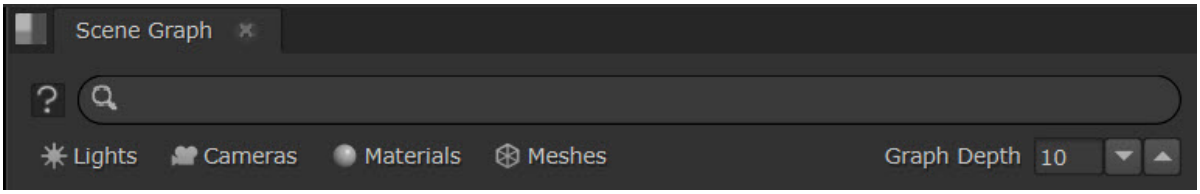
The Scene Graph has styling to help you visualize the hierarchy of your stage. Prims with Arcs, Instances and Protypes appear in different colors:



Geometry Highlight and Selection Synchronization

Synchronization between the Scene Graph and the viewer enables easy selection and navigation. Selecting an item in the Scene Graph highlights the respective item in the viewer and vice versa.

Searching and Filtering in the Scene Graph



Searching

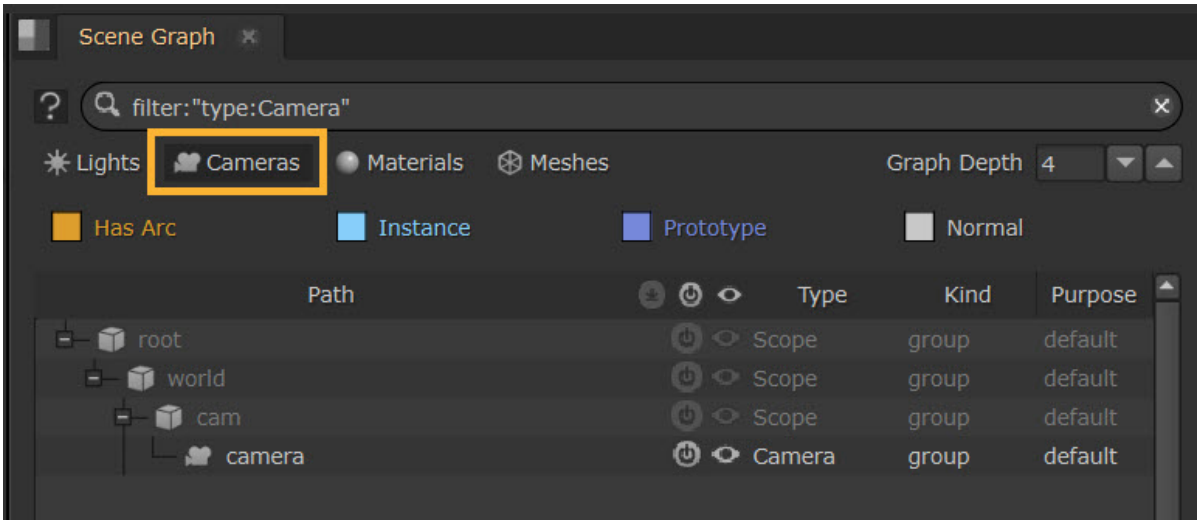
A search bar at the top of the Scene Graph lets you swiftly locate items in the scene, highlighting the search results in bold. Cycle through the results by repeatedly pressing the **Enter** key, or pressing **Shift** + **Enter** to cycle back.

Matches highlight in bold. The number of the selected results and the total number of search results are shown to the right of the search bar.

Filtering

There are a variety of preset filters which you can enable using the buttons just below the search bar - these show all **Lights**, **Cameras**, **Materials**, and **Meshes**. You can toggle them on and off, and use multiple at once.

For example, **Cameras** filter is applied to this stage. Other types of prims are grayed out.



You can also apply filters by typing the syntax in the Scene Graph search bar. For example, for filtering by Type for Cameras, the syntax would be:

Filter:"Type:Camera"



Columns in the Scene Graph

The Scene Graph panel contains various columns representing different data facets. These include:

- **Path**: Displays the hierarchical structure of objects.
- **(Payload)**: A USD specific concept to control memory usage by deciding the parts of a scene to load.
- **(Active)**: Allows the activation and deactivation of specific prims.
- **(Visibility)**: Offers control over the visibility of 3D objects in the viewer.
- **Type**: The data category of an element. For example, **mesh** represents meshes, and a **float** attribute stores numerical values.
- **Kind**: The organizational role of an element in the hierarchy. It is categorized as values like **component** for individual objects or **assembly** for a collection of components.
- **Purpose**: Categorizes components or primitives based on their role in different stages of the production pipeline. For example, An object with a **render** purpose would be included in the final rendering pass . A **default** purpose generally means that it is a primary element of the scene. It

should be taken into account in various processes, such as rendering, simulation, and interaction within the viewport.

Payload Management

Payloads let you choose which parts of a scene are loaded into memory. This helps you save on memory and speed up load times, which can be particularly useful for complex scenes with high levels of detail.

If you're using payloads, you can load individual items by clicking on the icon corresponding to each item in the payload column. Payloads are loaded by default, which is indicated by the light gray icon . The icon will switch to a dark gray icon  when the payload is unloaded.

Prim Visibility and Active Status

You can turn prims off (deactivate them) in the Scene Graph or simply not show them in the Viewer. Changing a prim's Visibility or Active overrides in the Scene Graph will only affect those states for the Viewer, not for any upstream nodes, e.g. Scanline Render nodes or Write nodes.

An 'on' icon  in the Active column indicates that the prim is active. You can toggle between the 'on' icon and a dot to choose if the prim is active or deactivated. When deactivated, whilst the slot for the prim is still technically in the Scene Graph, the prim is effectively removed from the stage. Children of a deactivated prim will not be shown in the scene graph.

An eye icon  in the Visibility column indicates that the prim is visible in the viewer. Toggle the eye to show/hide the prim. Unlike deactivation, the Scene Graph is not affected, only the Viewer.

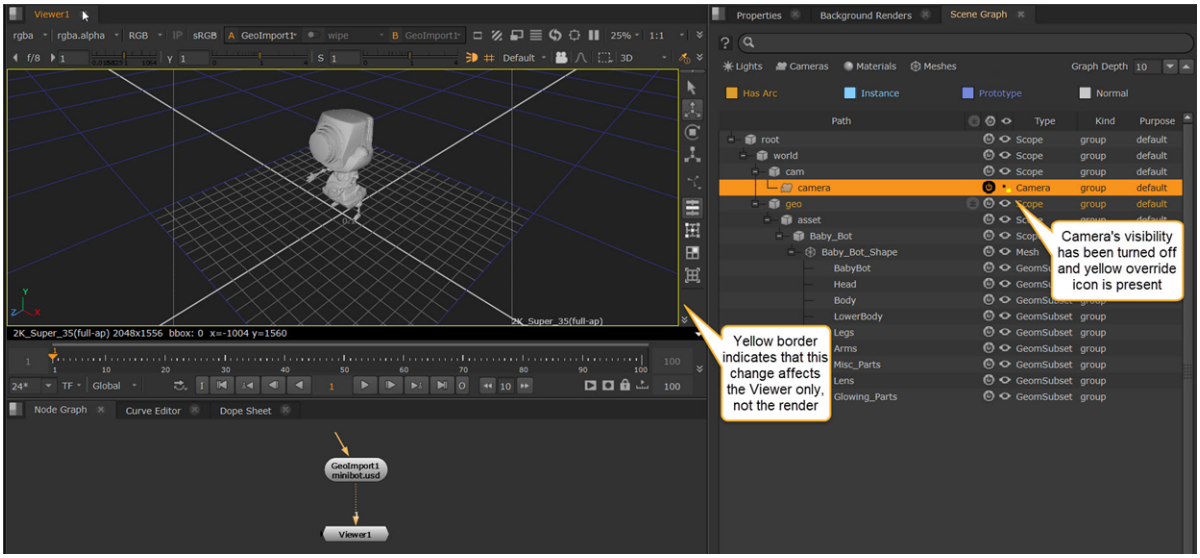


Note: Any changes to the prim Active and Visibility state which affects the Viewer only is signalled by a yellow border around the edge of the Viewer.

Right-click on a prim and use the context menu to **Show** or **Hide** the visibility of the prim.

Overrides: Indicators and Management

A small yellow dot on a visibility or active icon indicates that this control for the prim has been overridden, in other words it has changed from the state it was in when imported.



If you no longer require the overrides and their indicators, you can use the right-click context menu and select **Clear overrides.** All yellow dots then disappear to indicate that these are no longer overridden.

The Clear overrides option is not context specific and always applies to the whole scene graph.

Tree Operations

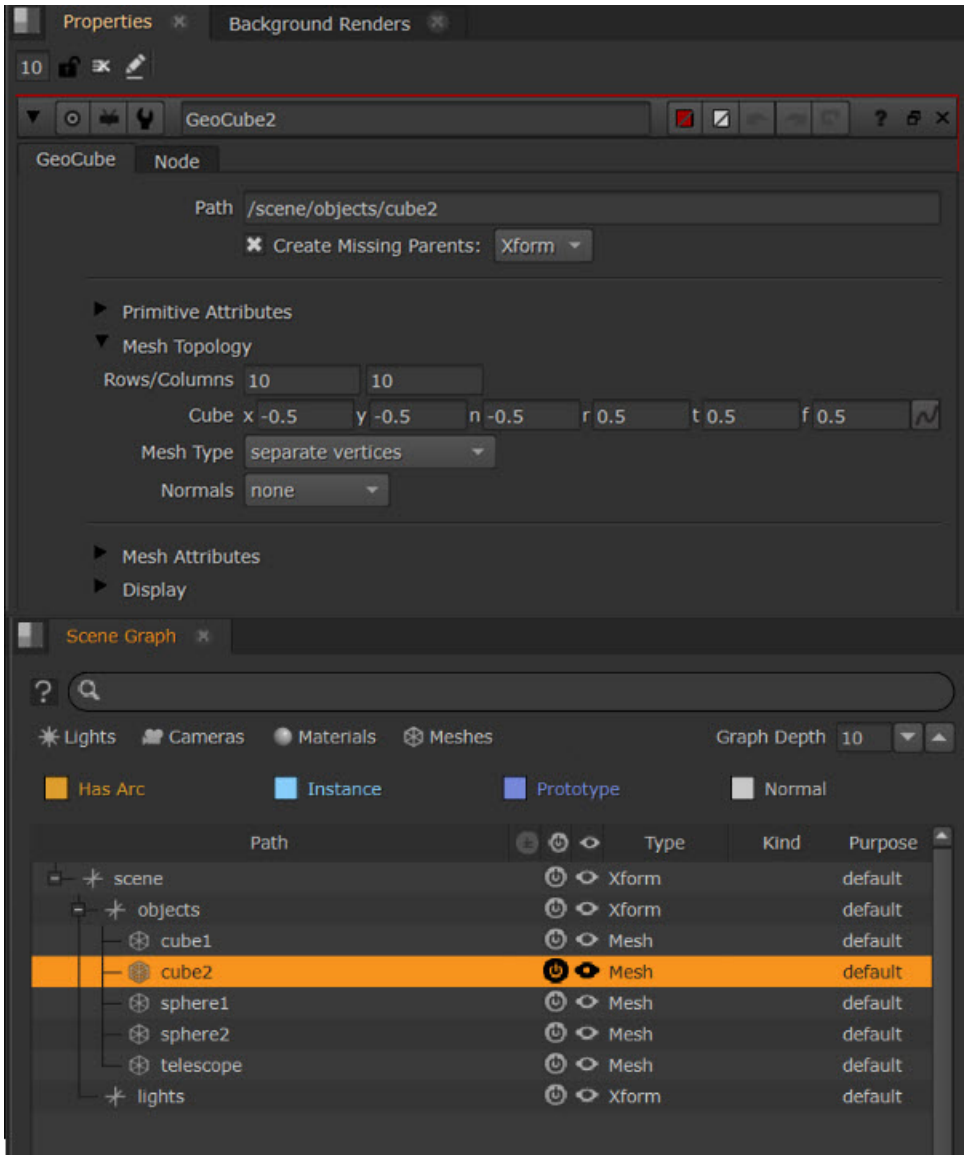
Expand and Collapse: Use the plus and minus icons next to the prim name to expand or collapse the scene structure. Hold **Shift** when expanding to also expand all nested elements within a parent.

Multi-Selection: Select multiple items using **Shift** to select a range or **Control** to multi-select individual prims.

You can also use the right-click context menu on a prim to choose **Collapse all** or **Expand all.**

Using Masks and Paths in the Scene Graph

Path and Mask knobs on certain 3D nodes can be used to place items in the Scene Graph hierarchy. **Path** knobs are located on creation nodes like GeoCube, allowing you to specify the desired position in the scene's hierarchy.



Modifier nodes such as GeoTransform have a **Mask** knob, to define the prims in the scene that you want the node to affect.

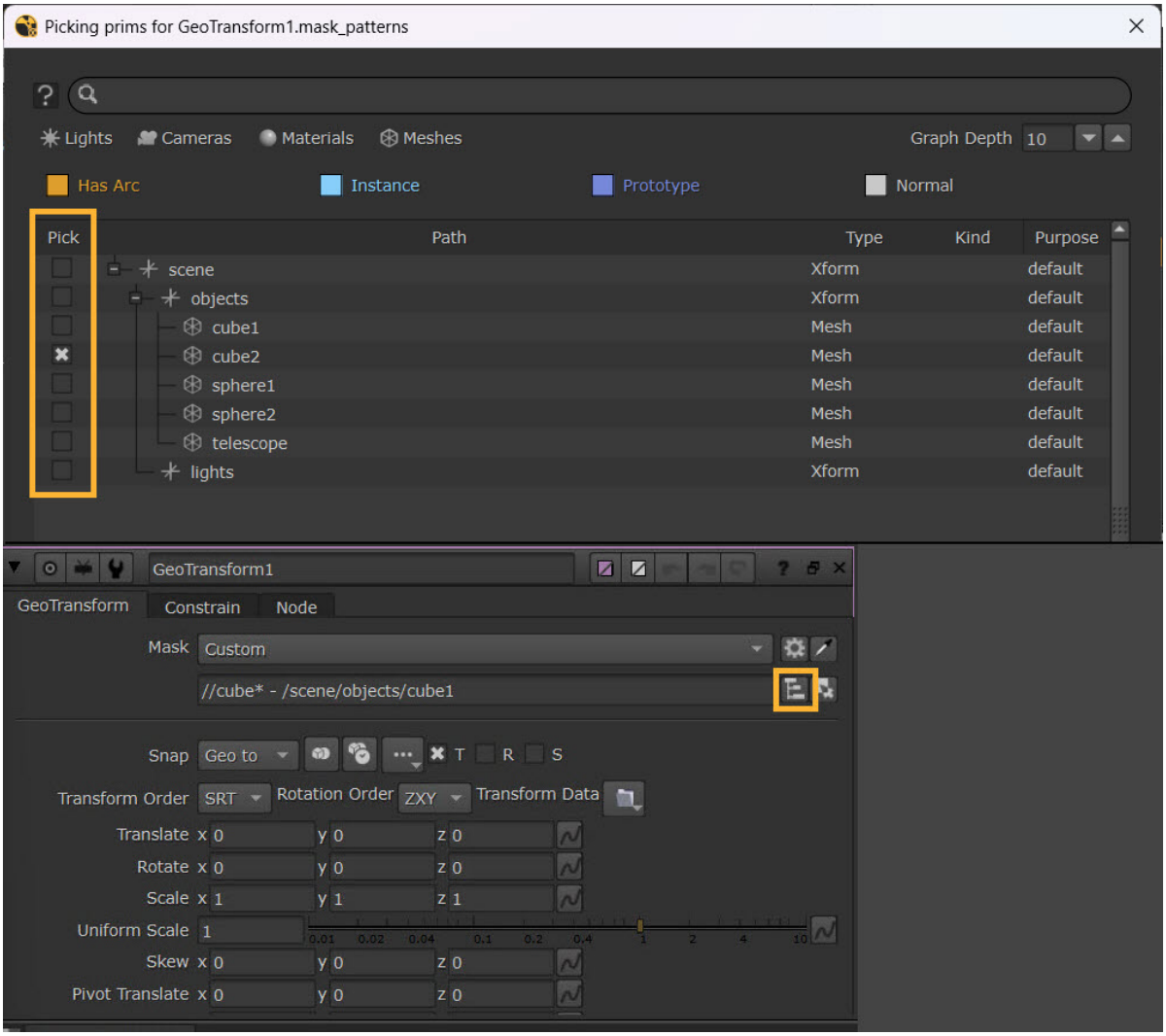
Note: For more information see Using Paths and Masks.

The **Mask** knob is a dropdown with the most common mask actions displayed, such as All Cameras and All Lights, to make it easier to select the most appropriate mask for your needs. The default is All roots or All meshes, depending on the node, which selects the top most level Xforms in your scene graph.

You can also set Mask knobs by:

Using the Scene Graph picker
Copying the prim paths from the prims you currently have selected
Select prims in the Viewer using the eyedropper tool
Drag and drop from the scene graph onto the knob
Type into the path(s) directly

The Scene Graph picker button next to the **Mask** knob opens a pop-up scene graph, showing the hierarchical scene at that point in the node graph. The pop-up shows the scene's hierarchy from the node and everything upstream.



Use the **Pick** column to select items.

Making Selections

Make your selections using the checkbox in the left **Pick** column, or double-click on a prim to check the box automatically. Press **OK** to confirm the selection.

The selection(s) are input into the mask knob. You can select individual or multiple prims using common multi-select hotkey modifiers such as **Shift**+click.

Some nodes require a single selection, such as in the import prim path knob on Camera nodes. In this case, the pop-up scene graph will offer a single selection only. Double-click the chosen camera, to select it as input to the mask knob, which closes the pop-up scene graph automatically.

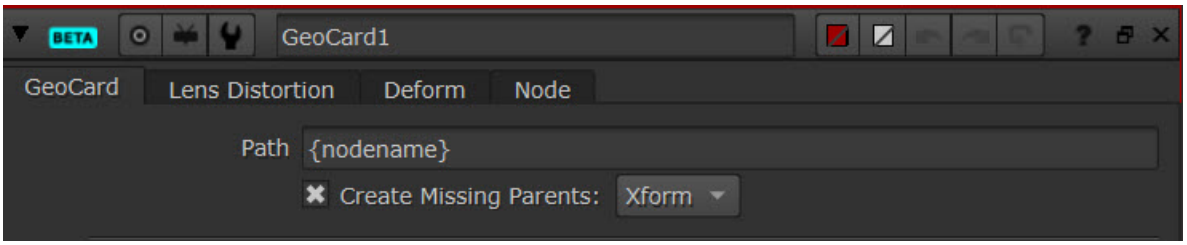
Tip: The pop-up scene graph uses the same search feature and filtering system as the main scene graph. Filters are automatically applied on certain nodes. For example, when choosing a prim path for a Camera node, only Cameras will be filtered, and only a single selection is available.

Using Paths and Masks

In the New 3D system, **Path** and **Mask** knobs are available for nodes that create or import geometry, and allow you to specify where imported objects live in Nuke's Scene Graph based hierarchy. This guide explains how to set paths and masks, and works through some more advanced workflows involving CEL, which may help with organizing your 3D scene.

Path Knobs

Path knobs allow you to specify which parts of your Scene Graph you want to affect. When inserting a 3D node to modify geometry such as GeoCard, the Path knob defaults to **{nodename}**. This expression is created by Nuke, to make sure that unique ID is always created and you don't have to worry about populating this yourself if you don't want to.

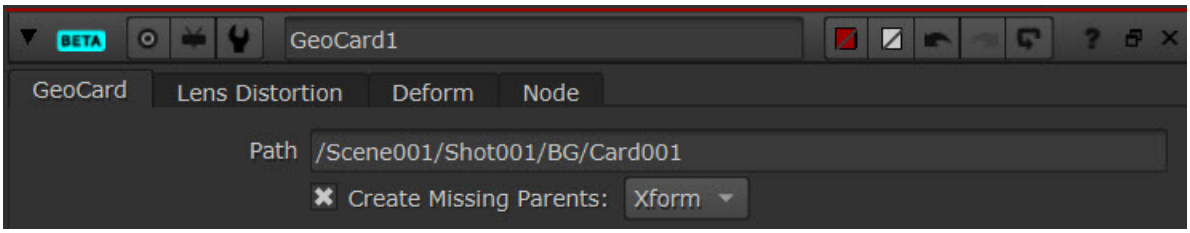


Tip: Using the default **{nodename}** can help speed up geometry creation.

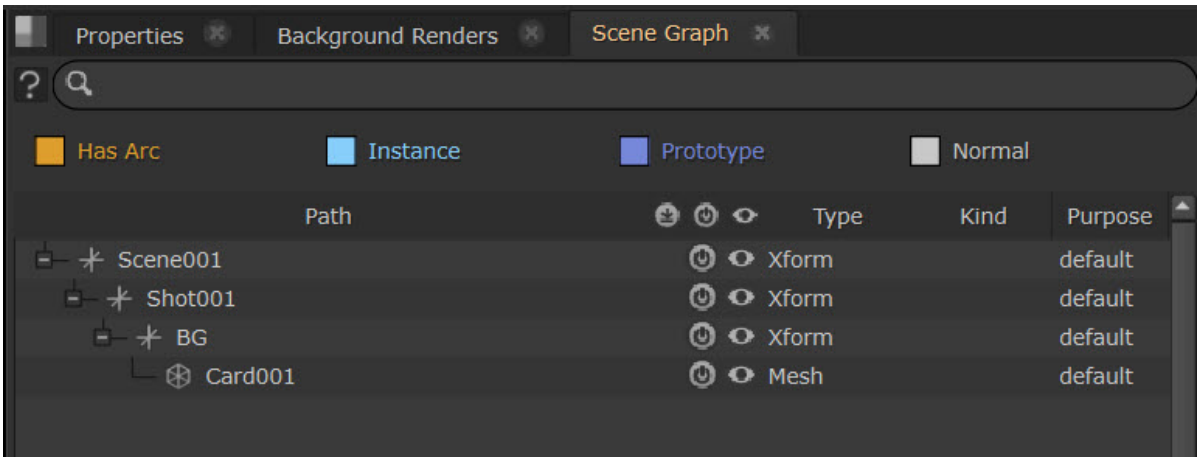
Custom Paths

If you would like to customize where your geometry lives in your Scene Graph, you can do this by typing to edit the **Path** field.

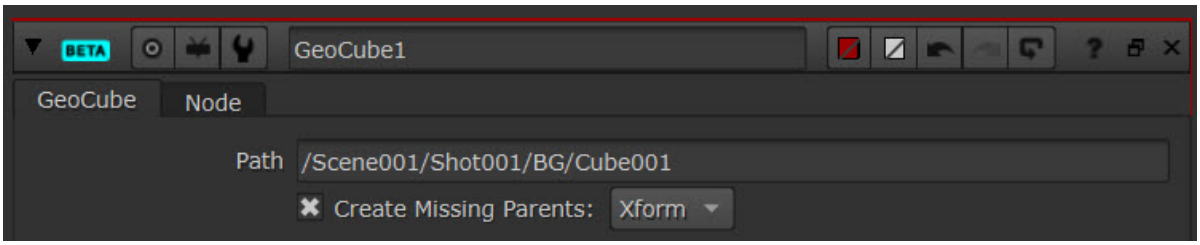
For example, here we have added a couple of higher levels in the hierarchy above the Card which has been named Card001.

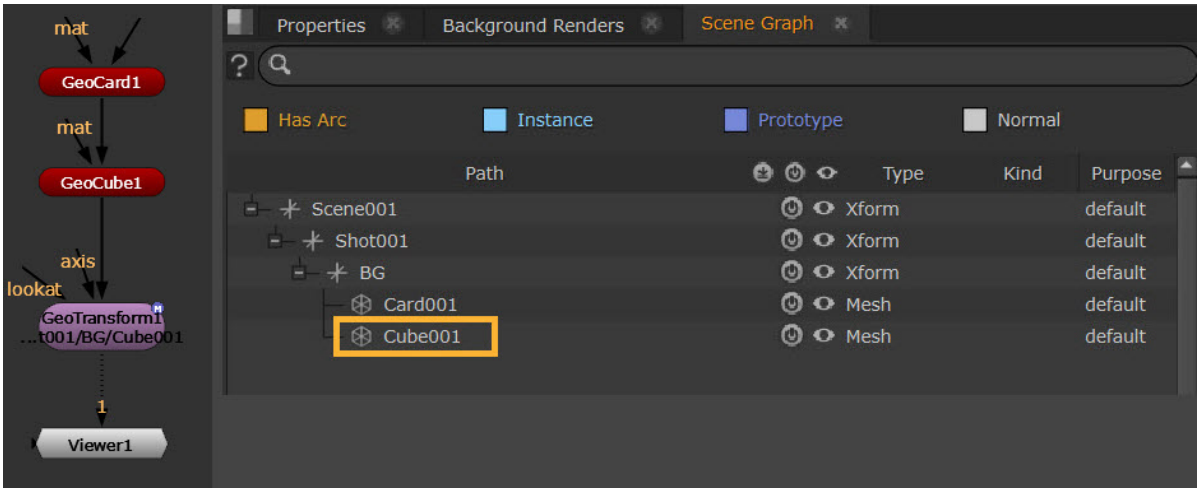


This hierarchy is automatically updated in the Scene Graph and any missing parents are automatically created.



If you then create a second object like a Cube, and edit the Path to be under the same hierarchy, you'll be able to connect it directly under the GeoCard node without the need for a Merge node, similar to how Nuke's 2D nodes function.

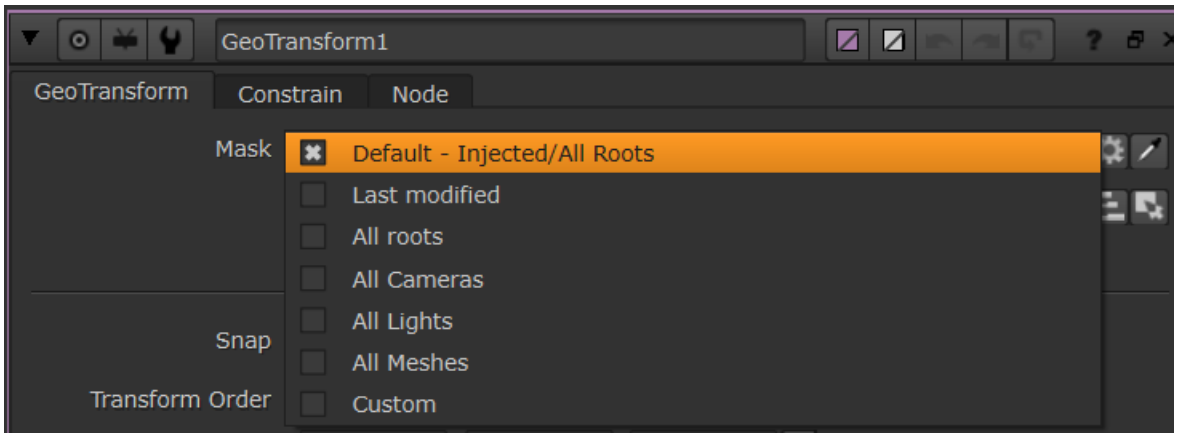




Mask Knobs

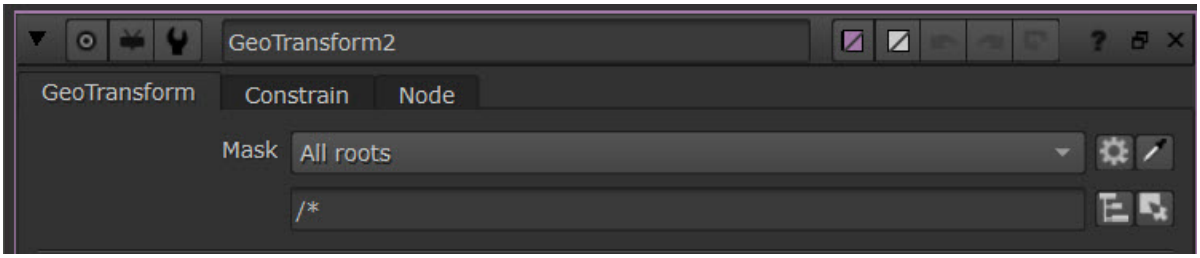
You can find the **Mask** knob in modifier nodes such as GeoTransform. The Mask knob is a dropdown, allowing you to select the prims you want to affect. You can choose the most appropriate mask from a range of common options or choose your own Custom mask.

You may not necessarily need to change this as the **Default** option selects the default prim(s) for the node - this will be the injected mask prim paths if there is an upstream node with **Inject Mask** enabled, otherwise the default mask for this node type is evaluated.



All roots option is a convenient way to have this node just affect the top most level Xforms in your Scene Graph.

For example, if you have both a Card and a Cube, this expression is telling the GeoTransform node to select all objects in Scene001:



All Meshes, **All Lights** and **All Cameras** works in the same way, but with added specificity on the prim type for those nodes related to certain types of prims.

Tip: The All roots and All Meshes options can help to avoid issues with ordering in the node graph as it selects all objects in the hierarchy.

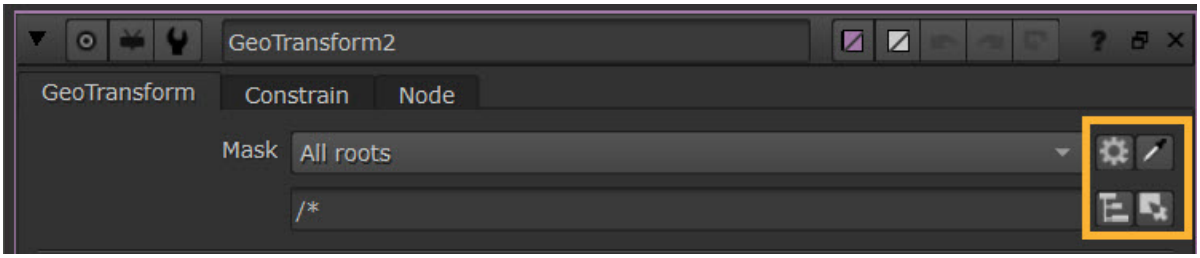
Underneath the Mask dropdown, the mask field automatically updates based on your selection in the dropdown, helping to show how the mask is being created with expression language, and allowing for easier manual editing if needed.

In this case, the field underneath displays **/*** as this is the equivalent expression for **All roots**. This expression essentially says to select everything from the root of the scene graph down and therefore we are able to modify all nodes in the chain.



Note: We could also manually add objects to the Mask path by dragging and dropping them from the Scene Graph, picking from the Scene Graph or Viewer, or copying your current selection. For more about this, see Using Masks and Paths in the Scene Graph.

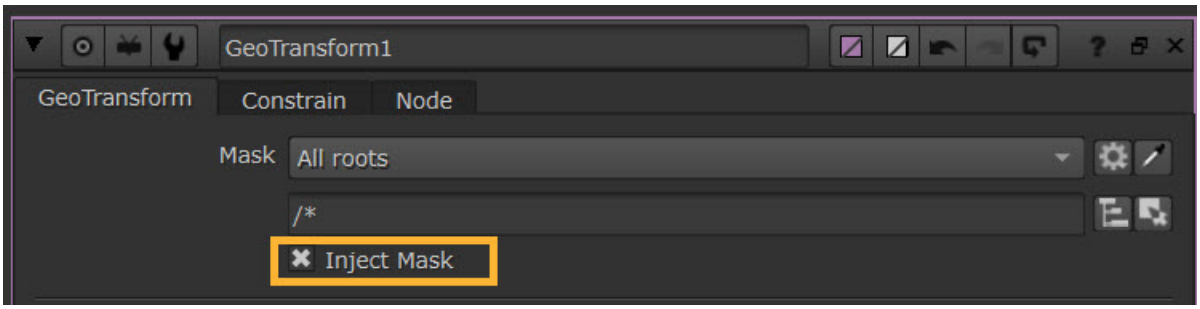
To help with setting up masks, there are several Select Path Mask buttons to the right of the mask field.



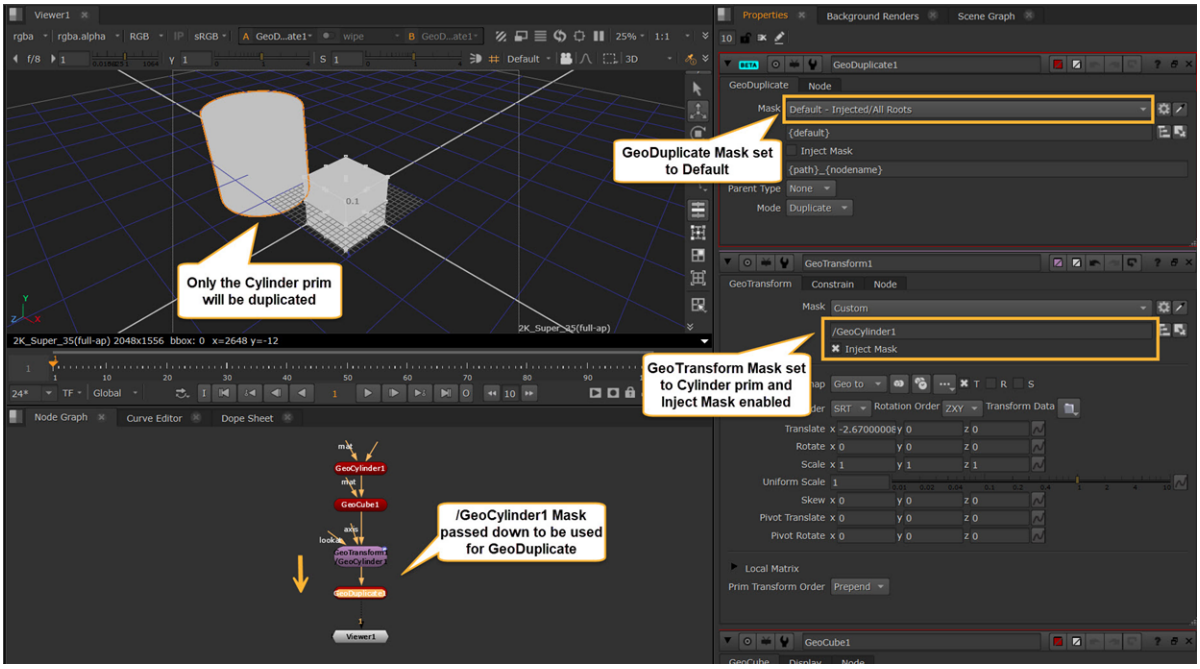
Copy prim paths from the prims you currently have selected
Select prims in the Viewer using the eyedropper tool
Using the Scene Graph picker, which opens a pop up scene graph
Select all the prims of your mask in the Scene Graph and in the Viewer, helping you to understand what your mask is acting on

Using Masks Downstream

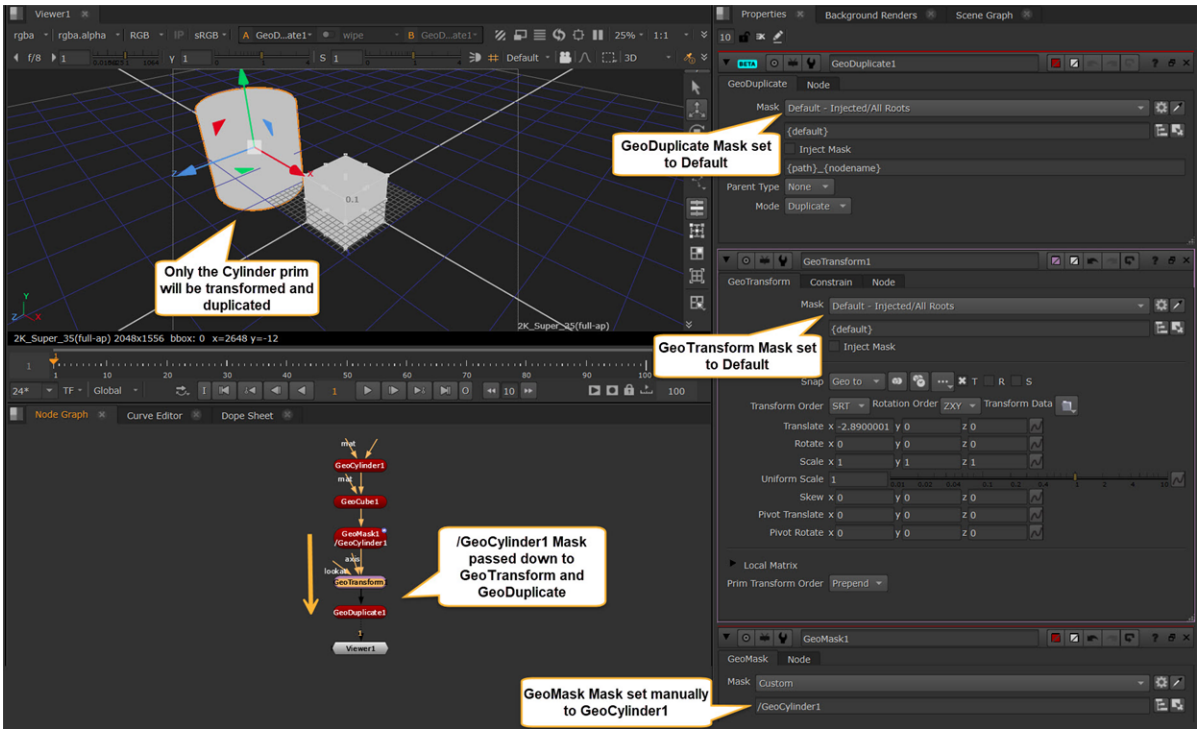
- You may wish to reuse the same masks further down the node graph. You can enable the **Inject Mask** checkbox to push the mask in the current node is to the geo nodes below it that have the **Default** mask set.



In this example we have two prims in our scene, a Cube and a Cylinder. We have set the GeoTransform Mask to act on the Cylinder only and want to reuse this Mask for a GeoDuplicate node downstream:



- GeoMask node is an alternative way to set a mask downstream. For GeoMask, the Mask should be set manually. When connected to other nodes downstream, the Mask is pushed onwards to all nodes below it.



Tip: The prim your Mask is directed towards will show on the GeoMask node label for easy identification.

Note: You can then clear this at any point by using a GeoClearMask node.

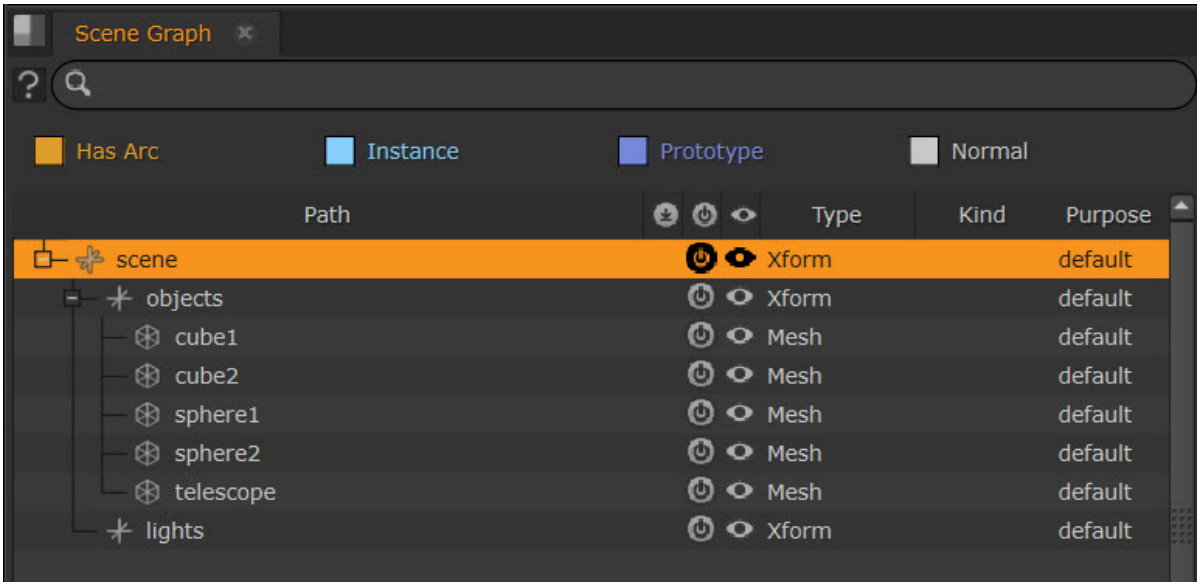
Advanced Path and Mask Workflows

You can create more complex selections with these knobs using expressions and tokens. The knob expression language is based on a simplified version of the Cell Expression Language (CEL) which is what we use in our sister product Katana.

For example, GeoTransform is a geo modifier node, so the Mask knob defaults to the option **All roots**. With the expression **/***, all we're doing is using **/** to say go to the root of the scene graph hierarchy and then ***** to indicate we want to select everything underneath the root.

We can expand on this selection by using ***** to indicate select everything with a given text string.

For example, taking a simple scene with various types of meshes:



We could edit the mask field in a variety of ways:

Expression	What it does
/scene/objects/cube1	Selects the single object “cube1”
/scene/objects/cube*	Selects all the cubes in the “objects” folder
//cube*	Selects any Leaf node starting with “cube”
/scene//cube1	Select anything called “cube1” inside of “scene”



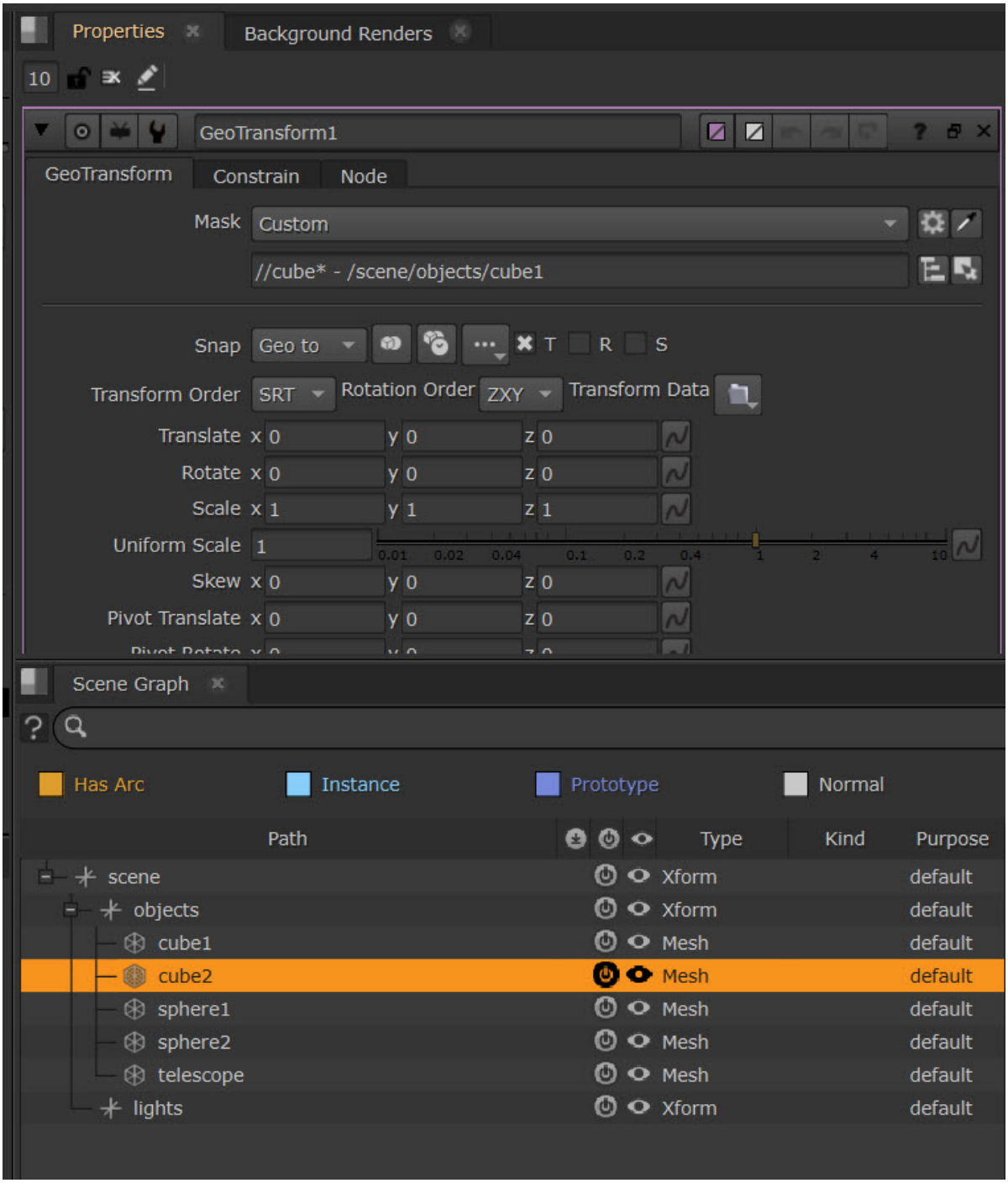
Note: These edits will set the Mask to **Custom**.

You are also able to add and subtract and add expressions or paths to one another.

For example:

Expression	What it does
//cube* - /scene/objects/cube1	Selects all the cubes except for "cube1"
//cube* + //sphere*	Selects all the cubes and all the spheres
//*telescope*	Selects all Leaf prims with telescope in the name
//*telescope*//	Selects all prims with telescope in the name plus their descendants
/scene/$lights	Selects all prims in the collection called "lights" on the object "scene"

For example, filling in this expression means "selects all the cubes except for "cube1". Clicking the Select Mask Path button would only select cube2 in the Scene Graph.



Merging 3D Data

There are some differences when working in a USD scene graph based system compared to the Classic 3D System which require some changes in workflow.

If we take two prims and combine them using a GeoMerge, this works as it always has and allows you to modify both prims at once. You don't have to change anything on the node itself as the default options should reflect the classic behaviour.

Note: To learn about the different types of merging in the GeoMerge node, see Merge Mode Options.

However the difference in behavior occurs when trying to make modifications to separate prims within the same stage. For example, if you separate out a prim from the stage using GeoPrune or GeoActivation, transform that separated prim and then try to merge it back via GeoMerge, you just get the separated prim and not the rest of the stage. This is because USD is merging the Activation and Visibility states of the stage. During the merge the A and B pipes compare to see who has the strongest opinion and show that pipe's Activation and Visibility settings.

If you want to work with a prim in a stage but not have it affect the rest of the scene, then you have various options:

- Use the Mask to only specify that prim in the modification. You can use GeoMask and GeoClearMask workflow, or the **Inject** workflow to pass any masks downstream. See Using Paths and Masks.
- Separate that prim out from the stage and bring it in as a new prim via GeoReference. See Referencing with GeoReference.

Note: It would only really be viable to use GeoPrune for merging if you don't need the rest of the scene at all. The GeoPrune node lets you deactivate a prim or selection of prims in your stage, so you can work on what's left.

Warning: By default when you create the GeoVisibility node it will hide everything in the stage, which means you then need to disable the node in order to pick intuitively.

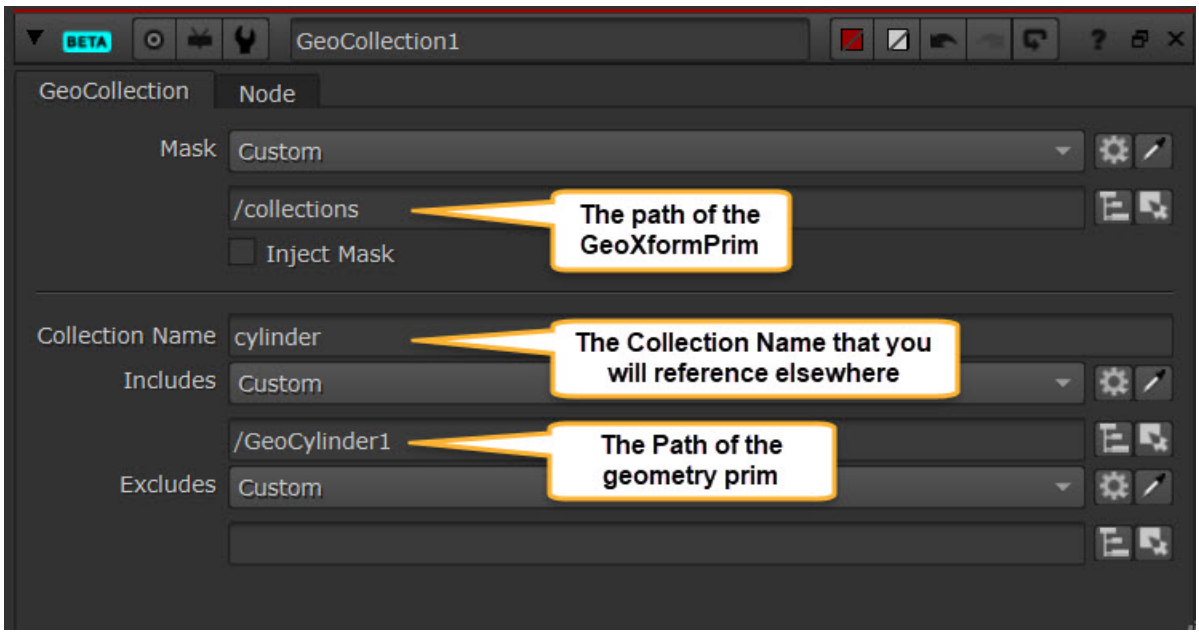
Using Collections

If you want to organize your scene by placing prims in collections, you'll need the GeoCollection node. This node is used to specify paths of prims to include, and can then be used as a mask input for future nodes down the stage.

For example, you might want to create collections for all the lights or cameras in the scene or any prims that have shadows cast on them.

In this example, we have just chosen one specific geometry prim to include in a collection.

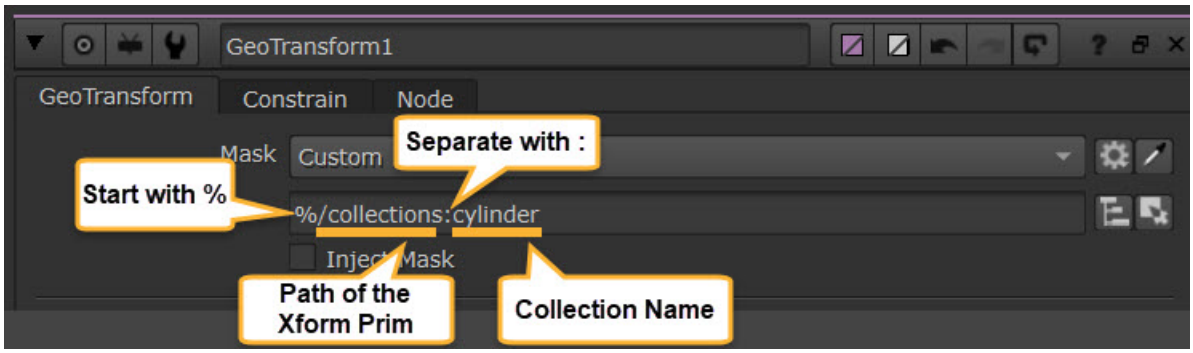
1. Add a GeoCollection node to your node graph.
2. You'll need to have created a Xform or Scope to point to (you can use GeoXformPrim, or GeoScopePrim node). Then on the GeoCollection node, set your Mask to this and this is where your collection will be saved to. In this case we've created a GeoXform and its prim path is **/collections**.



Note: Learn more about Xforms and Scopes at Creating Scope and Xform Prims.

3. Give the collection a name under **Collection Name**. It's worth keeping this short as you'll need to reference this elsewhere.
4. With the collection set up though you can now specify what prims you want in the collection. You can use the **include** knob to add prims to the collection and the **exclude** to remove prims from the collection. Typically you'll just leave exclude empty, but if you are using the expression language to make complex selections across a large stage, then having the exclude can be useful.
5. Now you have set up your collection, you can reference the collection in other nodes. For example, if you wanted to transform the prims in the collection, you'll need to specify the Collection Name in the **Mask** knob of a GeoTransform node.

This needs to include some syntax and reference the above Xform beforehand.
For example, %[GeoXform]:[CollectionName]



Warning: Using the GeoTransform node directly with the Scope or Xform you created in the node graph will not work with the collection set on it. You always need to reference the **Collection Name**.

Working with Cameras in the New 3D System

To work with cameras in the New 3D System, you have several options as an artist.

1. You can use the Camera node to create a new camera, or bring in camera data from a file or from your scene:
    - The Camera node creates 3D data in the same manner to the classic Nuke Camera node, which allows us to use it as a way to connect the new 3D system to all the nodes across Nuke that use 'cam', 'axis', or 'lookat' inputs.
    - The Camera node creates cameras that are unique from other DCCs, as it allows us to provide functionality like spherical and cylindrical projection modes that are unavailable outside of Nuke.
    - While this node does not directly create USD data by default, if you want to add this camera to your USD stage, it can be connected via GeoScene, which injects it into the scene as a new

USD camera prim via the **Output Prim Path** knob. This essentially creates a duplicate camera in your scene.

- When **Live Read** is enabled, data can be pulled live from a USD file allowing you to simply pass the data through the node, or to go in and start editing the data. Any edits you might make to the camera being fed into the Camera node, say via the **scene** input pipe, will be automatically passed through the Camera node.
- This is perfect for when you want to use an imported camera that exists in your scene graph to work with the Project3DShader, or even if you have imported a render camera and want to simply pull it out of your 3D node graph network and connect to ScanlineRender **cam** input right at the bottom of your node network helping to keep your script much tidier.

See Using the Camera Node section below.

2. If you are working with a USD scene and wish to create a new camera, it may be more helpful to use the GeoCamera node:

   - The GeoCamera node differs from Nuke's Camera node in that it creates USD data directly.
   - For example, creating a camera with GeoCamera means that you can use the GeoTransform node and move everything in your scene, including the camera data you created. This means you no longer have to expression link transformation data to an Axis node connected to the axis input of a Camera node to have it transform with the rest of your scene.
   - The created cameras are USD cameras with the USD schema and don't have the ability to connect to all Nuke nodes with a **cam** input.

3. If you are bringing in a scene with a USD camera already in it, you can use the GeoEditCamera or GeoCamera node to edit an existing USD camera.
   (These two nodes have the same functionality. The only difference is that GeoEditCamera defaults to **Edit** Mode, and GeoCamera defaults to **Create** Mode. Both Modes are available in both nodes.)

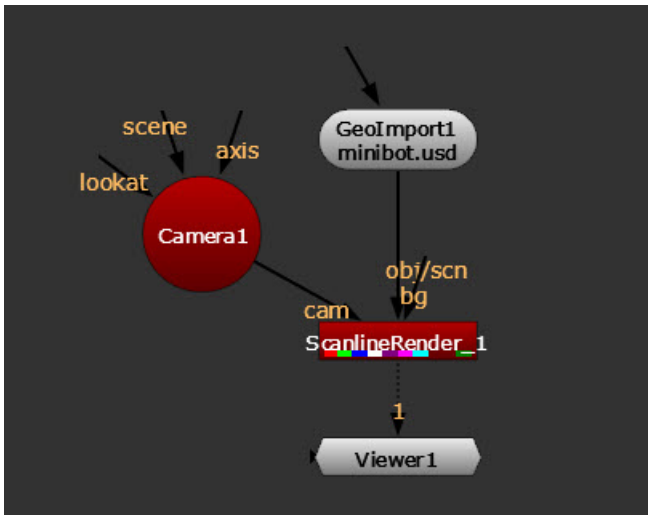
   - There may be instances where you have imported a camera into your scene and simply want to directly edit its values so that these edited values can flow to all other nodes across your node graph (including Camera nodes pulling this data in). This is where the GeoEditCamera allows you to load an imported camera from your stage and edit the values of that USD camera directly inside of Nuke, so you no longer have to duplicate it in a Camera node.

See Editing USD Cameras section below.

Using the Camera Node

Creating a Camera

If you wish to create a new camera in your scene, start by adding a Camera node.



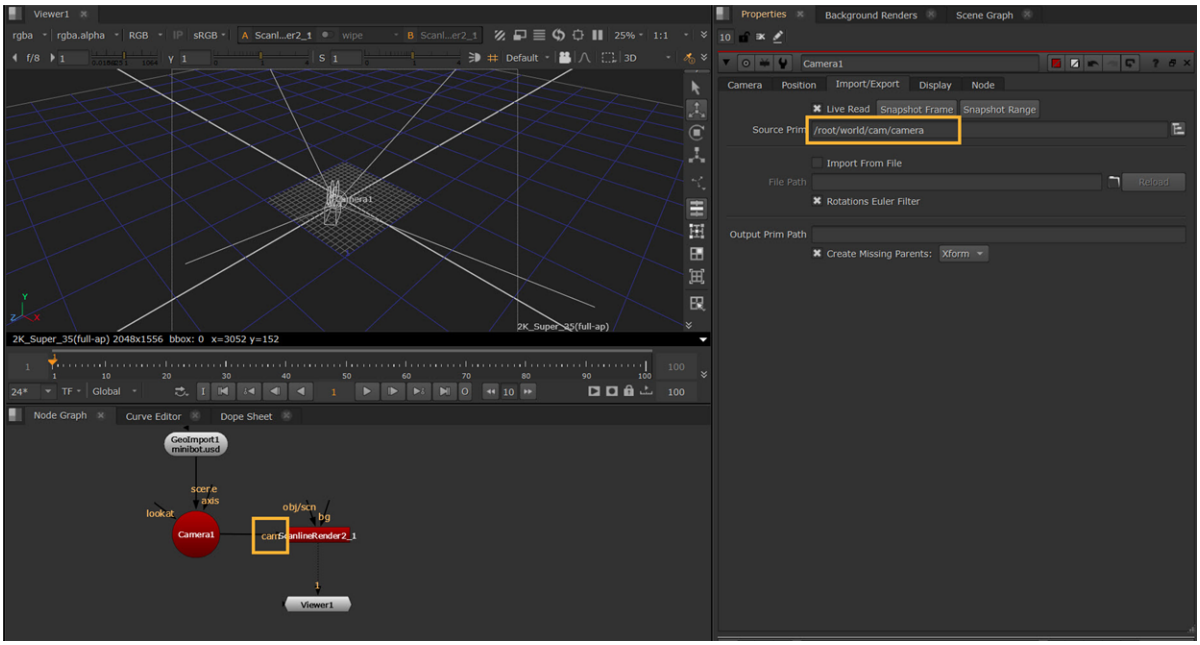
Cameras can then be connected to any node with a **cam** input, such as ScanlineRender. The camera connected to the ScanlineRender2 node is the camera used for rendering.

Bringing Data In

To bring existing camera data into the node, you can load a camera from file or from prim path:

- To load from a file, go to the **Import/Export** tab, and check to enable the **Import From File** knob. This allows you to then load a camera from a USD or Alembic file.
- To use an existing camera prim, use the **Source Prim** knob and pick from your scene.

For example, here we are picking an imported USD camera from **Source Prim**. Bringing data into the Camera node allows you to connect that camera to any node in Nuke with a **cam** input, such as ScanlineRender. The camera connected to the ScanlineRender2 node is the camera used for rendering.



Using the Camera

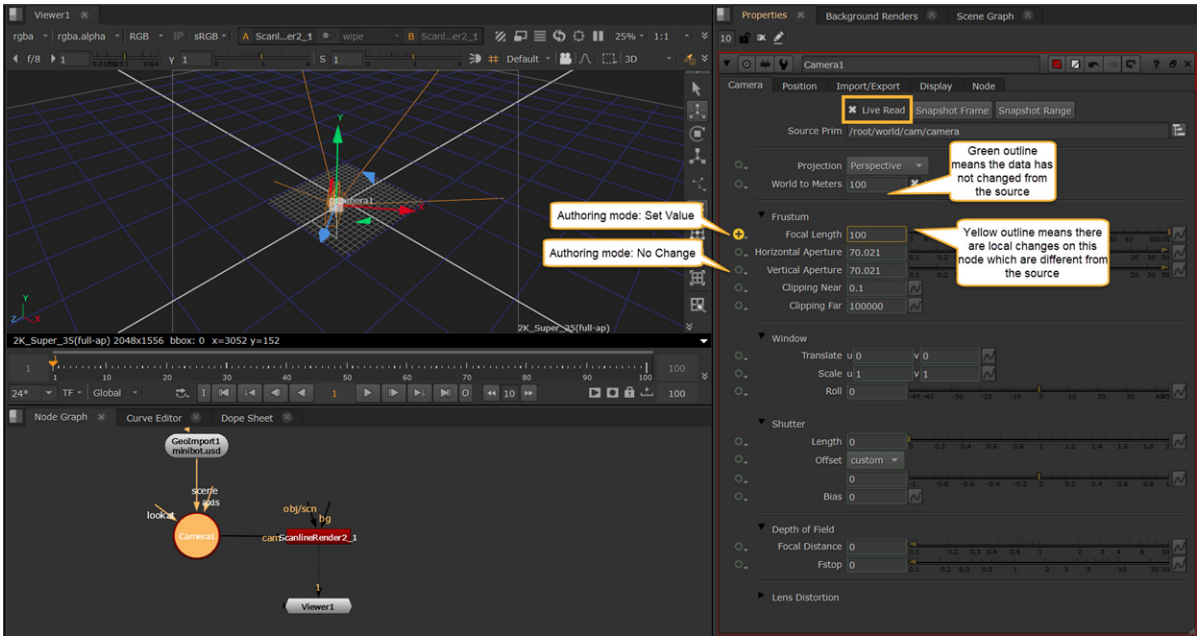
Once the camera is created and any data is brought in, the Camera node allows you to work with it and make any edits in Nuke.

If you have the **Live Read** knob enabled, any updates you make to the Camera knobs will be updated live. The Camera node is then able to be connected to other nodes where you want to use that data via a **cam** input such as ScanlineRender2.

If you are using an imported or referenced camera, and have enabled **Live Read**, you'll notice that the knobs have a **green** outline. This helps you to see which data is being read from elsewhere in their script and which data has been changed directly on the node itself.

- The **green** outline means the data has not changed from the source.
- The **yellow** outline means there are local changes on this node which are different from the source.

The authoring mode icons to the left of each knob allow you to control this. **Set Value** when there are local changes, and you can always revert back to **No Change** if you wish to reset to default (from source). Learn more about Authoring Modes



When **Live Read** is enabled, there is also an option to **Snapshot Frame** and **Snapshot Range** while using the Camera. This will essentially take any camera that is being live read into the node and bake the values as a local copy to that node, allowing you to have a permanent snapshot of the data at the moment in time.

Xform Syncing Options

When importing camera data, you may be importing a camera which has transformation data on the camera prim itself, but also has transformation data on the Xforms that the camera lives under. These knobs allow you to determine how you work with that data.

Note: An "**Xform**" prim stores transformation data that applies to its child prims.

- **Parent Xform Sync** contains the parent transformation data i.e. that extra Axis inside the Camera properties and will automatically update the transformation values based on changes of transformation in the stage. For example, if you move the entire scene using a GeoTransform, these values will be updated while the Cameras local transformation values stay the same. By unchecking this you can break that connection and moving the scene no longer also moves the camera but you still have the transformation data from before the desync baked into the knob values.
- **Local Xform Sync** allows you to sync or desync the local transformation data. When enabled this essentially just loads an imported camera's transformation and animation data into the local transformation knobs. Whereas disabling this allows you to edit these values yourself, so if you have a camera that you've imported with animation data, you can desync the values and then move the camera as

you need on a per frame basis. Essentially you have the ability to directly edit the camera's transformation data.

- **Combine With Parent** If enabled the parent prim xform is combined (concatened) with the local prim xform and placed in the local-xform TRS knobs. When not enabled the parent prim xform is extracted separately and placed in the local parent xform TRS knobs, while the local prim xform is extracted and placed in the local axis xform TRS knobs.
- **Rotations Euler Filter** checks for possible euler flips in the decomposed rotation curves, and removes them.

For more about transformations, see Transformations in 3D Scenes.

Other Operations

- You may also wish to use the **Projection** options when working with cameras. To read more about projections, see Projections in the New 3D System.
- To find constrain options, these are on the **Position** tab. To learn about constraining, see Constraining Prims.

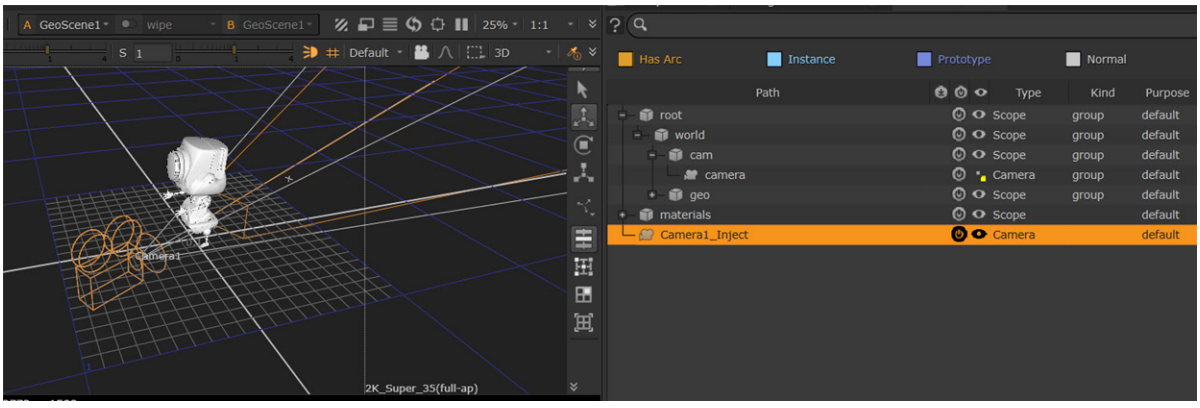


Note: Full controls and descriptions can be found in the Node Reference Guide - Camera.

Camera Data

You can push a Camera created in Nuke to your USD stage using the **Output Prim Path**. This creates a duplicate camera that is a USD prim in your stage under a path you have chosen in the **Output Prim Path**. If this path is left blank, it will create one for you when the camera is connected via GeoScene, and will be named *[CameraName]_Inject* by default.

Camera injected into stage:



By default when you connect a Nuke Camera to a USD stage using the GeoScene node, this is the prim path that will be used to generate a USD camera prim in your stage. This USD camera won't have the unique functionality of the Nuke camera node, like spherical or cylindrical projection modes, but will allow you to pass perspective or orthographic cameras and any other mappable attribute values between different DCCs.

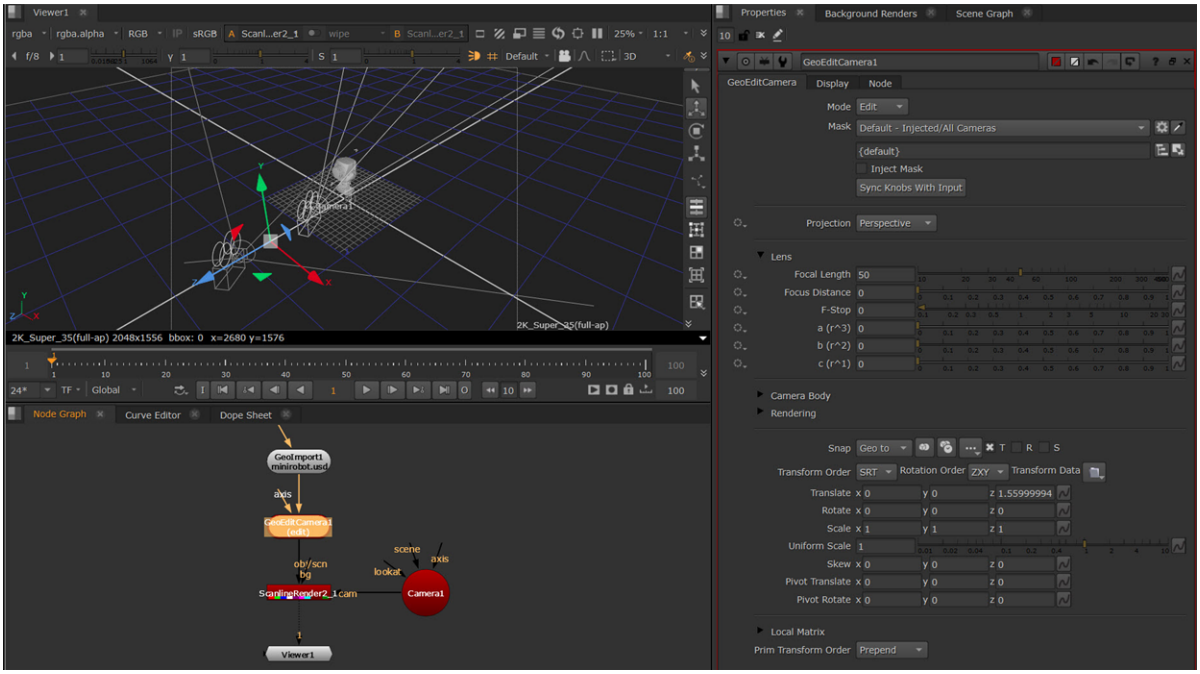
Editing USD Cameras

To directly edit a USD camera prim that has been imported into your stage, you can use GeoEditCamera node.

1. Add a GeoEditCamera node downstream of your imported USD stage.
2. By default the GeoEditCamera node is set to **Edit** in the **Mode**, and **All Cameras** in the **Mask**, meaning that any camera imported from your GeoImport will be targeted. You can also use the pick menu to verify which camera is targeted and change this if needed.
3. The **Sync Knobs With Input** button can be used to get the values of the imported camera and set the knobs to these.
4. You can then override values if needed. For example, you can use the camera locator in the Viewer with transform handles and snap menu to transform directly in the node. See more at Using Transformation Tools.



Note: When working with this node (and others in the 3D system) you may see some knobs greyed out with with the icon  to the left. This is because the Authoring mode is set to 'No Change' by default. Directly interacting with the knob automatically updates it to 'Set Value' mode and allows you to continue editing as usual. Learn more about Authoring Modes.

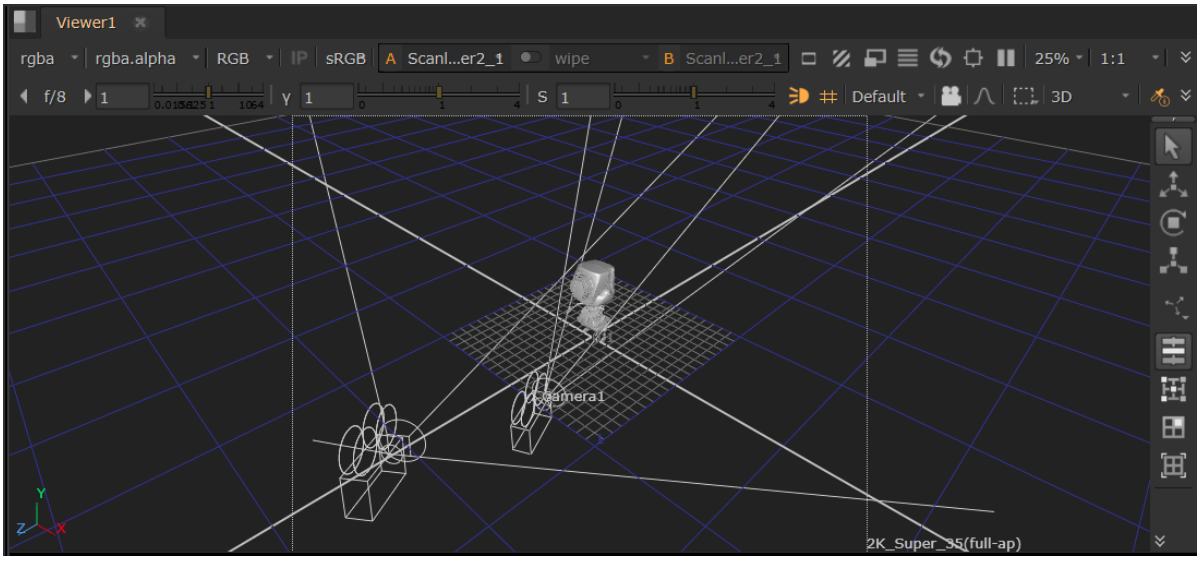


Tip: You can also constrain a camera within the stage using the **axis** input of the GeoEditCamera and the Axis node. See Using Axis Node in Your Scene.

Tip: In your scene you can include both USD cameras and Nuke “Axis op” cameras. If further cameras are needed in your scene you can still add new Camera nodes and connect them to ScanlineRender2.

Cameras in the Viewer

Both imported USD Cameras and Axis op Cameras have locators in the Viewer to allow you adjust positioning and see your scene in context.



You can also look through cameras in the 3D Viewer. See Look Through Camera.

Using Axis Node in Your Scene

The Axis node allows you to add input pipes, enabling more options to transform and constrain cameras.

This node allows you to pass data through from USD or Alembic files, or an existing prim in your scene.

Tip: To learn more about the different types of constraint in Nuke, see Constraining Prims.

Note: You can find more information on the controls in the Node Reference Guide - Axis.

Axis Data

When configuring an Axis in your scene you may wish to use imported or existing data.

- You can use **Import From File** or **Source Prim** knobs to import data either from an external file or pick a prim from your scene.
- If you have the **Live Read** knob enabled, any updates you make will be updated live. The Axis node is then able to be connected to other nodes where you want to use that data via an **axis** input, such as a camera.

Tip: See how the Axis node's **Source Prim** knob works in a constraint workflow in our Projections in 3D Video.

- You can push an Axis created in Nuke to your USD stage using the **Output Prim Path**. This happens by default when connected to the GeoScene node. If this path is left blank, it will create one for you when the camera is connected via GeoScene, and will be named *[AxisName]_Inject* by default.
- The Axis node's **Xform** syncing controls work the same way as the Camera node. See Xform Syncing Options to learn more.

Using Live Read

When **Live Read** is enabled, data can be pulled live from a USD or Alembic file allowing you to simply pass the data through the node, or to go in and start editing the data. This allows you to connect that Axis to any node in Nuke with an **axis** input.

If you are using an imported or referenced Axis, and have enabled **Live Read**, you'll notice that the knobs have a **green** outline. This helps you to see which data is being read from elsewhere in their script and which data has been changed directly on the node itself.

- The **green** outline means the data has not changed from the source.
- The **yellow** outline means there are local changes on this node which are different from the source.



Video: See how the data pass through works in a constraint workflow in our Projections in 3D Video.

The authoring mode icons to the left of each knob allow you to control this. **Set Value** when there are local changes, and you can always revert back to **No Change** if you wish to reset to default (from source). Learn more about Authoring Modes

When **Live Read** is enabled, there is also an option to **Snapshot Frame** and **Snapshot Range** while using the Axis. This will essentially take any Axis that is being live read into the node and bake the values as a local copy to that node, allowing you to have a permanent snapshot of the data at the moment in time.

Using the Look At Input - Example

You can also use more traditional constraint workflows using Axis. For example, you can connect an Axis node to a Camera's **look at** input so that the camera rotates to face the Axis, wherever it is moved.

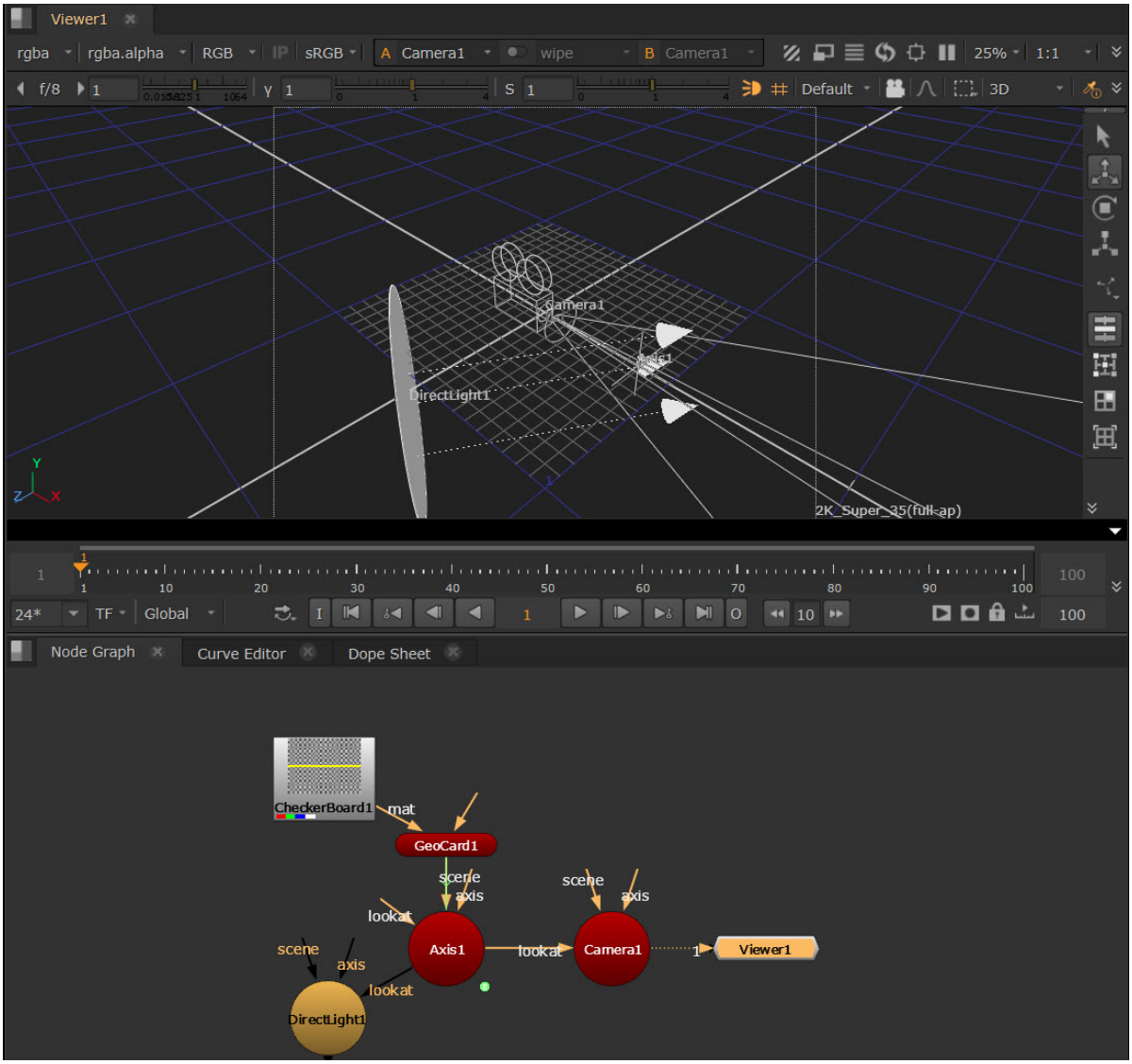
If you animate a card to move along the x axis, you can attach a camera and a light node with the **look at** inputs so that they automatically rotate and face the card as it moves. To do this, complete the following steps:

1. After animating your GeoCard, insert an Axis node.
2. Expression link the Axis node to the GeoCard node by holding **Ctrl**, clicking on the translate animation button in the GeoCard node's properties, and dragging it to the translate animation button in the Axis node's properties. The expression link is displayed as a green line with an arrow denoting the direction of the expression. See Linking Expressions for more information.
3. Insert a Camera node and either drag the camera to the required position in the Viewer, or use the Camera node properties to adjust the position of the camera.
4. Connect the Camera's **look at** input to the Axis node.



Note: If you amend the **Type** in the **Constrain** section, the input pipe switches label depending on which constrain type is selected. Learn more about constraint at Constraining Prims.

5. Insert a light node, such as DirectLight, and drag the light to the required position in the Viewer, or use the light node properties to adjust the position of the camera.
6. Connect the light node's **look at** input to the Axis node.
7. Playback your animated card and notice that the camera and light follow the animated card.



Projections in the New 3D System

Projecting Textures onto Objects

You can use the GeoUVProject and Project3DShader nodes to project texture images onto your 3D objects. This way, you can add detail, surface texture, or color to your geometry, making the geometry more realistic and interesting. The GeoUVProject node changes the UV values of the vertices whereas the Project3DShader node is a material shader.

Notes if you've used the Classic 3D System:

Workflow wise, GeoProjectUV is very similar to the UVProject node in the Classic 3D System, and most of the knobs are the same. There are however a few knob updates for better USD compatibility and updated properties ordering, for example:

- **Cull Behind Projection Source** allows you to stop any projections occurring behind the projection plane.
- The attrib name knob from the classic node, has been updated to include a **Generate Perspective** knob which generates an 'stw' attribute containing the perspective (w) component of the uv and the **Attribute Name** allows you to specify the name for this.
- **Reference Frame** essentially allows you to perform sticky projections. By enabling this knob the projection will only be applied at the frame being referenced, making it easy to set up projections that stick to moving prims.



Note: GeoProjectUV projects onto vertices so the tessellation of the mesh is important especially when viewing in 3D Viewer as the standard texture mapping in Storm, especially with UsdPreviewShader, does not handle perspective projections and so you see distortions across a surface with large triangles/quads. The smaller the triangle/quad, the more vertices are projected on and the lower the visible distortion.

The Project3DShader node has the same behavior of the classic Project3D node, but with two additional knobs with the ability to clip the projection beyond the cameras near and far planes.

Projecting Textures with the GeoProjectUV Node

The GeoUVProject node sets the UV coordinates for the object, allowing you to project a texture image onto the object. If the object already has UV coordinates, this node replaces them.

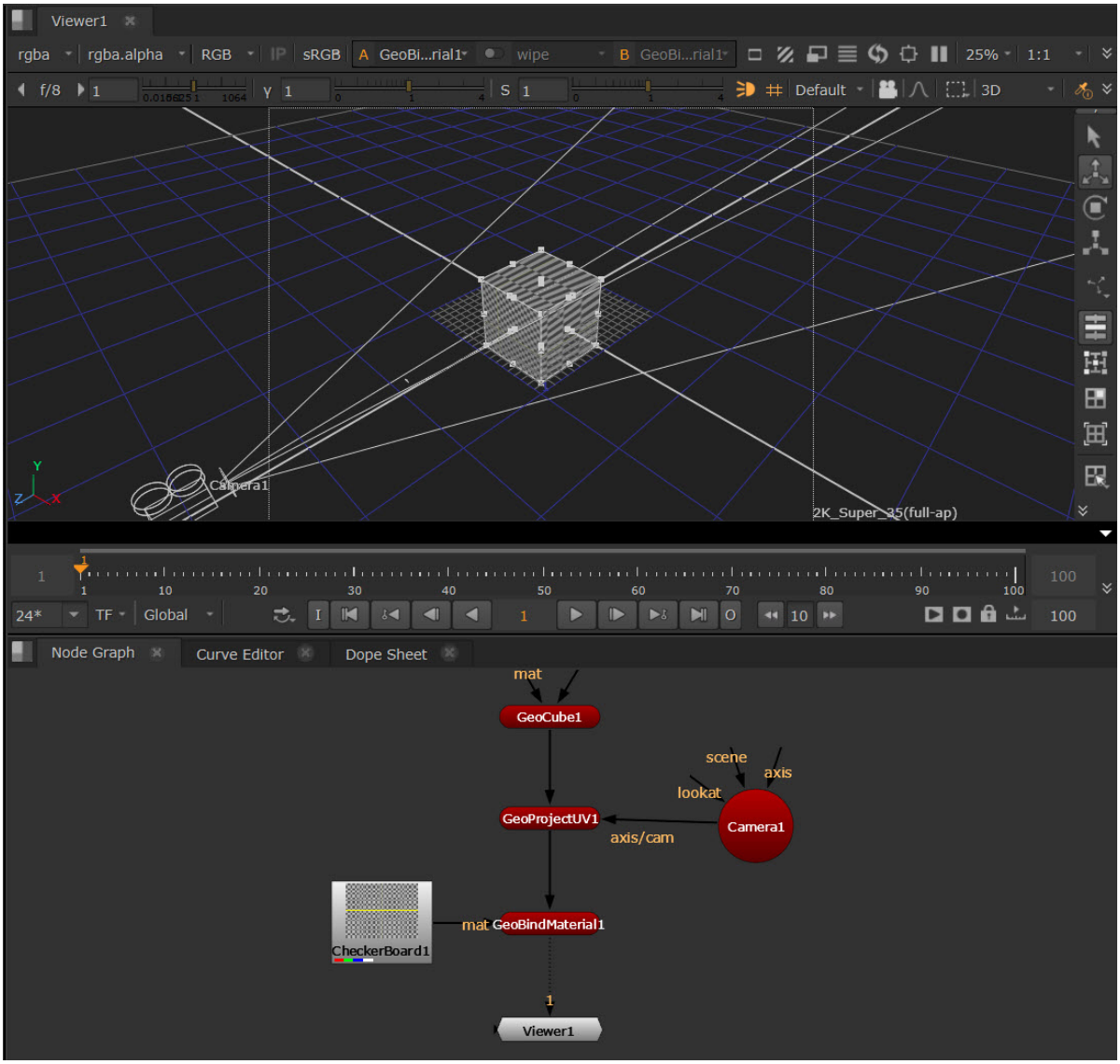
1. In the node graph, select **3D** > **3D** > **Modify** > **GeoProjectUV** to insert a GeoUVProject node anywhere after the 3D object you want to modify. In this example, we will place the GeoUVProject node downstream of a GeoCube node.
2. Attach a **Viewer** to the GeoUVProject node.
3. Connect an Axis or a Camera node to the GeoUVProject node's **axis/cam** input. If you connect an Axis node, project the texture UV coordinates onto the object using the axis transform values (that is,

translation, rotation, scale, etc.). If you connect a Camera node, do a similar projection as with the axis but also use the camera lens information, such as the aperture.

4. In the GeoUVProject node Properties, adjust the following parameters as needed:

- From the **Projection** dropdown menu, select the projection type. Usually, it's best to select a type that's close to the object's surface shape. For example, if your object is a sphere, like a football or a planet, select Spherical.
- From the **Plane** dropdown menu, select the projection direction: XY, YZ, or ZX to project the texture image along the z, x, or y axis. This dropdown menu is only available if you selected planar as the projection type.
- From the **Project On** dropdown menu, select Both, Front or Back depending on whether you want to project the texture on the front face of the object, its back face or both. The front face of an object is the one facing the camera and similarly the back face is the one furthest away from the camera.
- Check **Cull Behind Projection Source** if you want to stop any projections occurring behind the projection plane.
- • Check **View Frustum Culling** if you want the node to affect only the vertices inside the camera view frustum. Any vertices outside the view frustum are not affected and they still keep their original UV coordinates. Uncheck if you want the node to affect all vertices.
- To mirror the texture UV coordinates in the horizontal direction, check **Invert U**. To mirror them in the vertical direction, check **Invert V**.
- To scale (stretch or squash) the texture UV coordinates in the horizontal direction, adjust the **U Scale** slider. To scale them in the vertical direction, adjust the **V Scale** slider. The higher the value, the more the texture is stretched.
- To change the name of the attribute that's used as the vertex's UV coordinates to find the image pixel, enter a name in the **Attribute Name** field. If your geometry has more than one set of UV coordinates, you can choose the attribute that suits the projection type. Check the **Generate Perspective** option to generate an 'stw' attribute containing the perspective (w) component of the uv.
- Check the **Reference Frame** option if you want the projection to only be applied at the frame being referenced, for example if you want to set up projections that stick to moving prims.

5. Now you'll need to add the image you wish to project and bind this as a material to the object downstream. First, add an image, for example by adding a CheckerBoard or your own image via the Read node.
6. Connect the image to the **mat** input of a GeoBindMaterial node to project the image.



7. You can then transform the Camera to position the material.

Video: See a sticky projection example with GeoProjectUV in our Projections in 3D Video.

Projecting Textures with the Project3DShader Node

The Project3DShader node projects an input image through a camera onto the 3D object.


See Project3DShader: Projecting Textures.

Transforming and Modifying

This section covers the various ways you can edit the geometry, camera, lights and other prims in your New 3D System scene.


Transformations in 3D Scenes - how to use GeoTransform.


Constraining Prims - how to use GeoConstrain and the Constrain tab.


Using Transformation Tools - how to use transform handles, snap menu, tracks, and bounding boxes.


Manipulating Geometry Data - how to use GeoDisplace, GeoNoise, GeoNormals,GeoRadialWarp, and GeoTrilinearWarp.


Using Geometry Data - how to use GeoDrawMode, GeoDuplicate, GeoSelection, and GeoNormals.


Instancing Geometry - how to use GeoInstance.

Transformations in 3D Scenes

In the New 3D System, GeoTransform allows you to specify the prims you want to move in your stage. By default the mask will default to **Default - Injected/ All Roots**, which will pick up any Inject Masks from

upstream, or will revert to All Roots which moves everything in your stage, similar to how the classic TransformGeo in the Classic 3D System would work.

Note: Learn more about Inject Mask knobs at Using Masks Downstream.

The main knob to be mindful of is the **Prim Transform Order** knob. This knob allows you to specify the point at which this transform is applied to the masked prims, whether GeoTransform should be applied before (Prepend) or after (Append) the upstream transforms. **Prepend** matches the most closely to the Classic 3D System behavior, but because of the new scene graph hierarchy, prims transformations could be set up in a way where **Append** might make more sense in some scenarios and **Prepend** in others.

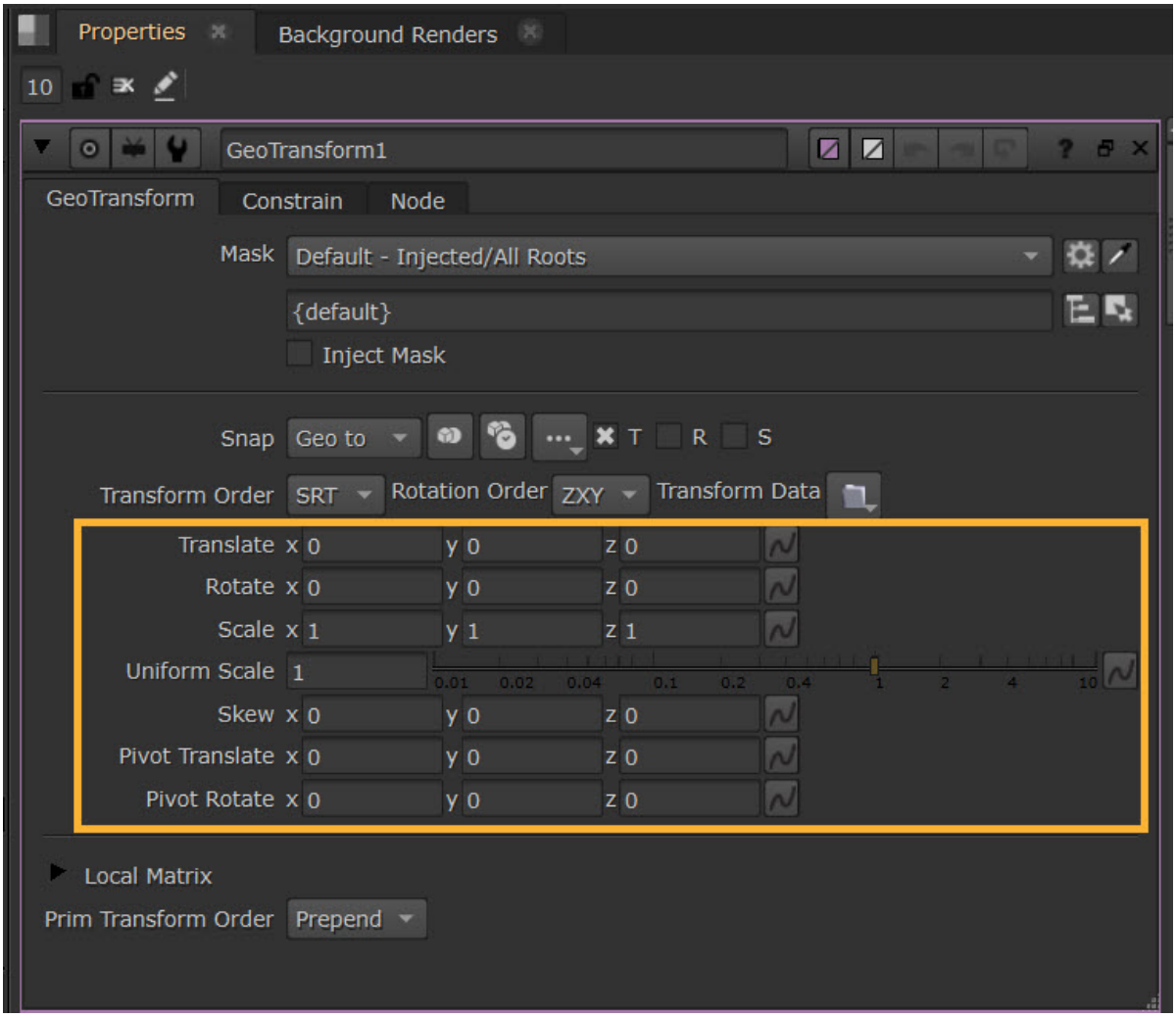
Tip: As an artist if your transform is not functioning as you were expecting, it is worth switching between Append and Prepend, and exploring how the transforms for the prim you want to move are set up in the scene graph hierarchy.

Tip: There are some useful tools that are part of GeoTransform node, such as the **Snap** menu directly in the node. See more about these tools at Using Transformation Tools.

Note: Via the **Constrain** tab, you can directly constrain to other prims without needing to add an additional node. This tab works in the same way as GeoConstrain - learn more at Constraining Prims. Note that the constrain input, by default 'look at', only works with Axis ops currently and we are exploring allowing you to connect this input pipe to other geo ops.

Editing Transformation Values

To perform transformations in Nuke's New 3D System, you can edit the transformation values in the GeoTransform node's Transform options in the **Properties** panel.



If you connect this to the Viewer, transform handles are also available in the Viewer, which you can use to edit these values. See more about these tools at Using Transformation Tools.

Tip: You may not always need to add a separate GeoTransform node, as some prim creation nodes also allow you to edit their transformation values in their Properties such as GeoXformPrim and GeoSphere.

Note: In the Camera, Axis and Nuke light nodes, you'll notice that there are various types of transformations. See Xform Syncing Options to learn more about this.

Constraining Prims

Constraining allows you to link the transformations or two or more prims together. In Nuke, this is done either by the GeoConstrain node, or via the constraint input and **Constrain** tab options in various other nodes, as outlined below.

Using GeoConstrain Node

The GeoConstrain node allows you to constrain geometry, cameras, lights, and more, to other prims in your scene.

1. Add a GeoConstrain node downstream of your geometry and in the **Properties**, set what you want to constrain via the **Mask** knob.
2. Set the **Target Path** to the prim you want to constrain to i.e. the moving prim you want your original prim to look at or move with.
3. The **Point** knob lets you set where on your target prim you want to constrain to, by default this is set to the **Origin**, but you can change this to be a specific **Vertex** or **Face** on the geometry and then use the picker knob to pick a relevant point. Note that if you are constraining to a prim selection that is not a mesh then you may only be able to constrain via origin as the target object needs to be a point-based prim.

Tip: The **Time** knob lets you set a time offset for your constraint by a relative amount in the past or future, or an absolute value which basically acts like a FrameHold which you can key as needed.

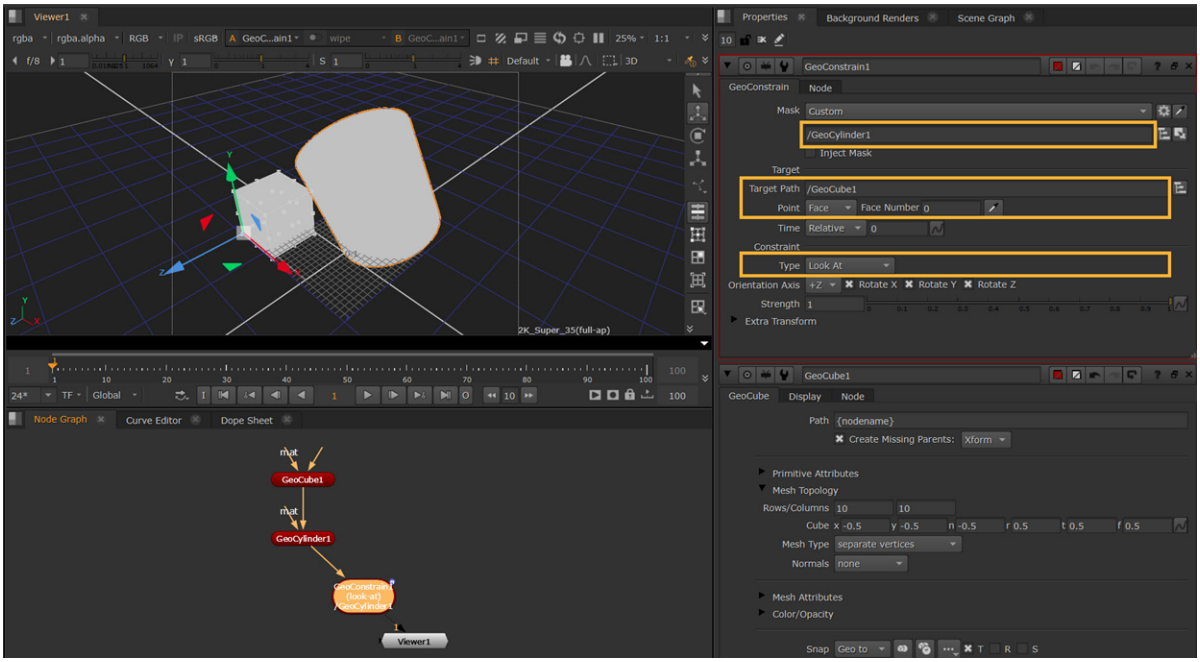
4. Choose the Constraint **Type**.

- This knob defaults to the **LookAt** constraint, allowing your masked prim to look at the target prim as it moves. You can set which axis you want to be able to follow on and the strength, which is useful when you want to blend camera movements together.
- The **Transformation** constraint matches the location, rotation and scale of the prim to those of the target, so whatever the target prim does, the masked prim will also do. You can toggle the Translate, Rotate and Scale attributes on and off independently and by default we have Scale turned off because most often we found artists are just trying to match the translation and rotation of a prim. If you ever just want to do one of those constraints individually then you can either deselect these knobs, or just select the **Translation**, **Rotation**, or **Scale** in the **Type** options from the dropdown.

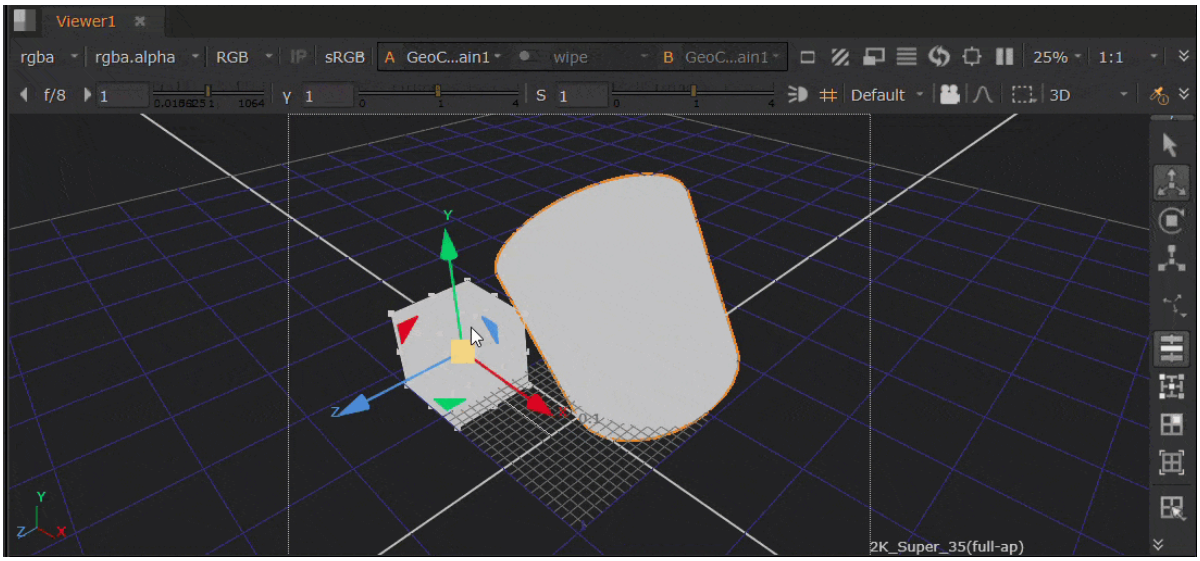
- The **Maintain Offset Frame** knob allows you to keep a relative offset between itself and the target prim. This basically means the prim either constrains from its starting position, or goes exactly to the position of the target prim.
- You can also constrain using the **Parent** constraint which modified the prims transformation as if the target was the parent hierarchically. Most of the time this will work the same as the transformation constraint, but it is easiest to tell the difference when you start rotating your target prim, as the hierarchical nature will mean the masked prim will rotate around the target prim like planets in a solar system, rather than matching the targets prim like a Transformation constraint would do.

Note: To learn more about the types, including Translation, Rotation and Scale, see GeoConstrain in the Reference Guide.

In this example, we are constraining a GeoCylinder to the face of a GeoCube, using **LookAt**. The **Mask**, **Target Path**, **Point** and **Type** have all been selected.



5. Now translating the GeoCube will automatically translate the GeoCylinder to look at the chosen face of the GeoCube.



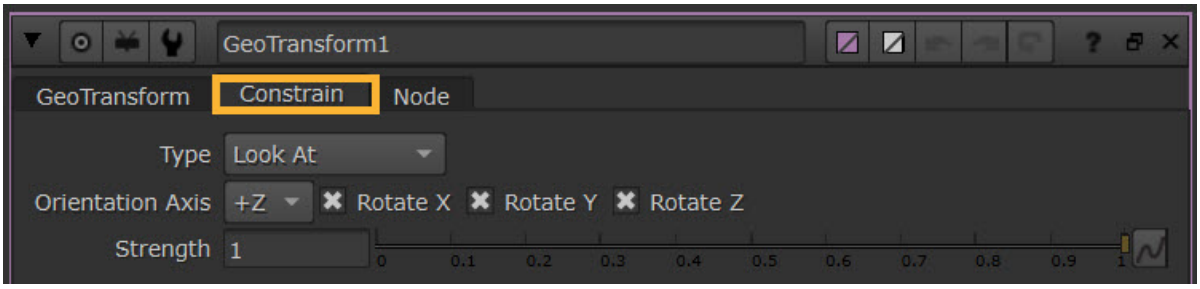
Note: The **Extra Transform** section does as the name implies, allows you to add any additional transformation offsets that you would like, but with the addition of a **Coordinate System** knob, which lets you specify whether you would like the extra transformers to exist in a **Point**, **Prim**, or **World** coordinate system.

Video: See our Projections in 3D Video to see how you can use GeoConstrain for sticky projections.

Using the Constrain / Position Tab

Some other nodes allow you to constrain directly in the node via constraint inputs and the **Constrain** tab, such as GeoTransform, as well as Axis, Camera, and all of the Light nodes on the **Position** tab.

This tab works in the same way as the GeoConstrain node, without the need to add an extra node to your node graph. You won't need to select a Mask or a Target, as you will be constraining using the current node and the nodes connected via the constraint input pipe, by default 'look at'. The label on the input pipe will adjust depending on which Constraint **Type** you select.

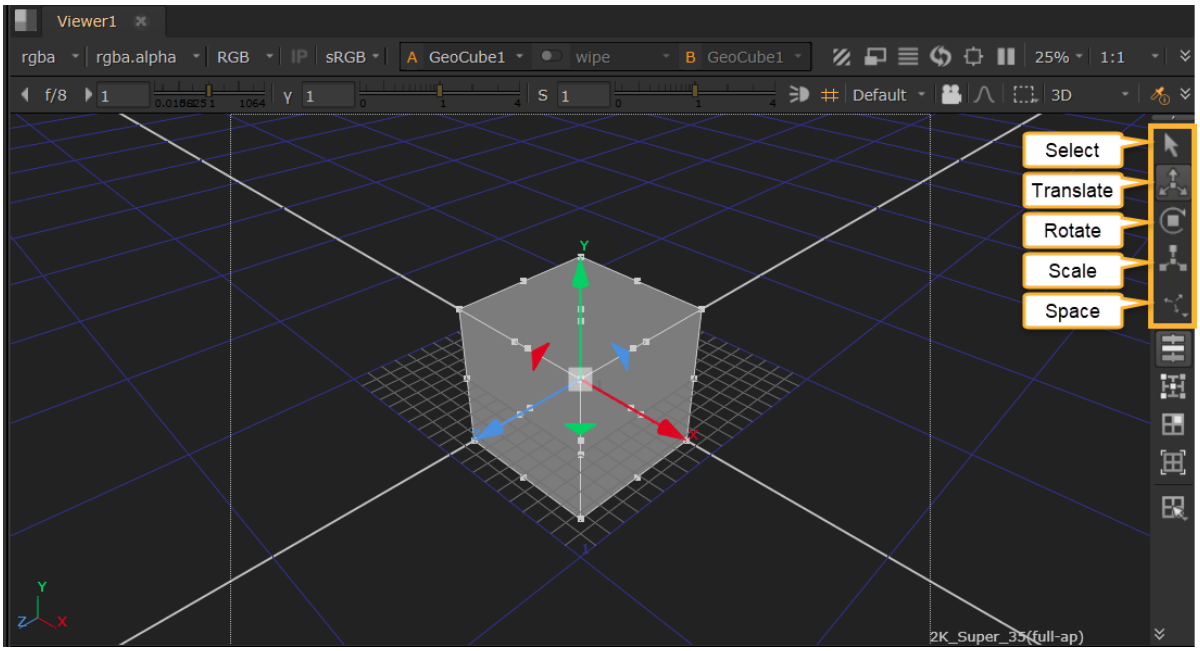


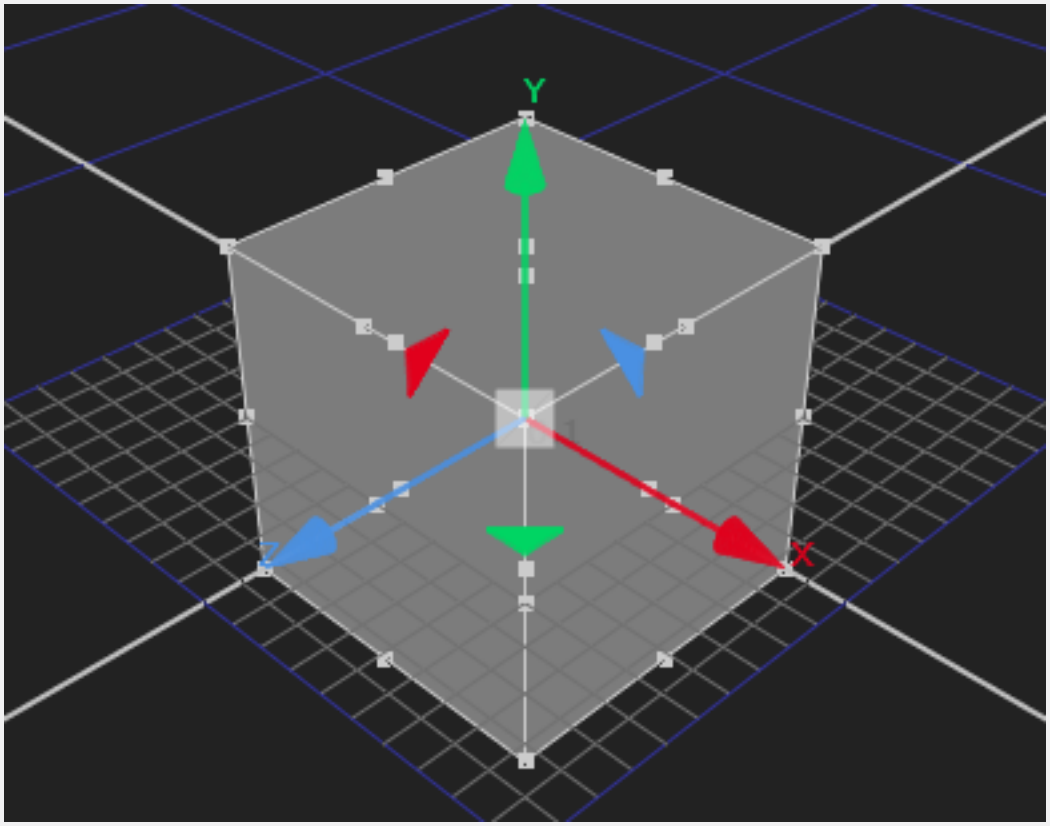
An example of using the **Look At** with Axis, Camera and Lights can be found at Using the Look At Input - Example.

Using Transformation Tools

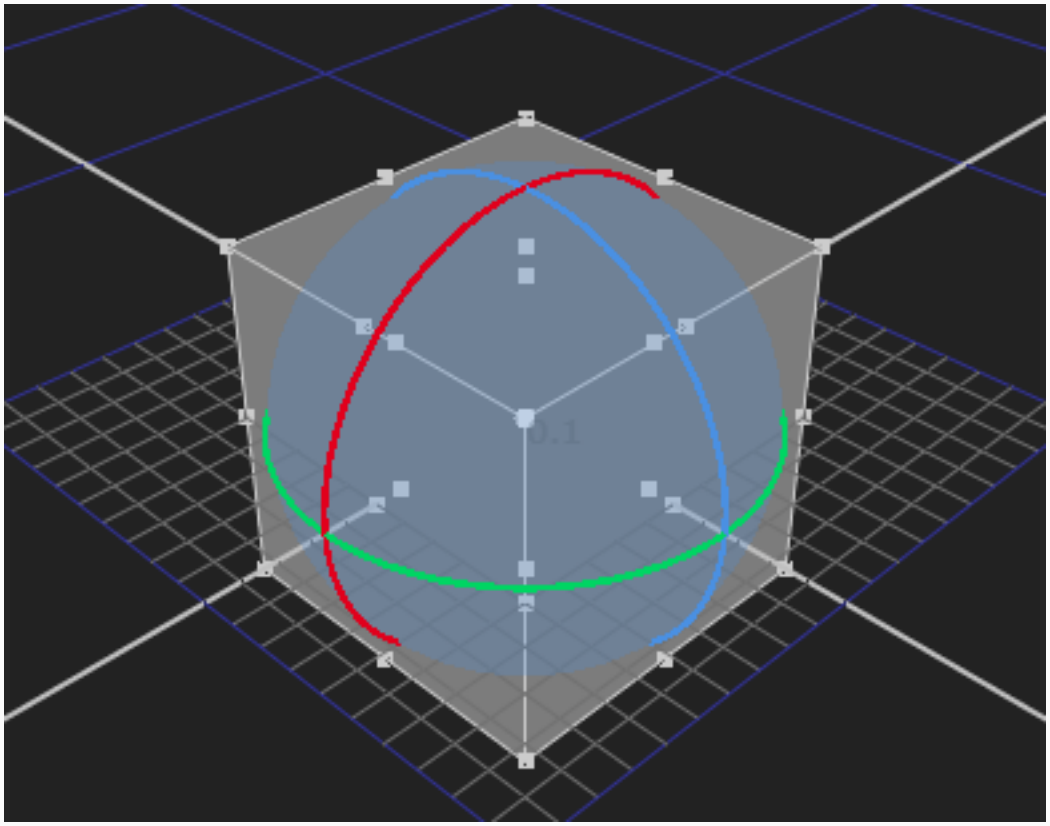
Transformation Tools in the Viewer

Each transform option brings up the relevant handles inside the Viewer so you can transform your prim by eye.

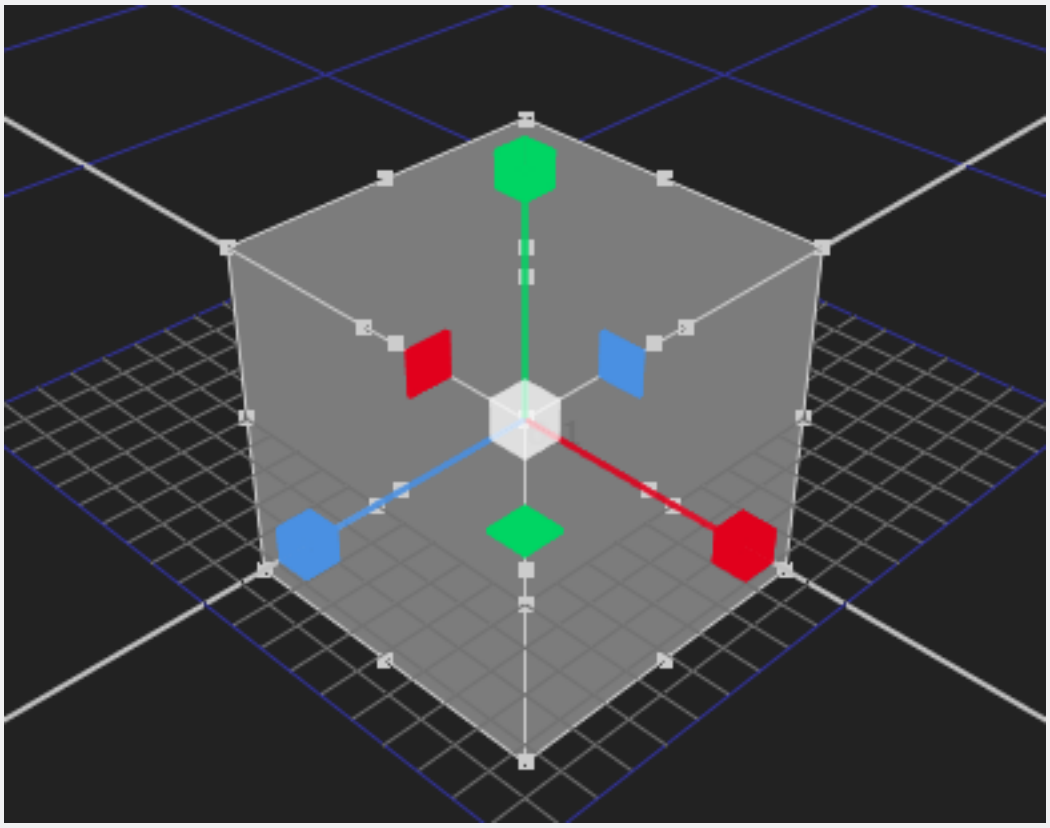




Translate handles



Rotate handles



Scale handles

Transforming objects in the Viewer is available in three modes: **object**, **screen**, and **world** space. You can select which transform space you're using by clicking the icon.

Transforms behave differently depending on which transform space you're using:

• **object** space - transforms in this space are relative to the object itself and its orientation. For example, translating on the X axis in object space might shift its position on the Z axis in world space.

• **screen** space - transforms in this space are relative to the screen, the flat plane of your monitor, and its orientation. You can only translate on the X and Y axes, the Z axis is disabled.

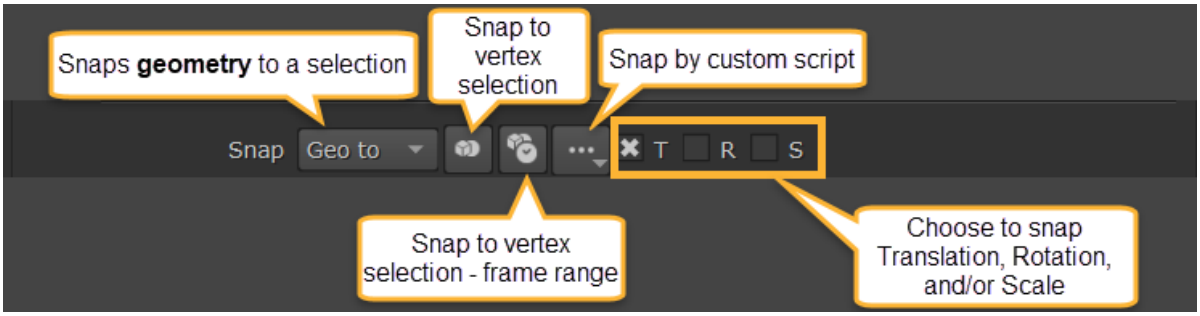
• **world** space - transforms in this space are defined by the XYZ axes of the scene in Nuke's 3D Viewer. The mini axes drawn at the bottom left of the Viewer always show world space no matter what space you're working in.

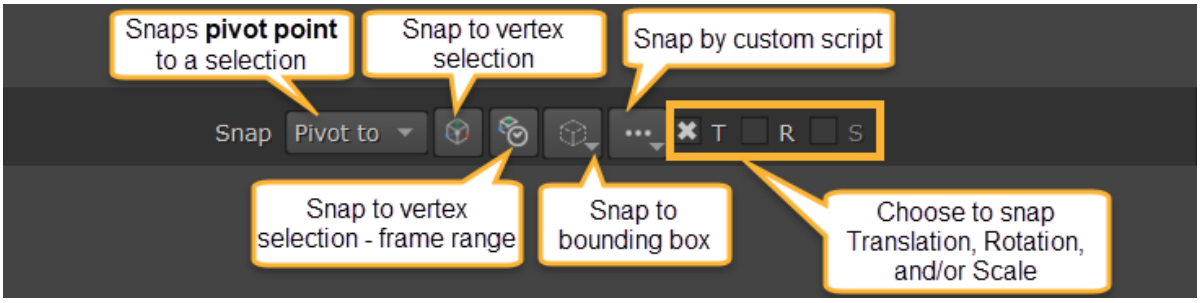
See more about transformations at Transformations in 3D Scenes.

Snap Menu

You can use the **Snap** menu, to snap Geo or Pivot to vertex selections, custom scripts, and bounding boxes.

This menu is found in node properties of geometry nodes (GeoCube, GeoCylinder, GeoCard, GeoSphere), GeoTransform, as well as Camera, GeoEditCamera, Axis and Lights (GeoDiskLight, GeoDistantLight, GeoDomeLight, and GeoSphereLight. SpotLight, DirectLight, PointLight, and EnvironmentLight.)

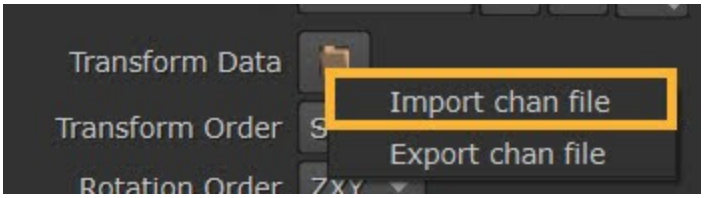




Channel Files

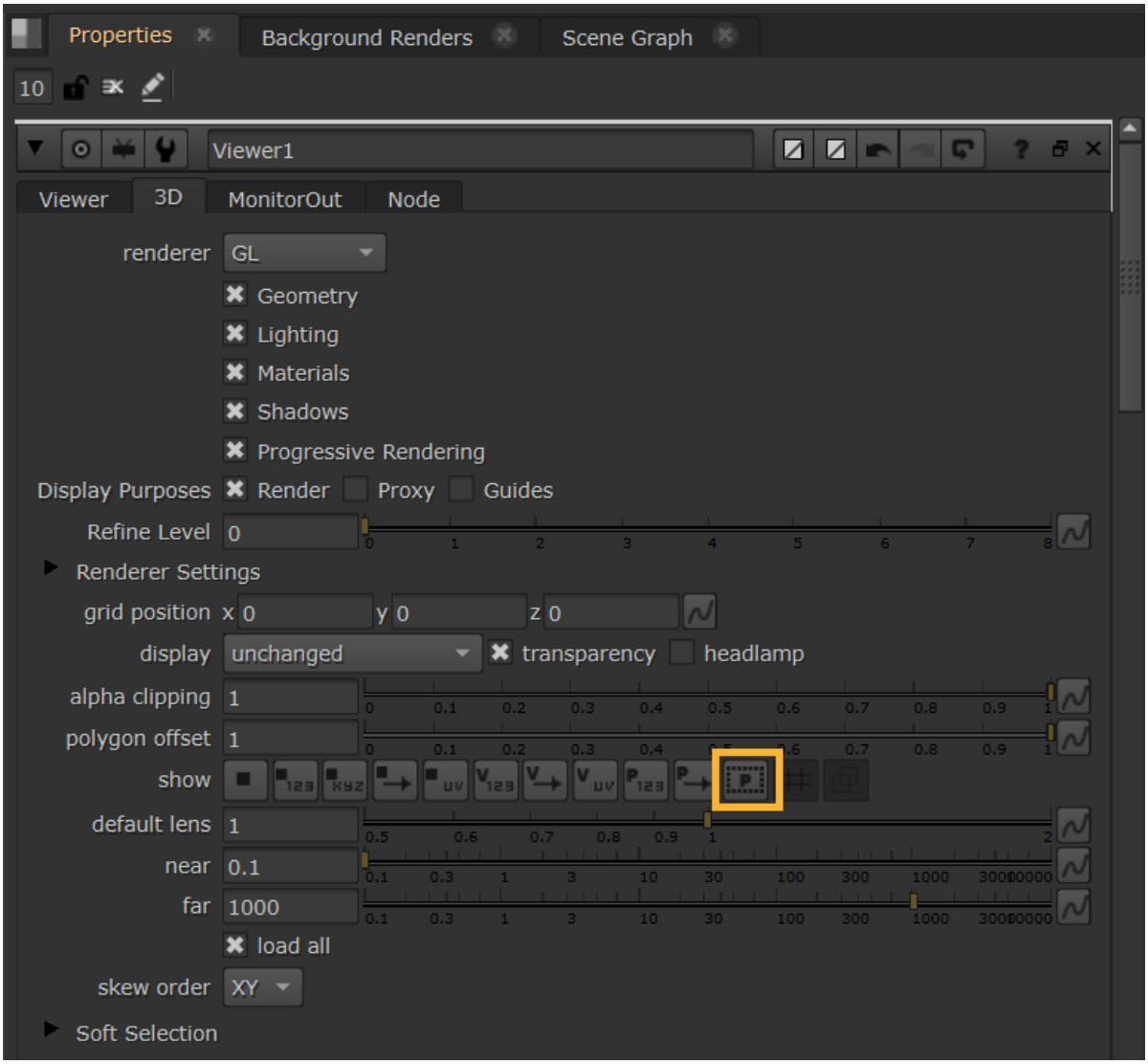
You can also apply tracks to prims. Nuke can import channel files and apply the motion data to the transformation parameters of any prim. To do this you can use the **Transform Data** button and select **Import chan file.**

This menu is found in node properties of geometry nodes (GeoCube, GeoCylinder, GeoCard, GeoSphere), GeoTransform, as well as Camera, GeoEditCamera, Axis and Lights (GeoDiskLight, GeoDistantLight, GeoDomeLight, and GeoSphereLight. SpotLight, DirectLight, PointLight, and EnvironmentLight.)

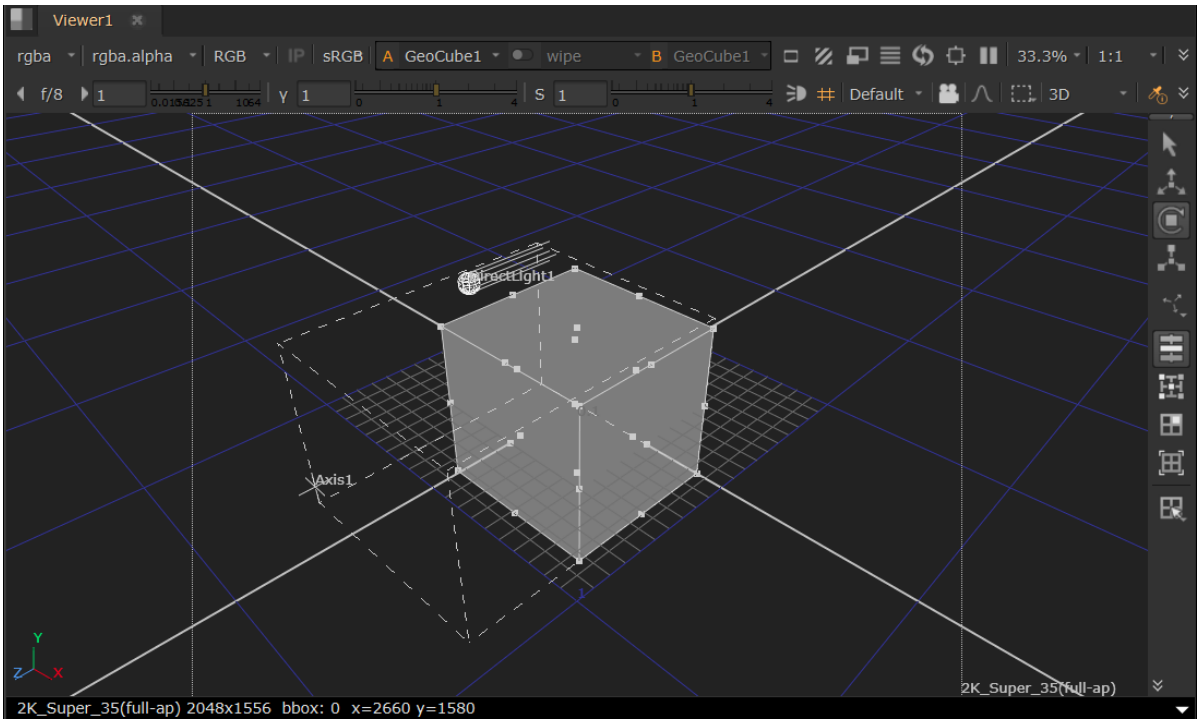


Bounding Boxes

You can turn on prim bounding box in the **Viewer** Properties under the **3D** tab > **show**, and selecting this button:



The bounding box will then be visible via a white dotted line in the 3D Viewer.



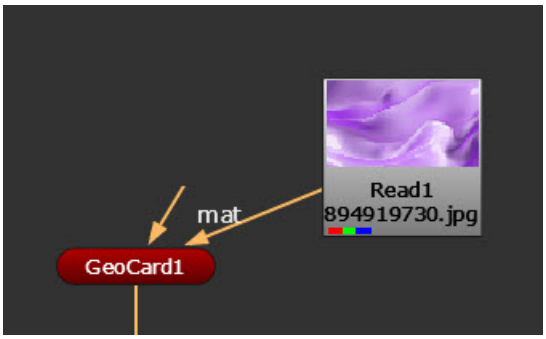
Manipulating Geometry Data

Using GeoDisplace

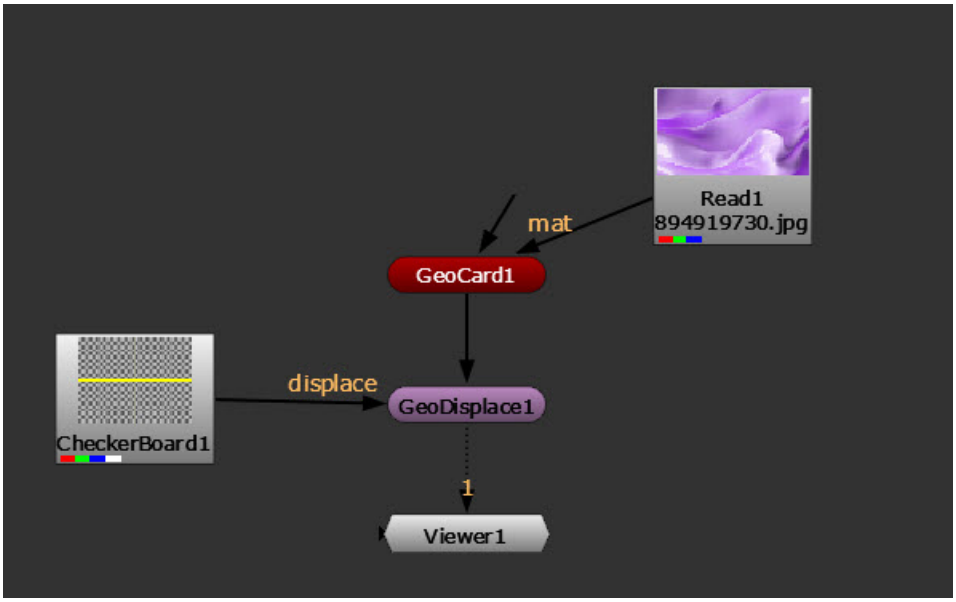
You can use displacement to quickly add detail to geometry. The GeoDisplace node moves each vertex along the normal to the surface according to channels in a texture map connected to the **displace** input.

For example, you can create a bumpy surface on a GeoCard:

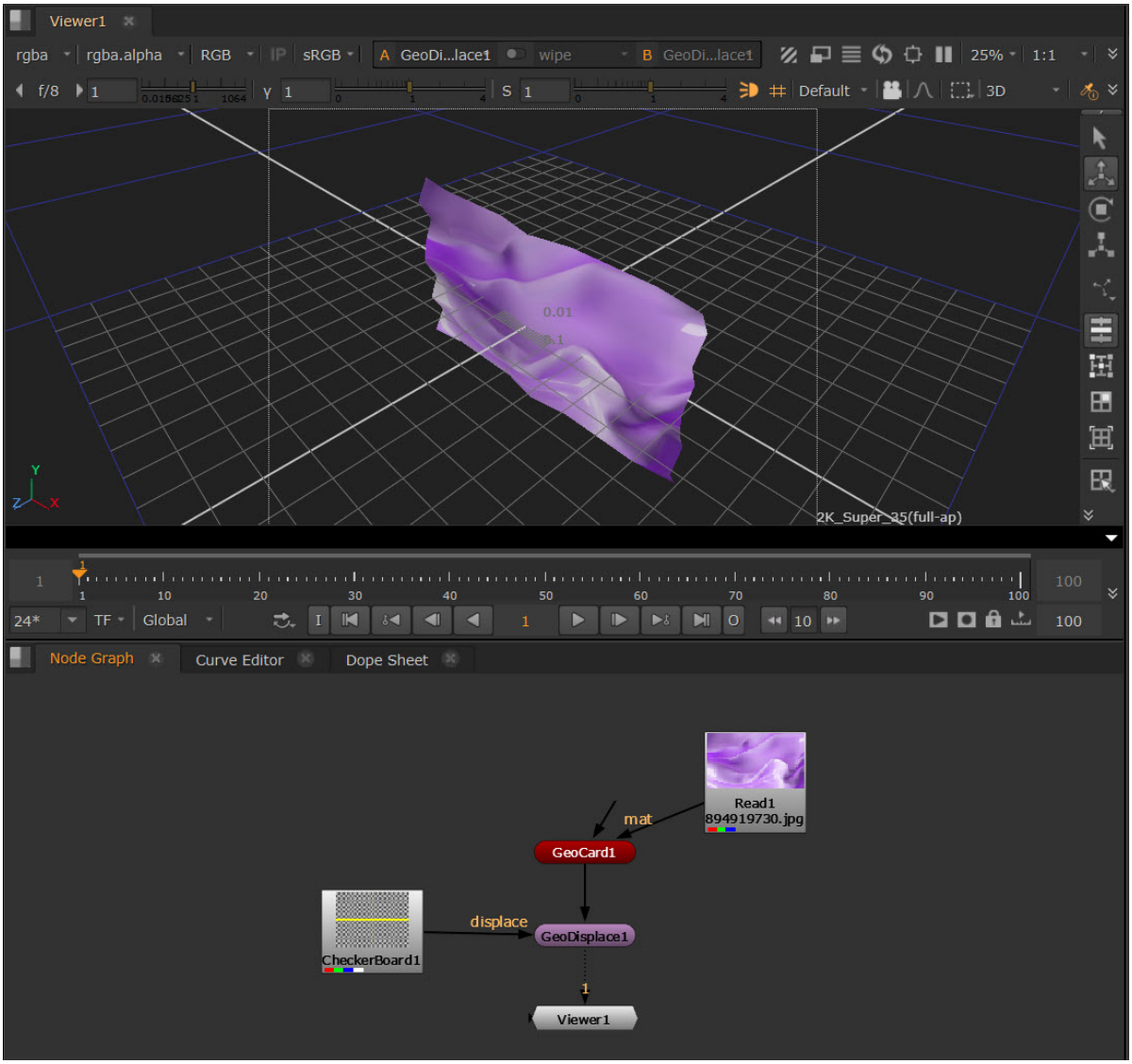
1. Add geometry to your scene using GeoCard node.
2. Connect a texture image to your GeoCard using the Read node and its **mat** input.



3. Add a GeoDisplace node downstream of your GeoCard node.
4. Add a CheckerBoard node connected to the GeoDisplace nodes' **displace** input.



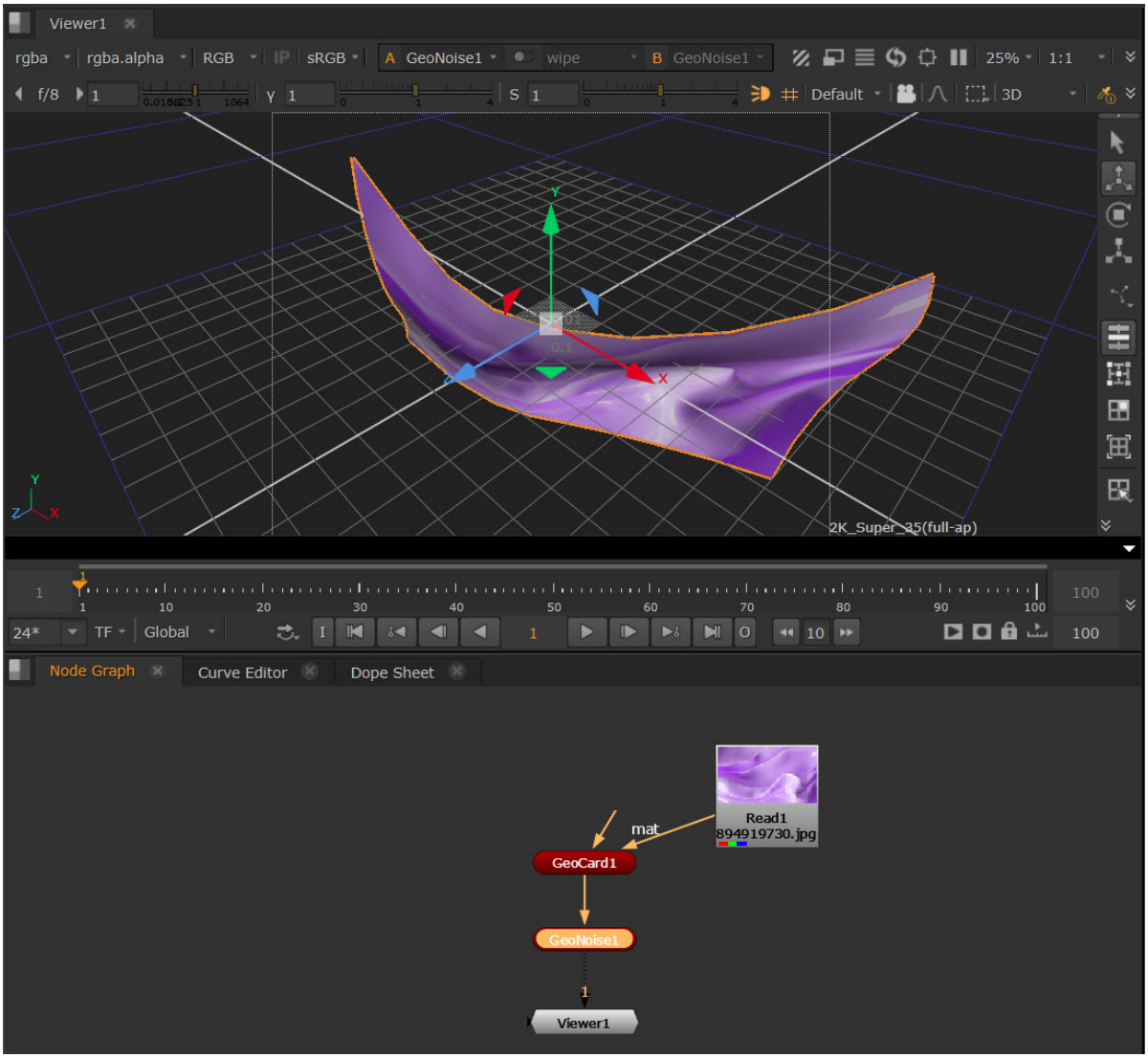
5. Now you can view the displaced geometry in the Viewer and edit the GeoDisplace node Properties to refine the effect. See GeoDisplace in the Node Reference Guide to learn about the controls.



Using GeoNoise

The GeoNoise node warps meshes using a Perlin noise function that creates seemingly random noise. For example, you could use this node to generate terrain from a card.

Using the same example as the above, this time we are using GeoNoise downstream of the GeoCard.

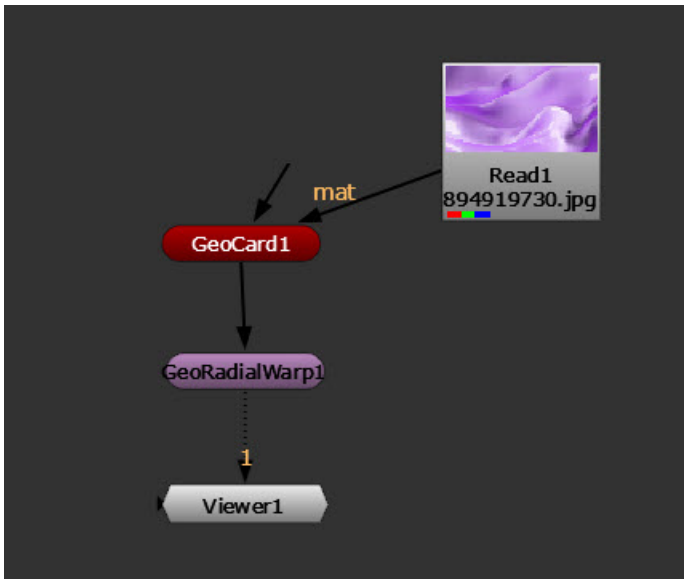


- You can use the transformation handles in the Viewer to refine its shape.
- Edit the knobs in the Node Properties, such as **Noise Type** and **Time**, to adjust the effect. See GeoNoise in the Node Reference Guide to learn about the controls.

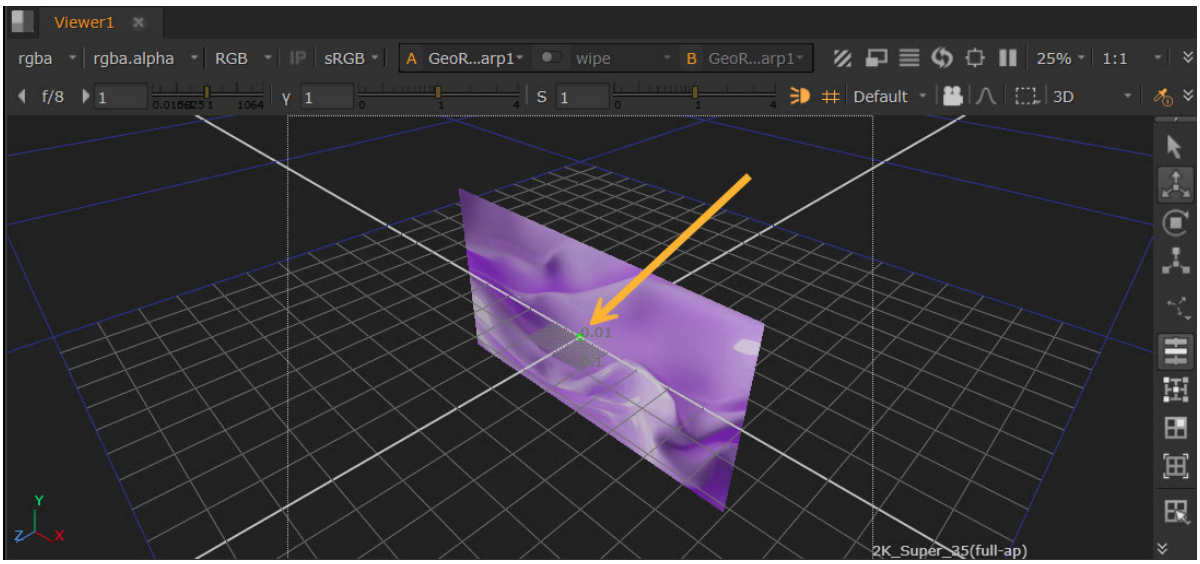
Using GeoRadial Warp

The GeoRadialWarp node warps meshes away from a center point.

1. Connect a GeoRadialWarp node downstream of your geometry.



2. You can use the **Rotation Center** control in the Properties to set the location of the warp center. Or you can drag the center widget around in the Viewer as below. The center widget is the green dot here.

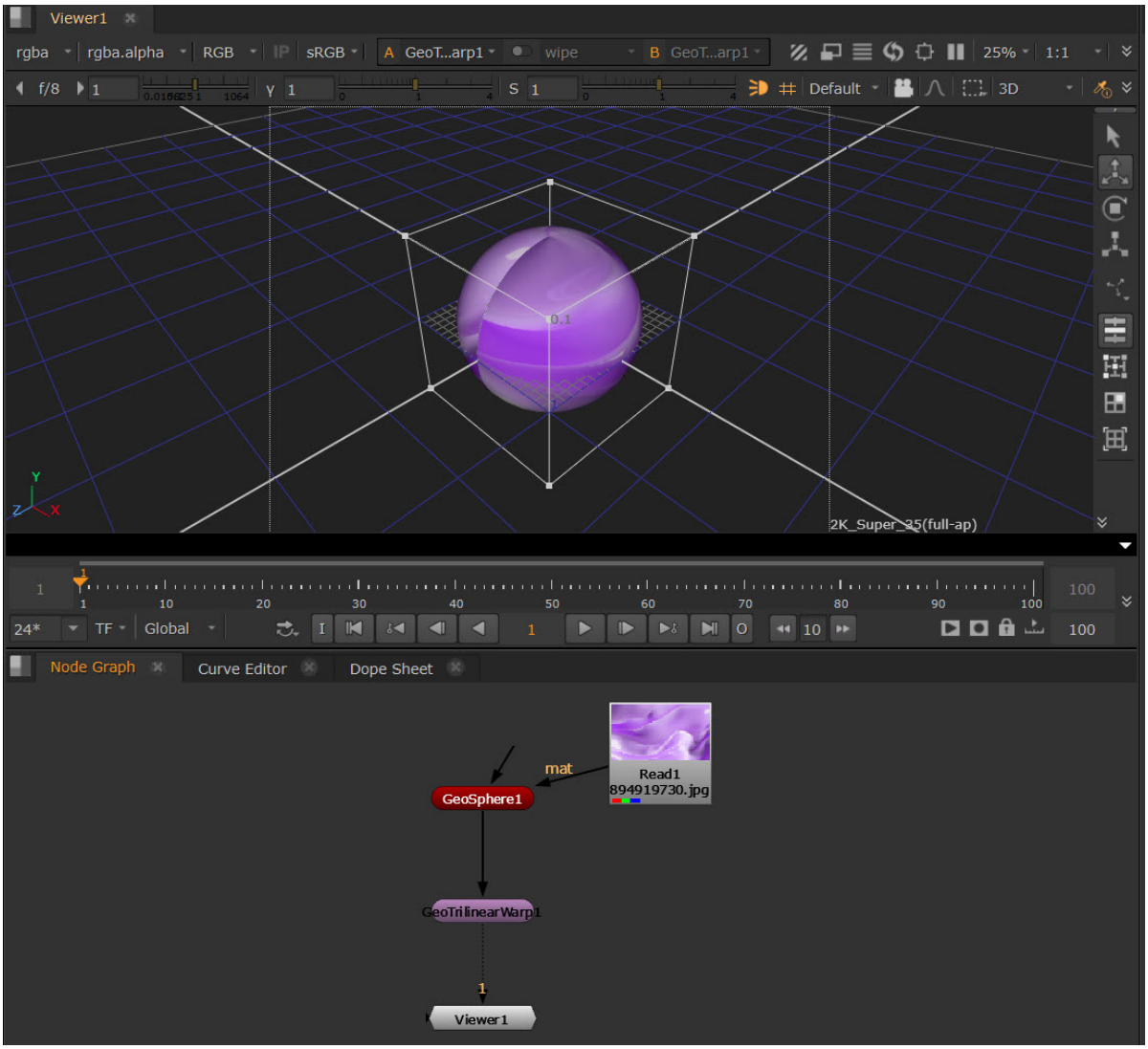


Using GeoTrilinearWarp

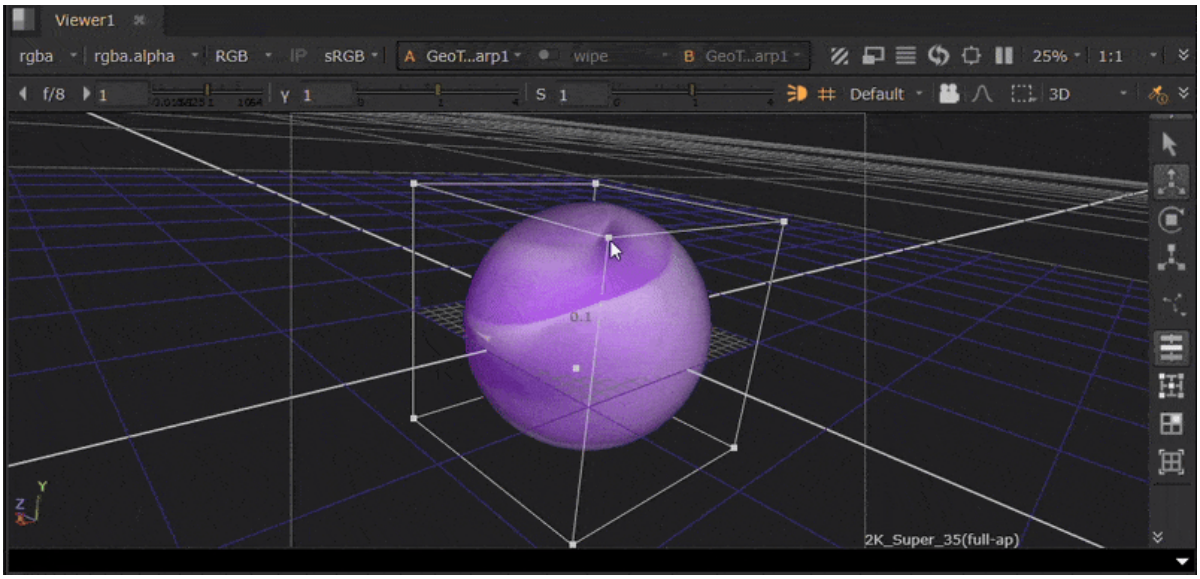
The GeoTrilinearWarp node warps meshes by moving the corners of its bounding box separately to create the warp.

For this example, we will use a GeoSphere node so that the bounding box is visibly distinct.

1. Connect a GeoTrilinearWarp node downstream of your geometry. The bounding box will then be visible in the Viewer.



2. Click and drag the corners of the bounding box in the Viewer to warp the mesh.



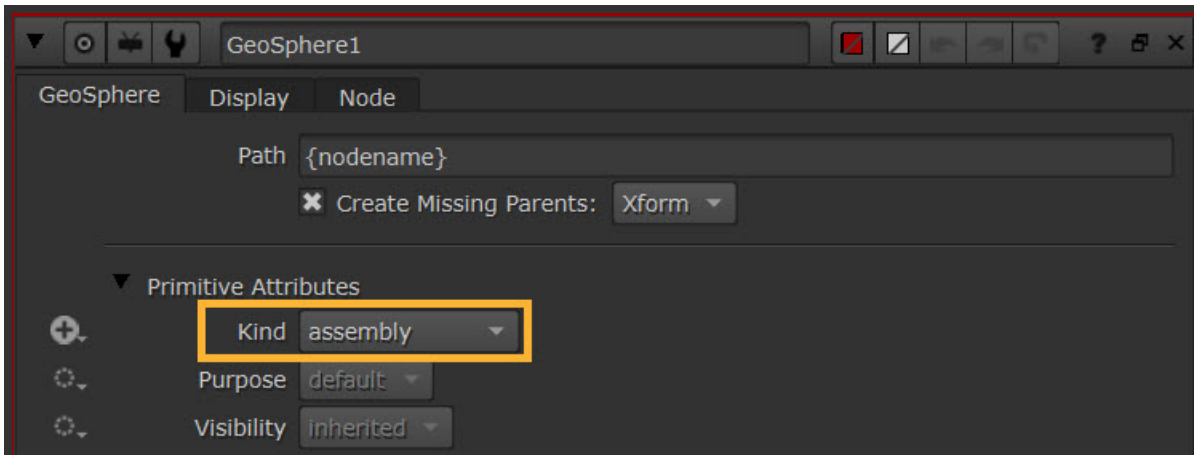
Using Geometry Data

Using GeoDrawMode

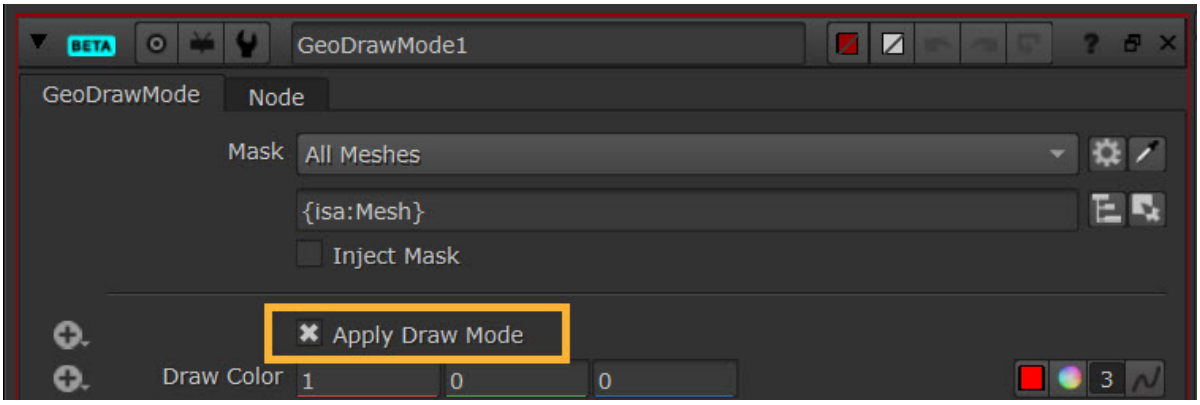
GeoDrawMode is a node that allows you to draw your geometry as bounding boxes, as cards, as origin points.

Note: GeoDrawMode only works on prims that have their **Kind** attribute set.

1. Make sure your geometry prim has the **Kind** attribute set. In this case we are using a GeoSphere.

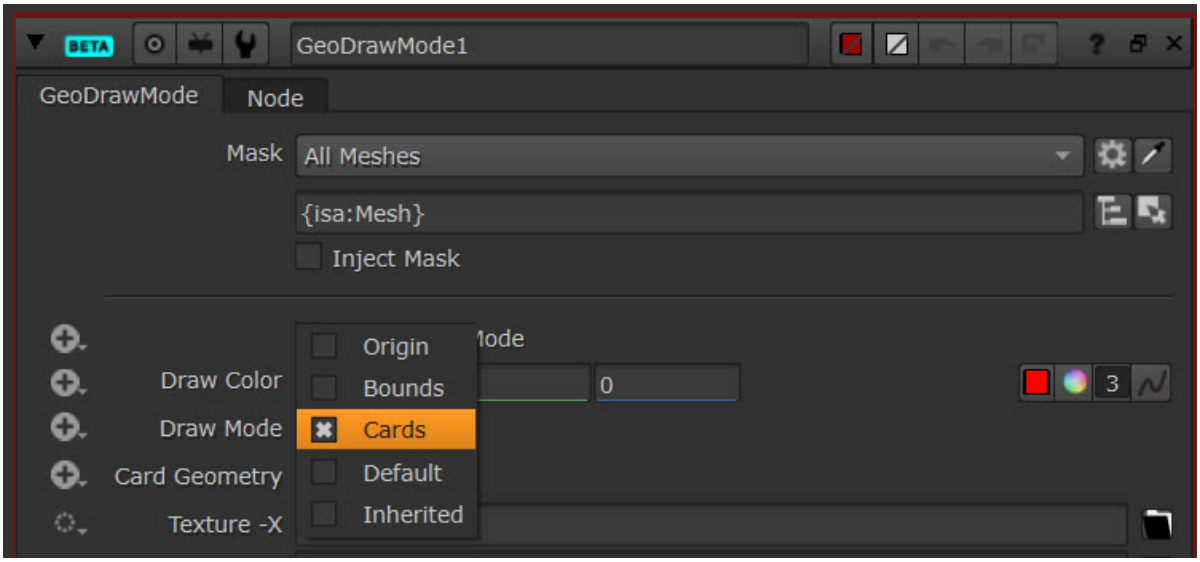


2. Add a GeoDrawMode node downstream of your geometry. Make sure, in the GeoDrawMode Properties, that the **Mask** is pointing to your prim. In this case we will use the option **All Meshes** to target the GeoSphere.
3. The **Apply Draw Mode** knob allows you to essentially set the draw mode at the level your mask specifies. You'll need this option checked on to be able to visualise the draw mode.

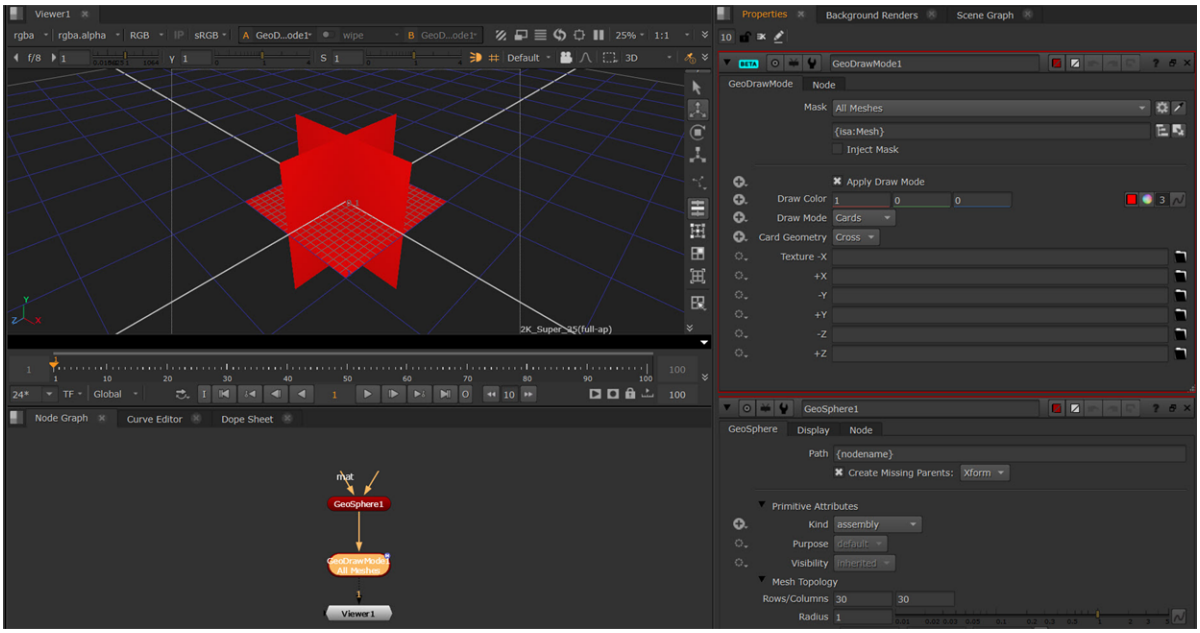


4. The **Draw Colour** knob allows you to set the color of the bounding box or cards you are creating. The Default color is red.
5. The **Draw Mode** is what allows you to change between the different draw options.

   For example, when you set it to the **Cards** mode, this enables various options in the **Card Geometry** knob below.



6. Set the **Card Geometry** knob to visualise the card as either a **Box** or a **Cross**. Here we have the card as a **Cross**.

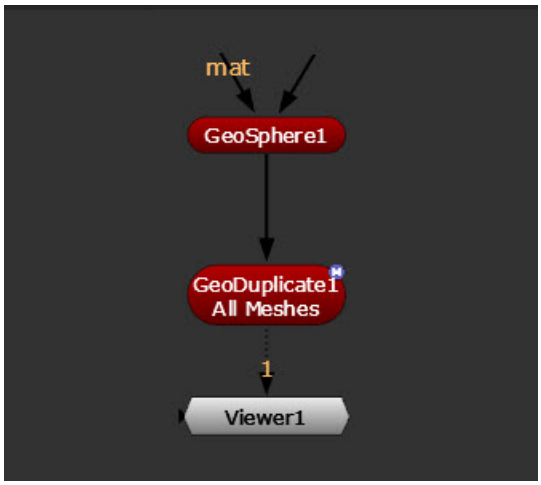


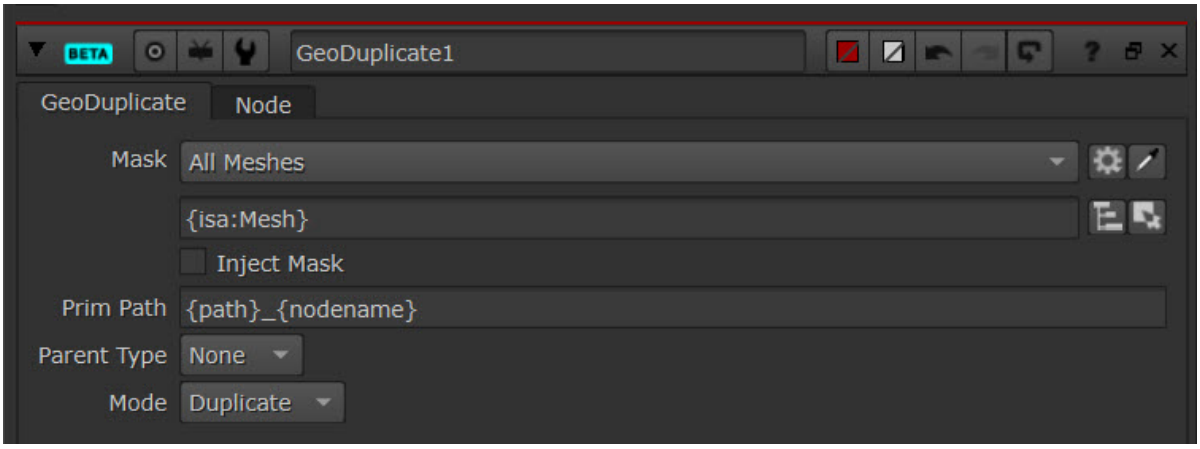
7. You can then use the **Texture** knobs to apply textures to your card.

Using GeoDuplicate

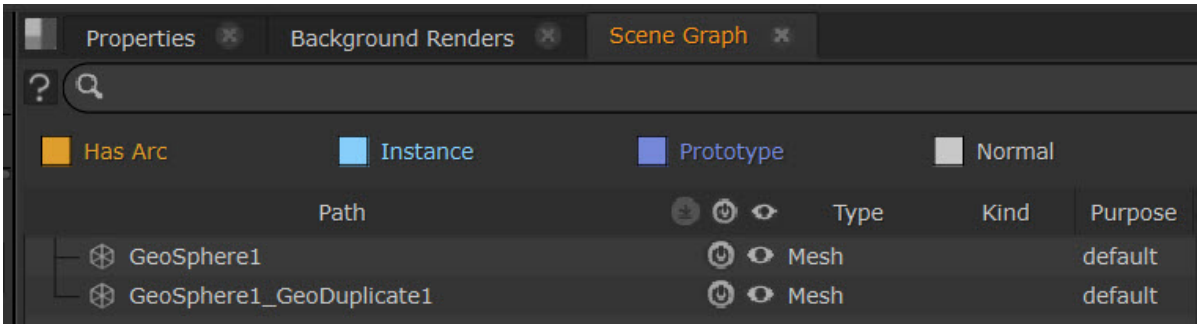
GeoDuplicate allows you to duplicate a prim or a number of prims.

1. Add a GeoDuplicate node downstream of the prim you wish to duplicate. Make sure that the **Mask** knob of the GeoDuplicate node is pointing to the prim you wish to duplicate. In this case, we are pointing to **GeoSphere1** using the option **All Meshes**.





2. This will create a new scene graph location at the location in the **Prim Path** knob. The new location is required because each object in USD must have a unique path. By default it will name the duplicated prim by the name of the prim you are masking and the name of the GeoDuplicate node: **{path}_{nodename}**. For example when duplicating **GeoSphere1**, you would get **GeoSphere1_GeoDuplicate1**:



Tip: You can update the duplicate prim's name if you want to, by modifying what is in the **Prim Path** knob and any parent groups you can set to be either Scopes or Xforms.

3. By default the **Mode** is set to **Duplicate**. Duplicate mode makes a flattened copy of the prims. This is the default behavior as it is the most similar to the Classic 3D System.

   Or you can duplicate the masked prims as a **Reference**. This essentially makes an instance of the original prim data, so you are using the source prim data as a live input and any changes you make to that data will also apply to the instance.

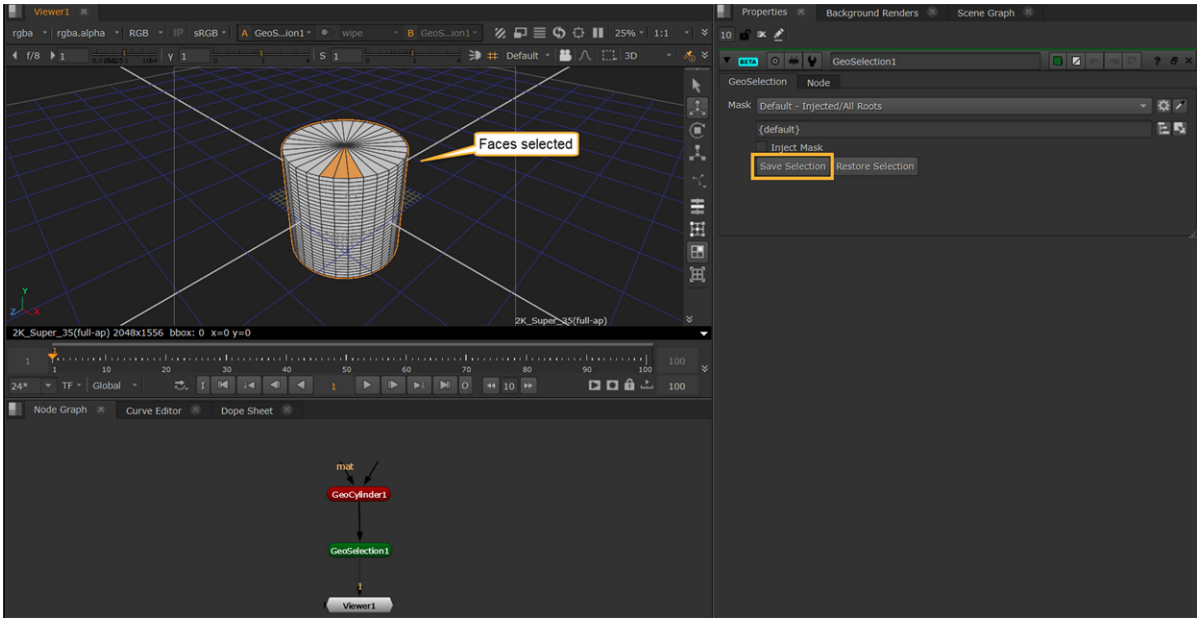
Note: The best way to think of it is if you want a wholly new duplicate of the geometry that only changes when you make specific changes to it, use **Duplicate** mode. If you want to make changes to the original geometry and have those changes also show up in the duplicate, then use **Reference** mode.

Note: See more about instancing at Instancing Geometry.

Using GeoSelection

GeoSelector node allows you to store a face or vertex selection in the Viewer and then reapply the selection later.

To use the node you simply create a face or vertex selection, then select **Save Selection**.



Then, if you make a different face or vertex selection and want to get back to your saved selection, you simply hit **Restore Selection**.

Using GeoNormals

The GeoNormals node creates or rebuilds mesh normals. Subdivision surfaces don't generally need normals, but there are occasional cases where creating or modifying normals is beneficial, such as when shaders need detailed lighting information to render correctly.

- Use the option **Action** > **build** to rebuild each normal specified in **Attribute Name** based on the surrounding vertices.
- The **Angle Threshold** determines the break angle where two faces no longer constitute a smooth surface.

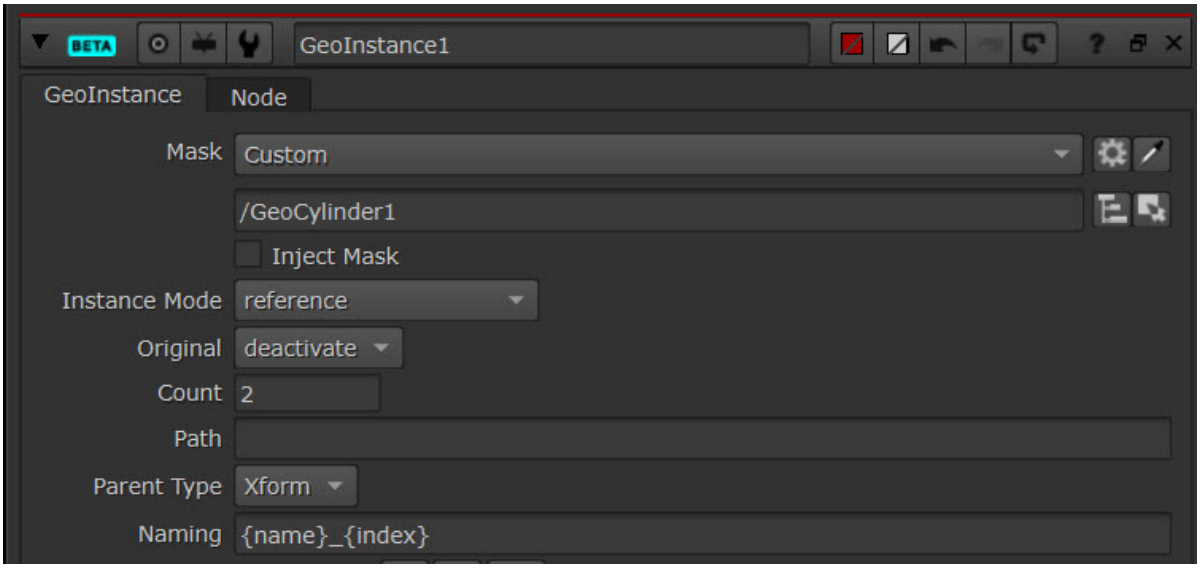
Instancing Geometry

GeoInstance allows you to create multiple copies of a prim, like GeoDuplicate, however all the copies from this node will be a form of instance, so any changes to the original prim will affect the instanced prims.

1. First, add a GeoInstance node downstream of your geometry. Set what you want to instance in the GeoInstance node Properties under **Mask**. In this example, we are selecting **/GeoCylinder1**.
2. Next, set the **Instance Mode**. The default Instance Mode is **reference**, which basically references the source prims for all the new instance copies being made. The second mode **instanceable reference** is a cheaper method of referencing the source prims but does not allow the results to be edited. Think of reference as a "read and write" mode and instanceable reference as a "read only" mode.

   The **point instancer** creates a single scene graph location containing all instances, which is computationally cheaper than **instanceable reference** mode, but the instances cannot be modified individually by downstream nodes. This mode is mostly useful for particles or other geometry with thousands of parts as you can only modify the whole instance at once.
3. You can then set if you want the **Original** prim you are instancing to be active or not in your scene. By default this will be set to **deactivated**, as the first instance will usually sit on top of the original prim data.
4. The **Count** lets you set the number of instances you want to generate, and the **Path** knob lets you set where you want your instances to live. By default this is blank and the instances will live in the same location as the originals.

   In this example, we've created two instances of the /GeoCylinder1:



5. The **Naming** knob also lets you set how you want your instances to be named. By default, the index token adds the instance number to make it an intuitive default name.



Note: You can also create instances using the GeoReference node. See Referencing with GeoReference.

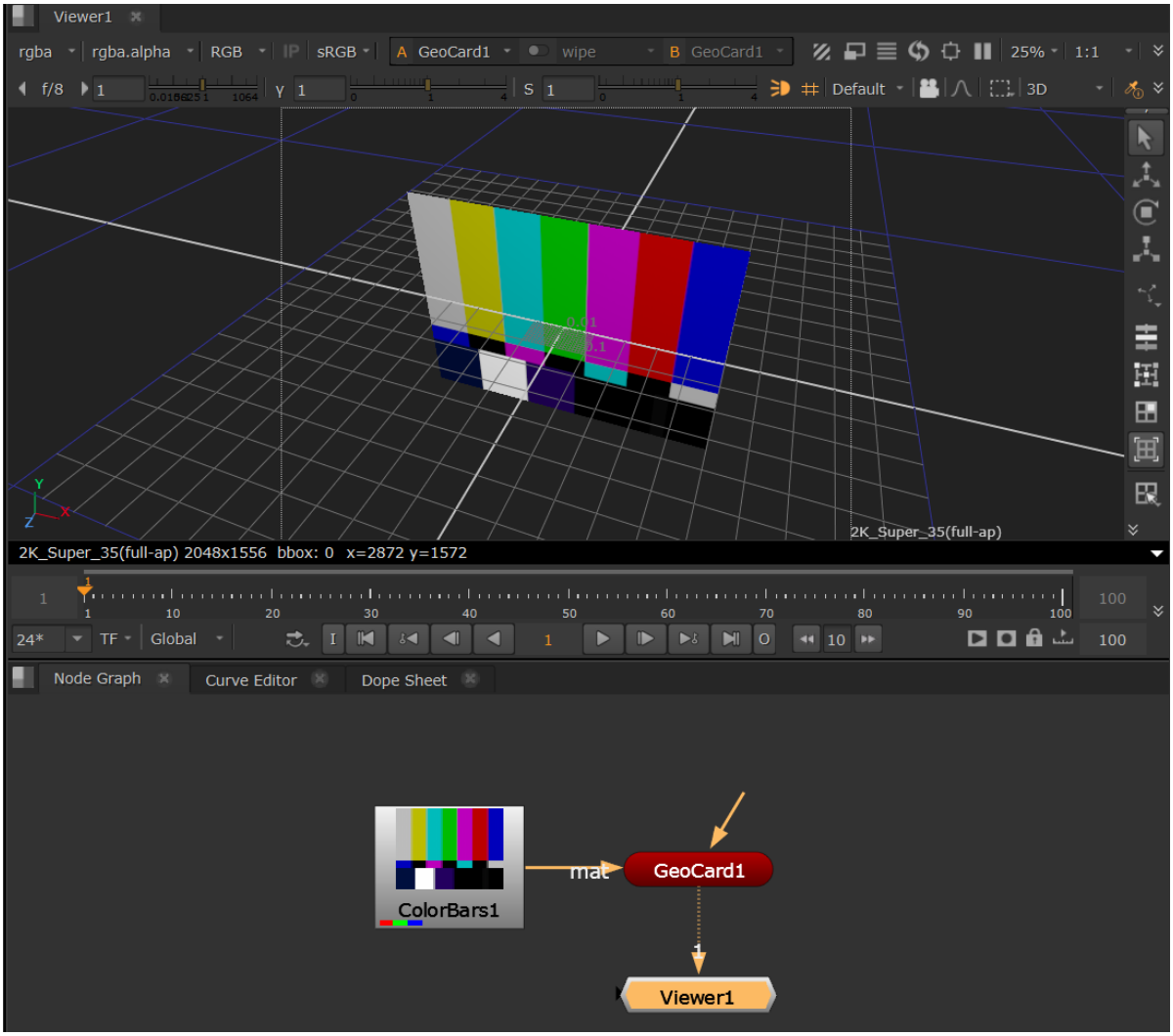
Considerations with Instancing

- Nuke doesn't support the import of instanced prims, so you'll need to either use GeoReference or GeoInstance, rather than GeoImport for these prims. There will be a warning message if you try to do this.
- Instances restrict what you can control in the scene graph. For example, you can't deactivate instanceable prims in the scene graph.
- Instances restrict other node processes. For example, you can't prune if your prim is a child of an instance, or bind materials to an instanced prim. There will be warning messages if you try to use GeoBindMaterial with an instanced prim.

Assigning Materials and Shaders

Working with materials and shaders is mostly consistent with the classic 3D system. If you want to attach a texture such as Checkerboard or ColorBars to a piece of geometry then there are two main workflows.

- The first being with a Nuke geometry node like GeoCard, where you take the **mat** input and connect this to any texture you want.



- The second workflow is for geometry that you may have imported or referenced using the GeoImport or GeoReference node. As these nodes don't have a **mat** input you can use the GeoBindMaterial node, which is the new 3D system alternative to ApplyMaterial.

See the following pages:

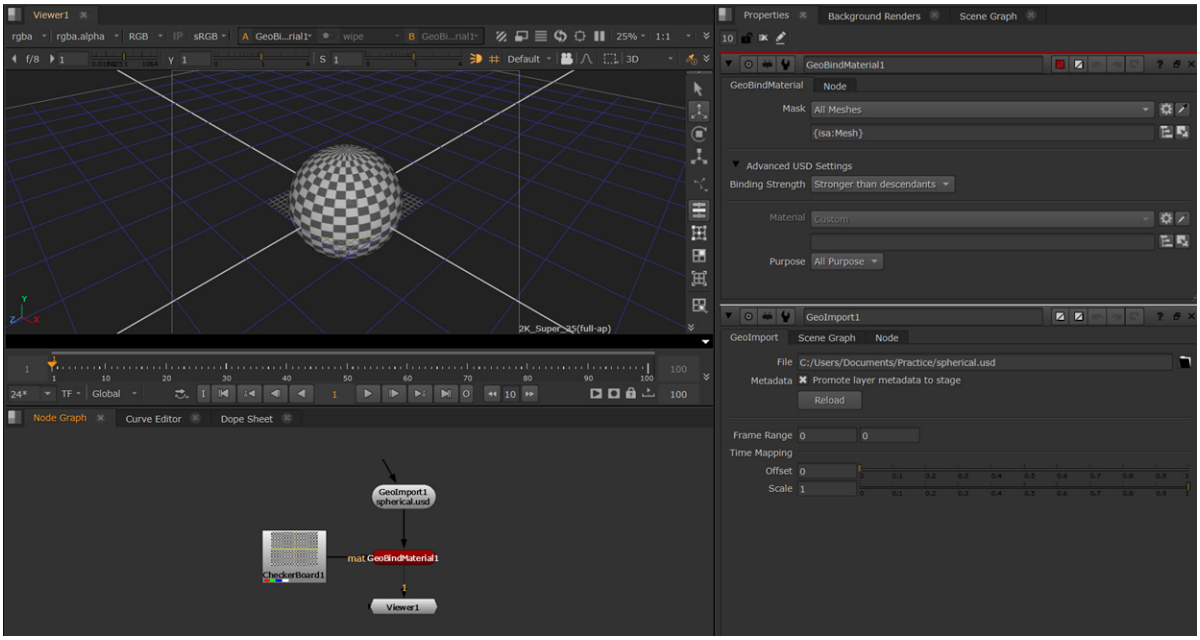
Using GeoBindMaterial to Assign Materials - how to bind materials to geometry.

Shading Geometry - how to use the nodes in the Shader menu to control what your objects seem to be made of.

Using GeoBindMaterial to Assign Materials

The GeoBindMaterial node allows you to bind a material to geometry if you've imported or referenced a USD file into your scene.

1. Use the GeoImport or GeoReference node to import geometry. In this example we are importing a simple sphere prim via the file **spherical.usd**.
2. Connect a GeoBindMaterial node downstream of your imported geometry.
3. Add a texture image via the Read node or using Nuke's generators such as the CheckerBoard node. Connect it to the GeoBindMaterial node's **mat** input.
4. The **Mask** knob defaults to 'All Meshes' essentially allowing any mesh that exists in the stage to have your desired material bound to it. If you want a specific target, you can choose custom and use the picker to select it. The material should now show up on your prim.



5. In USD, material bindings have their own purpose strengths and if you import geometry with existing material bindings, this may interfere with your material being applied. Under **Binding Strength**, the default is set to **Stronger than descendants**, so that your materials will show up regardless of this. However, if you want the opposite of this, you can choose **Weaker than descendants**, or if you want to use native USD behaviour, you can choose **Passthrough**.
6. The **Material** knob allows you to bind an existing material to a piece of geometry in your scene.

7. You can also use the **Purpose** dropdown to determine which rendering context the material applies to. By default, **Preview** bindings only show up in the Viewer, and **Full** bindings only show up in ScanlineRender. **All Purpose** can be viewed in both.

Note: In USD, Preview and Full bindings will always take precedence over All purpose bindings. By default, ScanlineRender will display Full bindings over All Purpose ones, but if something only has an All Purpose binding, it will display that.

Note: Instanced prim restrict how they can be edited and therefore you can't bind materials to them. See more about this at Instancing Geometry.

Shading Geometry

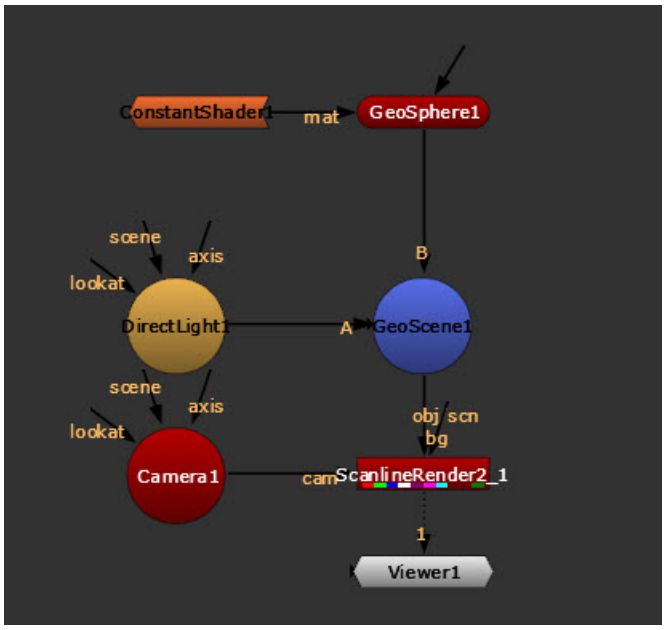
The nodes under the **3D** > **3D** > **Shader** node menu let you define the material attributes of geometric objects in a scene, including the quality of light reflected back to the camera from an object's surface. Using these nodes, you can control what material your objects seem to be made of.

You can also add several Shader nodes one after the other to produce more complex effects. For this, you should use the unlabeled inputs on the Shader nodes.

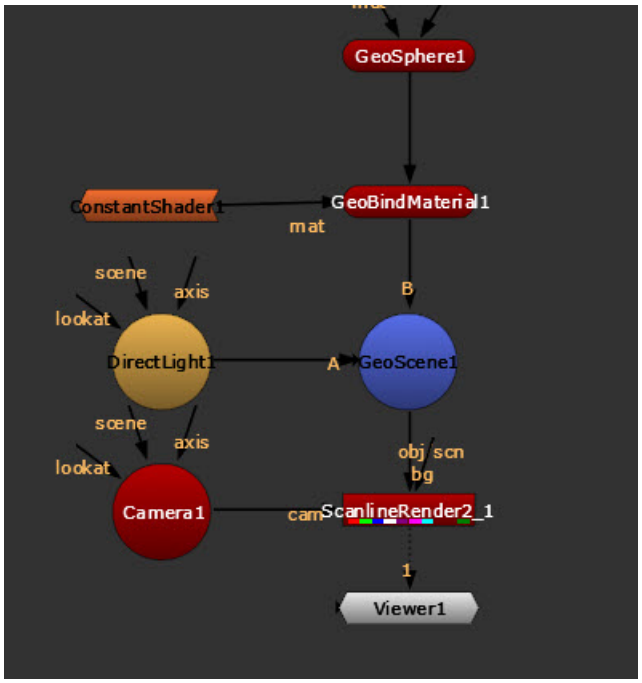
Node Graph

You can insert 3D shader nodes in the following places in your scripts:

• connected directly to the 3D object node's **mat** input, as in this image:



- Or, connected to the GeoBindMaterial node's **mat** input. This is a good way to apply a global material to all objects. See steps below.



Shaders with GeoBindMaterial

The GeoBindMaterial node applies a material from the mat input to your 3D object(s), so you can add a shader node here in order to change how they will appear.

1. Add a GeoBindMaterial node.

2. Connect the unnamed input of the GeoBindMaterial node to your geometry (for example, a GeoSphere node).



Note: If you want to apply a global material to several objects, you can also connect the unnamed input to a GeoScene node. This overrides any materials applied to the individual geometry nodes before they were merged.

3. Connect your shader, such as a ConstantShader node, to the GeoBindMaterial node's **mat** input. See more information about different types of shaders below.

Types of Shaders in the New 3D System

Shader Node	Usage	User Guide
ReflectiveSurface	The ReflectiveSurface node has additional controls for rendering reflections.	ReflectiveSurface: Generating Reflections
BasicSurface	The BasicSurface node creates a basic material where you can adjust diffuse, specular, and emission.	BasicSurface: Shader Usage
Project3DShader	You can use the Project3DShader node to project texture images onto your 3D objects. This way, you can add detail, surface texture, or color to your geometry, making the geometry more realistic and interesting.	Project3DShader: Projecting Textures
ConstantShader	The ConstantShader node is used to apply a material with constant color to a prim. ConstantShader has no inputs.	ConstantShader: Color
FillShader	The FillShader node lets you replace selected material channels with a constant color. Typically, you would use this node to make one object hold out the others. When you set the FillShader color to 0, it acts as a "3D cookie cutter" and makes a black hole where the material would otherwise be.	FillShader: Cut Out
WireFrameShader	The WireframeShader node allows you to render a	WireframeShader:

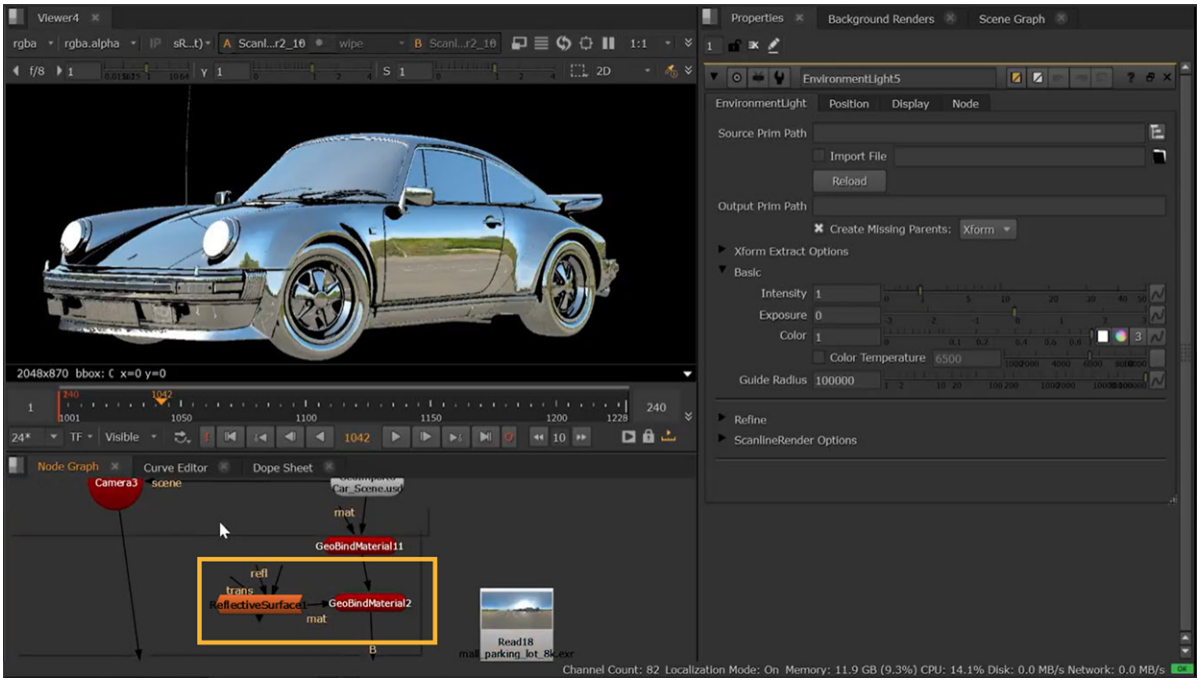
	wireframe overlay on the surface of your geometry object or particle simulation.	Applying Wireframe Overlay
MergeLayerShader	The MergeLayerShader allows you to combine two shader nodes together, using compositing algorithms like none, replace, over, and stencil. This is particularly useful for combining multiple Project3DShader nodes, allowing you to composite 2D images projected onto the 3D geometry atop each other.	MergeLayerShader: Merging Shaders
MtlXStandardSurface	The MtlXStandardSurface node supports MaterialX rendering.	MtlXStandardSurface: MaterialX Shader
PreviewSurface	The PreviewSurface node represents the preview shader from the Universal Scene Description (USD) standard. It is a basic material that can be used as a standard solution when switching between pipelines and different applications. For example, a game pipeline may rely on metallic workflows and film on specular workflows. **Warning:** PreviewSurface node is still in Beta and may error when a texture input is connected. We suggest the use of the MtlXStandardSurface node.	PreviewSurface: USD Workflows

ReflectiveSurface: Generating Reflections

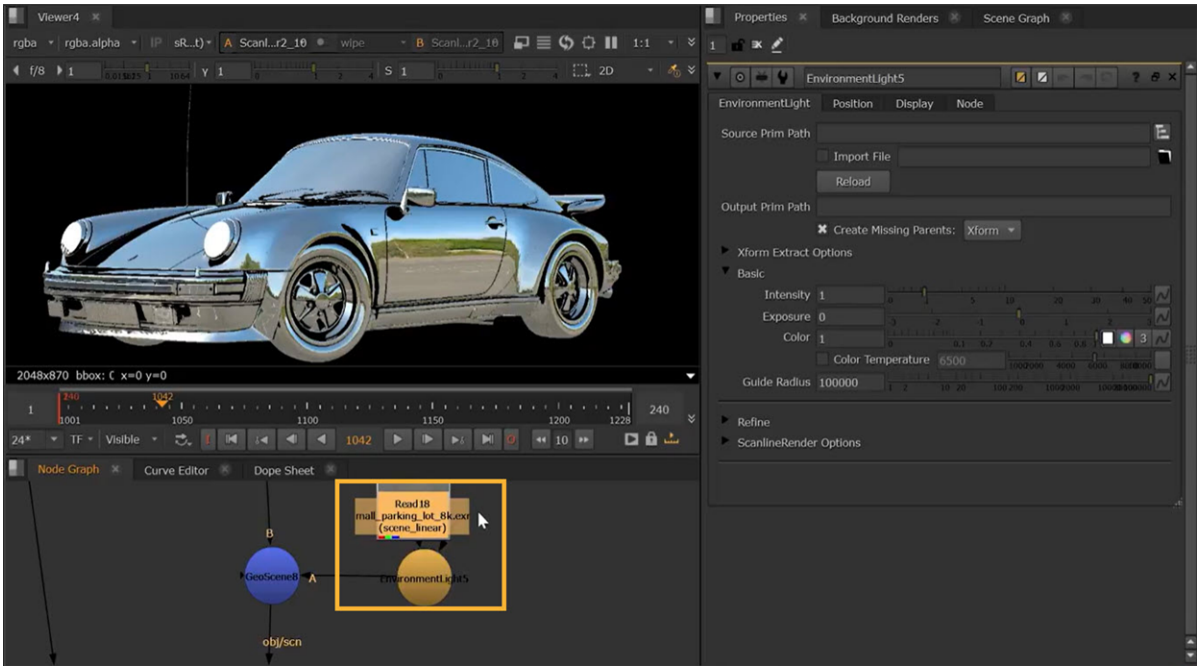
The ReflectiveSurface node allows you to generate mirror-like reflections, translucency, transmission and more from ScanlineRender2's ray-tracing architecture.

1. Connect the ReflectiveSurface shader node to the GeoBindMaterial's **mat** input, and place both downstream of your geometry.

Note: To view the reflections, you'll need to connect a ScanlineRender2 and a Camera node and view the results in the **2D** panel.



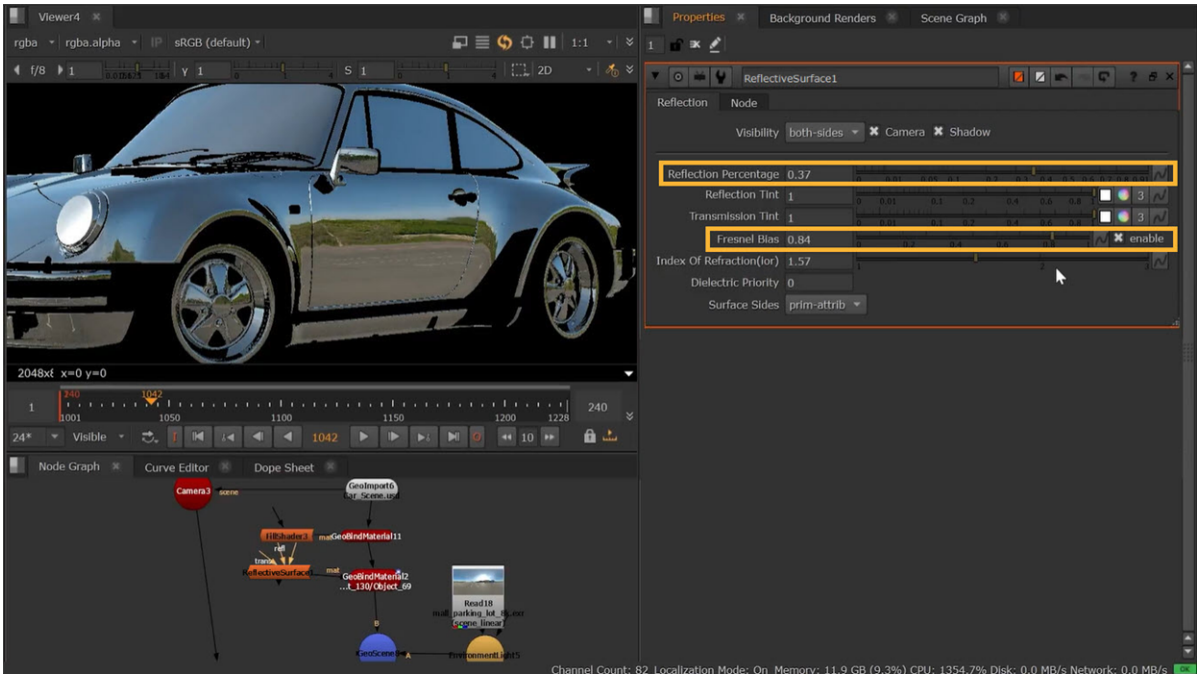
2. Add a GeoScene node and connect any lights that will be reflected in your scene. For example, you could add an EnvironmentLight to reflect an image map off your geometry. See Light Your Scene.



3. In the ReflectiveSurface node Properties, the **Visibility** knobs allow you to set if the material is visible to each of the respective knobs, so enabling **Camera** means the material will be visible to the render

camera and disabling it would make the material invisible to the render camera which could be useful if setting up a HDR dome that you want in reflections, but not in the background.

In this example, we are using the **Fresnel Bias** option and lowering the **Reflection Percentage** to avoid visual artifacts:



The **Reflection Percentage** knob lets you balance between reflection and transmission with 1 being a fully reflective surface and 0 being a transmissive surface. The **Fresnel Bias** sets the amount of reflectivity you see, typically based on the angle that you look at a surface.

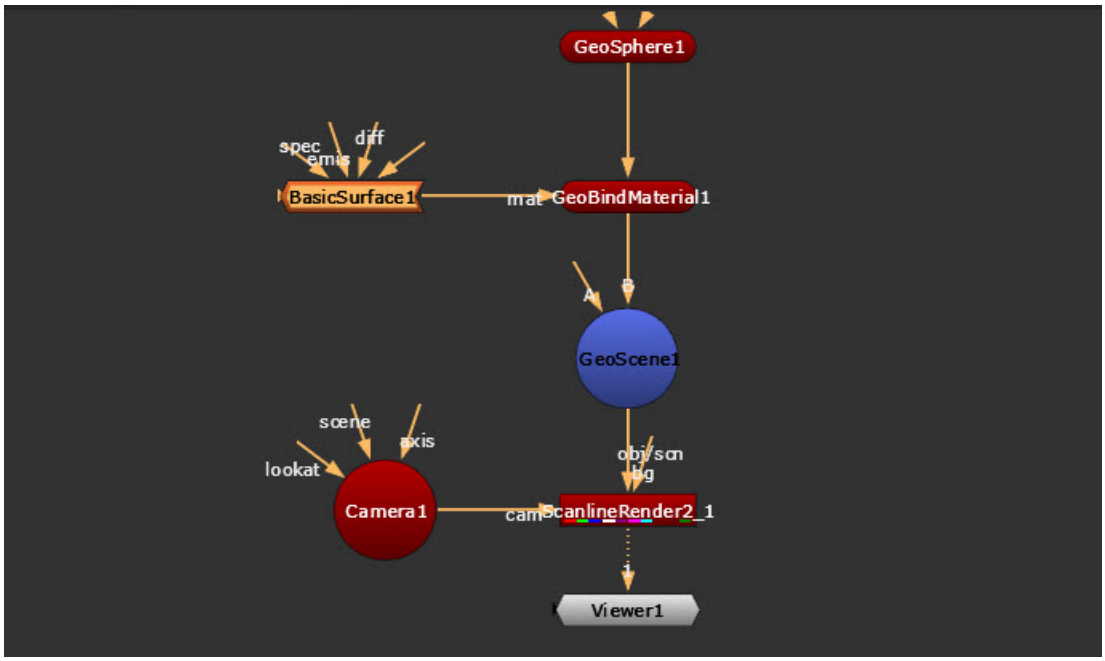
4. Adjust further options as needed. The **Reflection Tint** alters the reflection color. **Transmission Tint** doing the same for transmission. The **Index of Refraction** lets you control how much the light is bent when going through the material. The **Surface Sides** knob essentially lets you set if the renderer should render a single side or both sides.
5. You can then view your 2D render and make further adjustments in the ScanlineRender node, such as adjusting **Ray Options**. See Ray-tracing with ScanlineRender to learn more about the options for controlling reflections and refractions inside the ScanlineRender node.

BasicSurface: Shader Usage

The BasicSurface node combines diffuse, specular, and emission controls, allowing you to adjust all three aspects of the material from a single properties panel.

1. Insert a BasicSurface node into your script.

2. Place the BasicSurface node between your 2D texture image and your 3D object node, or connect it to a GeoBindMaterial node's **mat** input.



3. The **Visibility** knobs allow you to set if the material is visible to each of the respective knobs, so enabling camera means the material will be visible to the render camera and disabling it would make the material invisible to the render camera which could be useful if setting up a HDR dome that you want in reflections, but not in the background.
4. The BasicSurface node has several map inputs you can use to mask the effect of the node. Use:

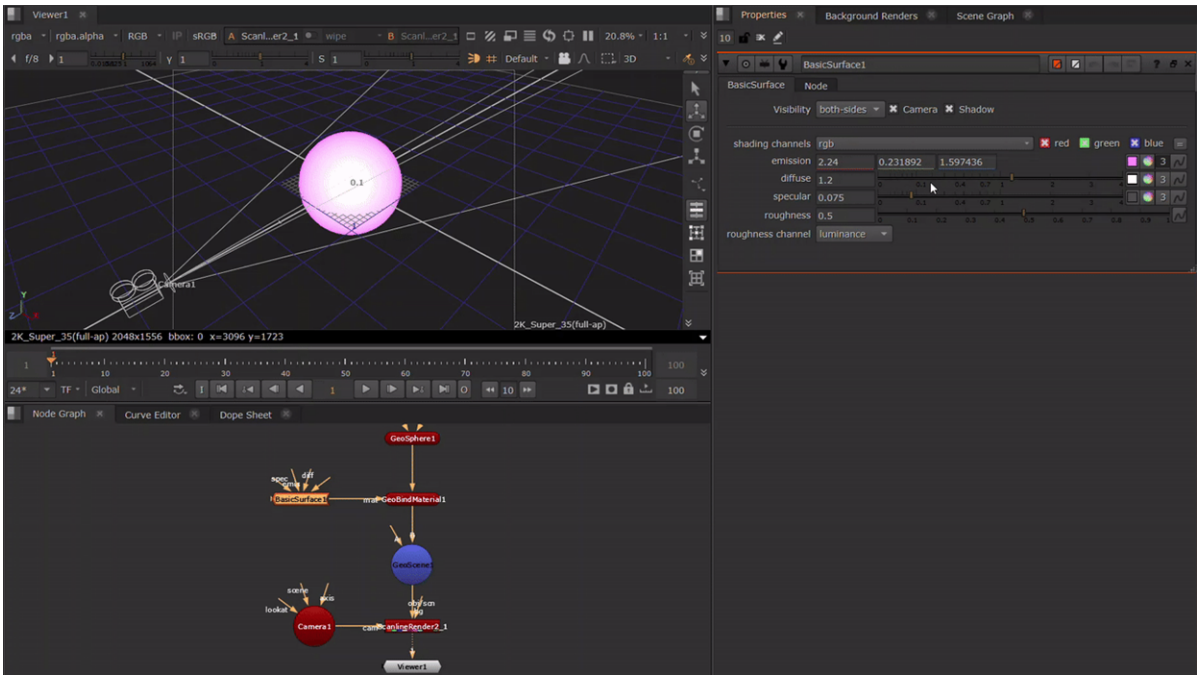
• **diff** to modulate the diffuse component,

• **spec** to modulate the specular component,

• **emis** to modulate the emission component,

• **rough** to module roughness. Note that this input is hidden on the left side of the node.

5. In the BasicSurface properties, use the **shading channels** dropdown menu to select the channels you wish to process.
6. Adjust **emission** to change the color of the light the material emits. Note that when you have an image connected to the unlabeled input of the BasicSurface node and adjust this value, you need to look at the rendered 2D image to see the effect of your changes.
7. Adjust **diffuse** to control the color of the material when illuminated.

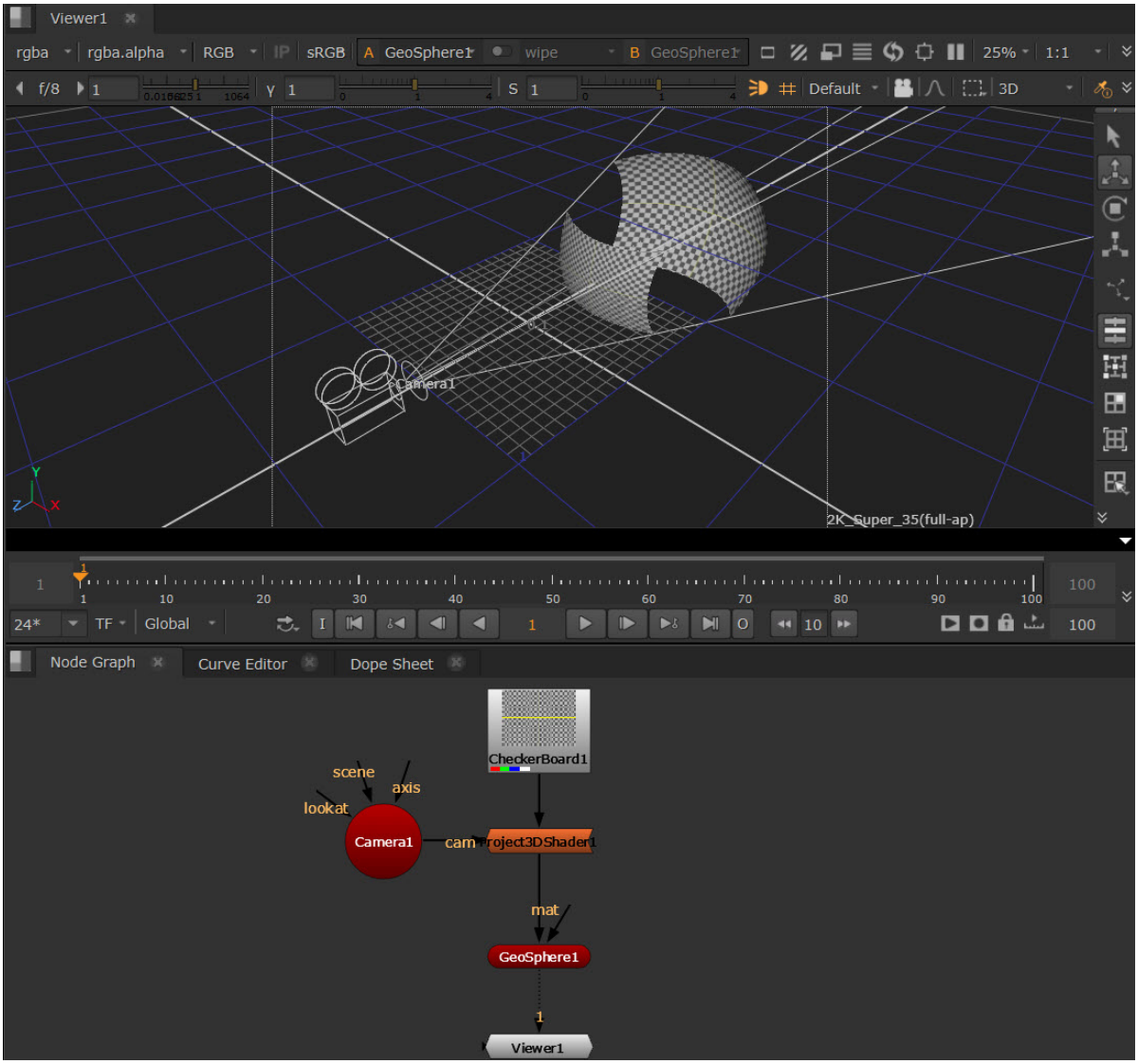


8. Use **specular** to control how bright the highlights on the material seem, and **roughness** to adjust roughness.

Project3DShader: Projecting Textures

The Project3DShader node projects an input image through a camera onto the 3D object.

1. Select **3D** > **3D** > **Shader** > **Project3DShader** to insert a Project3DShader node downstream of the image you want to project (for example a CheckerBoard).
2. Connect a Camera node to the Project3DShader node's **cam** input.
3. Insert a 3D geometry node (a GeoSphere for example) after the Project3DShader node.
4. Attach a **Viewer** to the 3D geometry node.



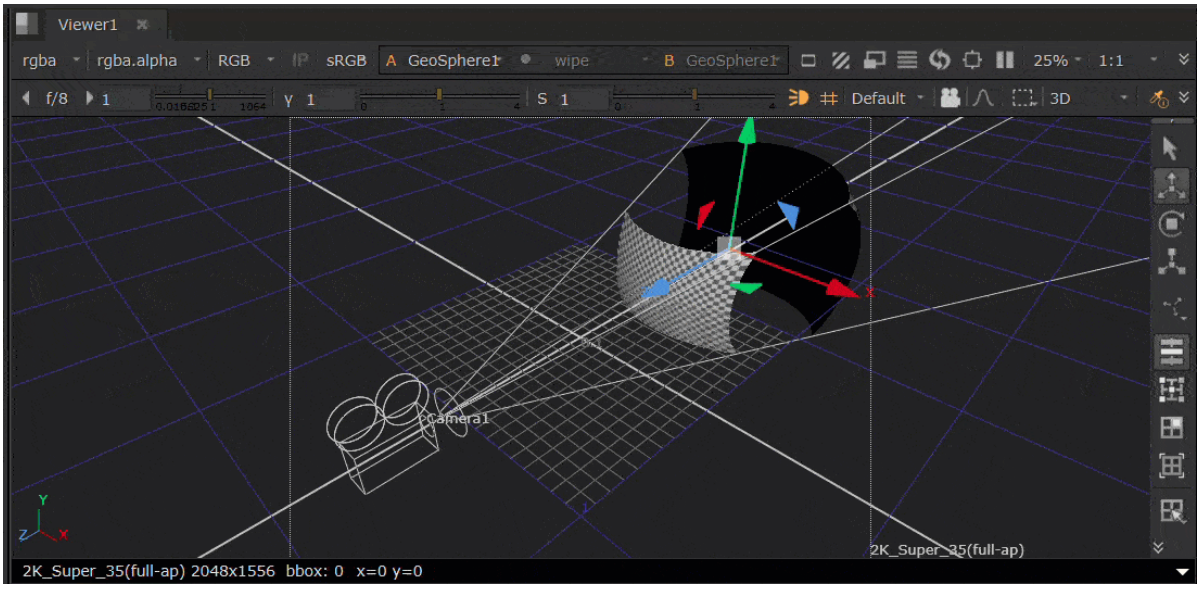
5. From the **Project On** dropdown menu, select to project the image on either the front facing, back facing, or both polygons.
6. To extend the input image at its edges with black, check **Crop To Format**. To extend the image with the edge colors, uncheck this. You can also check or uncheck **Near Plane Clip** or **Far Plane Clip** to clip the projection beyond the cameras near and far planes.
7. If you want to use ray casting to test the projection and find out which parts of it are occluded, you can use the **Occlusion Mode** dropdown. Select from:

• **none** - to disable occlusion testing.

• **self** - to tell Project3DShader that only the geometry connected to it can cause occlusion.

• **world** - to tell Project3DShader that other objects in the scene can cause occlusion.

8. You can then transform the Camera to position the material.



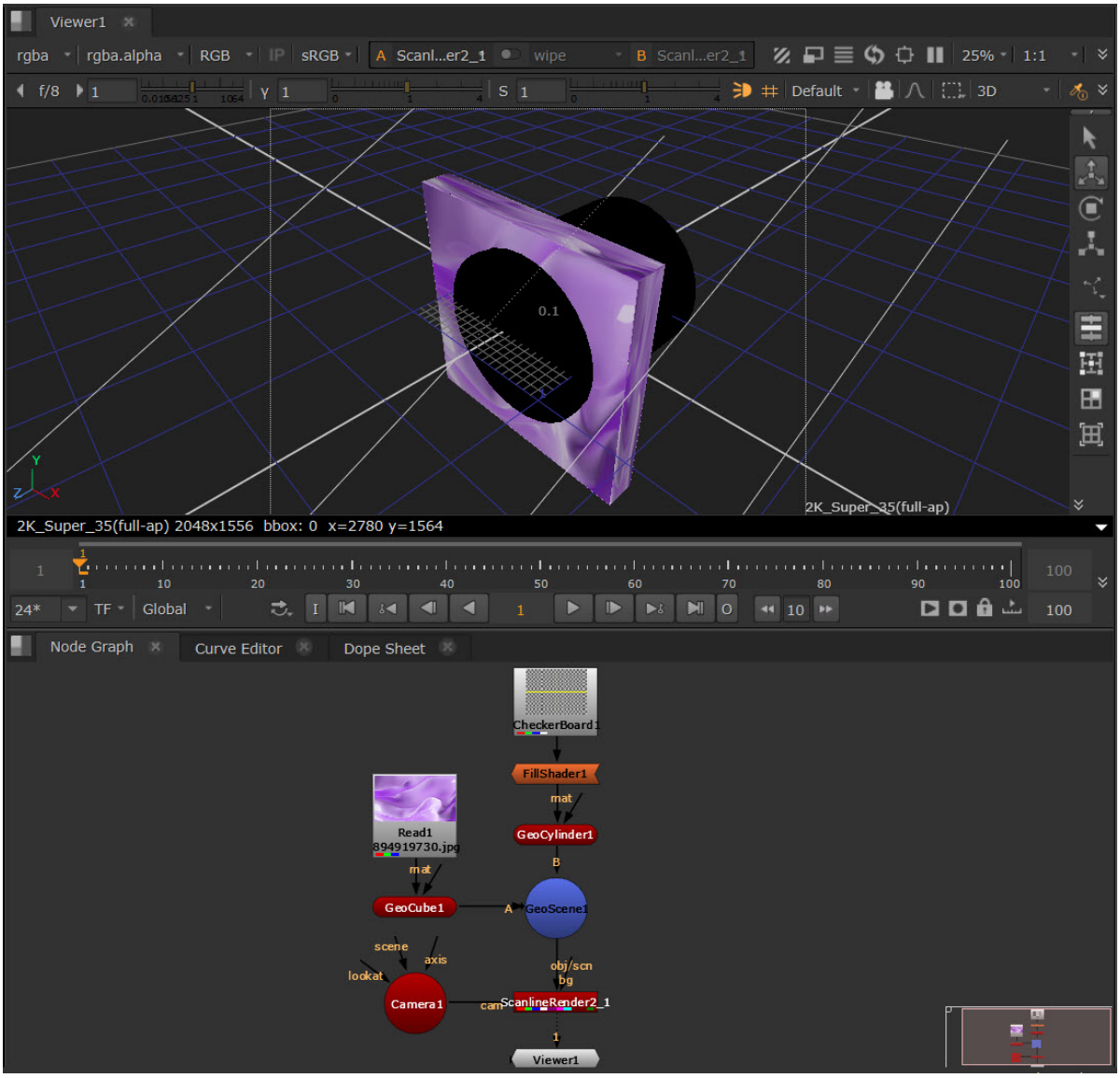
Video: See some examples of using the Project3DShader in our Projections in 3D Video.

FillShader: Cut Out

The FillShader node lets you replace selected material channels with a constant color. This is similar to using a Constant node as the input texture. However, the advantage of using the FillShader node is that you can easily apply it to the alpha channel in addition to the rgb channels. Another advantage is that the FillShader node doesn't break the shading sequence, so you can insert it after other material nodes in your node graph.

In this example, we are connecting the FillShader to the **mat** input of a GeoCylinder in order to cut out the center of a GeoCube.

1. Insert a FillShader node between the 2D image you're using for the surface texture and the 3D object node that creates the surface.



2. In the FillShader controls, use the channels controls to select the **channels** you want to replace with a constant color.
3. Use the **color** control to select the constant color. By default, this is set to black (0).

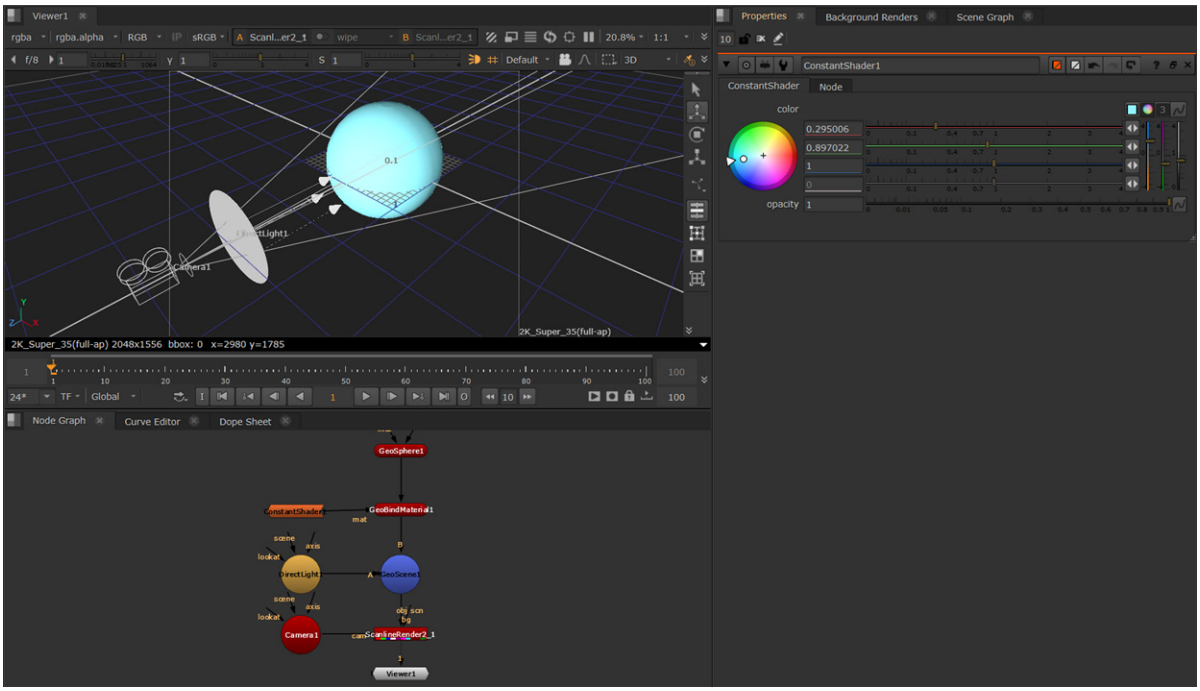
ConstantShader: Color

The ConstantShader simply allows you to control the color and opacity, and has no additional inputs.

In this example we have applied a blue constant color to the GeoSphere.

1. Connect a ConstantShader node to a GeoBindMaterial node's **mat** input.

2. In the Properties, adjust the **color** values or use the color wheel to pick a color to apply.
3. Adjust the **opacity** value if needed.



WireframeShader: Applying Wireframe Overlay

The WireFrameShader node allows you to render a wireframe overlay on the surface of your geometry object or particle simulation. This can be useful, for example, if you want to:

• check that your texture projection correctly lines up with your geometry.

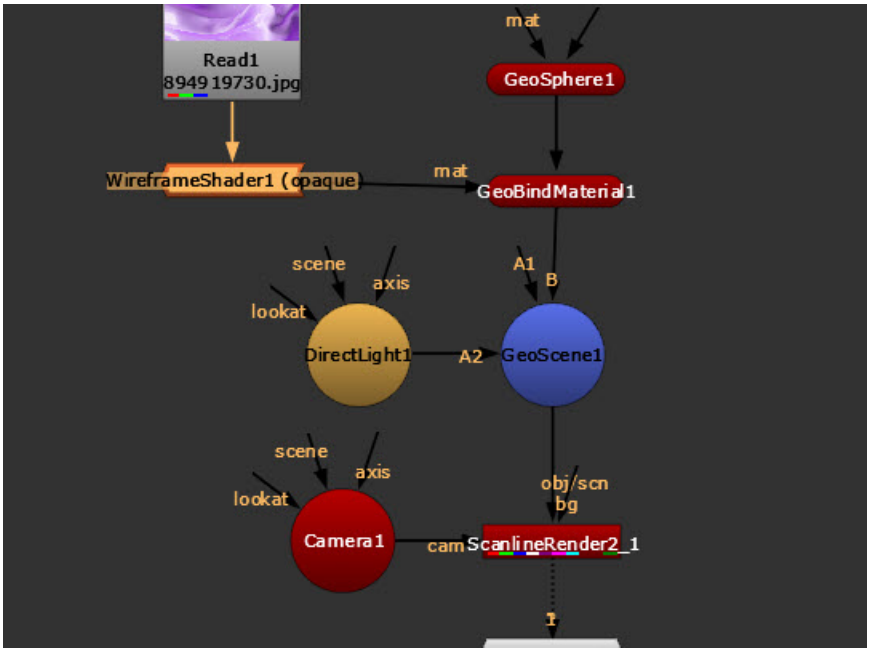
• create a quick render of your 3D scene to check the positioning of objects.

• create motion graphics.

• create "making of" videos.

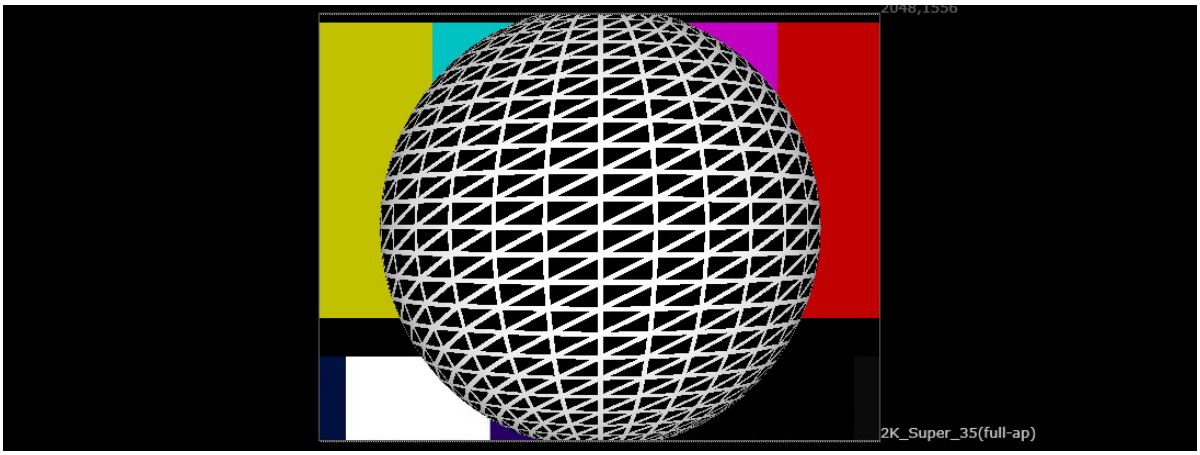
Note: The WireframeShader node currently only works if you are rendering your 3D scene using ScanlineRender.

1. Insert WireFrameShader node into your script.
2. Place the WireFrameShader node between your 2D texture image and your 3D object node, or connect it to a GeoBindMaterial node's **mat** input.

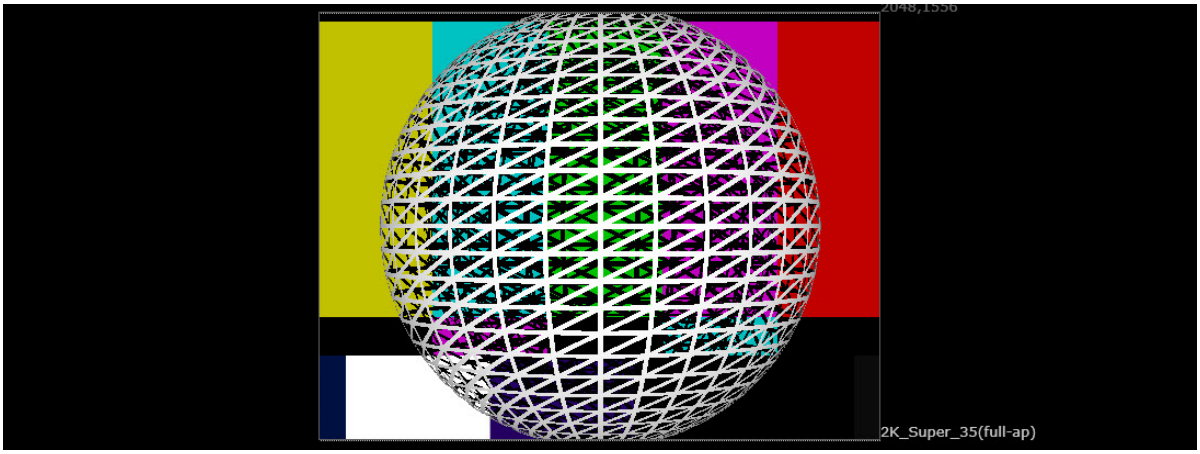


3. In the WireFrameShader properties, use the **channels** dropdown menu to select the channels you wish to process.
4. From the **operation** dropdown, select how to apply the wireframe overlay to your geometry:

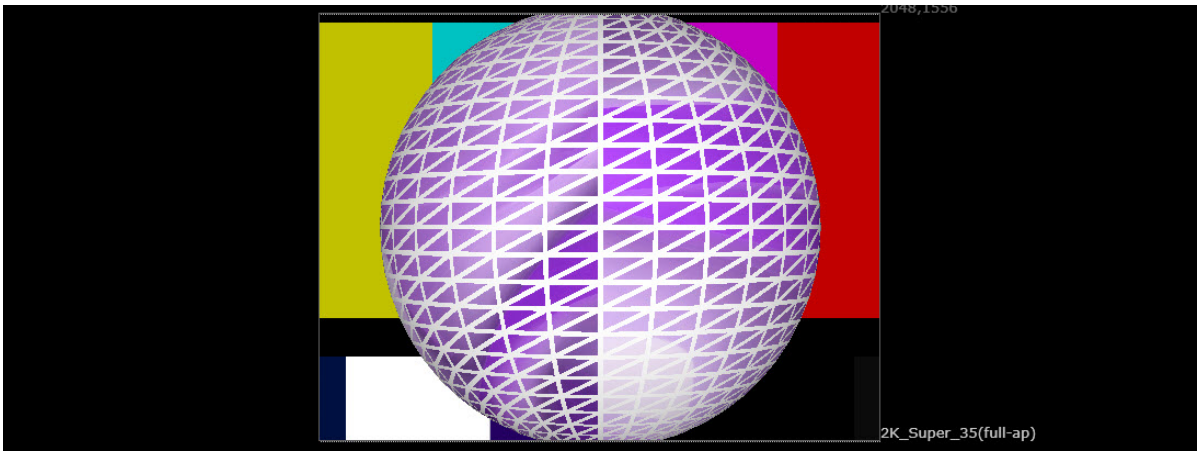
• **opaque** - display the wireframe on fully opaque black input geometry.



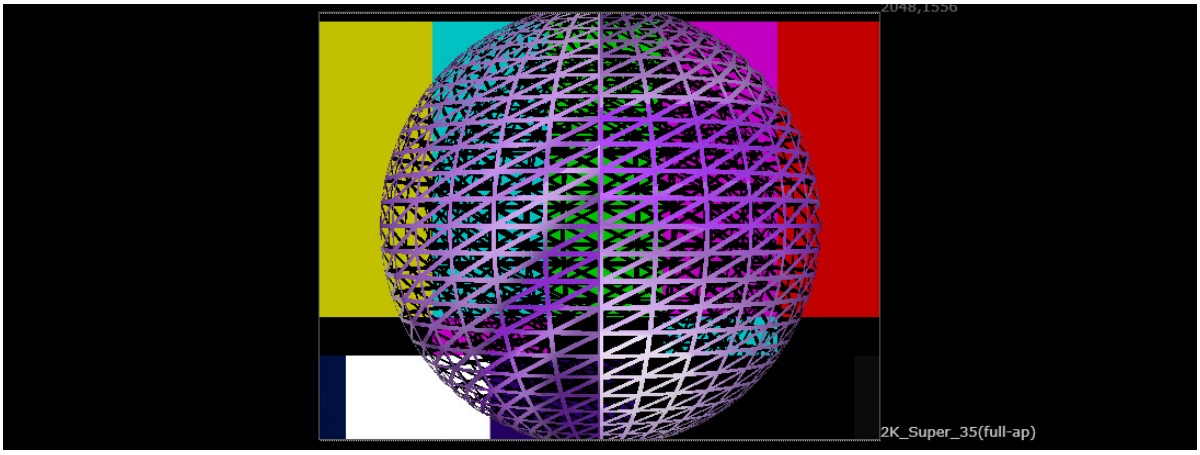
• **see through** - displays wires over a fully transparent geometry.



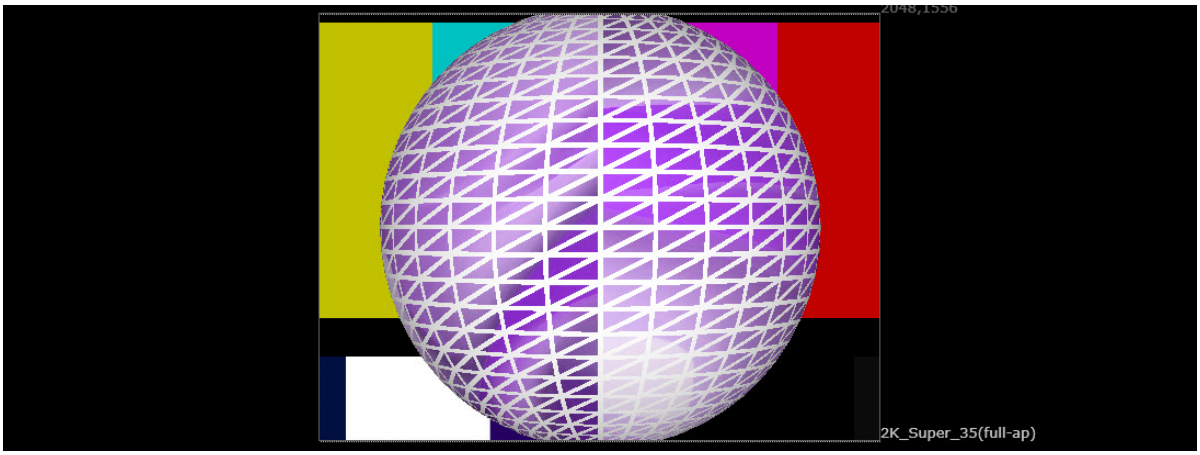
- **over** - display the wireframe on top of the input shader or texture.



- **multiply** - multiply the wireframe by the input shader or texture and display it on fully transparent geometry.



- **modulate** - apply standard diffuse shading to the wireframe and display it on top of the input shader or texture. This takes into account any lights in the scene.



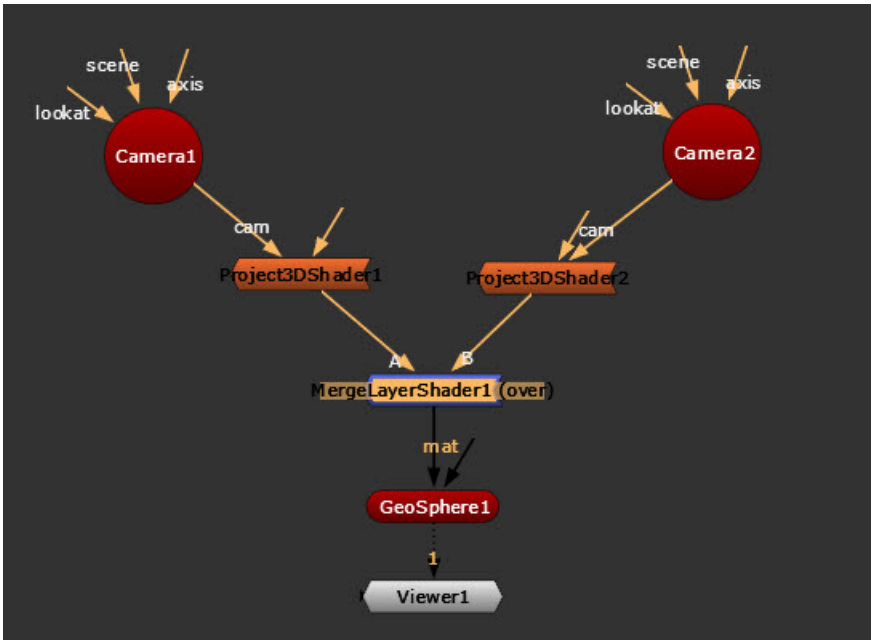
5. To set the width of the wireframe lines (in pixels), adjust **line width**.
6. To set the color and transparency of the wireframe lines, adjust **line color**.

MergeLayerShader: Merging Shaders

With the MergeLayerShader node, you can combine two shader nodes together, using compositing algorithms like none, replace, over, and stencil. The MergeLayerShader node is particularly useful for combining multiple Project3DShader nodes, allowing you to composite 2D images projected onto the 3D geometry atop each other.

1. Add a MergeLayerShader(over) node after the two shader nodes you want to combine.
2. Connect the MergeLayerShader node to the **mat** input of the 3D object you want to project the images on.
3. Connect the shader nodes to the MergeLayerShader node's **A** and **B** inputs. **A** refers to the foreground element, and **B** to the background element.

   For example, if you wanted to combine two Project3DShader nodes and composite their results onto a GeoSphere, your node tree would look something like the following:



4. From the operation dropdown menu, select how you want to composite the results of the two shader nodes together:

- to only use input **B** in the composite, select **none**.
- to only use input **A** in the composite, select **replace**.
- to composite input **A** over input **B** using a mask, select **over**.
- to use input **B** outside the mask area, select **stencil**.
- to use input **B** inside the mask area, select **mask**.
- to add input **B** to input **A**, select **plus**.
- to use input **A** if it is greater than input **B** or else use input **B**, select **max**.
- to use input **A** if it is less than input **B** or else use input **B**, select **min**.

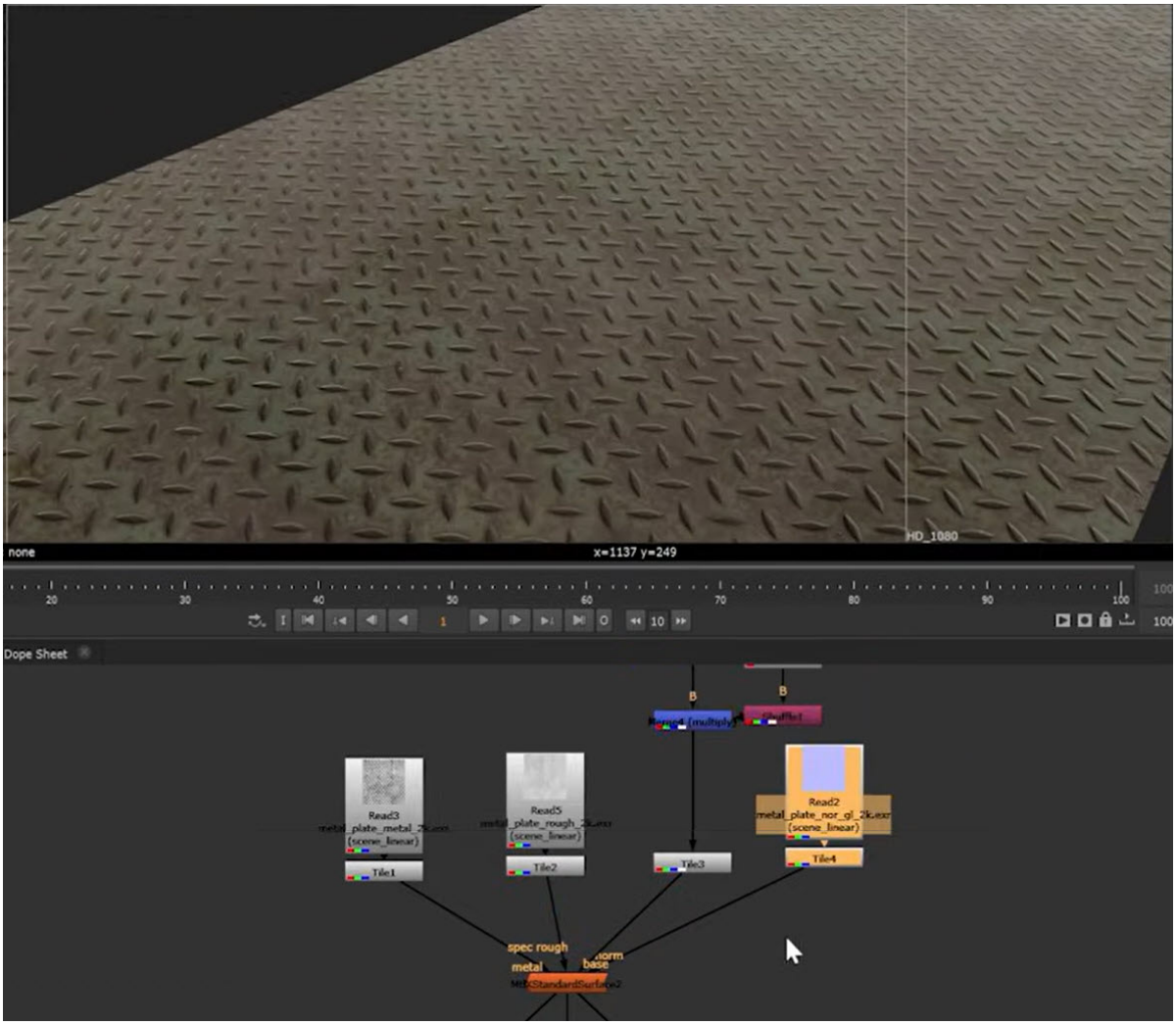
5. For operations (such as **over**) that need an alpha channel (mask), select which channel to use for the alpha from the alpha **channels** dropdown menu.

MtlXStandardSurface: MaterialX Shader

The MtlXStandardSurface node allows you to generate your own MaterialX shaders in Nuke.

In this example, we will use MtlXStandardSurface to add base, normal, specular roughness, and metalness texture maps to a metal floor sheet to give a realistic metallic look.

1. Insert a MtlXStandardSurface node into your scene's node graph.
2. Use Read nodes to add a texture source images and connect them to the MtlXStandardSurface node's **base**, **norm**, **spec rough**, and **metal** inputs.

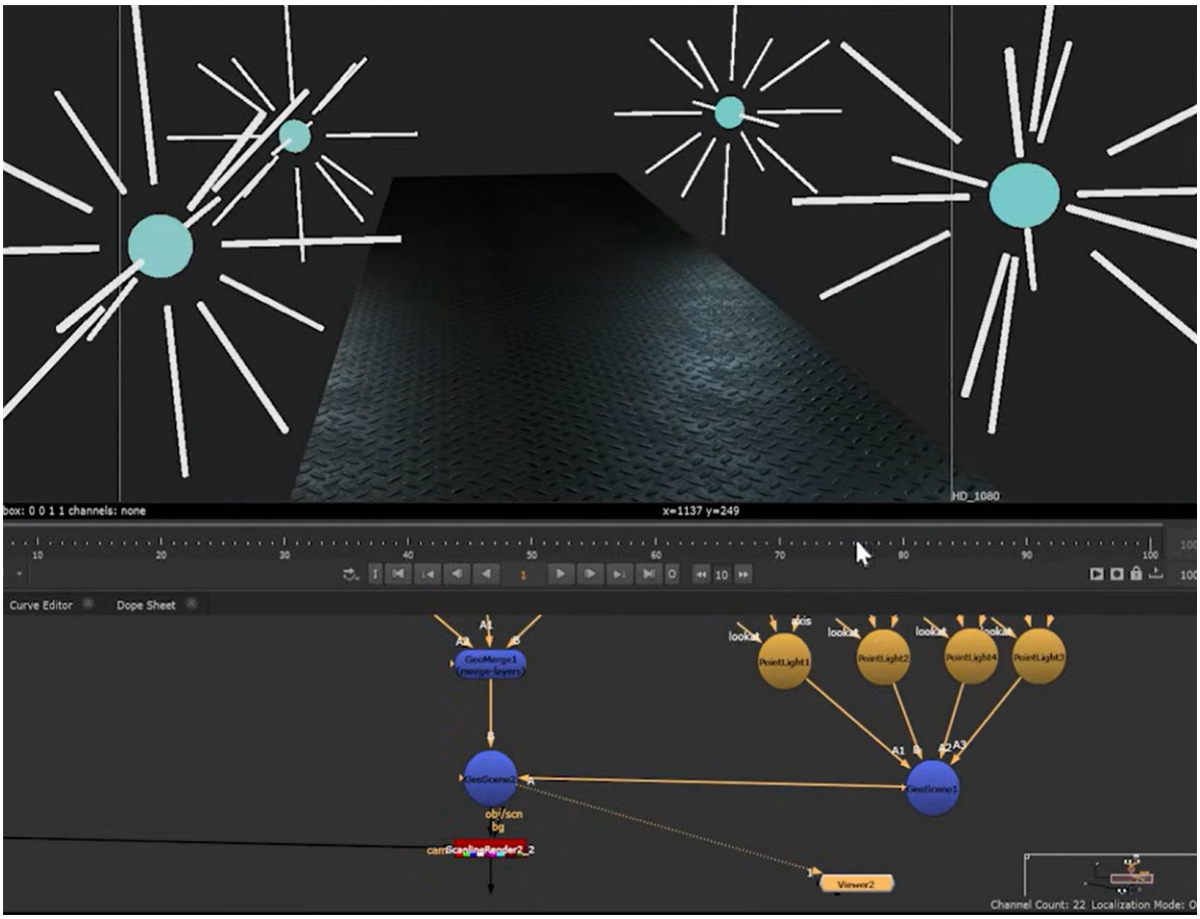


Base drives the main texture of the surface. Norm controls the main surface normal used for lighting and reflections, helping to protrude and break up the surface.

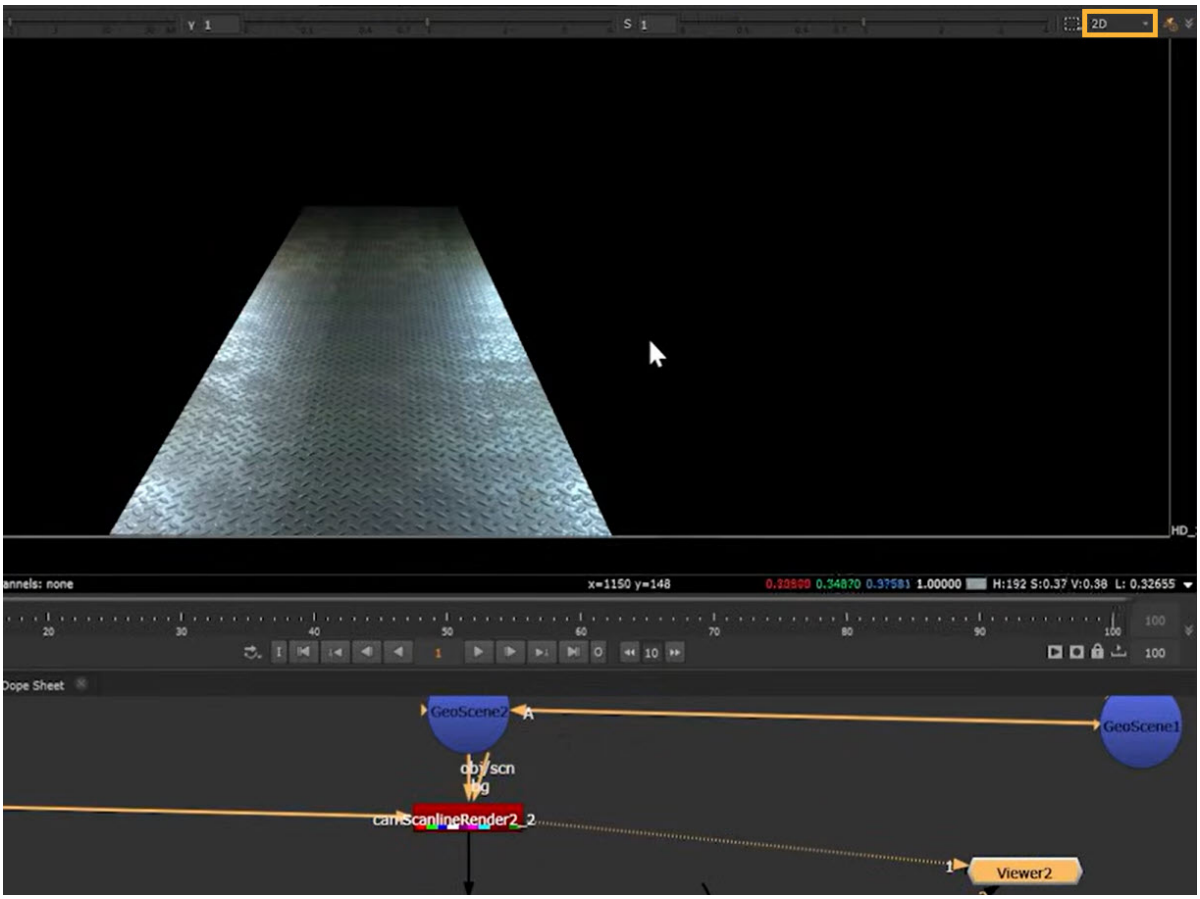
The specular roughness controls how sharp or soft reflections appear across the surface, and metalness controls how metallic the surface behaves.

Note: There are several other inputs such as coat and emission - see MtlXStandardSurface in the Reference Guide to learn more about these.

3. By adding lights to the scene, you can see the textures and reflections more realistically. Lights can be connected via GeoScene. Here we have added several PointLight nodes. Read more about how to use lights at Using Nuke Lights.



4. Add a ScanlineRender node downstream of your scene and switch to **2D** mode in the Viewer to see how the render will look.



Video: Watch our Lighting MaterialX for better renders Video to see this example in action, as well as further examples using coat and coat normals, and workflows referencing in pre-existing MaterialX materials.

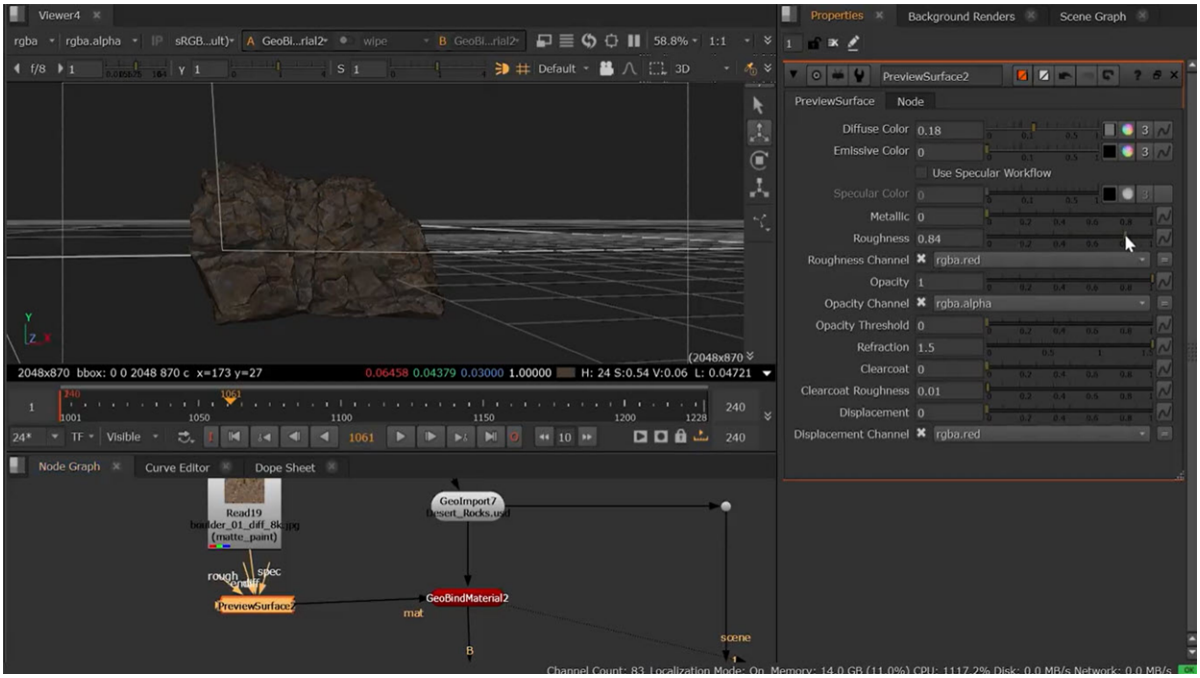
PreviewSurface: USD Workflows

The PreviewSurface node is a representation of the "USDPreviewSurface" shader which is meant to model a "modern" physically based surface with both specular and metalness workflows. PreviewSurface supports these either by enabling or disabling the Use Specular Workflow control. It has a variety of inputs such as roughness, specular, emission, diffuse, as well as hidden inputs for opacity, normals, AOVs, displacement.

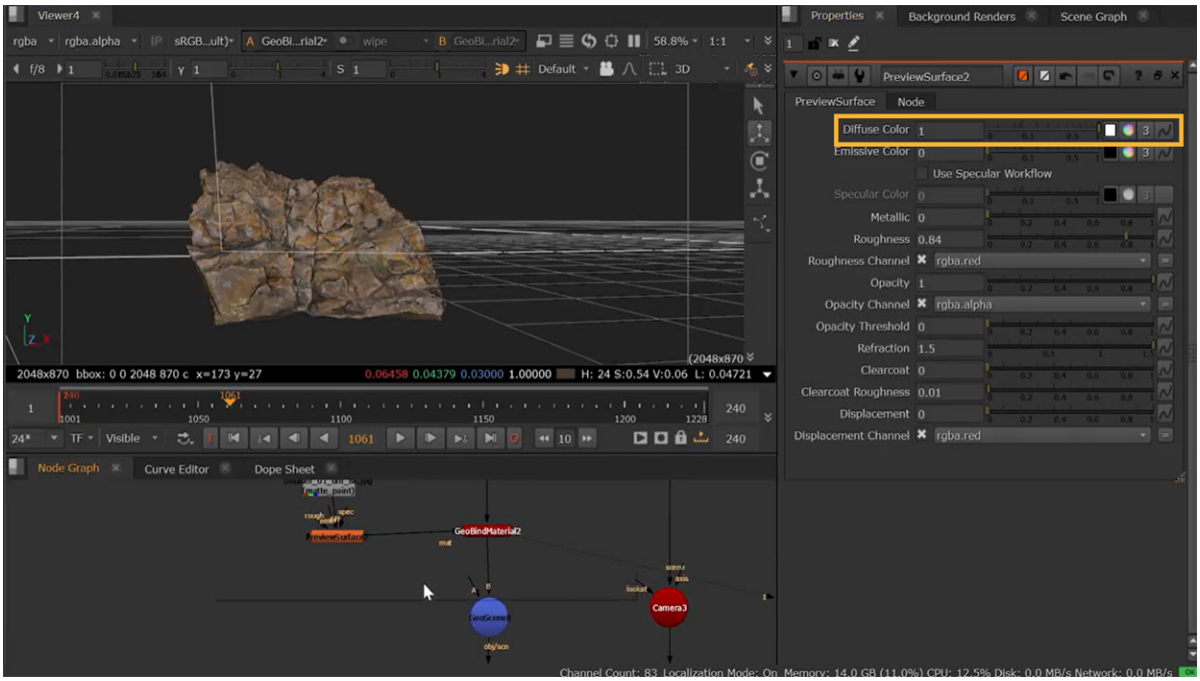
Warning: PreviewSurface node is still in Beta and may error when a texture input is connected. We suggest the use of the MtlXStandardSurface node.

In this example, we are using a PreviewSurface shader to add **diffuse** elements to an imported **.usd** prim.

1. Import a **.usd** file into your scene with GeoImport. See Importing with GeoImport to learn more about this.
2. Use a Read node to add your texture source image and connect it to the PreviewSurface node's **diff** input.
3. Connect the PreviewSurface shader node to the GeoBindMaterial's **mat** input.



4. Assemble the rest of your scene, for example with GeoScene and any lighting needed.
5. Adjust the knobs to achieve your desired effect. For example, here we are turning up the **Diffuse Color** to achieve a brighter surface.



Note: With the **Use Specular Workflow** checkbox turned **off**, increasing the **Metallic** knob can create a shiny metal look, preferred by games pipelines. For a similar effect for film pipelines, you may prefer turn the **Use Specular Workflow** checkbox **on**, and instead adjust **Specular Color**.

Light Your Scene

The New 3D System includes:

- **Nuke** lights: SpotLight, DirectLight, PointLight, and EnvironmentLight.
- **USD** lights: GeoDiskLight, GeoDistantLight, GeoDomeLight, and GeoSphereLight.

What are Nuke lights?

You can create Nuke lights via SpotLight, DirectLight, PointLight, and EnvironmentLight. These lights are versatile and will fit most use cases, but they involve some conversion when working with USD. As the light parameters are based on USD values, you will see equivalent knobs for each light, but they may not be identical. When you use these lights in a USD stage, you'll need to output the light to a new prim path in

order to include it in your stage. This is done automatically when you connect to a GeoScene node. If you don't set a prim path yourself - they will appear in your stage as **{nodename}_Inject**.


See more at Using Nuke Lights.

What are USD lights?

You can create and edit USD-native lights directly in Nuke via GeoDiskLight, GeoDistantLight, GeoDomeLight, and GeoSphereLight. This means that the parameters of the lights map over USD values seamlessly, without need to convert the data. These are intended for when you have USD lights in your stage that you want to edit directly, for example, if you have imported a USD stage into Nuke. To edit imported light prims, you can also use the GeoEditLight node, which has the same set of options, once you pick the light type under **Schema**.


See more at Using USD Lights.

Using Nuke Lights

There are four Nuke light nodes in Nuke's New 3D System. You can create and fully edit these lights inside Nuke. Their node parameters are based on USD values to make sure they align across your pipelines, but this involves some conversion, as explained below.

Light Node	USD Equivalent
DirectLight	DistantLight
EnvironmentLight	DomeLight
PointLight	SphereLight
SpotLight	DiskLight

How does the conversion work?

The lights are converted to the USD stage to the location in the **Output Prim Path** knob. You can set the prim path where the light will be created, or this is done automatically once you connect them to a GeoScene node. By default, the converted light will appear in your scene graph as **{nodename}_Inject**.

This process essentially converts the light from a Nuke shader light to a USD light prim for use in other DCCs, which values matching as close as possible, but note these won't necessarily be 1:1. See details about each light below.

Tip: If you want to avoid conversions to USD, you can create and edit USD-native lights directly, see Using USD Lights.

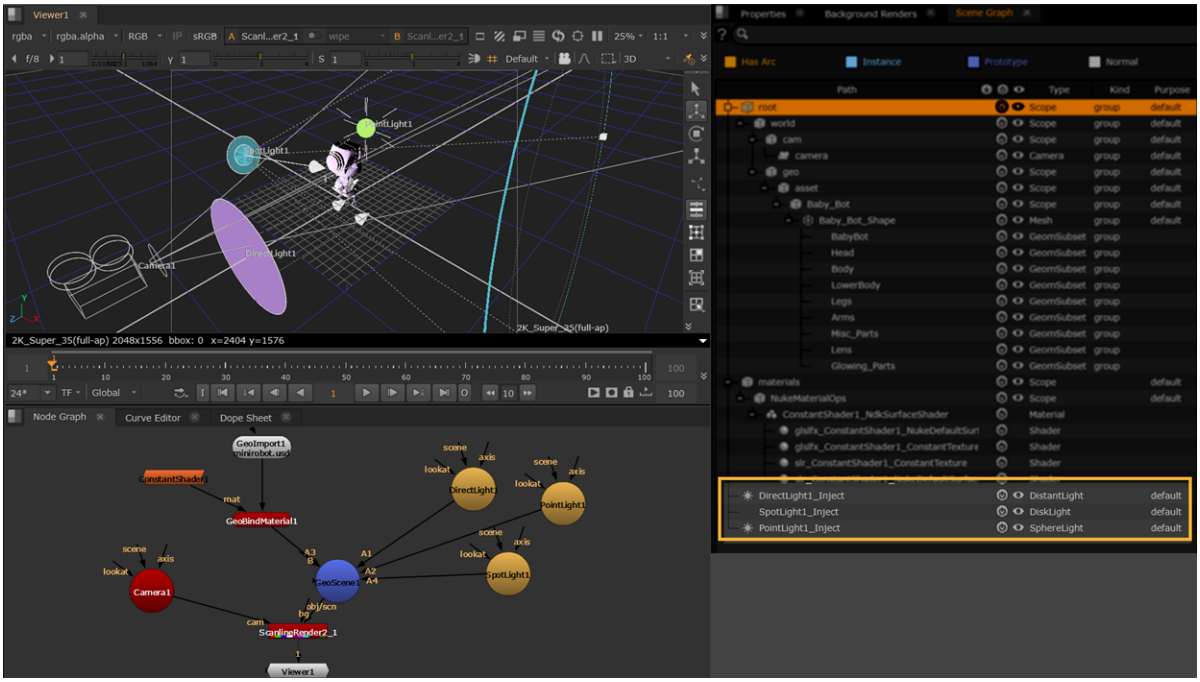
Note: Learn more about the different types of light at Light Your Scene.

Adding Lights to Your Scene

- To add a new light, add any of the light nodes to your node graph and connect it to the GeoScene node that is downstream of your geometry.

Note: At this point, you can also import light data to use - see Import/Export below.

- Once connected to GeoScene, the lights will then be added to the Scene Graph automatically under **{nodename}_Inject**. Or if you wish to adjust the prim path, edit the **Output Prim Path**.
- In this example, we have added three of the Nuke lights: DirectLight, PointLight and SpotLight. (We have also changed the **Color** of each of them in the Node **Properties**).
- The scene is then connected to a ScanlineRender2 node and the **Viewer** in order to view it. The **Position** tab includes Constrain and Xform options, and the **Display** tab allows you to change how the lights look in the Viewer. You can then use the transformation handles to transform the lights interactively in the Viewer. See more about how you can use the Viewer at Lights in the Viewer.



Adjusting Light Properties

All lights have **Basic**, **Refine**, **ScanlineRender Options**, **Position**, **Import/Export**, and **Display** options listed here. Further light-specific knobs are outlined in the sections at the bottom of this page.

Basic

- You can set the **Intensity** of the light, which scales the power of the light linearly.
- You can also set an **Exposure** value which scales the power of the light exponentially as a power of 2, similar to an FStop control.
- You can enable the **Normalize** checkbox to keep the power of the light constant when the size of the light is altered. This makes it easier to independently adjust the intensity/exposure and size of the light because the total illumination provided will not vary with the area or angular size of the light. May not be respected by all light shader implementations.
- Use the **Color** wheel to set the color of the light.
- Use the **Color Temperature** checkbox to enable color temperature. In degrees Kelvin, this represents the white point of the light. When enabled the rgb converted result multiplies against the 'Color' control. The default is a common white point 6500k, or D65. Lower values are warmer and higher values are cooler with a valid range from 1000k to 10000k.

Refine

- The **Diffuse Amount** knob is a multiplier for the effect this light has on diffuse response of materials, so changing this essentially makes a diffuse shader respond differently to the light.
- The **Specular Amount** knob does the same for the specular response of materials.

ScanlineRender Options

- You can set an **Illumination Mask** and **Receive Shadow Mask** if you want the light to cast shadows on other objects. See Lights and Shadows to learn more about the workflow.

Position

- Each light has the usual transformation options, plus Constrain and Xform options. Learn more about these at Using Transformation Tools, Constraining Prims, and Xform Syncing Options.

Import/Export

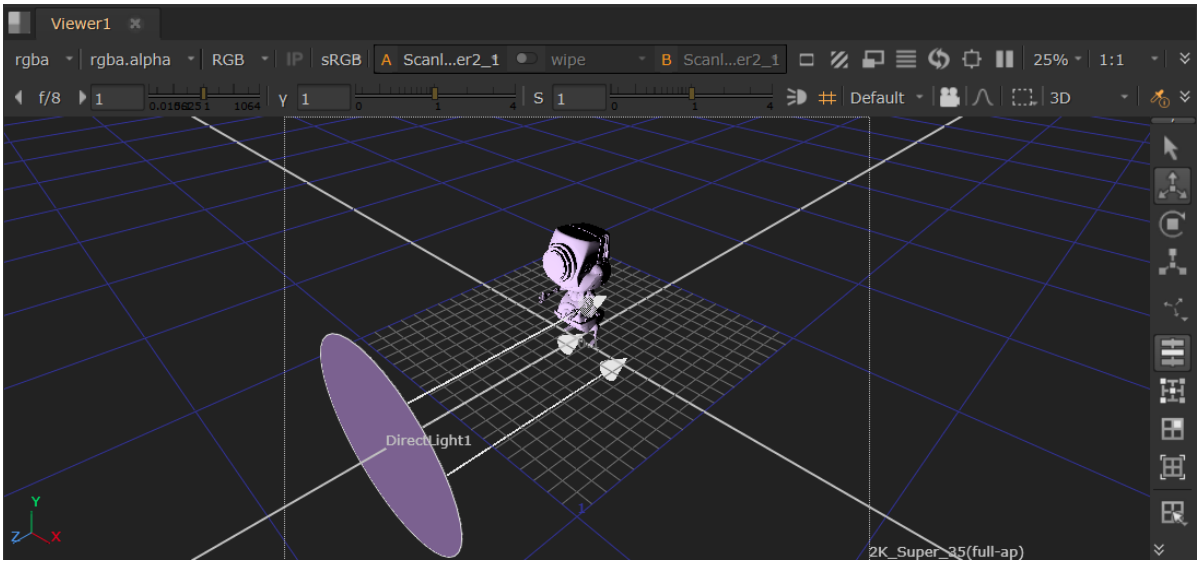
- If you wish to import an existing light, use the **Source Prim** (to pick from your scene) or **Import From File** (to use an external file) options.
- **Live Read** allows your changes to update live. Turning **Live Read** off resets the knobs to their local state. If you are using an imported or referenced Axis, and have enabled **Live Read**, you'll notice that the knobs have a **green** outline. This helps you to see which data is being read from elsewhere in their script and which data has been changed directly on the node itself.
    - The **green** outline means the data has not changed from the source.
    - The **yellow** outline means there are local changes on this node which are different from the source.
- If the **Output Prim Path** is left blank , the lights will then be added to the Scene Graph automatically under **{nodename}_Inject**. Or if you wish to adjust the prim path, edit the **Output Prim Path**.

Display

- The **Display** tab allows you to change how the lights look in the Viewer, including locator size and color.

DirectLight

This is a directional light which can be transformed to shine in the direction of your choice in your scene, equivalent to the USD DistantLight.



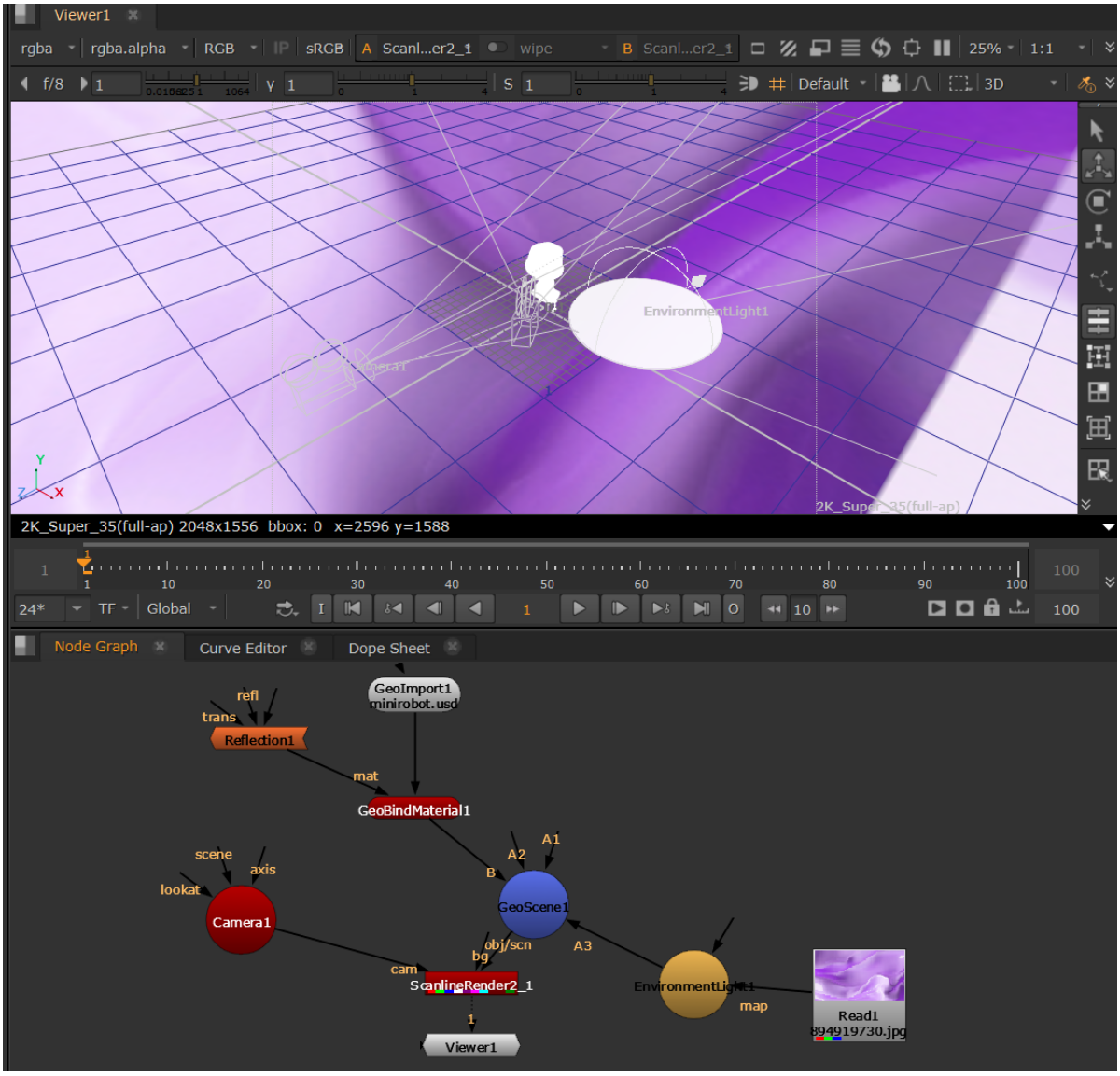
• **Spread Angle** lets you edit the angular diameter of the light in degrees.

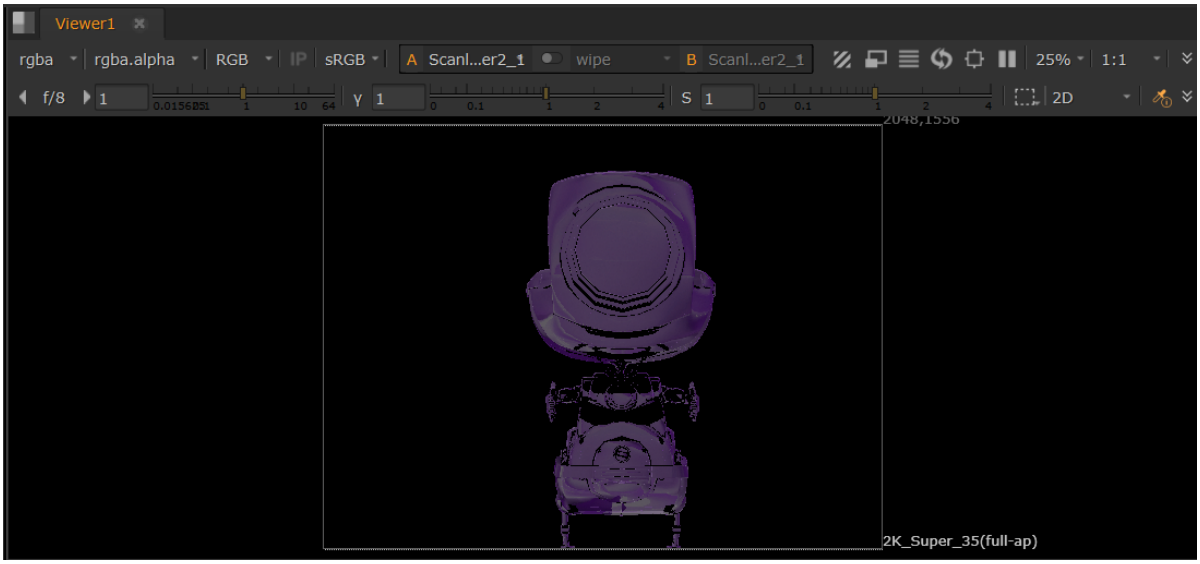
Note: All controls are listed in the Node Reference Guide - DirectLight.

EnvironmentLight

The EnvironmentLight is there to produce a mesh's specular color, so essentially anything plugged into the **map** input of the environment light will show up in the reflections of a mesh. This is equivalent to the USD DomeLight.

The easiest way to set this up is to connect a Read node with your image into the EnvironmentLight's **map** input.


Viewer1
rgba
rgba.alpha
RGB
sRGB
wipe
25%
1:1
f/8
Default
3D
EnvironmentLight1
2K_Super_35(full-ap) 2048x1556 bbox: 0 x=2596 y=1588
24*
TF
Global
Node Graph
Curve Editor
Dope Sheet
GeoImport1
minirobot.usd
refl
trans
Reflection1
mat
GeoBindMaterial1
A1
A2
B
scene
axis
lookat
Camera1
GeoScene1
obj/scn
A3
bg
cam
ScanlineRender2_1
EnvironmentLight1
map
Read1
894919730.jpg
Viewer1

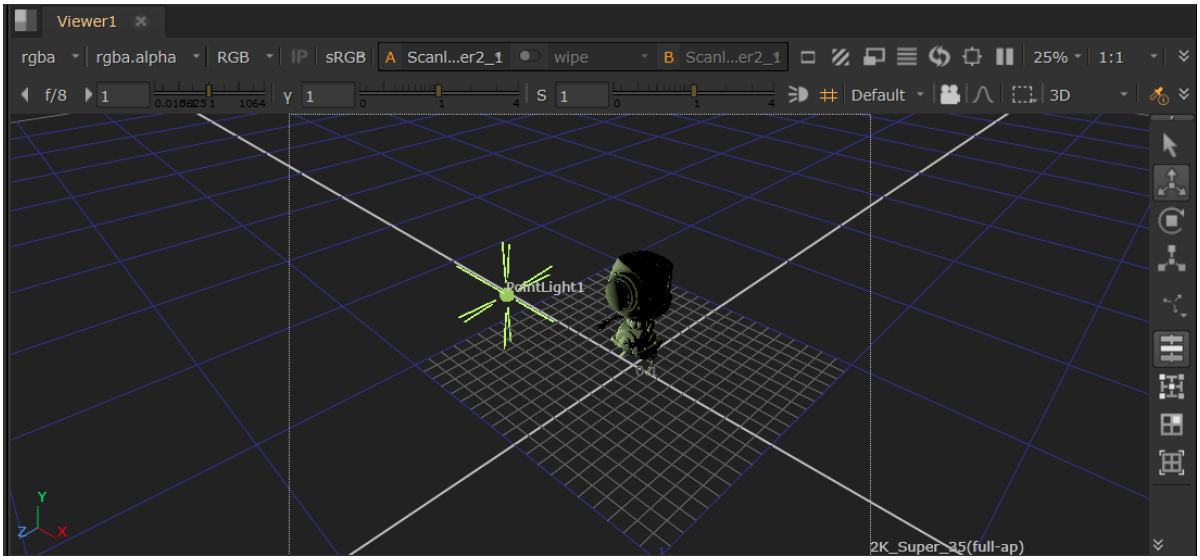

Viewer1
rgba
rgba.alpha
RGB
sRGB
wipe
25%
1:1
f/8
2D
2048,1556
2K_Super_35(full-ap)

- **Guide Radius** is intended to scale the locator in the Viewer. See Lights in the Viewer.
- **ScanlineRender Options**: These changes will not be visible in the 3D viewport via HDStorm as HDStorm doesn't provide controls for these yet, but will be visible in your ScanlineRender output. **Blur Size** allows you to set how much you want to blur the image you are mapping to the light, and **Mirror X** and **Mirror Y** will let you flip and flop the image being mapped. You can always opt to create a blur or mirror nodes to do this in the node graph also.

Note: All controls are listed in the Node Reference Guide - EnvironmentLight.

PointLight

This creates a point in 3D space that emits light in every direction, equivalent of the USD SphereLight but with a small radius.



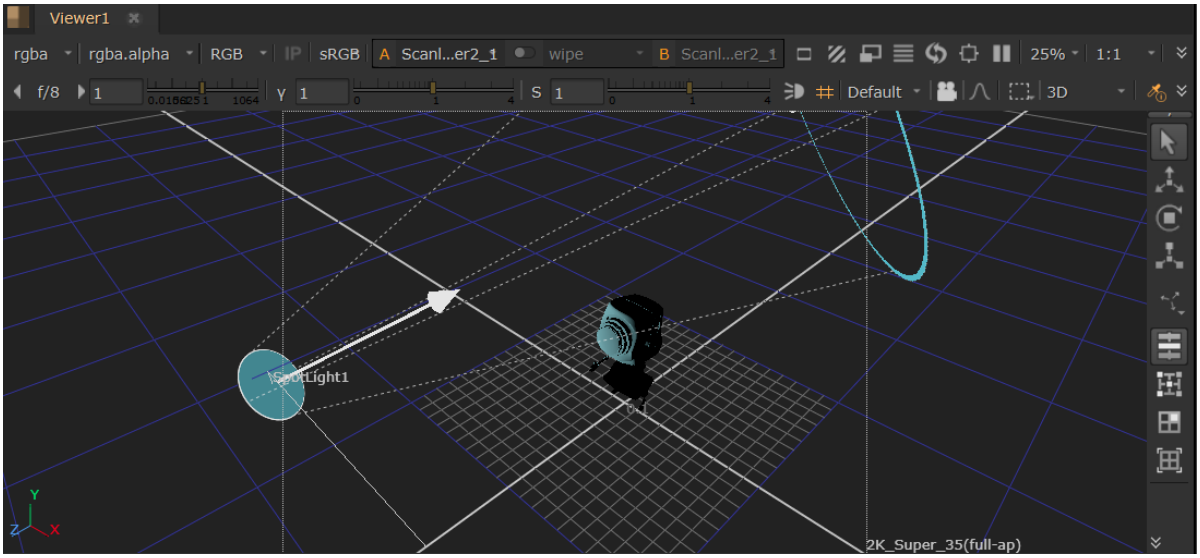
- The **Radius** knob controls the radius of the light, where a larger radius increases the overall reach of the light.
- The **Treat As Point** checkbox essentially reduces the radius to the equivalent size of a point. The Radius knob is greyed out when this Treat As Point is active as you have no need to change the radius in that case.
- **ScanlineRender Options**: These changes will not be visible in the 3D viewport via HDStorm as HDStorm doesn't provide controls for these yet, but will be visible in your ScanlineRender output. **Falloff Type** controls how much light the object gets from the light source. It is based on the distance between the light source and the object. A **Linear** type will diminish the light at a fixed rate as it travels from the object. **Quadratic** and **Cubic** types will diminish the light at an exponential rate. **FallOff Amount** also affects the scale of this.

Note: The **Output Prim Path** will export a Sphere Light to the USD stage. So this would convert the light from a Nuke shader light to a USD light prim for use in other DCC's, which values matching as close as possible, but note these won't necessarily be 1:1. For the PointLight for instance there is no equivalent USD light, so the best approximation is a SphereLight with a radius set to an extremely small amount.

Note: All controls are listed in the Node Reference Guide - PointLight.

SpotLight

The SpotLight node creates a point in 3D space that emits a cone-shaped light, equivalent to the USD DiskLight.



- The **Radius** knob controls the radius of the light. A larger radius increases the overall reach of the light.
- The **Cone Angle** controls the angular diameter of the light in degrees. The light will be cone shaped at smaller angles and behave more like a spotlight.
- The **Cone Softness** is the softness of the cone angle; typically a value of 1 is used for a soft gradation, while 0 would give the cone angle a hard cut-off. Values greater than 1 can be used to further soften the gradation.
- **Focus** affects the spread of the light. The greater the value the more focused the light is. Lower values result in a diffused effect.
- **Focus Tint** changes the color tint of the light in the falloff region determined by the focus property.

- **ScanlineRender Options**: These changes will not be visible in the 3D viewport via HDStorm as HDStorm doesn't provide controls for these yet, but will be visible in your ScanlineRender output. Falloff Type controls how much light the object gets from the light source. It is based on the distance between the light source and the object. A **Linear** type will diminish the light at a fixed rate as it travels from the object. **Quadratic** and **Cubic** types will diminish the light at an exponential rate. **FallOff Amount** also affects the scale of this.

Note: The **Output Prim Path** will export a DiskLight to the USD stage. So this would convert the light from a Nuke shader light to a USD light prim for use in other DCC's, which values matching as close as possible, but note these won't necessarily be 1:1.

Note: All controls are listed in the Node Reference Guide - SpotLight.

Using USD Lights

There are several USD-native light nodes to choose from in Nuke's New 3D System. Their node parameters map directly to USD values without any need for conversion. Each light node has a **Create** and an **Edit** mode so you can create new lights or edit imported light prims.

To edit imported light prims, you can also use the GeoEditLight node, which combines all four light types specially for editing purposes. You can select the light type under **Schema** and its options will dynamically populate.

Light Node	USD Equivalent	Use for:
GeoDistantLight	DistantLight	Creating or editing a DistantLight
GeoDiskLight	DiskLight	Creating or editing a DiskLight
GeoDomeLight	DomeLight	Creating or editing a DomeLight
GeoSphereLight	SphereLight	Creating or editing a SphereLight
GeoEditLight	DistantLight, DomeLight, SphereLight, DiskLight	Editing an imported DistantLight, DiskLight, DomeLight, or SphereLight

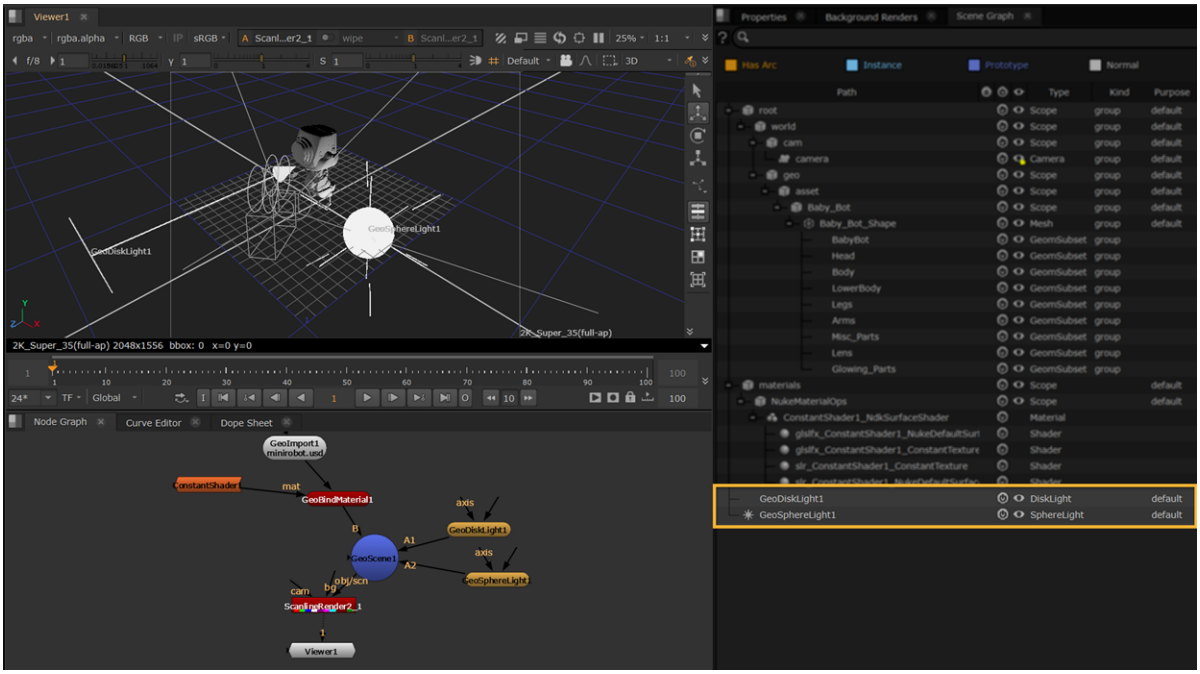
Note: See an overview of the different types of lights in Nuke at Light Your Scene.

Adding Lights

To add a light, simply add a light node to your node graph and connect it to the GeoScene node that is downstream of your geometry. In the node properties, under **Mode**, select either **Create** or **Edit** depending on whether you wish to create a new light or edit an existing one.

Creating a Light

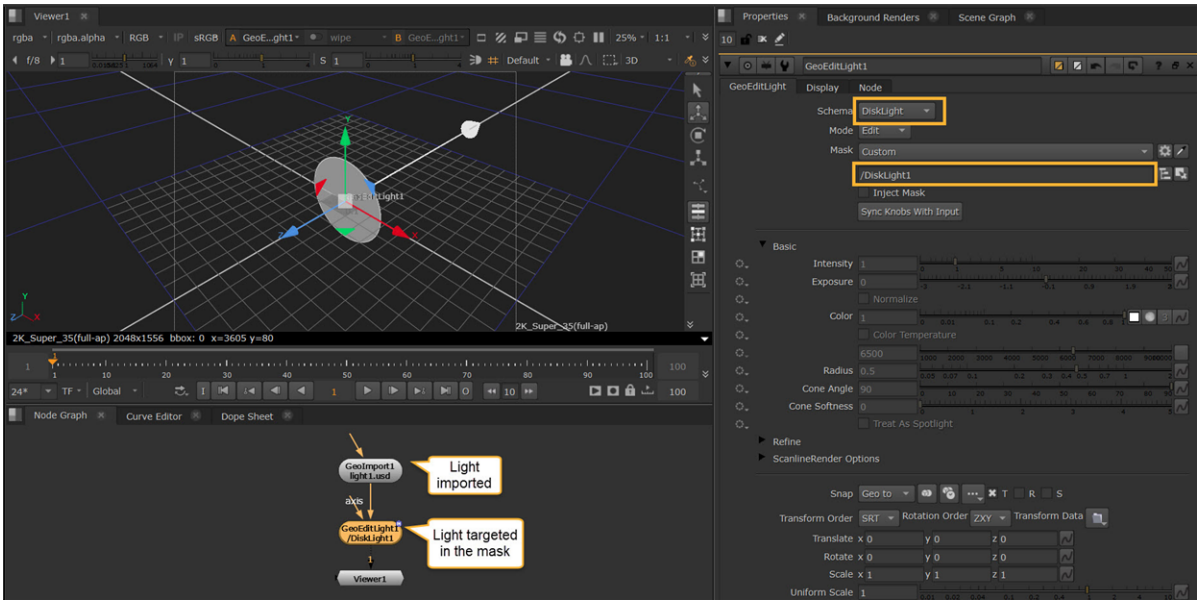
In this example, we are using **Create** mode, and we have added two of the USD-native light nodes (GeoDiskLight and GeoSphereLight). These appear directly in the Scene Graph as **{nodename}** - you can change the **Path** in the light node Properties to adjust this if needed.



Editing a Light

If you were using **Edit** mode, you would use the **Mask** knob to target your existing lights in your scene.

In this example we are using GeoEditLight to target a light from an imported stage by setting the **Schema** and the **Mask**.



Note: See Using Paths and Masks to learn more about masking.

Adjusting Light Properties

All lights have Basic, Refine, ScanlineRender Options, Transform, and Display options listed here. Light-specific knobs are outlined in the sections below.

Basic

- You can set the **Intensity** of the light, which scales the power of the light linearly.
- You can also set an **Exposure** value which scales the power of the light exponentially as a power of 2, similar to an FStop control.
- You can enable the **Normalize** checkbox to keep the power of the light constant when the size of the light is altered. This makes it easier to independently adjust the intensity/exposure and size of the light because the total illumination provided will not vary with the area or angular size of the light. May not be respected by all light shader implementations.
- Use the **Color** wheel to set the color of the light.
- Use the **Color Temperature** checkbox to enable color temperature. In degrees Kelvin, this represents the white point of the light. When enabled the rgb converted result multiplies against the 'Color' control. The default is a common white point 6500k, or D65. Lower values are warmer and higher values are cooler with a valid range from 1000k to 10000k.

Refine

- The **Diffuse Amount** knob is a multiplier for the effect this light has on diffuse response of materials, so changing this essentially makes a diffuse shader respond differently to the light.
- The **Specular Amount** knob does the same for the specular response of materials.

ScanlineRender Options

- You can set an **Illumination Mask** and **Receive Shadow Mask** if you want the light to cast shadows on other objects. See Lights and Shadows to learn more about the workflow.

Transform

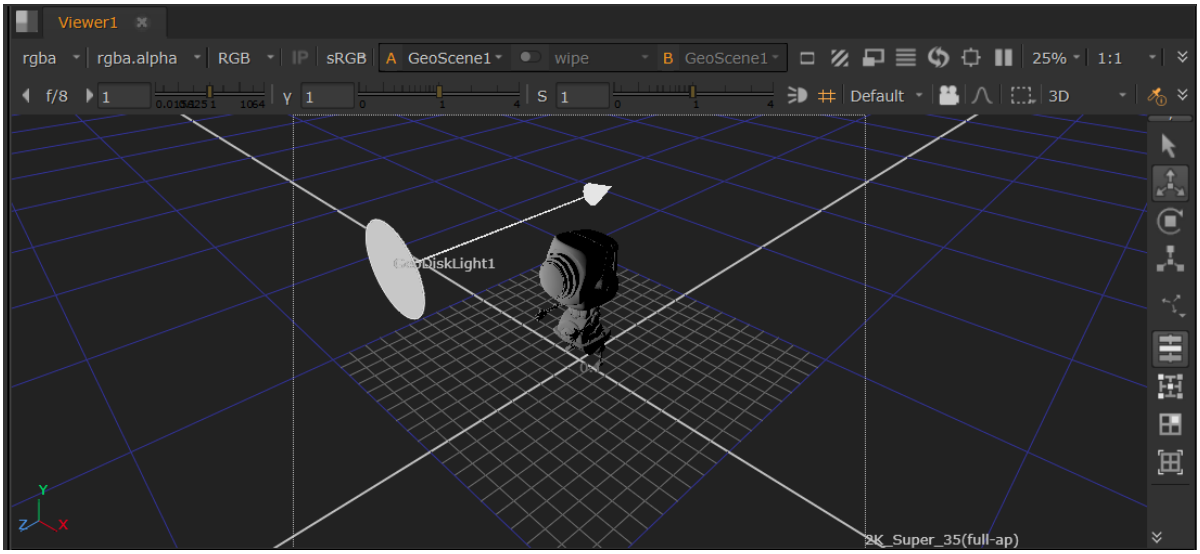
- Each light has the usual transformation options. You can use the transformation handles to transform the lights interactively in the Viewer. See more about how you can use the Viewer at Lights in the Viewer.

Display

- The **Display** tab allows you to change how the lights look in the Viewer, including locator size and color.

GeoDiskLight

This light node creates a point in 3D space that emits a cone-shaped light.



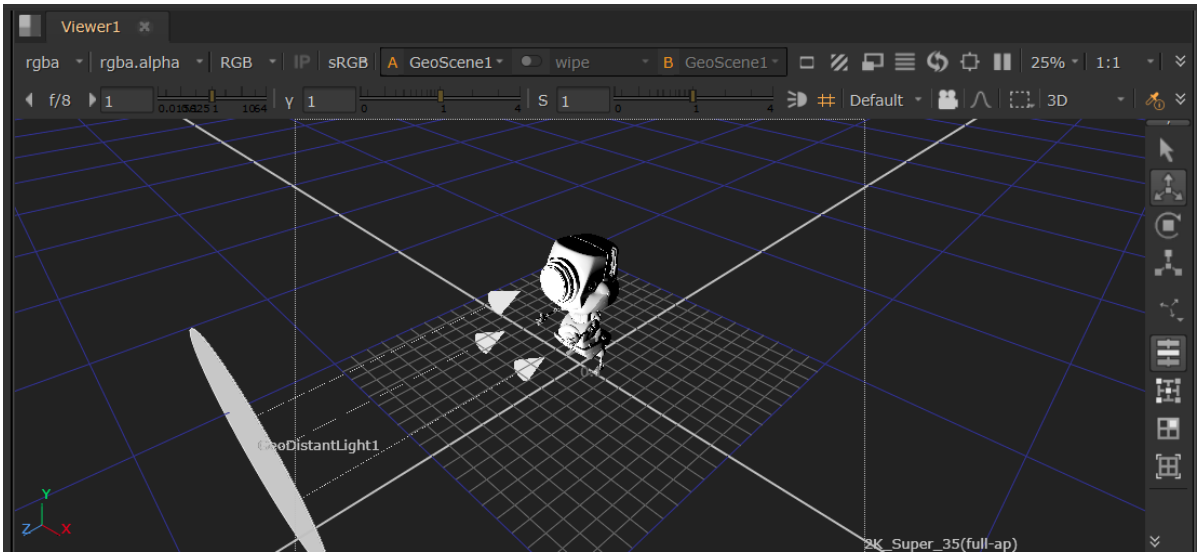
- The **Radius** knob controls the radius of the light. A larger radius increases the overall reach of the light.
- The **Cone Angle** controls the angular diameter of the light in degrees. The light will be cone shaped at smaller angles and behave more like a spotlight.
- The **Cone Softness** is the softness of the cone angle; typically a value of 1 is used for a soft gradation, while 0 would give the cone angle a hard cut-off. Values greater than 1 can be used to further soften the gradation.
- **Focus** affects the spread of the light. The greater the value the more focused the light is. Lower values result in a diffused effect.

- **Focus Tint** changes the color tint of the light in the falloff region determined by the focus property.
- **ScanlineRender Options**: These changes will not be visible in the 3D viewport via HDStorm as HDStorm doesn't provide controls for these yet, but will be visible in your ScanlineRender output. Falloff Type controls how much light the object gets from the light source. It is based on the distance between the light source and the object. A **Linear** type will diminish the light at a fixed rate as it travels from the object. **Quadratic** and **Cubic** types will diminish the light at an exponential rate.

Note: All controls are listed in the Node Reference Guide - GeoDiskLight.

GeoDistantLight

This is a directional light which can be transformed to shine in the direction of your choice in your scene.

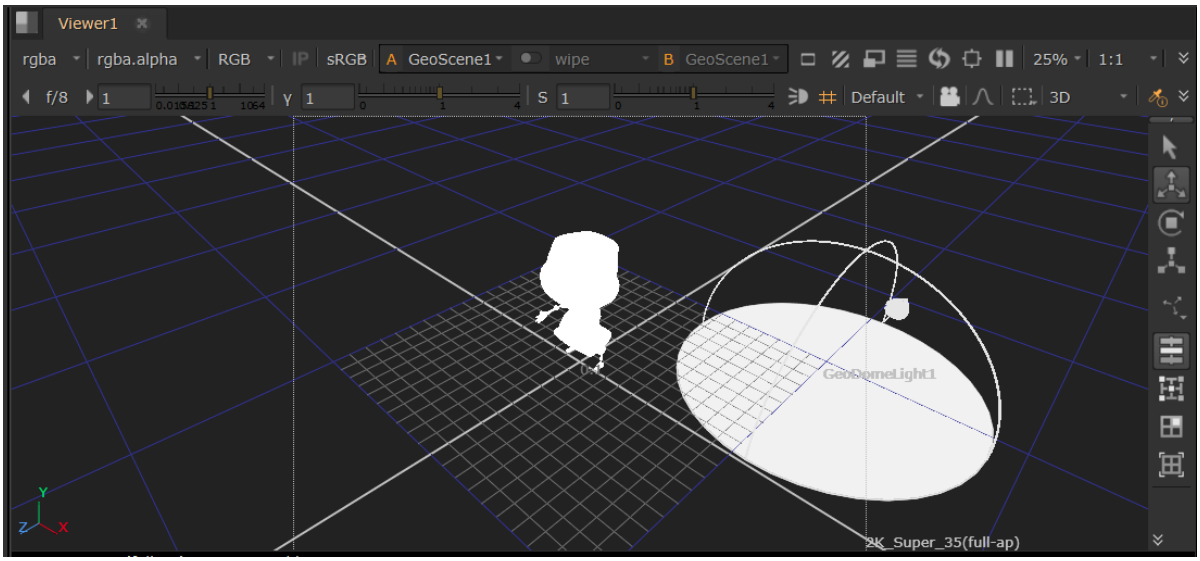


- **Spread Angle** lets you edit the angular diameter of the light in degrees.
- **ScanlineRender Options**: These changes will not be visible in the 3D viewport via HDStorm as HDStorm doesn't provide controls for these yet, but will be visible in your ScanlineRender output. **Falloff Type** controls how much light the object gets from the light source. It is based on the distance between the light source and the object. A **Linear** type will diminish the light at a fixed rate as it travels from the object. **Quadratic** and **Cubic** types will diminish the light at an exponential rate.

Note: All controls are listed in the Node Reference Guide - GeoDistantLight.

GeoDomeLight

This light node can produce a mesh's specular color, so essentially anything plugged into the **map** input of the light will show up in the reflections of a mesh.

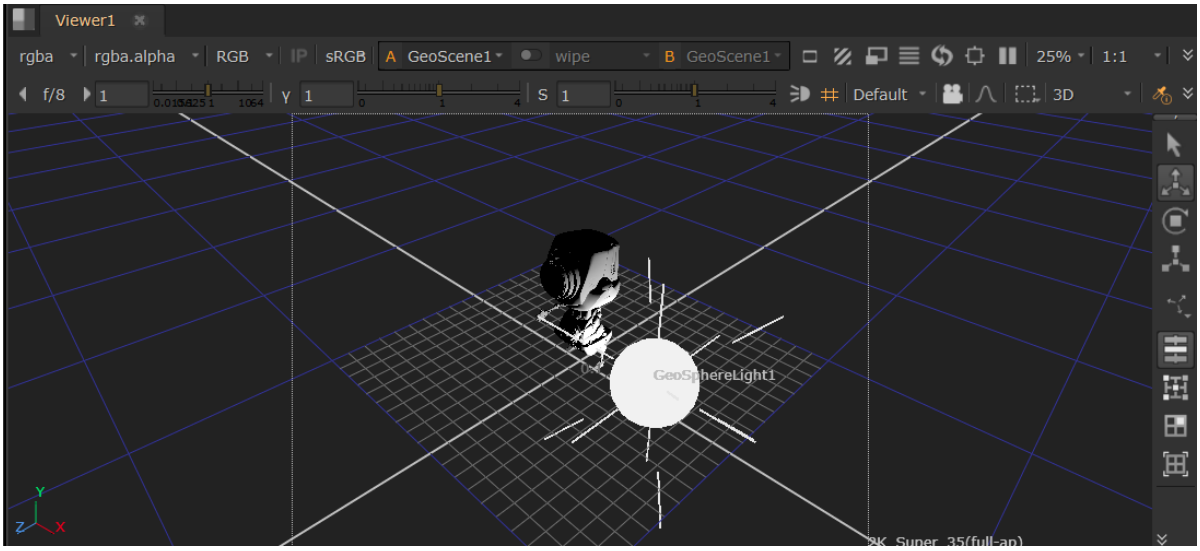


- **Radius** is intended to scale the locator in the Viewer. See Lights in the Viewer.
- **ScanlineRender Options**: These changes will not be visible in the 3D viewport via HDStorm as HDStorm doesn't provide controls for these yet, but will be visible in your ScanlineRender output. **Blur Size** allows you to set how much you want to blur the image you are mapping to the light, and **Mirror X** and **Mirror Y** will let you flip and flop the image being mapped. You can always opt to create a blur or mirror nodes to do this in the node graph also.

Note: All controls are listed in the Node Reference Guide - GeoDomeLight.

GeoSphereLight

This creates a sphere in 3D space that emits light in every direction.



- The **Radius** knob controls the radius of the light, where a larger radius increases the overall reach of the light.
- The **Treat As Point** checkbox essentially reduces the radius to the equivalent size of a point. The Radius knob is greyed out when this Treat As Point is active as you have no need to change the radius in that case.
- **ScanlineRender Options**: These changes will not be visible in the 3D viewport via HDStorm as HDStorm doesn't provide controls for these yet, but will be visible in your ScanlineRender output. **Falloff Type** controls how much light the object gets from the light source. It is based on the distance between the light source and the object. A **Linear** type will diminish the light at a fixed rate as it travels from the object. **Quadratic** and **Cubic** types will diminish the light at an exponential rate.

Note: All controls are listed in the Node Reference Guide - GeoSphereLight.

Lights and Shadows

All light nodes in Nuke contain shadowing controls under **ScanlineRender Options**:

- **Illumination Mask** sets which objects will be illuminated by the light.
- Use **Cast Shadow** checkbox to turn shadows on or off.
- **Receive Shadow Mask** sets which objects will receive shadowing from this light.

• **Shadow Channels** specifies the channel to output the shadowing.

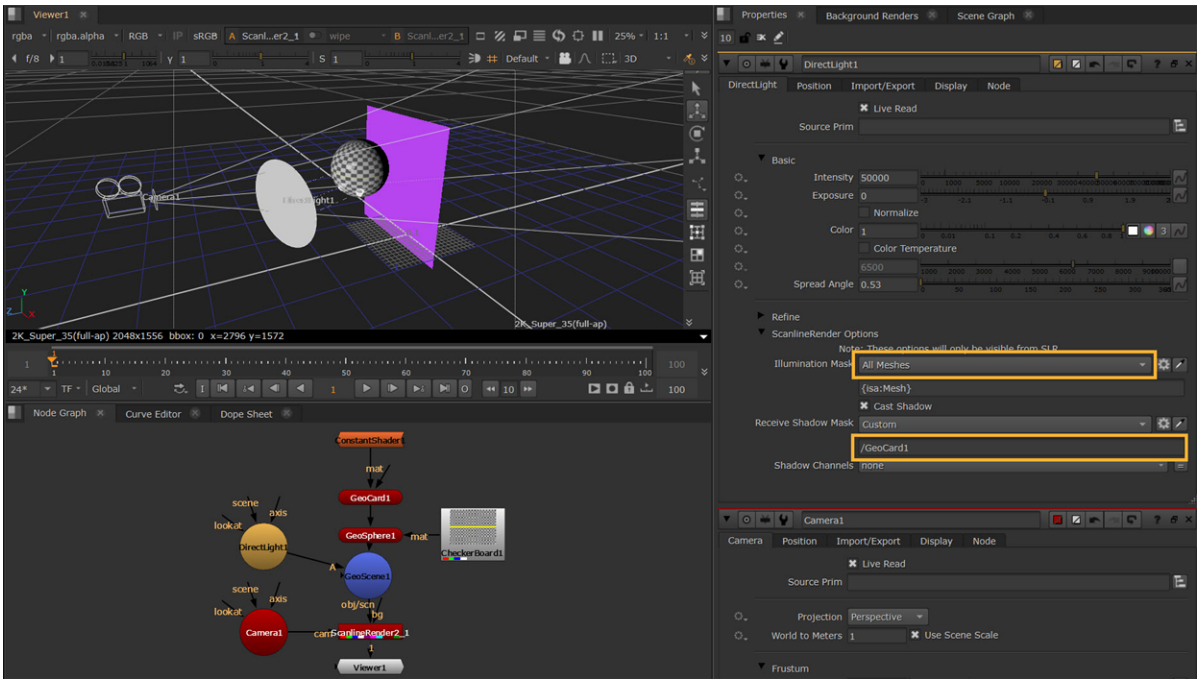
Shadow Example

In this example, we have set up a DirectLight which is shining onto a sphere (GeoSphere1) and a card (GeoCard1).

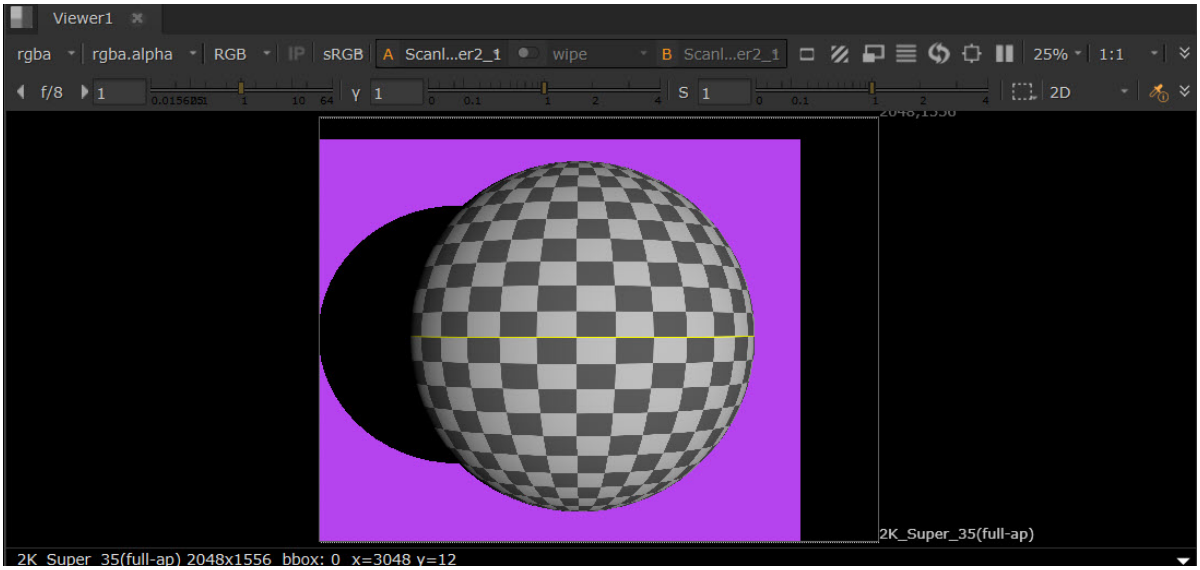
To affect the shadows created by the DirectLight, this can be controlled within the DirectLight's Properties.

By targeting **All Meshes** in the DirectLight's **Illumination Mask**, the sphere and the card are both illuminated by the DirectLight.

Then, by targeting the card's prim path (GeoCard1) in the **Receive Shadow Mask**, the sphere's shadow will show on the card behind it. Alternatively, using the **All roots** or **All Meshes** options would allow you to target all the objects and avoid the need to set this manually.



Viewing the ScanlineRender representation in 2D Viewer:



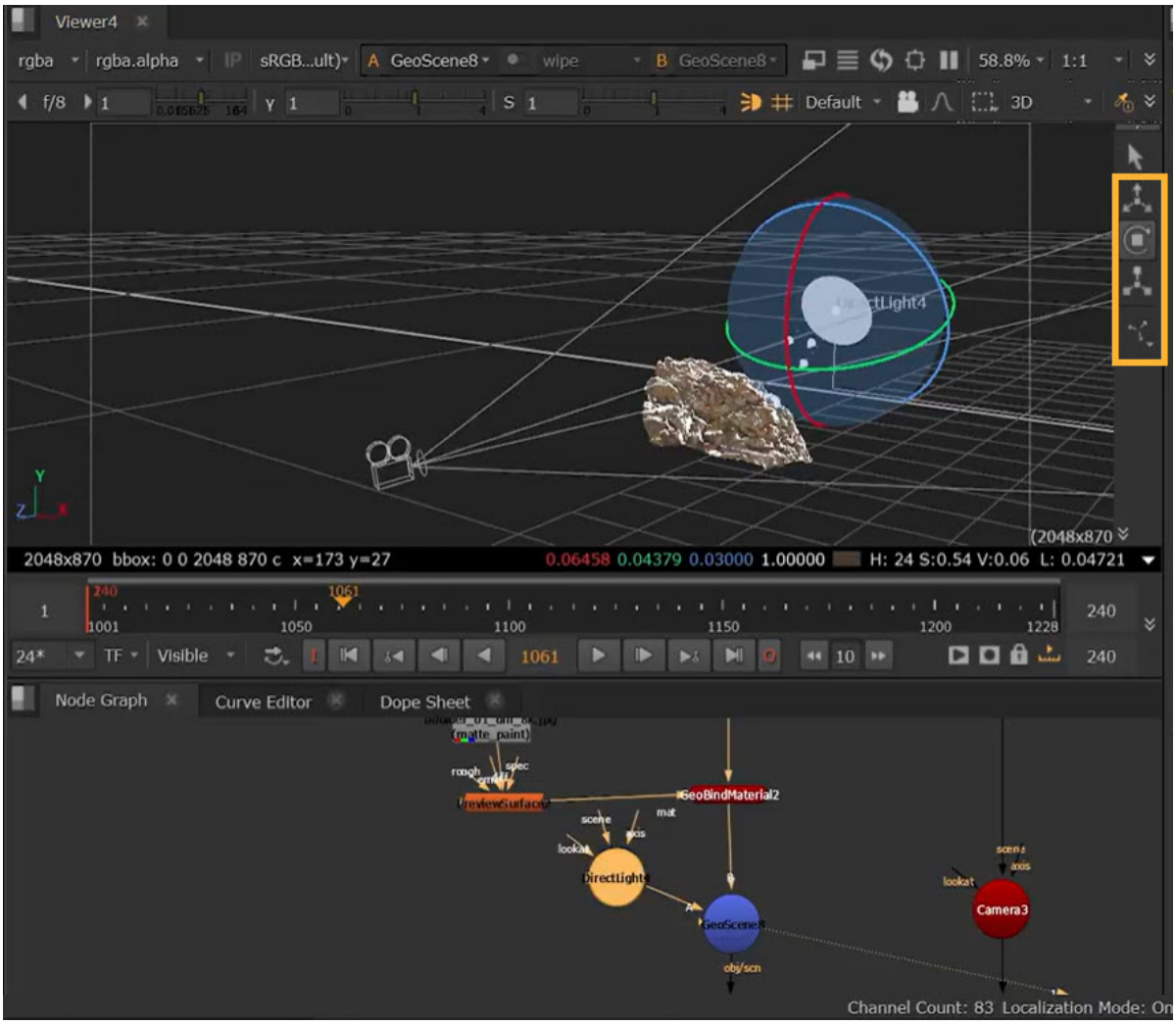
Lights in the Viewer

Light locators have been updated to give a better visualization and control in the Viewer.

You can now adjust various parameters using the locators, and transform the lights interactively in the Viewer.

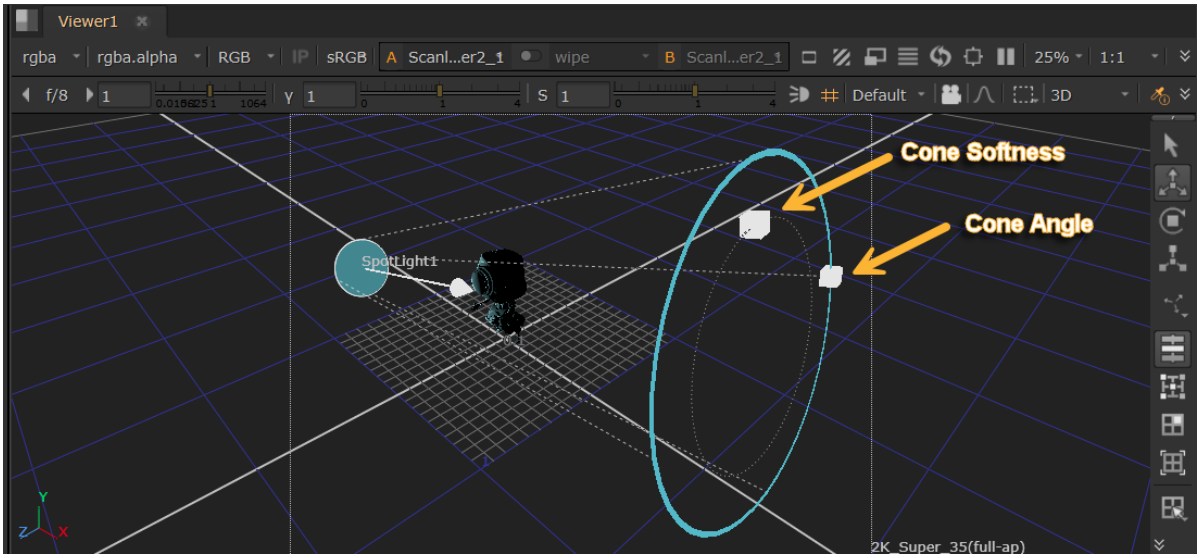
Transformations

All lights can use the Transformation handles in the 3D Viewer: Translate, Rotate, Scale. See Using Transformation Tools.



Light-specific Adjustments

When using the SpotLight or GeoDiskLight, you can drag the white square of the outer circle of the locator's cone to change **Cone Angle** value and the white square on the inner circle of the locator's cone to change **Cone Softness**.



Note: A few options are not supported by HDStorm, eg. Cone Softness, so changes will not show in the Viewer but they will show in your render. See Rendering Lights.

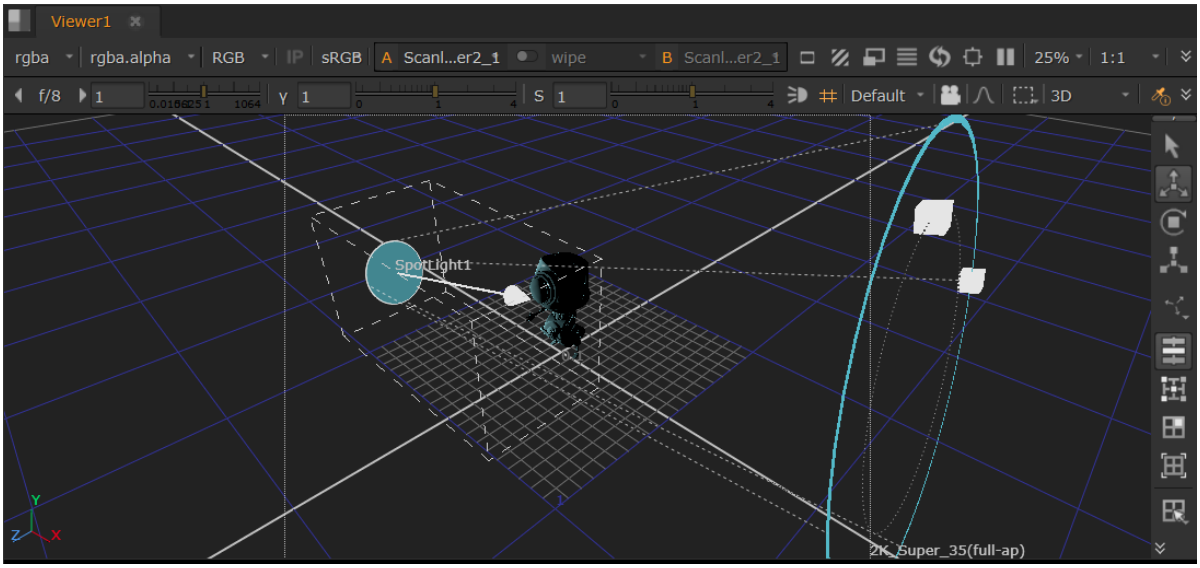
Dynamic Locator Updates

Some changes that you make in the node parameters will also be shown on the locator's display.

- **Intensity** increases will thicken the spokes on PointLight and GeoSphereLight, and thicken the arrow on the SpotLight, GeoDiskLight, DirectLight and GeoDistantLight.
- **Color** changes will be shown on the light locator - it will change color to match the selected light color, unless you choose to set this manually in the **Display** tab.
- **Radius** of the PointLight and GeoSphereLight, and **GuideRadius** of the EnvironmentLight and GeoDomeLight will change the locator's scale.

Bounding Boxes

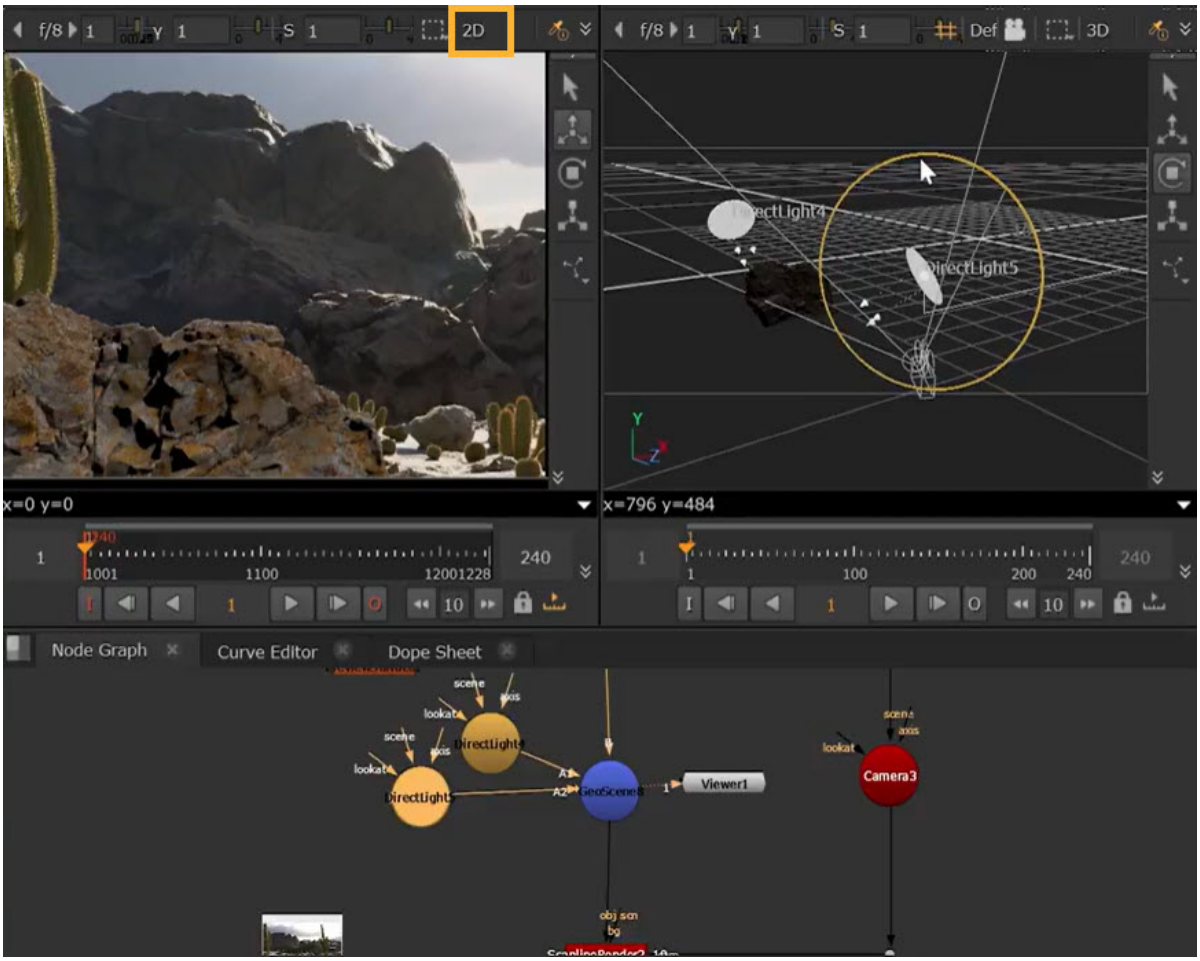
Lights display bounding boxes in the Viewer when enabled. To enable these, go to your Viewer node: **Properties** > **3D** > **show** >



Rendering Lights

Lights can be rendered with ScanlineRender2.

- First, you should connect the lights to a GeoScene node.
- Then connect the GeoScene into the **obj/scn** input of the ScanlineRender2 node.
- To view your render, simply toggle the **Viewer** back to **2D**. This shows you a representation of the render in the Viewer.



Some lights have specific **ScanlineRender Options** such as **Falloff Type**, **Blur Size**, **Mirror X** and **Mirror Y** which are not visible in the Viewer and will only be applied in the render.

Node Reference

DirectLight

Creates an adjustable, directional light in a 3D scene.

EnvironmentLight

Creates an adjustable, environment light in a 3D scene.

PointLight

Creates a point in 3D space that emits light in every direction.

SpotLight

Creates a point in 3D space that emits a cone-shaped light.

ScanlineRender

Renders all the objects and lights in a scene.

Rendering in the New 3D System

The ScanlineRender2 node operates in the same manner as the ScanlineRender you've always known and used, but can be used in conjunction with the New 3D System Nodes.

ScanlineRender2 has had updates to sampling and projection options, motion blur, AOV controls, and offers a new ray-tracing architecture under the hood.

ScanlineRender2 remains a utility renderer at heart, so a lot of focus has gone into providing and expanding on the workflows and controls artists expect, while providing new functionality.



Note: ScanlineRender's classic 3D system equivalent is also named ScanlineRender. For New 3D System scenes, make sure you are adding the New 3D System node (found in right click node menu under **3D** > **3D**), not the Classic one (**3D** > **3D Classic**).

Workflows and Usage

The workflow problems we want to solve are when you've comped in new elements into your shot, like explosions, fire, particle sims and need a way to better ground these comped elements by generating new render passes. The goal is that you can bring in as much of the scene geo as you need, set up your comp elements where you need them, add any additional lighting required for those new elements to interact and then render with reflection and refraction from ScanlineRender2.

As a utility renderer ScanlineRender2 is not intended to render entire production shots and instead focuses on what compositing artists need most to give them the ability to generate new render passes at the late stages of production, without having to go back to other departments.

Rendering with ScanlineRender

To render a New 3D System scene, you need three things: your scene, a camera, and a ScanlineRender2 node.

To render out a scene:

1. Set up your 3D scene as normal, including geometry, materials, lights, etc. Make sure you have a Viewer node.
2. Add a ScanlineRender2 node and connect the scene to the **obj/scn** input, usually via a GeoScene node.
3. Make sure the camera is connected to the render node's **cam** input.
4. Add a background to the **bg** input, if required. You can use the **bg** input to composite a background image into the scene and to determine the output resolution.

> **Note:** If the bg input is not used, the render node output defaults to the **root.format** or **root.proxy_format** defined in the **Project Settings**.

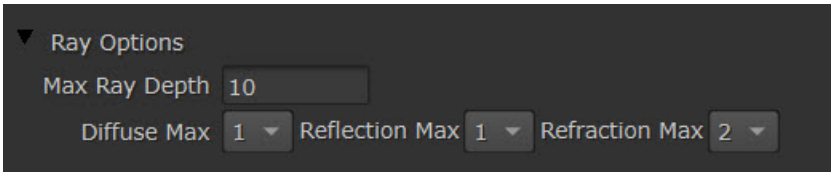
5. Toggle the Viewer back to **2D**. This shows you a representation of the render in the Viewer.
6. Connect the output of the ScanlineRender2 node to the Write node in order to render to disk.

> **Note:** Take a look at the ScanlineRender2 reference guide page to learn more about its controls.

Ray-tracing

You can now render reflections, refractions and more via ray-tracing. In the Properties panel, you can adjust **Ray Options**, including the ability to control your **Ray Depth** and adjust the options to the complexity you need.

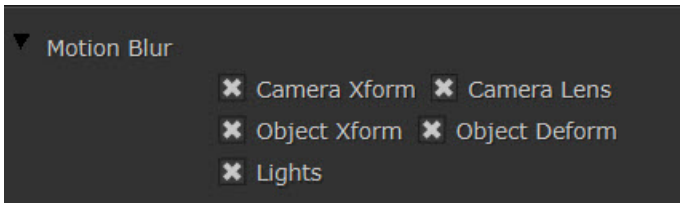
See more at Ray-tracing with ScanlineRender.



Motion Blur

In the **Motion Blur** section of the **Properties** panel, there are options to adjust different types of camera, object and light motion blurs.

See more at Motion Blur with ScanlineRender.



Projection Modes

In the **Camera/ Projection** Properties, the **Projection** knob lets you set the lens projector mode you want to use to output your render data.

- **Render Camera**: use the setting from the Camera node Projection knob
- **Perspective**: renders with objects in front of the camera having the illusion of depth defined by the camera's focal-length and aperture
- **Orthographic**: renders objects as if viewed using a parallel projection
- **Spherical**: renders the entire 360 degree world as a spherical map
- **Cylindrical**: renders the entire 360 degree world as a cylindrical map
- **UV Unwrap**: Every object renders its UV space into the output format. Used for creating texture maps

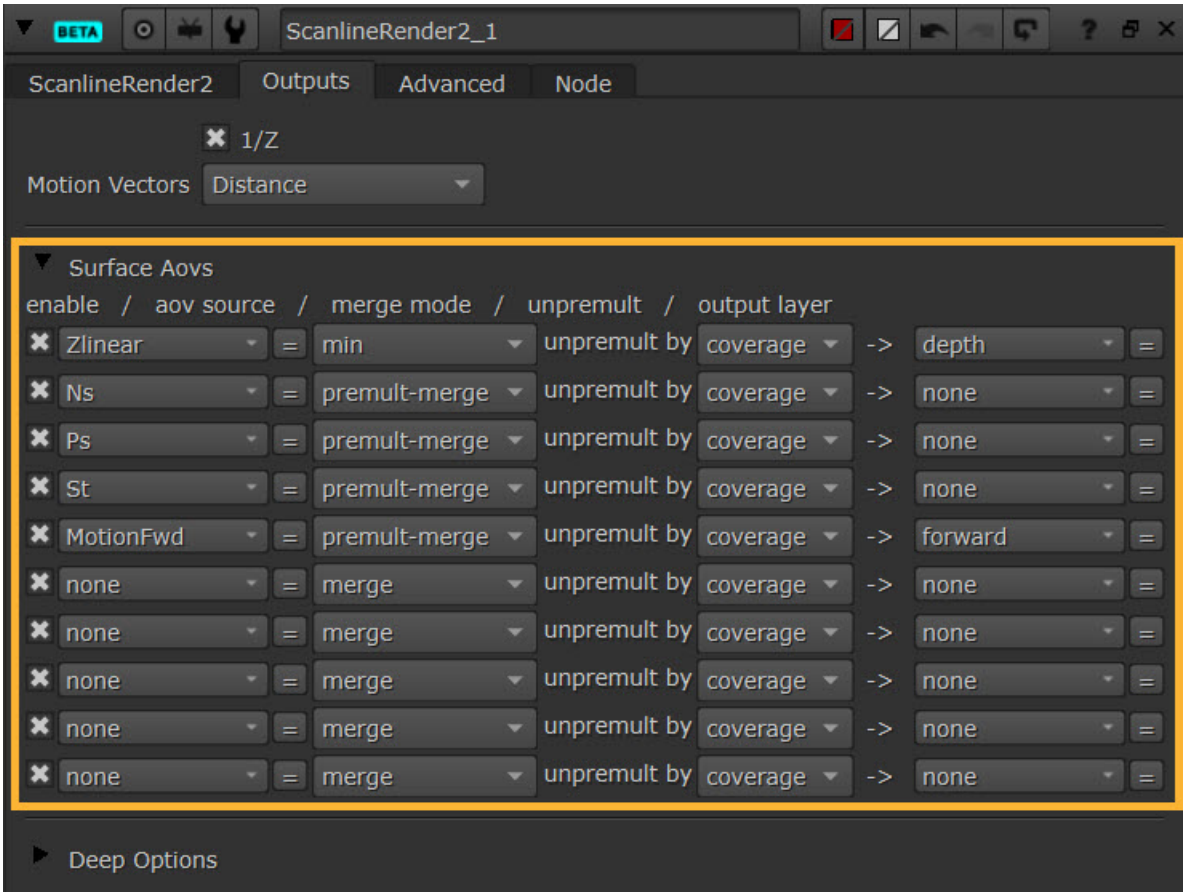
Note: The Camera node now no longer has UV Unwrap mode, because this is actually a special render mode, not a projection.

All the above options are now lens projector plugins, which means you can create as many as you want and add to this list. This means if you have a specific lens distortion map, you can add that as a plugin to undistort footage from a particular lens.

AOV controls

In the **Properties** > **Outputs** tab, you can edit the AOV outputs.





Render Nodes

Write

Renders the result of all upstream nodes and saves the result to disk.

Camera

Emulates real world cameras and can also project 2D textures.

GeoScene

Combines objects into a single scene for rendering.

Project3DShader

Projects an input image through a camera on to a prim.

ScanlineRender

Renders all the objects and lights in a scene.

Ray-tracing with ScanlineRender

Combining ScanlineRender2's controls with the ReflectiveSurface shader node can be an effective way to generate ray-tracing in your render passes.

For example, inside the ReflectiveSurface shader, you can control various key elements such as the amount of reflection and its tint.



Note: See ReflectiveSurface: Generating Reflections to learn how to set up this shader in your scene.

Then, inside the ScanlineRender2 node, you can control the ray depth across your scene at render time:

Ray Options

May Ray Depth

The 'Max Ray Depth' knob sets the number of times a ray can bounce around the scene.

The higher the number the more 'accurate' the render is, but at the cost of render time.

Diffuse Max

The 'Diffuse Max' knob sets at what depth count diffuse rays will stop.

Reflection Max

The 'Reflection Max' knob sets at what depth count glossy rays will stop.

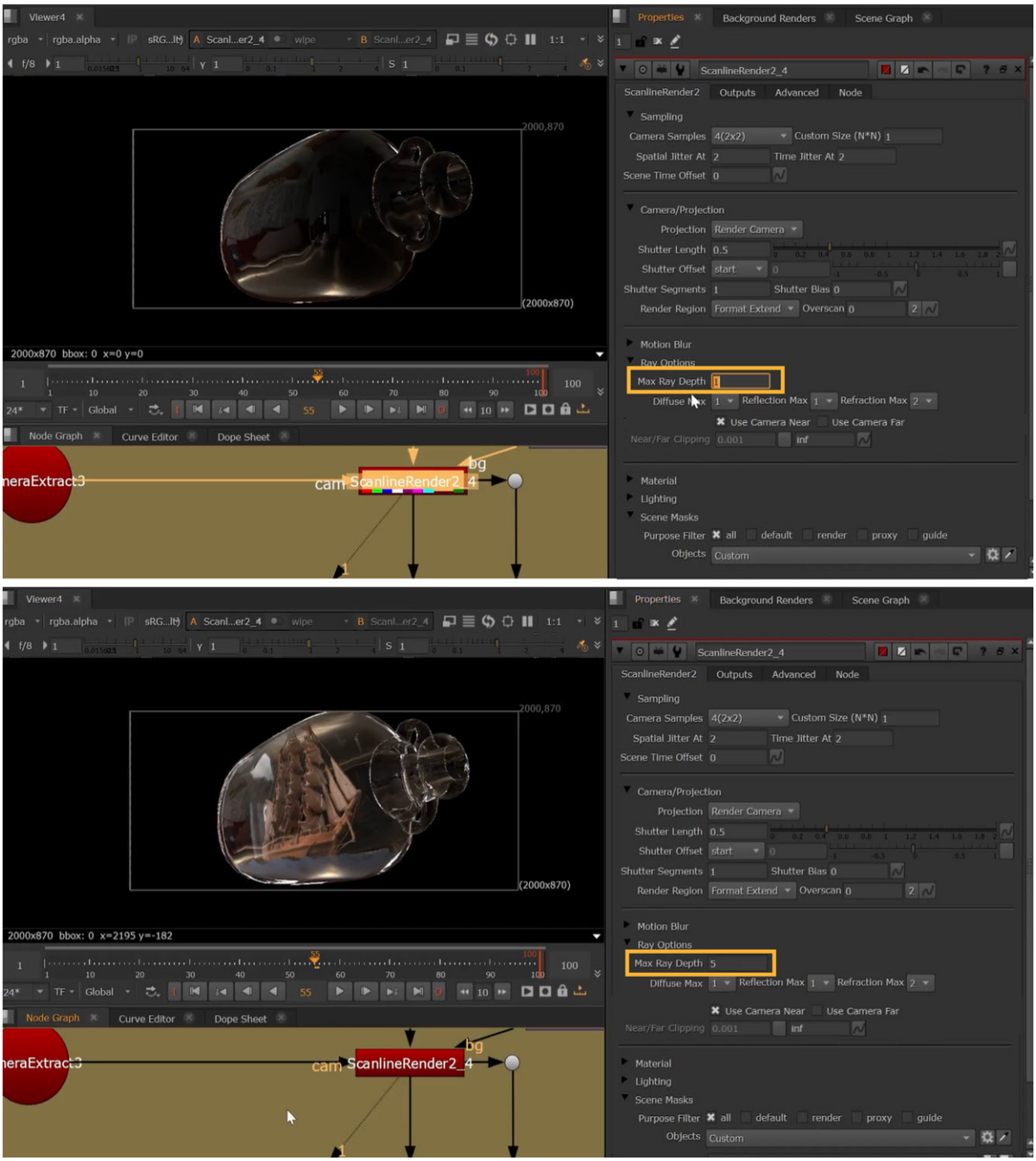
Refraction Max

The 'Refraction Max' knob sets at what depth count refraction rays will stop.



Note: You'll also find clipping options in this section.

For example, changing the **Max Ray Depth** from 1 to 5 makes a strong effect on the reflections in this scene:



Node Reference

ReflectiveSurface

Applies mirror-like reflections to a prim.

ScanlineRender

Renders all the objects and lights in a scene.

Motion Blur with ScanlineRender

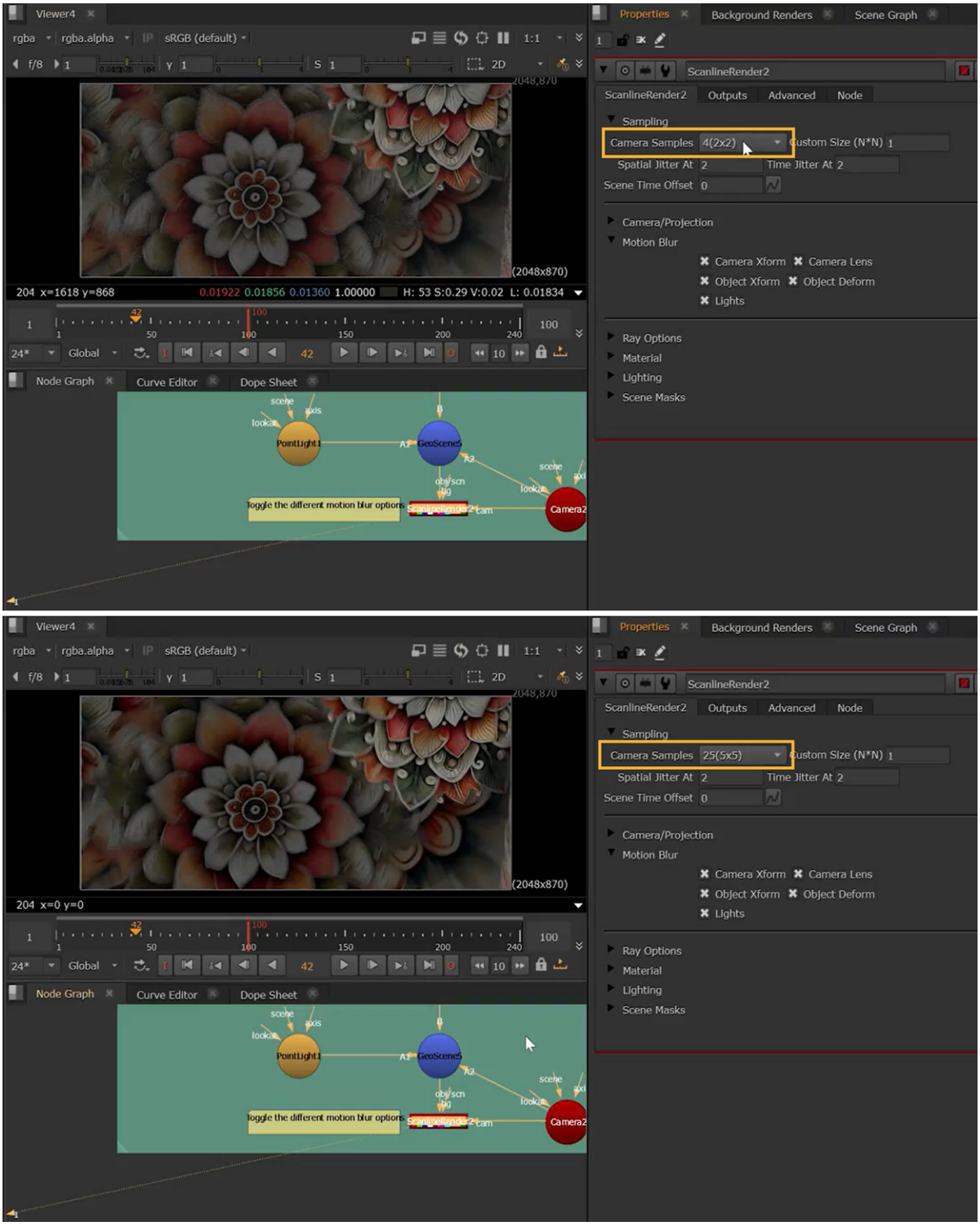
The most common way to affect the motion blur in your 3D scene is to adjust the **Sampling** controls in the ScanlineRender2 node Properties.

Sampling Controls

Camera Samples:

The image is sampled multiple times over the shutter period. Increasing this value is the most accurate, but also the slowest, way of adding motion blur, because the full rendering calculation is done multiple times for each pixel. More samples means smoother motion blur and less aliasing, but a longer render time. However, ScanlineRender2 node doesn't require nearly as many samples as the classic ScanlineRender node, allowing you to get to a smoother result faster.

For example, turning Camera Samples from 4 to 25 results in a cleaner image:



Additionally, because there was no jittering in the classic ScanlineRender mode, you got a "stacking" like effect for motion blur, but jittering can now be controlled making this more effective.

Spatial Jitter At

The 'Spatial Jitter At' knob allows you to control at which sample levels ScanlineRender2 will jitter spatial subpixel locations when doing anti aliasing so that you break up the patterns for better results.

By default the location of the sampling within the pixel itself is being randomly changed at a value of 2 or higher, so only turns on when samples are above 1. When turned off only the exact centre of the pixel is being sampled when samples is 1, or in a uniform grid if greater than 1. This means you have control over which sample levels have uniform sampling (samples at a value less than the 'Spatial Jitter At' value) or jittered sampling (samples at a value equal to or more than the ''Spatial Jitter At' value).

Time Jitter At

The 'Time Jitter At' knob determines at what sample number you start randomly jittering the time for motion blur.

The default value is 2, which means jittering to produce motion blur occurs when samples are at a value 2 or higher. If samples are at 1, no motion blur will be created, but motion vectors will be produced so that you can still blur the image.

Scene Time Offset

The 'Scene Time Offset' knob allows you to shift the frame you are rendering and can be done in subframe increments.

This means you can slip your render in time (with subframe accuracy) without affecting the camera or the objects in the stage, which can be really helpful when you are trying to slip the camera to get vector motion blur aligned properly.

Motion Blur Controls

Of course, the **Motion Blur** section of the Properties also gives you further control over the motion blur in your scene. Toggling the following options affects the motion blur of different elements:

Camera Xform

The 'Camera Xform' knob enables and disables the motion blur of a camera being transformed i.e.

animating a Camera through space

Please note that subsampling matrices produce artifacts currently, as this requires quaternions for the matrix interpolation method which has not yet been implemented. As a workaround increase the Shutter Segments to chop up the frame you are generating the matrix for, into multiple matrices.

Camera Lens

The 'Camera Lens' knob enables and disables the motion blur of a camera projection control being changed i.e. motion blur generated from animating the focal length of the camera

Object Xform

The 'Object Xform' knob enables and disables the motion blur of an object being transformed i.e. animating a GeoCube through space (the mesh transform matrix is animated)

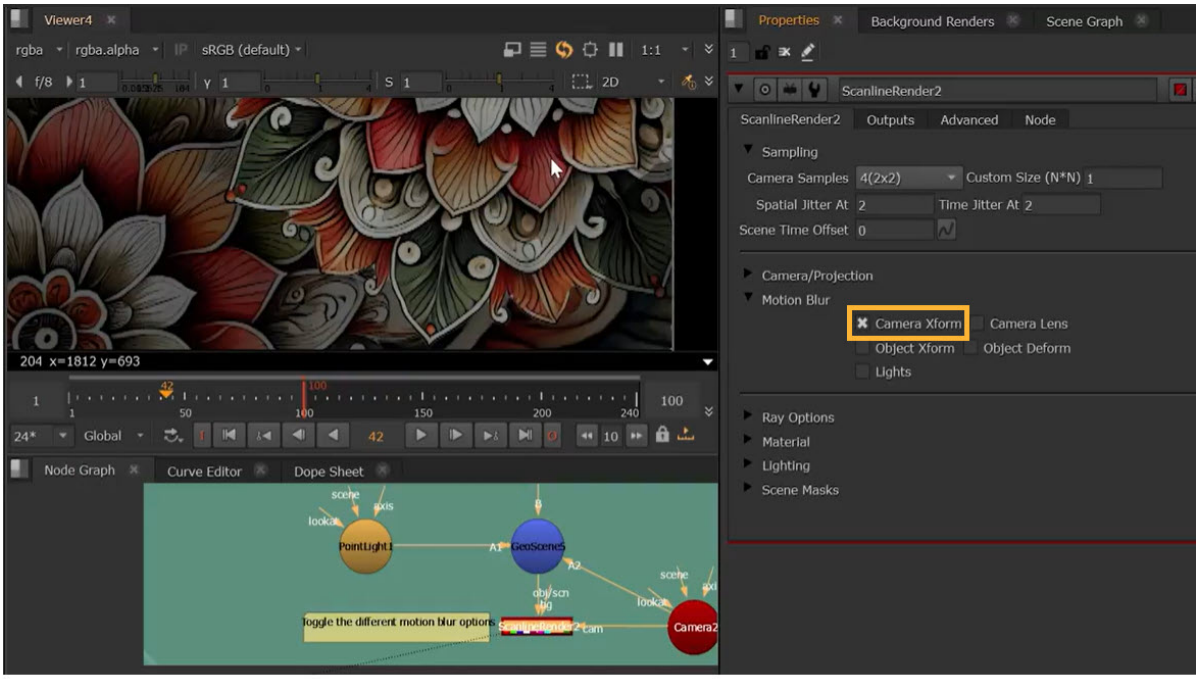
Object Deform

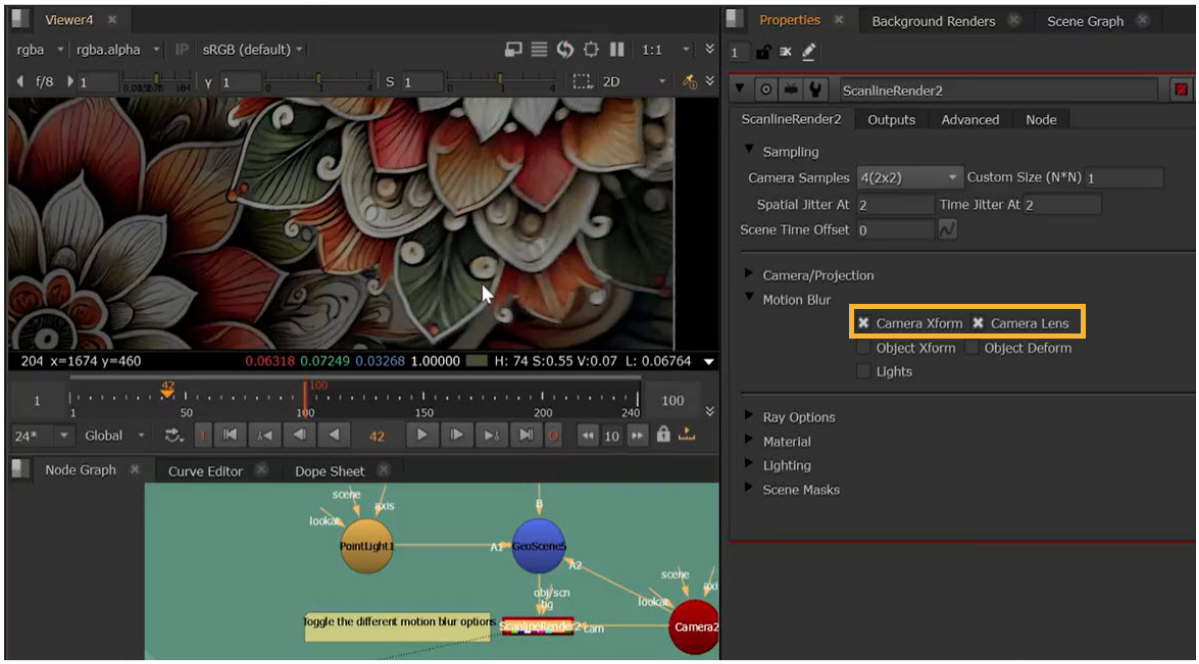
The 'Object Deform' knob enables and disables the motion blur of an object being deformed i.e. an animated radius knob on a GeoSphere (the point xyz locations are animated)

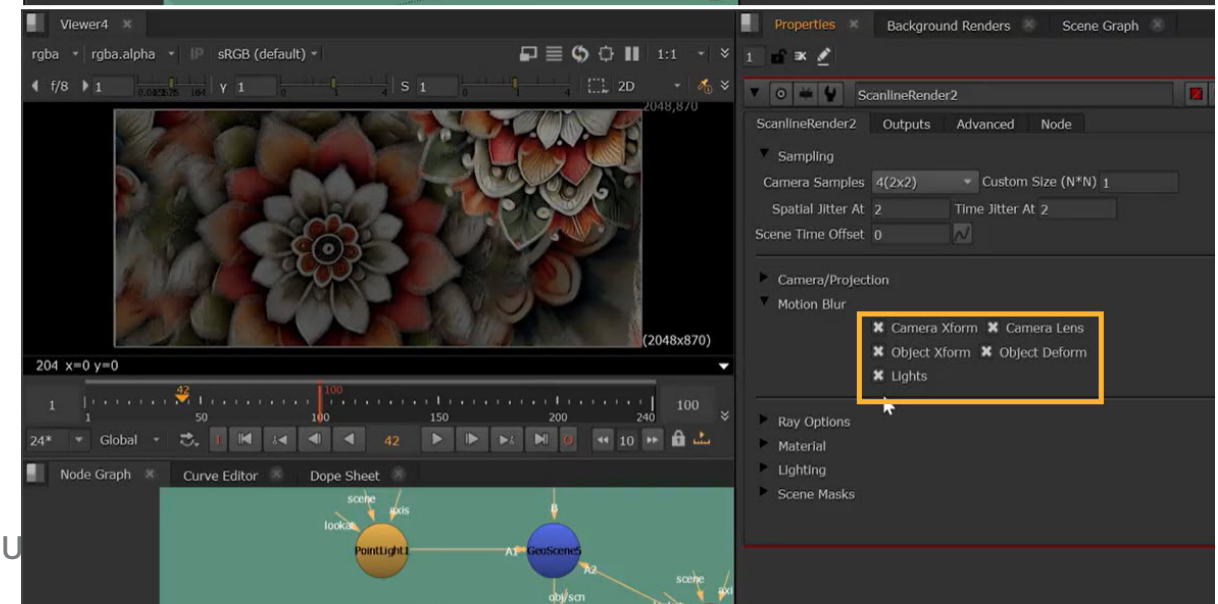
Lights

The 'Lights' knob enables and disables the motion blur of a light being transformed i.e. animating a light through space.

For example, turning on the motion blur of different elements at once:


Viewer4
Properties
Background Renders
Scene Graph
ScanlineRender2
Outputs
Advanced
Node
Sampling
Camera Samples
Custom Size (N*N)
Spatial Jitter At
Time Jitter At
Scene Time Offset
Camera/Projection
Motion Blur
Camera Xform
Camera Lens
Object Xform
Object Deform
Lights
Ray Options
Material
Lighting
Scene Masks
Node Graph
Curve Editor
Dope Sheet
PointLight1
GeoScene5
Camera2


Viewer4
Properties
Background Renders
Scene Graph
ScanlineRender2
Outputs
Advanced
Node
Sampling
Camera Samples
Custom Size (N*N)
Spatial Jitter At
Time Jitter At
Scene Time Offset
Camera/Projection
Motion Blur
Camera Xform
Camera Lens
Object Xform
Object Deform
Lights
Ray Options
Material
Lighting
Scene Masks
Node Graph
Curve Editor
Dope Sheet
PointLight1
GeoScene5
Camera2


Viewer4
Properties
Background Renders
Scene Graph
ScanlineRender2
Outputs
Advanced
Node
Sampling
Camera Samples
Custom Size (N*N)
Spatial Jitter At
Time Jitter At
Scene Time Offset
Camera/Projection
Motion Blur
Camera Xform
Camera Lens
Object Xform
Object Deform
Lights
Ray Options
Material
Lighting
Scene Masks
Node Graph
Curve Editor
Dope Sheet
PointLight1
GeoScene5

Shutter Controls

You'll also find that some of the controls in the **Camera/Projection** section of the ScanlineRender2 node Properties will affect motion blur:

Shutter

The 'Shutter' knob is how long the shutter should be open for when motion blurring.

Shutter Offset

The 'Shutter Offset' knob is how the shutter behaves with respect to the current frame value.

- centered: acts as a snapshot in the middle of the frame, so motion blur would be shown as if you paused halfway through the frame
- start: opens the shutter at the start of the frame, so you see the motion blur ahead of the geo animation
- end: opens the shutter at the end of the frame, so you see the motion blur after the geo animation
- custom: opens the shutter at a time specified by the user

Nuke has always defaulted to 'start' and going forward in time, so at a given frame the shutter is open. However often it can be useful to change this to 'end' when you want a motion blur trail to an impact point on a specific frame.

Shutter Segments

The 'Shutter Segments' knob is the number of time 'segments' used to interpolate motion blur. The greater the number of segments the greater the quality of rotational motion blur.

This helps with rotational motion blur, which gives more realistic results for motion blur on objects with rotational movement like, car wheels, helicopter blades etc.

Shutter Bias

The 'Shutter Bias' knob moves your motion blur to be weighted more to the start of a frame or end of a frame, or one end of the shutter.

Node Reference

ScanlineRender

Renders all the objects and lights in a scene.

Outputting AOVs with ScanlineRender

Instead of just rendering a single, final image as a render, Nuke can be configured to output various AOVs, each containing specific information. AOVs allow compositors to fine-tune elements like lighting and colors, without re-rendering the entire scene.

The ScanlineRender2 node enables greater control over your AOV outputs from Nuke.

In the **Outputs** tab of the ScanlineRender node, you have the following options.

- **1/Z**

  The '1 over z' knob determines how you would like to output your Z depth.
  - On: outputs classic Nuke Z distance, which 1/Z distance
  - Off: outputs absolute Z distance. This is the world space distance of the camera to the object.

- **Motion Vectors**

  The knob 'Motion Vectors' allows you to select the motion vector output type.
  - Off: don't create any motion vectors
  - Distance: stores the distance between the samples in the motion vector channel
  - Distance Normalized: normalized distance between the samples in the motion vector channel
  - Velocity: stores the velocity in the motion vector channel
  - Velocity Normalized: normalized velocity in the motion vector channel

- **Surface AOVs**

  Enable

The ‘enable’ knob sets whether the AOV is rendered.

AOV source

The ‘aov source’ knob are hard coded AOVs that are the source of the output layer

A list of shading variable layers extended if geometry prims provide primvars. For example if a geometry mesh object provides a primvar named ‘myUV’ then it will appear at the end of this list along with its size (float, vec2, vec3, vec4.)

Predefined shading attributes:

• Cf- Geometry color (from primvar if available)

• Co- Geometry opacity (from primvar if available)

• presence - Surface presence (physical solidity)

• Oi - Surface rgb opacity (optical transparency)

• Zlinear - U near-projected distance (classic Z)

• Zdistance - Camera ray distance

• Ps- Shading position

• Pg- Geometric position

• Ns- Shading normal

• Ng- Geometric normal

• Ts- Shading tangent

• UV- Surface parametric-uv coordinate (from primvar if available)

• UVW - Surface parametric-w (Z for volumes, from primvar if available)

• St- Texture coordinate (different than parametric-UV if modified by shader)

• MotionFwd - Forward motion-vectors

• MotionBwd - Backwards motion-vectors

Merge mode

The ‘merge mode’ knob sets how the ‘aov source’ wants to merge with other surfaces along the same ray in depth.

Math to use when merging multiple surface samples in depth:

- premult-blend - Premult A then Under(B + A*Aa*(1-Ba)) or Over(B*(l-Aa) + A*Aa) - best for data Aovs
- blend - Under(B + A*(1-Ba)) or Over(B*(l-Aa) + A) -
- best for color Aovs
- plus -B+A
- min - min(B, A) - best for Z
- mid -(B+ A)/2
- max - max(B, A)

Unpremult

The ‘unpremult’ knob sets whether you would like to unpremult the final combined output.

Output layer

The ‘output layer’ knob sets the output layer I want to send this data to on render

Once you’ve rendered, this data will be included in the output layer selected.

Export From Your Scene

To export your 3D data from the New 3D system, you can use the GeoExport node. Currently this node allows for the exporting of a flattened USD stage as **.usd**, **.usda**, **.usdc**, **.abc**, or **.ply** files.



Note: The export of **.ply** files is intended for the export of Gaussian Splats but it also supports exporting meshes. However, note that the two cannot be combined - i.e. you can't export a scene which is a mix of meshes and splats.

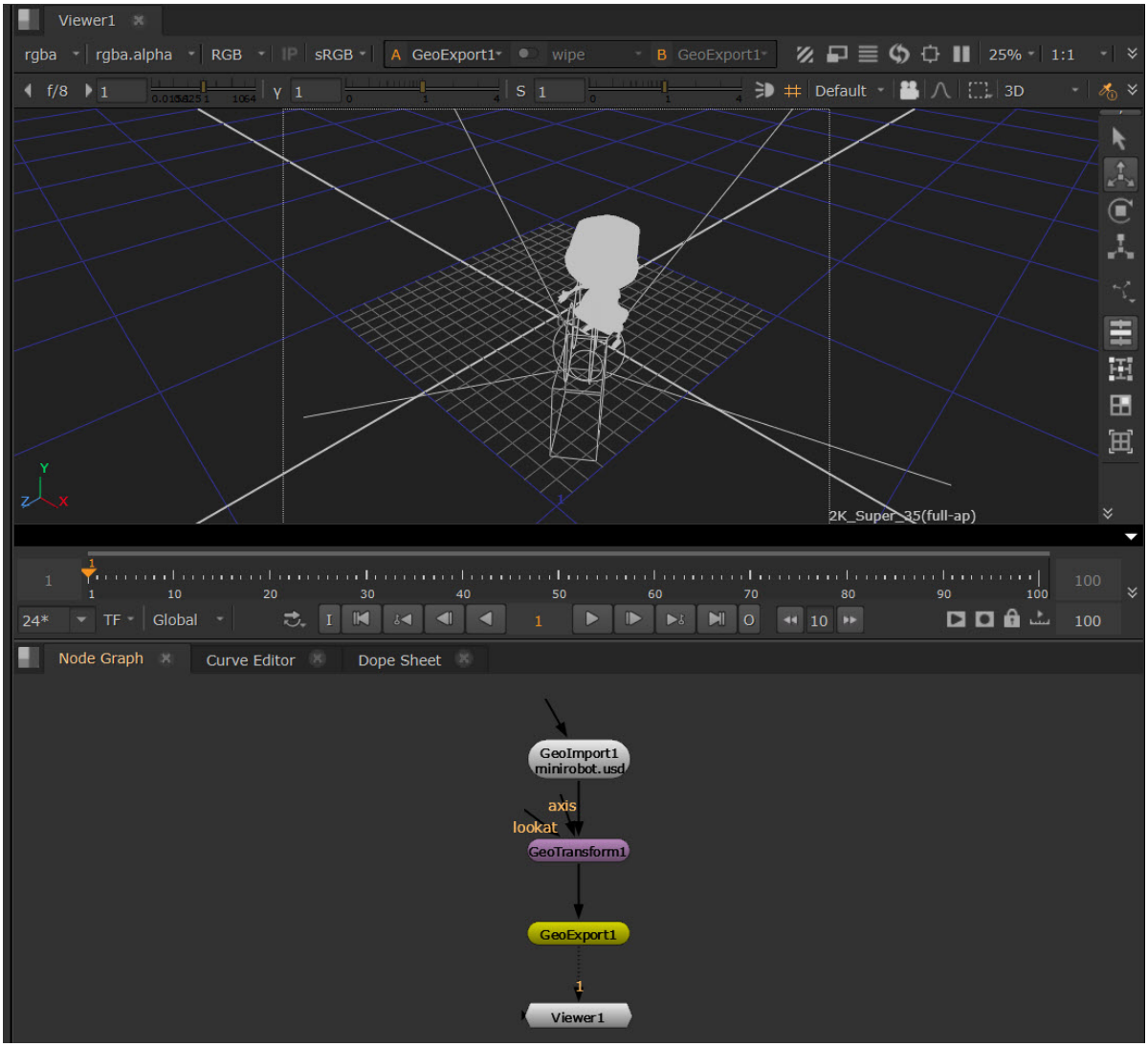


Note: Skew and non-uniform scales applied to Gaussian Splats will introduce artifacts into higher-order spherical harmonic data, when exporting to the **.ply** file format.

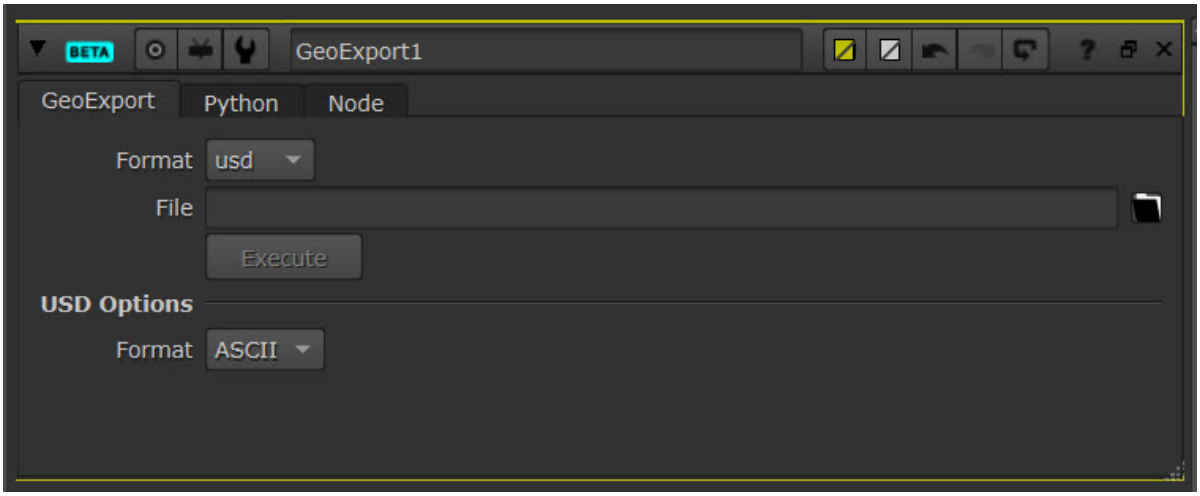
Note: For those looking to export geometry to the Classic Nuke 3D System, from the New 3D System, you can export a flattened USD stage and import the USD file via the ReadGeo node.

To export your scene:

1. Add a GeoExport node downstream of your scene in the node graph.



2. In the GeoExport node Properties, select a **Format** for your export files from Alembic (**.abc**), USD (**.usd**, **.usda**, **.usdc**), or **.ply** for Splats. If you select **usd**, you'll also get further **USD Options** below where you can select from **ASCII** or **Crate**.



3. Choose a file path in the **File** field.
4. Select the **Execute** button and your export will be saved in the location specified above.

Export Node

GeoExport

Exports a flattened stage to a USD file.

Debugging Your Scene

If something isn't behaving as expected, there are a few ways for you to inspect the scene and diagnose the issue in your New 3D Scene.

- Digging into the scene's code may provide the information you need to pinpoint any mistakes. To do this, you can use the GeoViewScene node.

See Inspecting with Inspecting with GeoViewScene Node.

- If you have some knowledge of USD and Python, then the GeoPython node can help with editing your USD stage.

See more at Editing with GeoPython Node.

- Nuke has added new error messages which will appear on various panels, to prompt you as to what might be the issue.

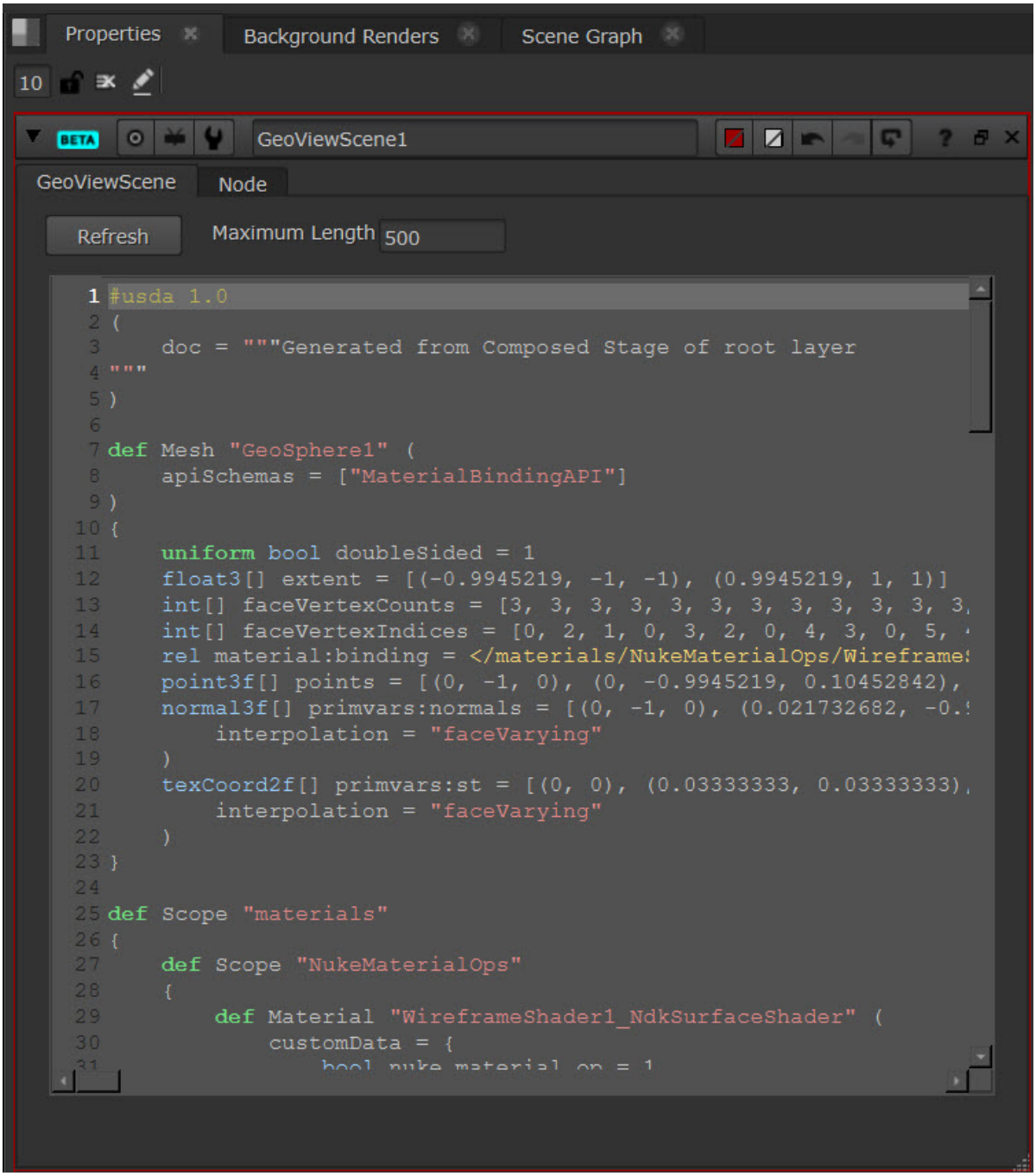
Error Message Examples

- If you try to import an invalid scene via GeoImport, you'll get a red warning bar at the top of the Viewer.
- If you try to bind materials to instanced prim via GeoBindMaterial, you get a yellow warning triangle in the Mask knob.

Inspecting with GeoViewScene Node

GeoViewScene node that lets you view the USD stage in a read-only text form and is primarily used for debugging.

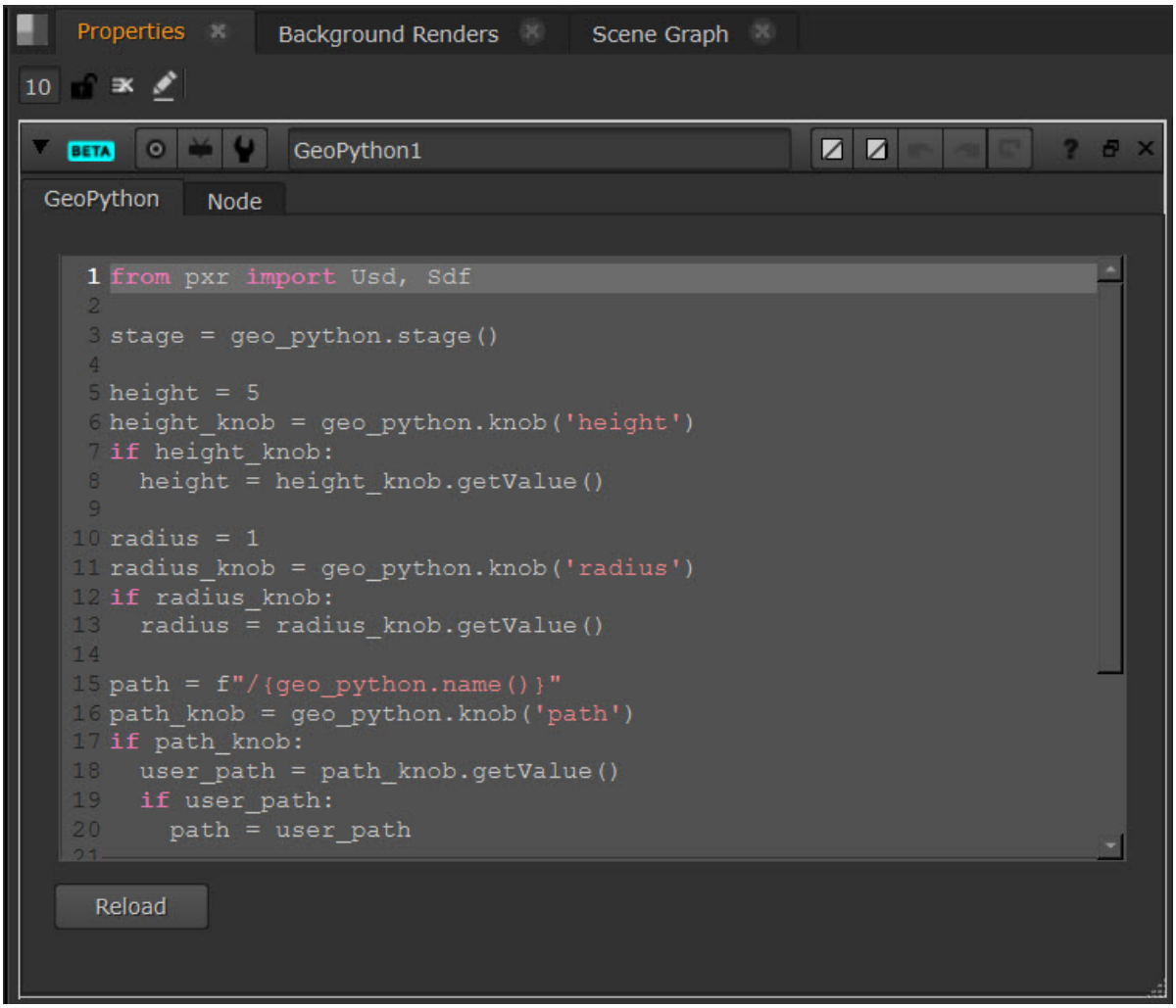
1. Connect a GeoViewScene node downstream of your scene in the node graph.
2. Open the GeoViewScene Properties panel and select **Refresh** to see the scene's code.
3. You can update the **Maximum Length** knob and then hit **Refresh** to increase or decrease the amount of lines of code shown.



Editing with GeoPython Node

The GeoPython node allows you to directly edit the Python code in your stage. If you have some knowledge of USD and Python then this node can be very powerful in creating new schemas and editing your USD stage.

1. Connect the GeoPython node downstream of your stage in the node graph.
2. You can then inspect and edit the code in the GeoPython node Properties panel.
3. If you make changes to your stage and want to view the updates in the GeoPython node, you'll need to select **Reload**. This refreshes the code.

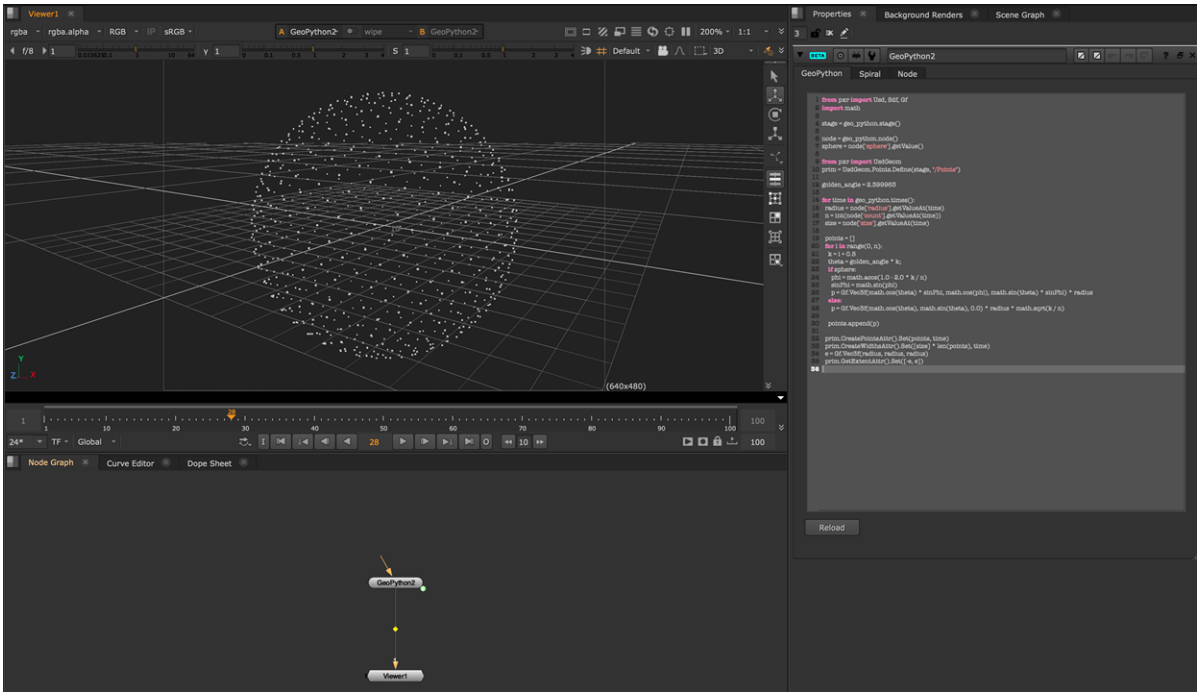


Example Code

For example, this would make a 3D sphere of points:

```
from pxr import Usd, Sdf, Gf
import math
```

```
stage = geo_python.stage()
node = geo_python.node()
sphere = node['sphere'].getValue()
from pxr import UsdGeom
prim = UsdGeom.Points.Define(stage, "/Points")
golden_angle = 2.399963
for time in geo_python.times():
  radius = node['radius'].getValueAt(time)
  n = int(node['count'].getValueAt(time))
  size = node['size'].getValueAt(time)
  points = []
  for i in range(0, n):
    k = i + 0.5
    theta = golden_angle * k;
    if sphere:
      phi = math.acos(1.0 - 2.0 * k / n)
      sinPhi = math.sin(phi)
      p = Gf.Vec3f(math.cos(theta) * sinPhi, math.cos(phi), math.sin(theta) * sinPhi) * radius
    else:
      p = Gf.Vec3f(math.cos(theta), math.sin(theta), 0.0) * radius * math.sqrt(k / n)
    points.append(p)
  prim.CreatePointsAttr().Set(points, time)
  prim.CreateWidthsAttr().Set([size] * len(points), time)
  e = Gf.Vec3f(radius, radius, radius)
  prim.GetExtentAttr().Set([-e, e])
```



Learn more about Python in the Nuke Python Developer's Guide.

Working with Gaussian Splats

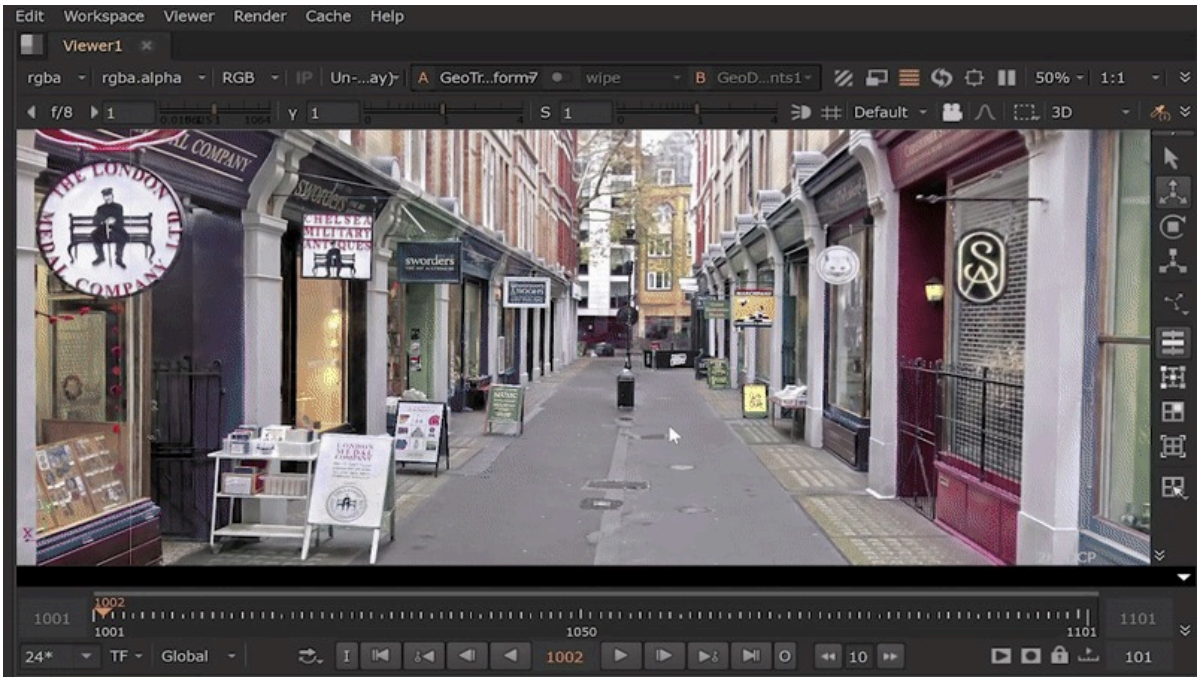
You can work natively with Gaussian Splats in the New 3D System. It's possible to import, view, manipulate, and render Gaussian Splats in Nuke.

This opens up new environment workflows for set extensions and matte painting, as well as greater flexibility for element integration.

See the pages below:

- Importing Gaussian Splats
- Manipulating Gaussian Splats
- Rendering Gaussian Splats

Tip: To see these features in practice, also see our Intro to Gaussian Splats video.



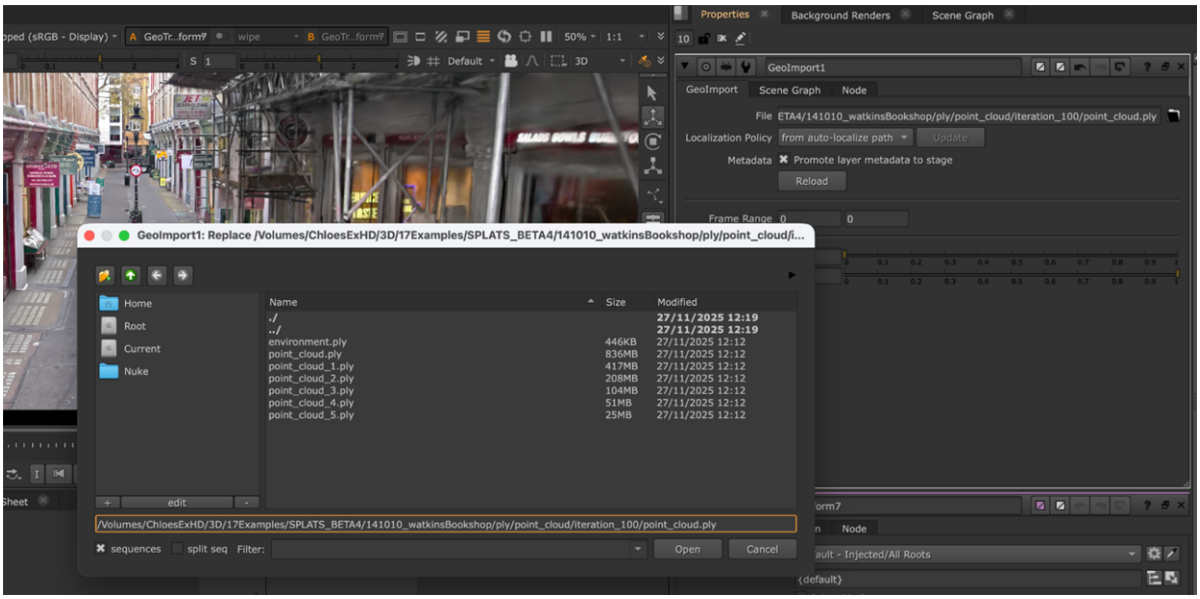
Importing Gaussian Splats

Gaussian Splats (**.ply** and **.splat** formats) can be imported into Nuke using the GeoImport and GeoReference nodes.

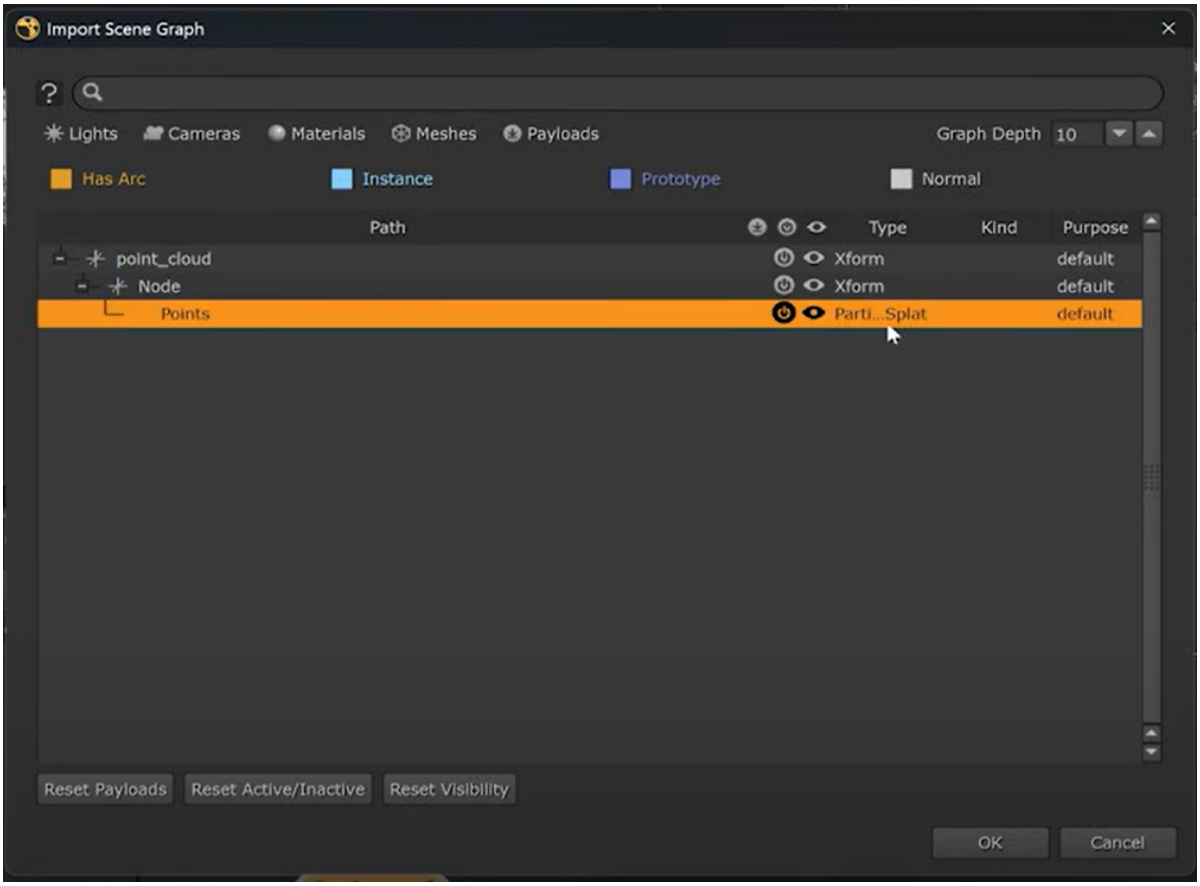
Note: Learn about the difference between GeoImport and GeoReference at Importing into Your Scene.

In this example, we are importing the Splats with GeoImport.

1. Insert a GeoImport node. In the node properties, select the file icon to open your file browser.
2. Select a **.ply** or **.splat** file to import into Nuke.



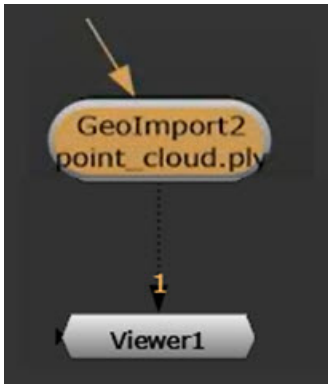
3. The **Import Scene Graph** dialog will appear. Here you can set payloads, activation, and visibility before importing if needed. For more information about these options, see Import Scene Graph Dialog.



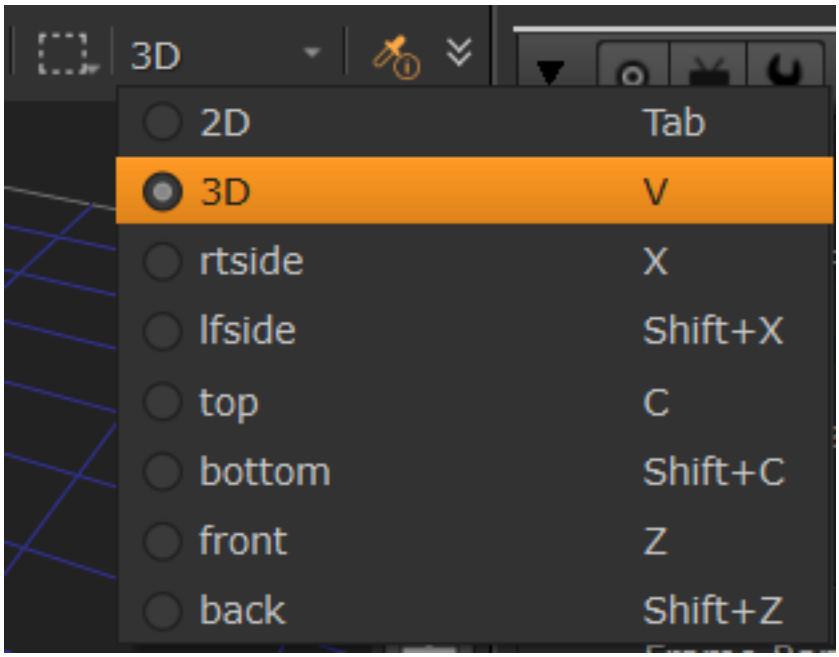
4. Once you have made your selections you can hit **OK**, and GeoImport will load your splat into Nuke with payloads, activation, or visibility that you have enabled.

Viewing Gaussian Splats

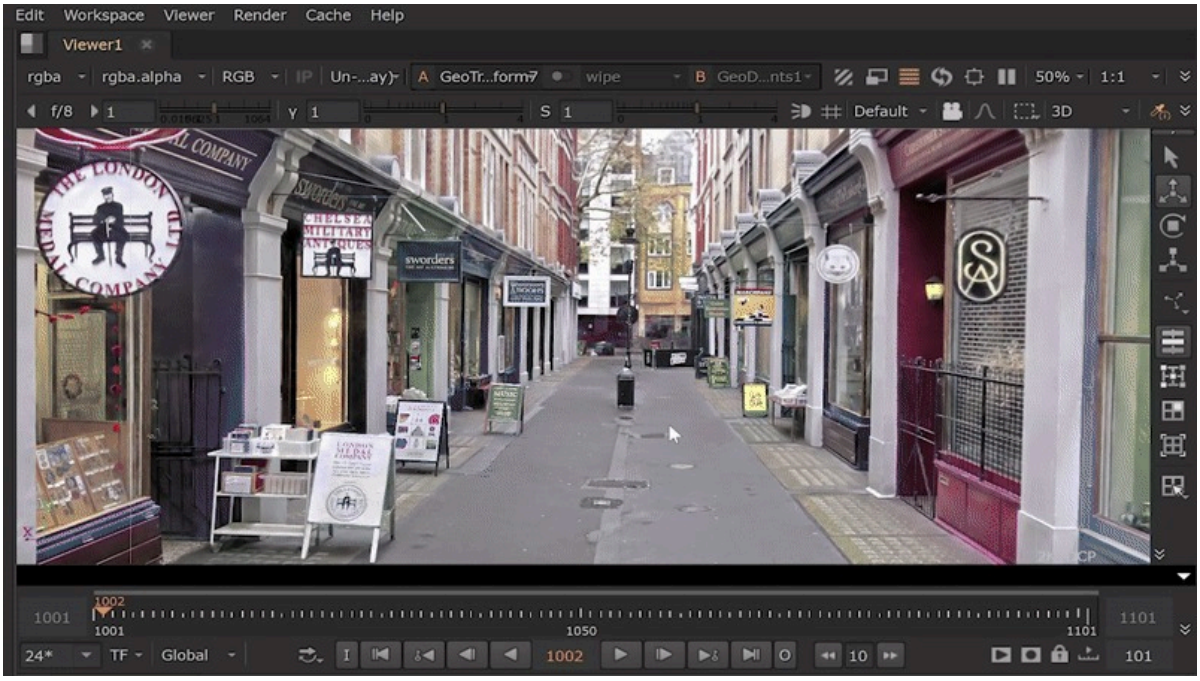
1. Connect a **Viewer** downstream of your imported splats to view them.



2. Make sure the **Viewer** is in **3D** mode to see this.



3. You'll see a performant Hydra based GPU rendering of the Gaussian Splat.



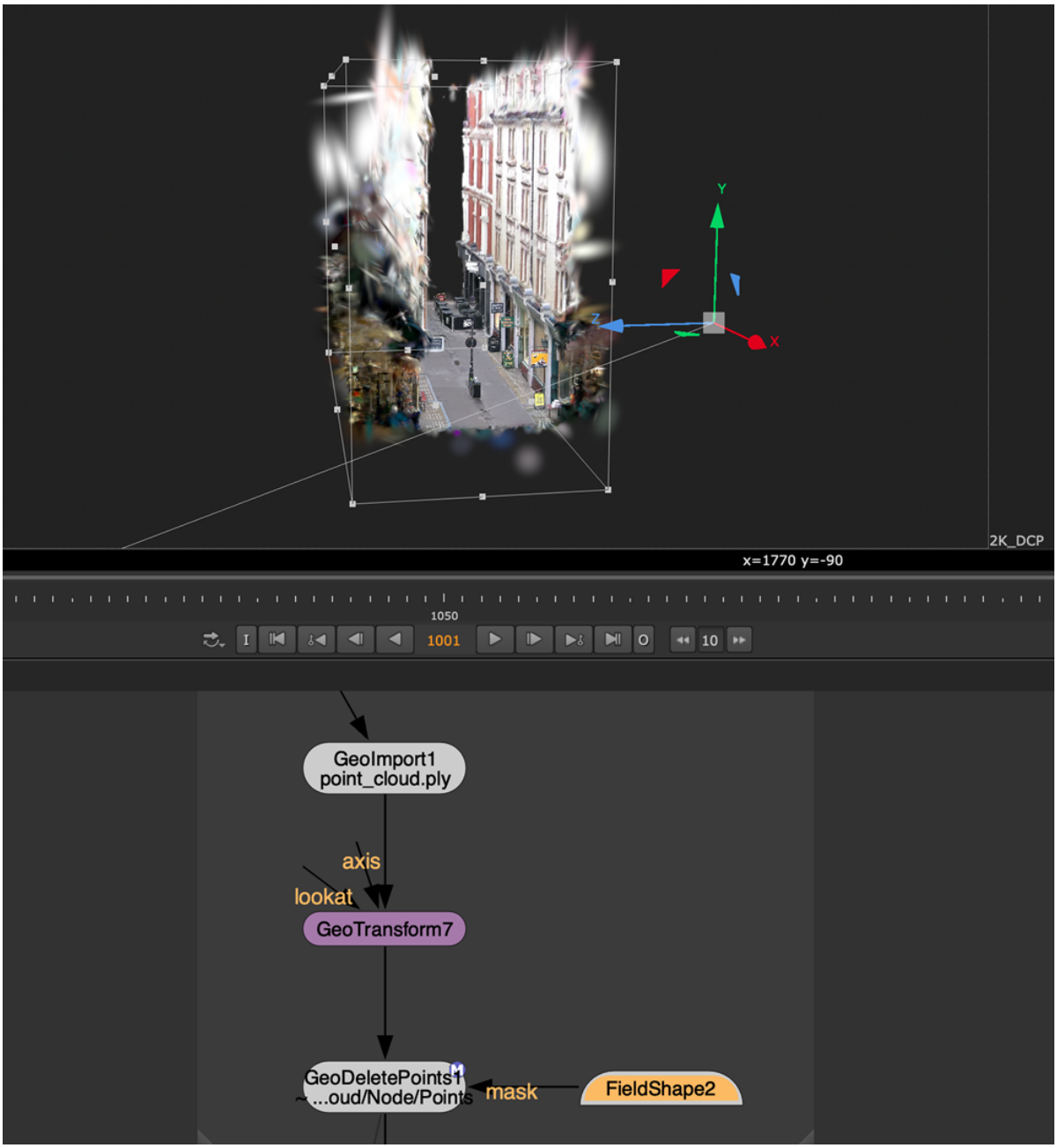
4. Next, see Manipulating Gaussian Splats to learn more about how to manipulate splats.

Manipulating Gaussian Splats

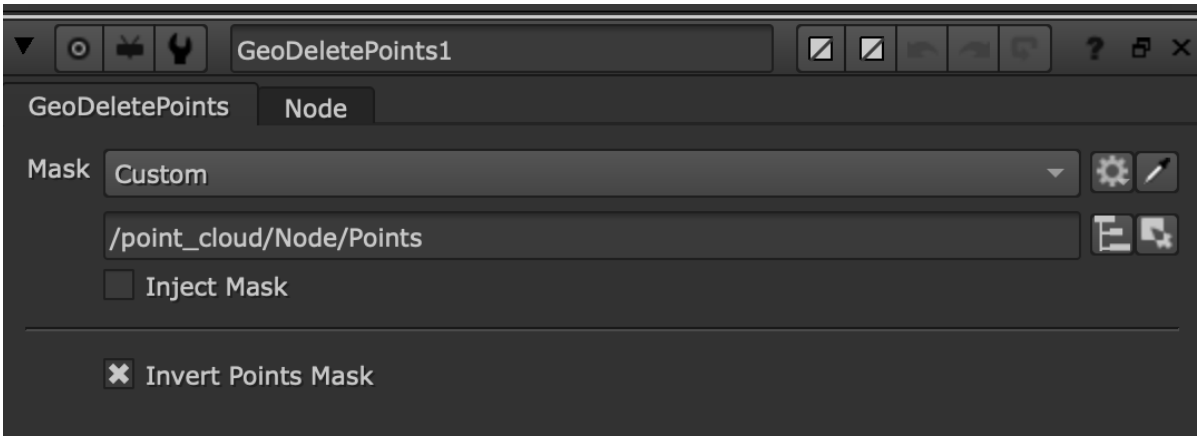
Manipulation of Gaussian Splats can be done using the Field Nodes. This is a way to create volumes that can be used to mask and manipulate 3D data.

In this example, we show removing and adding elements to an imported Splat scene. This Splat has been imported via GeoImport - see Importing Gaussian Splats if you haven't already imported a splat.

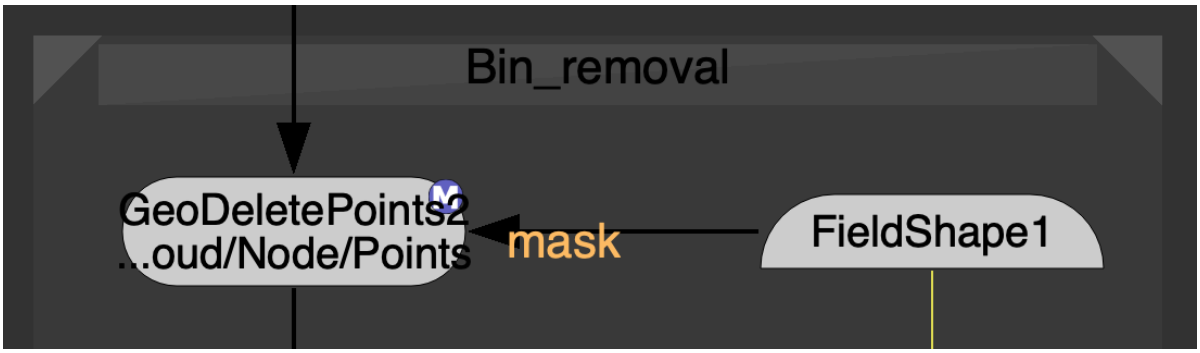
1. Before manipulating the splat, we are starting by using GeoTransform to rotate the imported splat. This can be helpful when setting up. See more about transformations at Transformations in 3D Scenes.
2. We first want to remove parts of the splat. To do this, first we are inserting a GeoDeletePoints and a FieldShape node. The FieldShape node is being used to create a mask for the GeoDeletePoints node, defining a section of our splat that we want to work within.



In the GeoDeletePoints, by enabling **Invert Points Mask**, points outside the mask will be deleted.

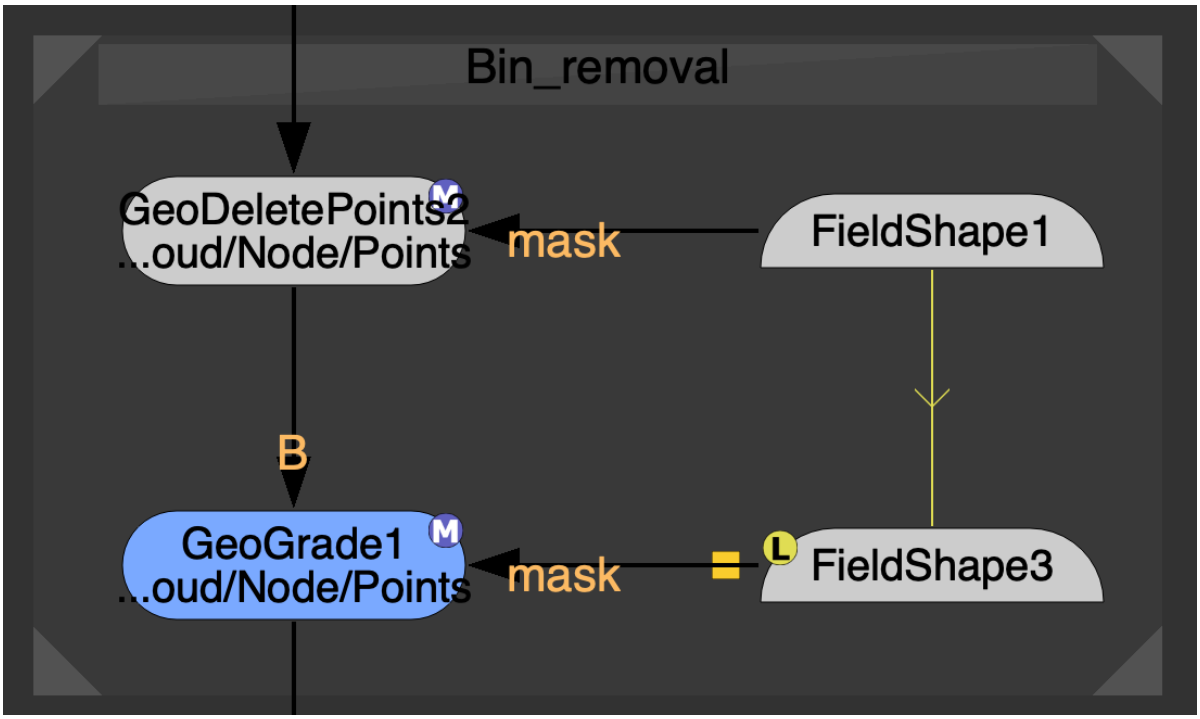


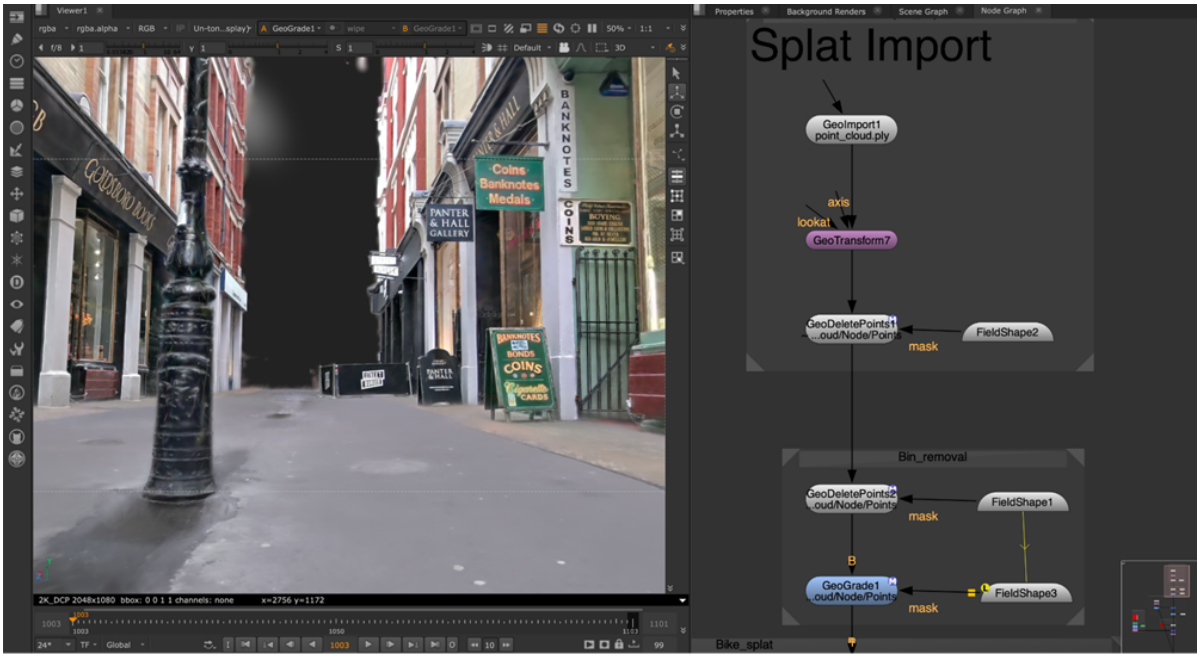
3. This can then be done a second time, using another FieldShape to create a mask for another GeoDeletePoints node. This time the FieldShape targets the specific object to remove in the **mask** - in this case, the bin in the scene.



4. To clean up the area where the bin no longer sits, we can use the GeoGrade node.

In this example, we have targeted a section of the pavement where the bin was as a **mask** for the GeoGrade.





5. Now that we've cleaned up our plate, we can also merge splats together as needed. To do this, we can use GeoMerge.

   Once you're happy with your scene, you can then render with SplatRender. See Rendering Gaussian Splats.

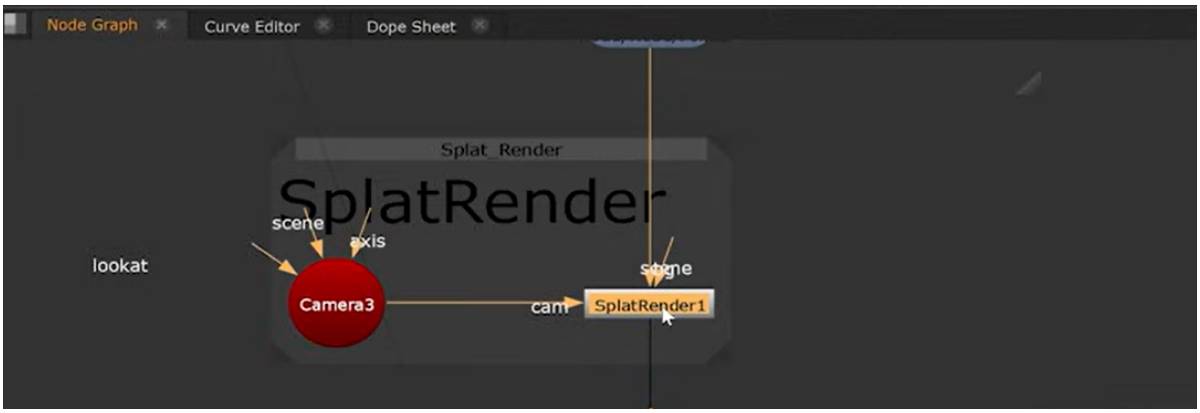
Note: See Field Nodes and LABS Nodes in the Reference Guide to see the full range of nodes you can use to manipulate splats.

Rendering Gaussian Splats

Use the SplatRender node to render splats to 2D pixels, along with Depth and Deep output.

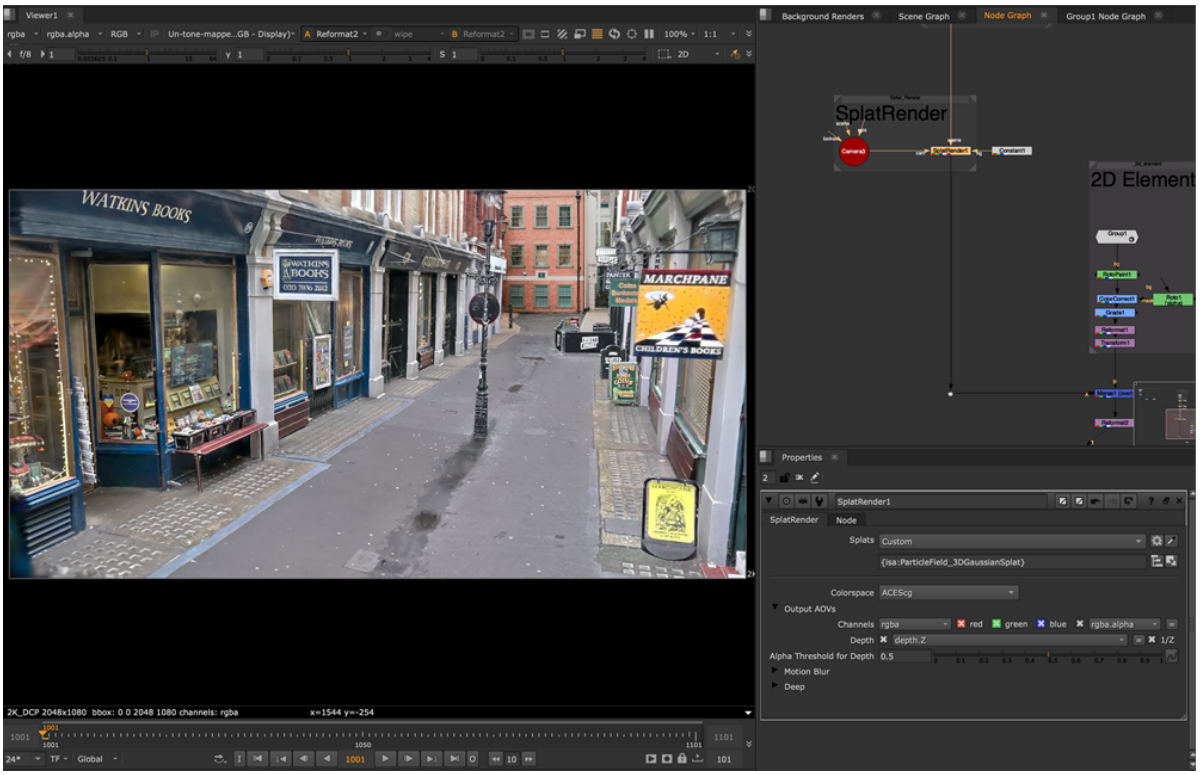
Note: See Importing Gaussian Splats and Manipulating Gaussian Splats to view earlier steps for working with splats.

Connect your splat to the **scene** input and add a Camera node to the SplatRender's **cam** input to render from the camera.



Inside the SplatRender node, you can adjust channel and colorspace output along with Motion Blur options. See the Reference Guide page to see all the controls - SplatRender.

Then change the Viewer to **2D** to see the result.



Once rendered, you can then easily use the splat in existing comp workflows, such as using the Depth or Deep channels to merge the splats with other elements in your shot.

For this, you could use a Read to add images and Merge them downstream of SplatRender. For example, you might want to add a 2D background or make other adjustments.

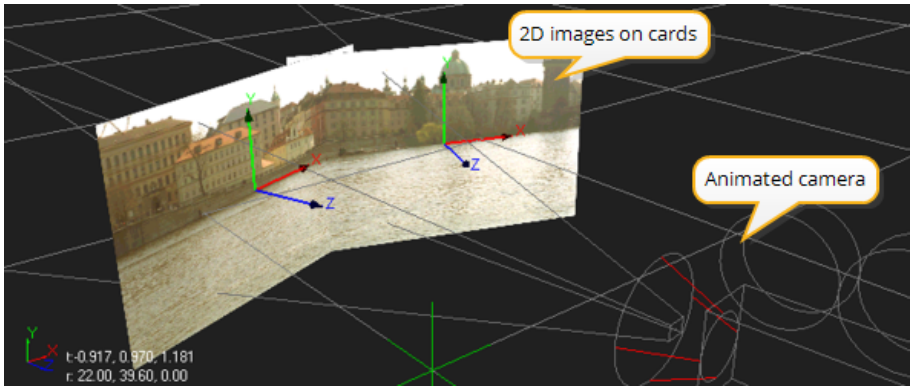
Classic 3D Compositing

Nuke's Classic 3D system workspace allows you to set up a 3D composite for camera moves, set replacement, and other applications where you need to simulate a "real" dimensional environment.

Overview

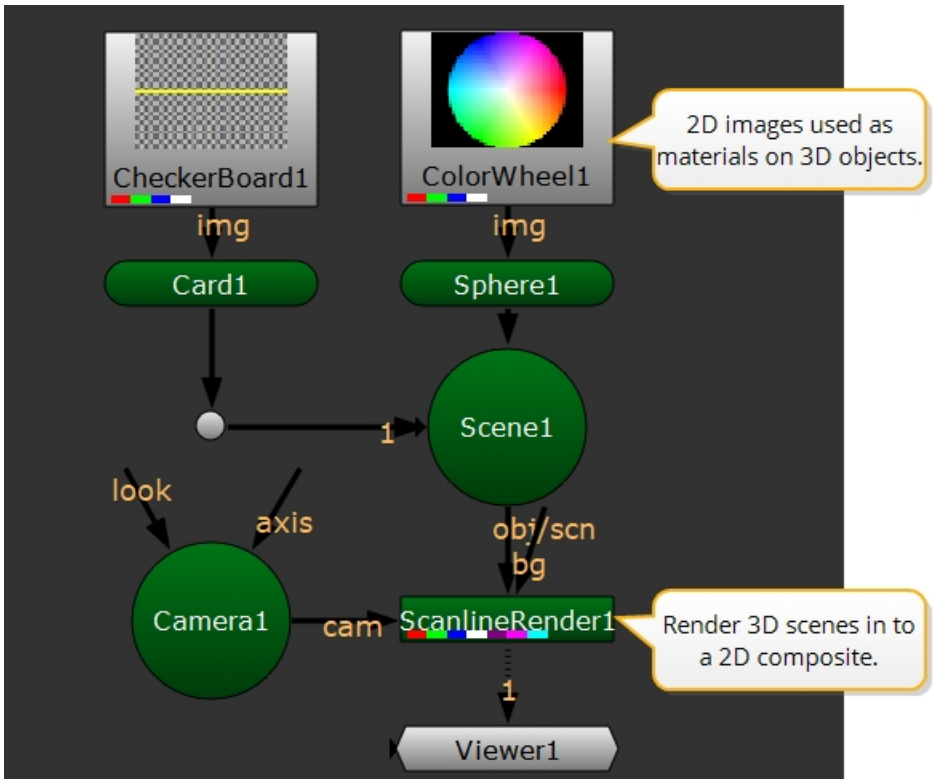
This chapter explains how to set up a 3D scene in Nuke's Classic 3D System, and how to add objects and cameras in the 3D workspace. You'll also see how to texture objects, transform objects and cameras, and render out scenes for use in other areas of your script.

Although the 3D workspace has many potential uses, you're most likely to use it - at least initially - to create pan-and-tile scenes. These are scenes with 2D image planes arranged into a curved shape, and then rendered out through an animated camera to give the illusion of a seamless environment.



Simple pan-and-tile scene.

The 3D objects in Nuke appear as round shapes to differentiate them from objects that perform 2D operations. As shown above, you can mix 2D and 3D objects together in the node tree. For example, you can texture a 3D object with a 2D clip, or take the rendered output from a 3D scene and use it as a 2D background.



Script with 2D and 3D operators

'Classic' 3D System

There are two 3D systems, and sets of nodes, in Nuke - the **Classic 3D System** and the **New 3D System**.

The Classic 3D System nodes are green by default, and New 3D System nodes are colored depending on the node type.

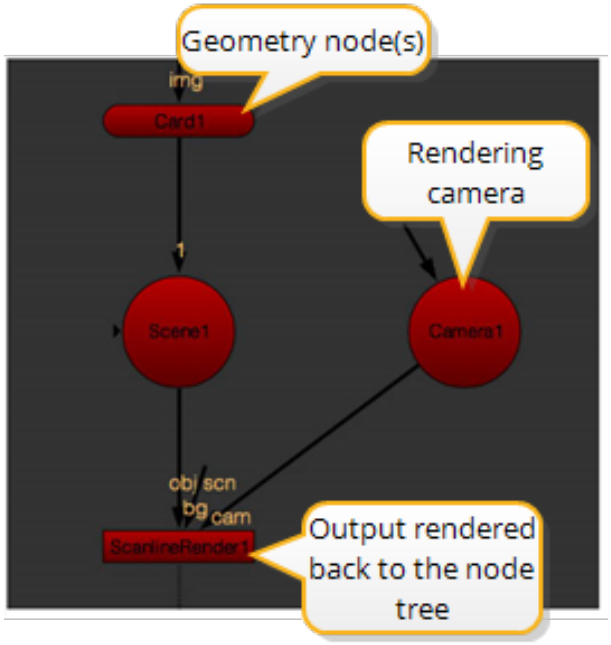
The Classic 3D System nodes are found in right click node menu under **3D** > **3D Classic**, whereas the New 3D System Nodes are under **3D** > **3D**.

Read more about the difference between the two systems at 3D Compositing with the New 3D System.

Setting Up a Scene

Each Classic 3D System scene includes the following objects: a Scene node, a Camera node, one or more geometry nodes (i.e., card, sphere, obj), and a ScanlineRender node. Examples of 3D scenes are shown in Overview and the figure below. In the example, the Scene node receives the output from the geometry

nodes and sends the composite of those objects to the ScanlineRender node, where the output is converted back to 2D.



Core nodes for a 3D composite.

Your script may contain multiple Scene nodes, cameras, and 3D render nodes. All 3D objects loaded in the Properties Bin appear in the 3D Viewer, regardless of whether they are connected to the same Scene node.

The Scene Node

Regardless of its location in your script, the Scene node is the highest-level node in the scene hierarchy because it references all the elements in a 3D workspace - all the geometric objects, cameras, and materials.

To Add a Scene Node

Select **3D** > **Scene** from the Toolbar.

The ScanlineRender Node

Every Scene node in a script should be connected to a ScanlineRender node, which tells Nuke to render the results of the scene. The ScanlineRender node also allows you to toggle between a 2D and 3D view of the scene.

To Add a ScanlineRender Node

1. Select the Scene node.
2. Select **3D** > **ScanlineRender** from the Toolbar.
3. Connect the **obj/scn** input to a Scene or geometry node.
4. Connect the **cam** input to the main camera.
5. Connect the optional **bg** input to composite a background image into the scene.
6. Press **Ctrl+I** (**Cmd**+**I** on a Mac) to open a new Viewer to display the output of the ScanlineRender node.

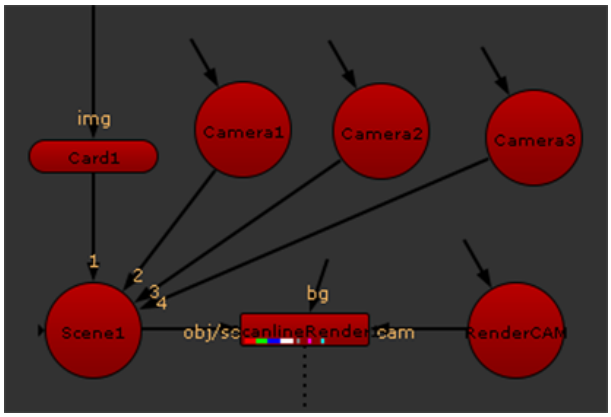
When an image is connected to the bg input, its resolution becomes the output resolution for the ScanlineRender node.

The Camera Node

Cameras may be connected to either the Scene node or the ScanlineRender node. The camera connected to the ScanlineRender node is the camera used for rendering.

1. Select **3D** > **Camera** to insert a camera node.
2. Drag an output connector from the Camera node to a Scene node or connect the Camera node to a ScanlineRender node's cam input.

When connecting cameras for the 3D scene, the camera you want to use for rendering should be connected to the ScanlineRender node, like this:

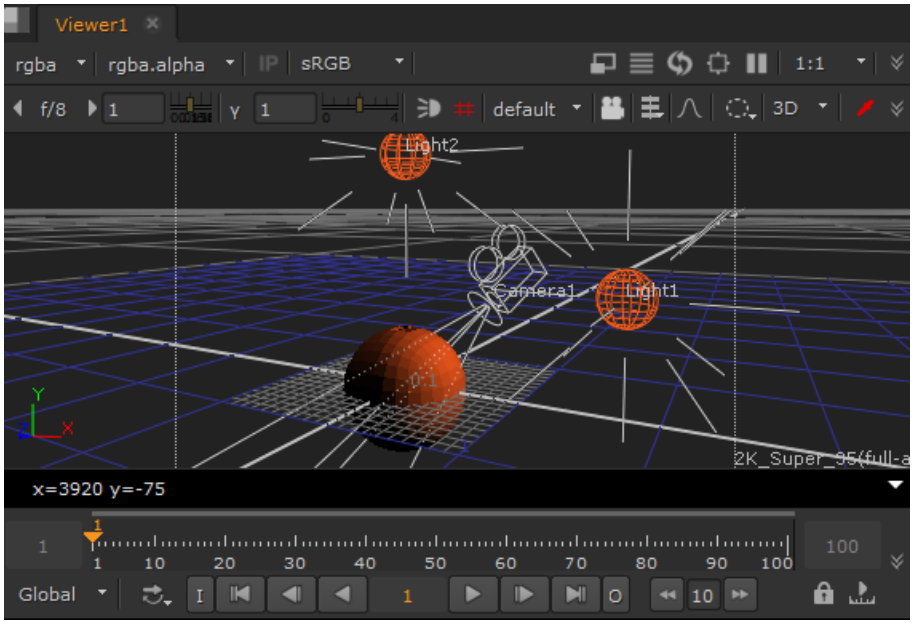


Connecting cameras to the scene.

Any additional cameras should be connected to the Scene node. When you have multiple cameras associated with a 3D scene, you can switch between them by selecting the viewing camera from the dropdown menu at the top of the Viewer. See the next section, Using the 3D Viewer, for more information.

Using the 3D Viewer

When you have a 3D setup in your script, any Viewer window can toggle between the 2D and 3D display modes. The 2D mode shows the result of a rendered scene, the 3D mode shows the perspective from cameras in the scene.



The 3D Viewer.

When you do not have a Camera node in your script, the 3D Viewer uses default views (see the figure below for the list of options). These views are similar to different cameras that you can look though, but they don't appear as objects that you can manipulate in the scene.

Choosing the 3D Viewer Type

Nuke 13.0 onwards includes a new Hydra 3D viewport, which uses hdStorm as a new renderer. The Hydra Viewer is more consistent with other applications in your pipeline, such as Katana, Solaris, or USDView, and the represents the output of the ScanlineRender node.

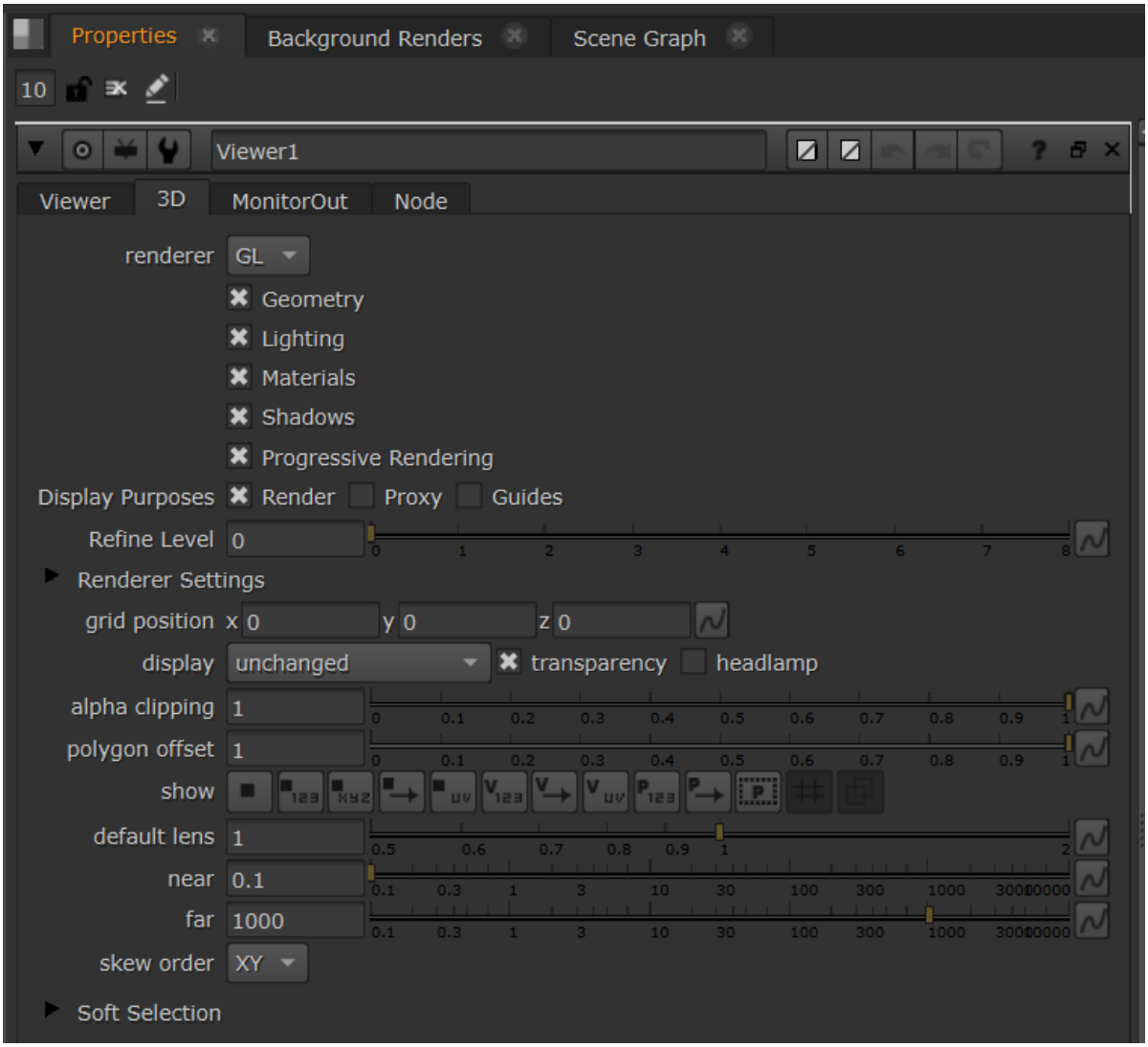


Nuke's **New 3D System** uses a Hydra renderer, and nodes in the **Classic 3D System** use a "Nuke" renderer.

As such, the Viewer's **renderer** knob only has one option: **GL**, which is both the Hydra and Nuke renderers, depending on if you are viewing New or Classic 3D nodes.

Note: Learn more about the New 3D System and USD at 3D Compositing with the New 3D System.

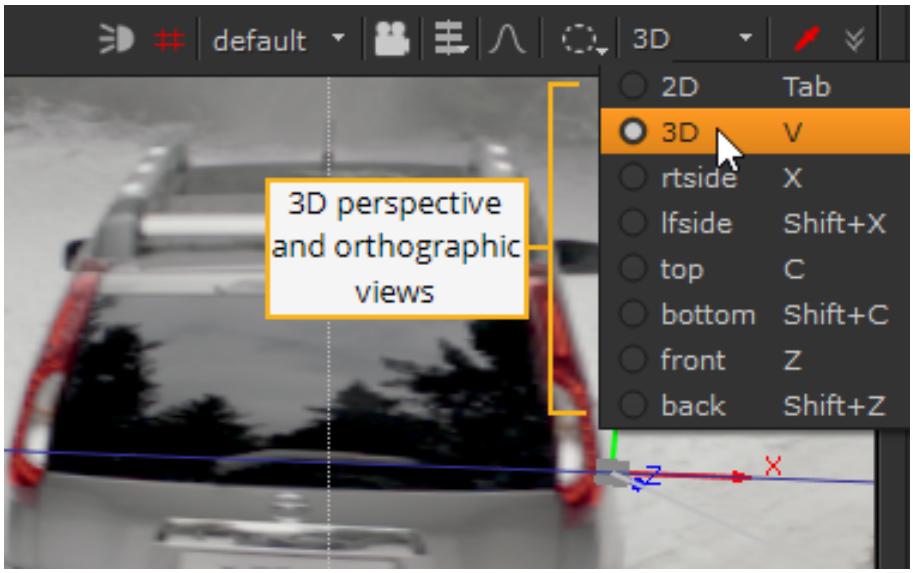
You can also select whether or not to display geometry, lights, materials and/or shadows in the 3D Viewer by toggling the relevant checkboxes.



The Viewer renderer option is also included in the **Preferences** under **Panels** > **Viewer** > **default renderer**. Changing the preference only affects new Viewer nodes.

Switching to the 3D Viewer

Open a Viewer and press **Tab** or **V** to toggle between the 2D and 3D modes - or select the view you want from the dropdown menu at the top right corner of the Viewer window.



Switching to the 3D Viewer.

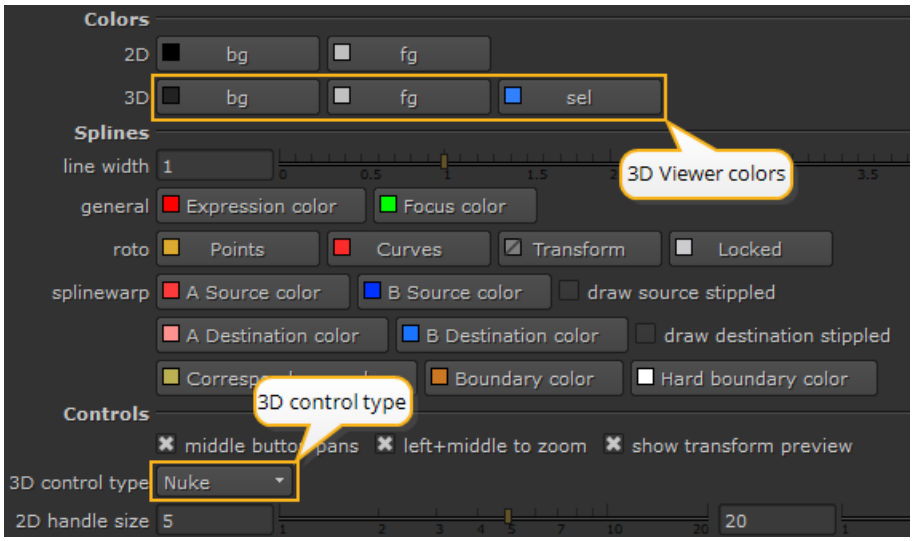
The "built-in" views give you different perspectives on your 3D scene. You can quickly switch between the views by pressing the keyboard shortcuts for right view (**X**), left view (**Shift+X**), top view (**C**), bottom (**Shift+C**), front (**Z**), back (**Shift+Z**), and three-quarter perspective (**V**).

To Navigate in the 3D Viewer

- Dolly: Press **Alt** and **middle-mouse-button** drag.
- Pan: Press **Alt** and **left-mouse-button** drag.
- Tilt: Press **Ctrl/Cmd** and **left-mouse-button** drag.
- Spin: Press **Ctrl/Cmd** and **left-mouse-button** drag.
- Roll: Press **Ctrl/Cmd+Shift** and **left-mouse-button** drag.
- Look through camera: Select a camera object, press **H**.
- Fit the scene: Press **F** to fit the entire 3D scene within the Viewer.

To Change the 3D Viewer Display Properties

1. Open the **Preferences** dialog (**Shift+S**), and select **Panels** > **Viewer Handles**.



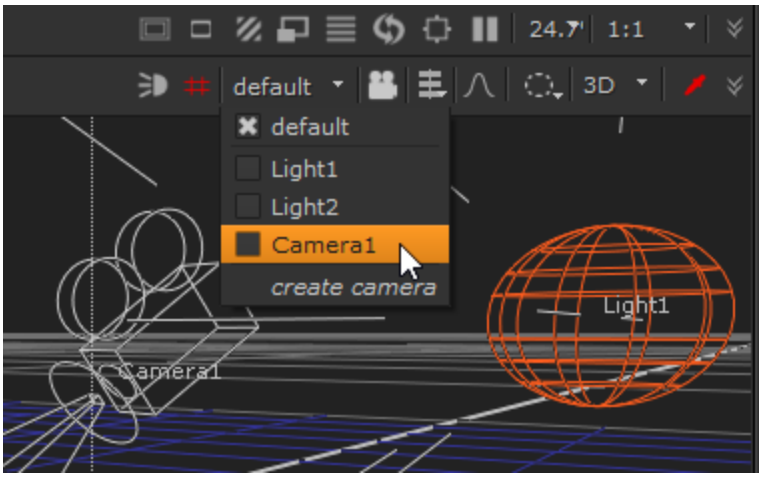
3D Viewer properties.

2. Make the desired changes to the 3D **bg** and **fg** colors.
3. From the **3D control type** dropdown menu, select the navigation control scheme you want to use (Nuke, Maya, Houdini, Lightwave, or Modo).
4. Click **OK**.

Note: The 3D control type also affects the mouse button assignments for panning and zooming in the node graph and 2D Viewers.

To Look Through a Camera

1. Press **V** to make sure you are looking through the 3D perspective view, and not one of the orthographic views.
2. From the 3D Viewer window, select the camera from the dropdown menu in the top right corner.



Note: This selection does not change the camera used for rendering. This changes only the camera to "look through" for the current 3D Viewer.

Cameras in the current data stream automatically appear in the dropdown menu of cameras you can select. To select a camera that doesn't appear in the menu, double-click the camera node to open its panel, and it is added to the menu.

To Lock the 3D Camera View

You can choose to lock the 3D view to the selected camera or light. You can toggle between the unlocked and locked modes by clicking the **3D view lock** button, or by pressing **Ctrl/Cmd**+**L**.

- **unlocked**: to freely move in the 3D view without restrictions. The 3D view lock button is gray.
- **locked**: to lock your movement to the camera or light you've selected in the dropdown menu on the right side of the 3D view lock button. The 3D view lock button is red.

To Use the Interactive 3D Camera View Mode

With the interactive 3D camera view mode you can change the camera or light values according to your movement in the Viewer. You can activate the interactive mode by **Ctrl/Cmd**+clicking the **3D view lock** button. When the interactive mode is on, the **3D view lock** button turns green. In order to activate the interactive mode, you need to have a Camera or a Light node selected in the dropdown on the right side of the **3D view lock** button.

When the interactive mode is on, you can use the plus (**+**) and the minus (**-**) keys to change the translate values of the camera or light you've selected. When the interactive mode is off, these keys zoom your view in and out.

3D Scene Geometry

There are several options for inserting 3D geometry into your Classic 3D scenes. You can:

- create primitive shapes: cards, cubes, cylinders, and spheres. See Using Built-in Primitive Geometry.
- import models and point clouds created in other 3D applications and exported into OBJ (Wavefront), FBX, or Alembic files. See Importing Objects from Other Applications.
- create your own geometry and point clouds from scratch. See Creating Point Clouds and Geometry from Scratch

Once you've inserted a 3D geometry object into your scene, you can:

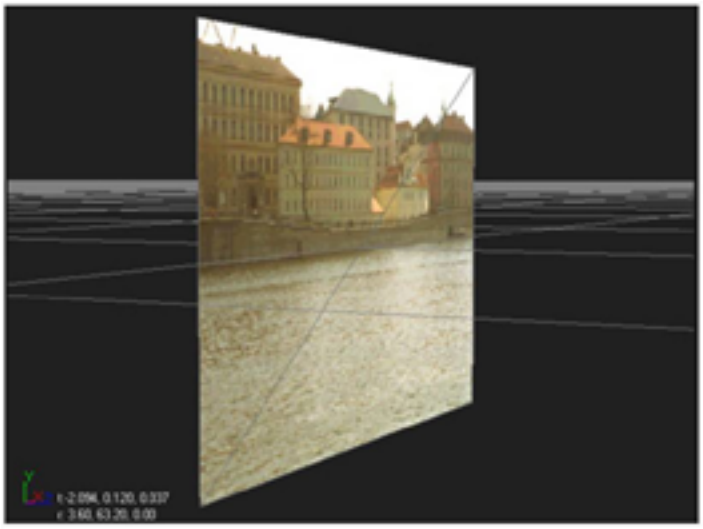
- adjust how the object is displayed in the Viewer. See Object Display Properties.
- select entire objects, as well as vertices or faces on objects. See 3D Selection Tools.
- merge objects together. See Merging Objects.
- modify object shapes. See Modifying Object Shapes.

Using Built-in Primitive Geometry

This section describes the primitive shapes built in with Nuke's Classic 3D System: cards, cubes, cylinders, and spheres. You can use them as the building blocks for other, more complex shapes, or place them in the background of your 3D scene where extra detail isn't visible. If you modify them with transforms, they can also be useful as foreground objects.

Working with Cards

A *card* is the simplest type of object you can add to a scene (and probably the type you will use most often). It’s merely a plane onto which you can map a texture - typically a clip you are using as part of a pan-and-tile setup.



A card object.

A card object may be deformed as a bilinear or bicubic object with controls contained in the card's parameters. You can also insert other 3D nodes, such as ProceduralNoise or RadialDistort, to change the card geometry.

Card nodes have extended bicubics (bicubics with more control points). They allow you to subdivide a card, giving you finer control for warping an area. You can subdivide the card into an evenly spaced grid or pick a location to add a row, column, or both.

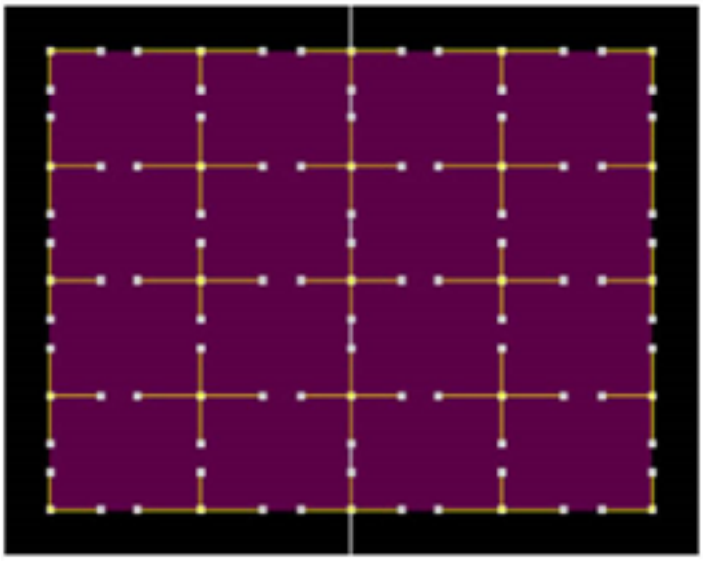
To Add a Card Object:

1. Click **3D** > **Geometry** > **Card** to insert a Card node.
2. Drag the Card node's **img** pipe to the Read node that has the image you want to apply to the card.
3. Connect the Card node to the appropriate Scene node to add it to the 3D scene.
4. Use the card object's transform controls to manipulate the position, scale, and rotation of the card in 3D space. For more information, see Transforming from the Node Properties Panel.

Deforming Card Objects

The Deform tab in the Card **Properties** panel lets you convert the card into a mesh surface that may be pulled and reshaped.

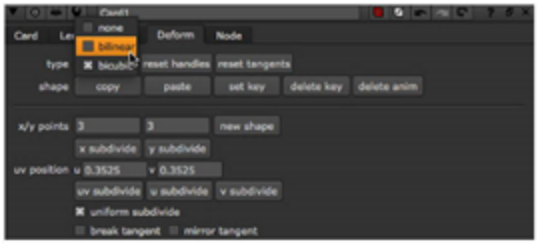
A *bicubic deformation* offers the greatest degree of surface elasticity. You can add any number of control points on the card and translate these points and their tangents in any direction. The control point tangents exert a magnetic-like influence over the objects surface.



The Card node can have any number of control points you can translate.

To deform a Card object:

1. Double-click the Card node to open its controls.
2. Go to the **Deform** tab, and select the mesh type for the deformation: **bilinear** or **bicubic**.



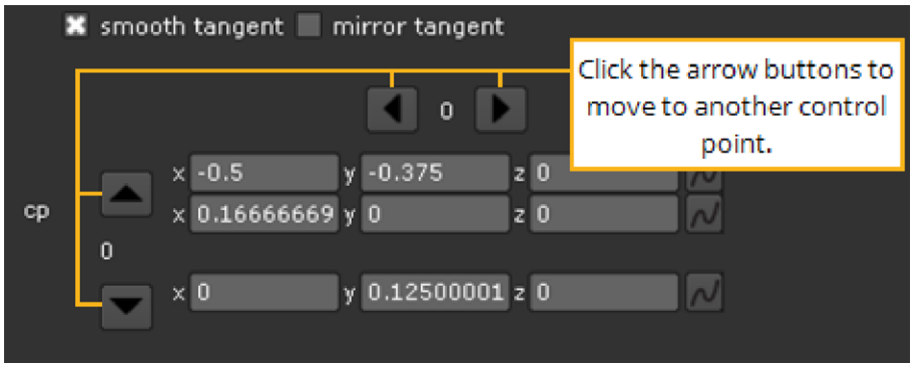
3. By default, the card has three control points on the x axis, and three on the y axis. To add more control points, do any of the following:
   - Enter new values in the **x/y points** fields and click the **new shape** button. For example, to create a shape with 4 points on the x axis and 6 on the y axis, change the **x points** value to 4 and the **y points** value to 6, and click **new shape**.
   - To evenly subdivide the current shape in the x or y directions, click the **x subdivide** or **y subdivide** buttons. This adds one control point between every existing control point in the selected direction. The **x/y points** fields are also updated to reflect the current number of control points.
   - To add one row or column of control points, adjust the **u** or **v** slider. The **u** slider specifies the position of new columns, and the **v** slider the position of rows. In the Viewer, a yellow cross marker indicates the position of the new row or column. You can also move the cross marker by dragging it to a new position in the Viewer. The **u** and **v** sliders' values are updated as you move the marker. When you are happy with the position, click the **uv subdivide** button. A row or column is added in the position you

specified. Clicking the button again has no effect, because there is already a subdivision at the specified position.

4. If you selected **bicubic** under **type**, you can adjust the way control point tangents behave when you are making your changes to the card. Do any of the following:
   - To have the original tangents adjusted to create a more uniform subdivision when you are using **x subdivide**, **y subdivide**, or **uv subdivide**, check **uniform subdivision**. If you do not check this, Nuke maintains the original tangents.
   - You can move the tangents in the Viewer by clicking and dragging. If you want to move a tangent together with the opposite tangent so that the two tangents form a continuous line, check **smooth tangent**. To break the tangent from the opposite tangent and move the tangent alone, uncheck **smooth tangent**.
   - To change the length of the opposite tangent to always match the length of the tangent you are moving, check **mirror tangent**. If you do not check this, the opposite tangent length is not changed.
5. Drag the points displayed in the mesh to deform the card.

To translate the control points and tangents:

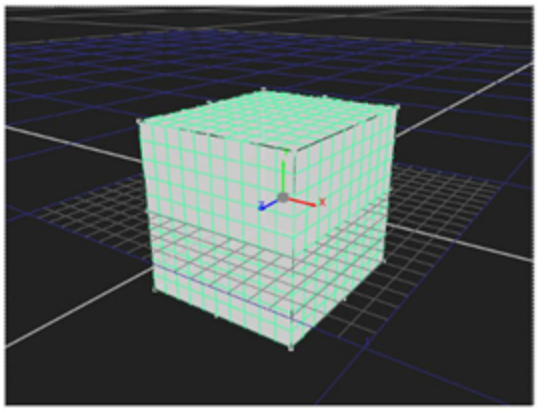
1. If necessary, double-click on the Card node to display its controls, and go to the **Deform** tab.
2. Only the controls for the selected point are displayed in the bottom of the Card properties panel. To translate another point, you can select a new point in the Viewer or use the arrow buttons in the bottom of the Card controls to move to the next control point.



3. To translate the control points and their tangents:
   - Increment or decrement the numbered **x**, **y**, and **z** fields. For each control point, the controls for translating the point itself are shown on top of the controls for translating the tangents.
   - Or drag on any control point or tangent to translate it relative to the current angle of view.

Working with Cubes

A *cube* is the familiar six-sided polyhedron. You can transform any of its sides (and, of course, texture it with clips).



A cube object.

To Add a Cube

1. Click **3D** > **Geometry** > **Cube** to insert a Cube node.
2. Drag the Cube node's **img** pipe to the Read node containing the clip you want to use as a texture.
3. Drag one of the Scene node's numbered pipes to the Cube node to place the cube in the scene.
4. Use the cube object's transform controls to manipulate the position, scale, and rotation of the cube in 3D space. For more information, see Transforming from the Node Properties Panel.
5. Translate any of the cube's sides to alter its shape.

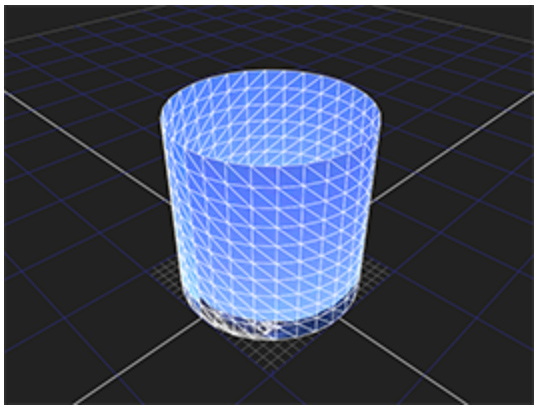
To Translate a Cube's Sides

1. If necessary, double click on the Cube node to display its parameters (and thereby select the object in the scene).
2. Increment or decrement the **cube** fields. (Assuming a positive z view of the object, **x** refers to the left side; **y**, the bottom side; **n**, the back side; **r**, the right side; **t**, the top side; and **f**, the front side.)

   Or drag on any side order to translate it relative to the current angle of view.

Working with Cylinders

A cylinder is an object with two identical flat ends that are circular or elliptical and one curved side. You can control its geometric resolution, or face count (and, of course, texture it with images).



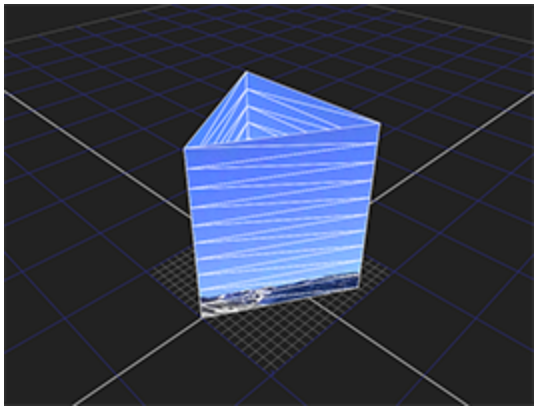
A cylinder object.

To Add a Cylinder

1. Click **3D** > **Geometry** > **Cylinder** to insert a Cylinder node.
2. Drag the Cylinder node's **img** pipe to Read node containing the clip you want to use as a texture.
3. Drag one of the Scene node's numbered pipes to the Cylinder node to place the cylinder in the scene.
4. Use the cylinder object's transform controls to manipulate the position, scale, and rotation of the cylinder in 3D space. For more information, see Transforming from the Node Properties Panel.

Adjusting Geometric Resolution

By default, a cylinder has 30 rows and 30 columns. You can, however, increase or decrease either number as appropriate. For example, the figure below shows a cylinder whose geometric resolution has been decreased to 10 rows and 3 columns.



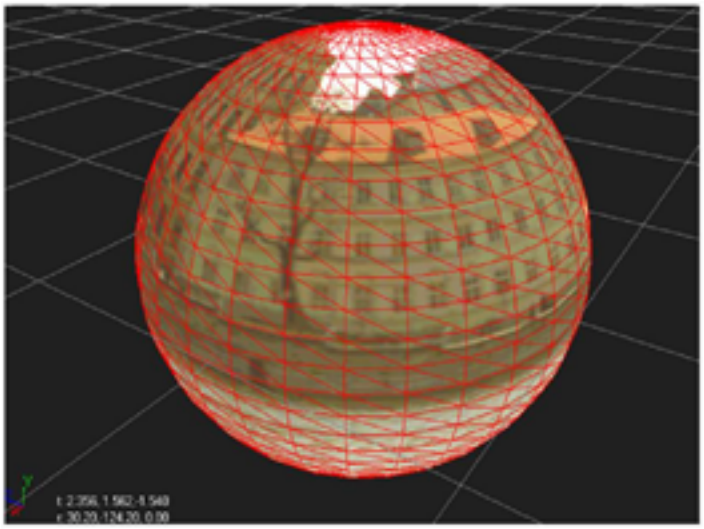
A low-resolution cylinder.

To adjust a Cylinder's geometric resolution:

1. If necessary, double click on the Cylinder node to display its parameters (and thereby select the object in the scene).
2. Increment or decrement **rows** field to adjust the number of latitudinal divisions on the cylinder.
3. Increment or decrement **columns** field to adjust the number of longitudinal divisions on the cylinder.

Working with Spheres

A *sphere* is the familiar globe-shaped polyhedron. You can control its geometric resolution, or face count (and, of course, texture it with images).



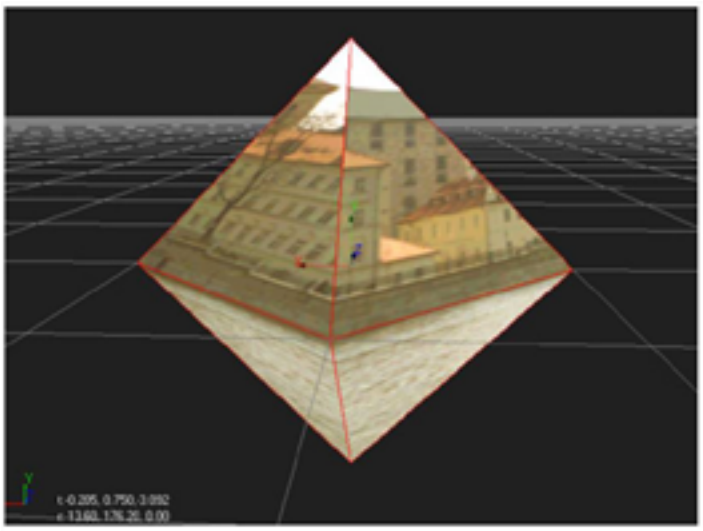
A sphere object.

To Add a Sphere

1. Click **3D** > **Geometry** > **Sphere** to insert a Sphere node.
2. Drag the Sphere node's **img** pipe to Read node containing the clip you want to use as a texture.
3. Drag one of the Scene node's numbered pipes to the Sphere node to place the sphere in the scene.
4. Use the sphere object's transform controls to manipulate the position, scale, and rotation of the sphere in 3D space. For more information, see Transforming from the Node Properties Panel.

Adjusting Geometric Resolution

By default, a sphere has 30 rows and 30 columns. You can, however, increase or decrease either number as appropriate. For example, the figure below shows a sphere whose geometric resolution has been decreased to 2 rows and 4 columns (making it, in effect, an octahedron).



An octahedron generated with
a low-resolution sphere.

To adjust a Sphere's geometric resolution:

1. If necessary, double click on the Sphere node to display its parameters (and thereby select the object in the scene).
2. Increment or decrement **rows** field to adjust the number of latitudinal divisions on the sphere.
3. Increment or decrement **columns** field to adjust the number of longitudinal divisions on the sphere.

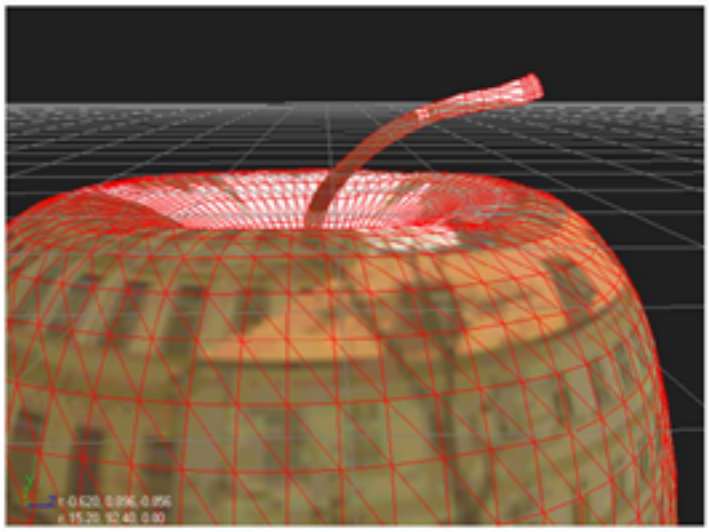
Importing Objects from Other Applications

Sometimes, you may need to import files or objects created in 3D applications, such as Maya or Boujou. Depending on what you want to import and from where, there are different ways of doing the import:

- To import geometry from OBJ (**.obj**) files, see Importing Geometry from OBJ Files.
- To import geometry or point clouds from FBX (**.fbx**) files, see Importing Geometry and Point Clouds from FBX Files. FBX is a standard file format many applications can export to, and **.fbx** files contain 3D scenes from which you can import cameras, lights, transforms, meshes, and point clouds into Nuke.
- To import geometry or point clouds from Alembic (**.abc**) files, see Importing ABC Meshes and Point Clouds. You can export to Alembic from most popular 3D applications.
- To import geometry from USD (**.usd**) files, see Importing USD Items.

Importing Geometry from OBJ Files

You can import into a Nuke scene 3D objects from other software programs that have been saved out in the **.obj** (Wavefront) format. You cannot manipulate **.obj** objects at the vertex level from inside Nuke, but you can texture and transform them.



An imported OBJ object.

To import an OBJ object

1. Click **3D** > **Geometry** > **ReadGeo** to insert a ReadGeo node.
2. In the ReadGeo parameters, click the **file** field's folder icon. The file navigation dialog appears.
3. Navigate to the OBJ file, then click **Open**. Nuke reads in the OBJ file.
4. Drag the ReadGeo node's **img** pipe to the Read node containing the clip you want to use as a texture.
5. Drag one of the Scene node's numbered pipes to the ReadGeo node to place the OBJ object in the scene.

Importing Geometry and Point Clouds from FBX Files

FBX is a standard 3D file format that gives you access to 3D scenes created in other applications supporting the same format. What you generally have in an **.fbx** file is an entire 3D scene containing cameras, lights, meshes, non-uniform rational B-spline (NURBS) curves, transformation, materials, and so on. From this scene, you can extract cameras, lights, transforms, and meshes into Nuke. This way, you can, for example, create a mesh in Maya, export it in a **.fbx** file, and use the same mesh again in Nuke.

Note: For the FBX SDK version used in Nuke, see Third-Party Library Versions.

Tip: If you have trouble with **.fbx** files, it may be because they were written with an older version of FBX. If they load very slowly, it is also possible that they are ASCII rather than binary. To get around these problems, you can use the FBX converter on the Autodesk website (http://usa.autodesk.com/fbx/download/). It converts between various different formats, including older FBX versions, ASCII, and binary, and is available on Windows, Mac, and Linux.

Importing Meshes from FBX Files

The ReadGeo node lets you import meshes (or NURBS curves/patch surfaces converted to meshes) from FBX files. Using one ReadGeo node, you can read in a single mesh or all the meshes in a **.fbx** file.

The mesh's vertices, normals, UVs, and vertex colors are read on a per frame basis or at frame 0. If there are any shape or cluster deformers, they are applied to the vertices. Materials or textures are not read in.

To import a mesh from an .fbx file:

1. Select **3D** > **Geometry** > **ReadGeo** to insert a ReadGeo node into your script.
2. In the ReadGeo controls, click the folder icon next to the **file** field and navigate to the **.fbx** file that contains the mesh you want to import. Click **Open**.
3. From the **animation stack** dropdown menu, select the take you want to use. FBX files support multiple takes. Usually, one of them is a default take that contains no animation.
4. From the **node name** dropdown menu, select the mesh you want to import from the **.fbx** file.
5. To adjust the frame rate used to sample the animation curves, enter a new value (frames per second) in the **frame rate** field. The frame rate you enter is only used if you check **use frame rate**. Otherwise, the frame rate from the **.fbx** file is used.
6. If you want to import all the meshes in the **.fbx** file rather than just one, check all objects. This overrides whatever you have selected under node name. If the objects are animated, check **read on each frame**. This bakes each object's transform into the mesh points and preserves the animation.
7. If you want to modify the transform properties imported from the **.fbx** file, uncheck **read transform from file** and make the necessary modifications. As long as **read transform from file** is unchecked, your changes are kept.
8. To reload the transform properties from the **.fbx** file, click the **Reload** button.

Importing Point Clouds from FBX Files

The ReadGeo node also lets you import point clouds from FBX files.

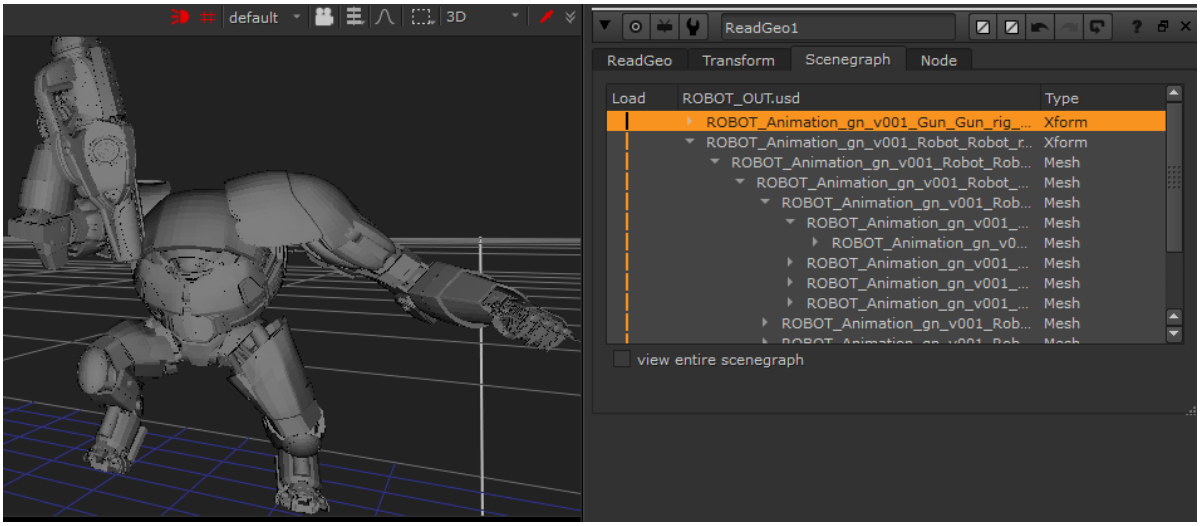
To import a point cloud from an .fbx file:

1. Select **3D** > **Geometry** > **ReadGeo** to insert a ReadGeo node into your script.
2. In the ReadGeo controls, click the folder icon next to the **file** field and navigate to the **.fbx** file that contains the point cloud you want to import. Click **Open**.
3. From the **animation stack** dropdown menu, select the take you want to use. FBX files support multiple takes. Usually, one of them is a default take that contains no animation.
4. In the **objecttype** dropdown, select **PointCloud**.
5. To adjust the frame rate used to sample the animation curves, enter a new value (frames per second) in the **frame rate** field. The frame rate you enter is only used if you check **use frame rate**. Otherwise, the frame rate from the **.fbx** file is used.
6. If you want to modify the transform properties imported from the **.fbx** file, uncheck **read transform from file** and make the necessary modifications. As long as **read transform from file** is unchecked, your changes are kept.
7. To reload the transform properties from the **.fbx** file, click the **Reload** button.

USD in Nuke's Classic 3D System

Universal Scene Description (USD) is an open source file type from Pixar that supports any number of assets organized into virtual sets, scenes, and shots. You can share **.usd** files freely between applications that support the format and edit them non-destructively.

Note: Nuke currently supports **.usd** and **.usda** files and can display color, normals, and UV information in the Viewer.



Note: For more information on the USD format, see https://graphics.pixar.com/usd/docs/index.html

You can import cameras, lights, meshes and point clouds from USD files (**.usd** file format) into a Nuke Classic 3D System scene. Data imported from **.usd** files are brought in using the relevant native 3D nodes in Nuke. For example, reading in lights from a **.usd** file are imported as Light nodes, and cameras are imported as Camera nodes. Meshes and point clouds are imported to ReadGeo node types.

This means if you've started using USD elsewhere in your pipeline, you can continue to use it to carry the data you need directly into Nuke, without converting to a different format.

Note: To learn more about USD in the **New 3D System**, see 3D Compositing with the New 3D System.

Choosing the 3D Viewer Type

Nuke 13.0 onwards includes a new Hydra 3D viewport, which uses hdStorm as a new renderer. The Hydra Viewer is more consistent with other applications in your pipeline, such as Katana, Solaris, or USDView, and the represents the output of the ScanlineRender node.

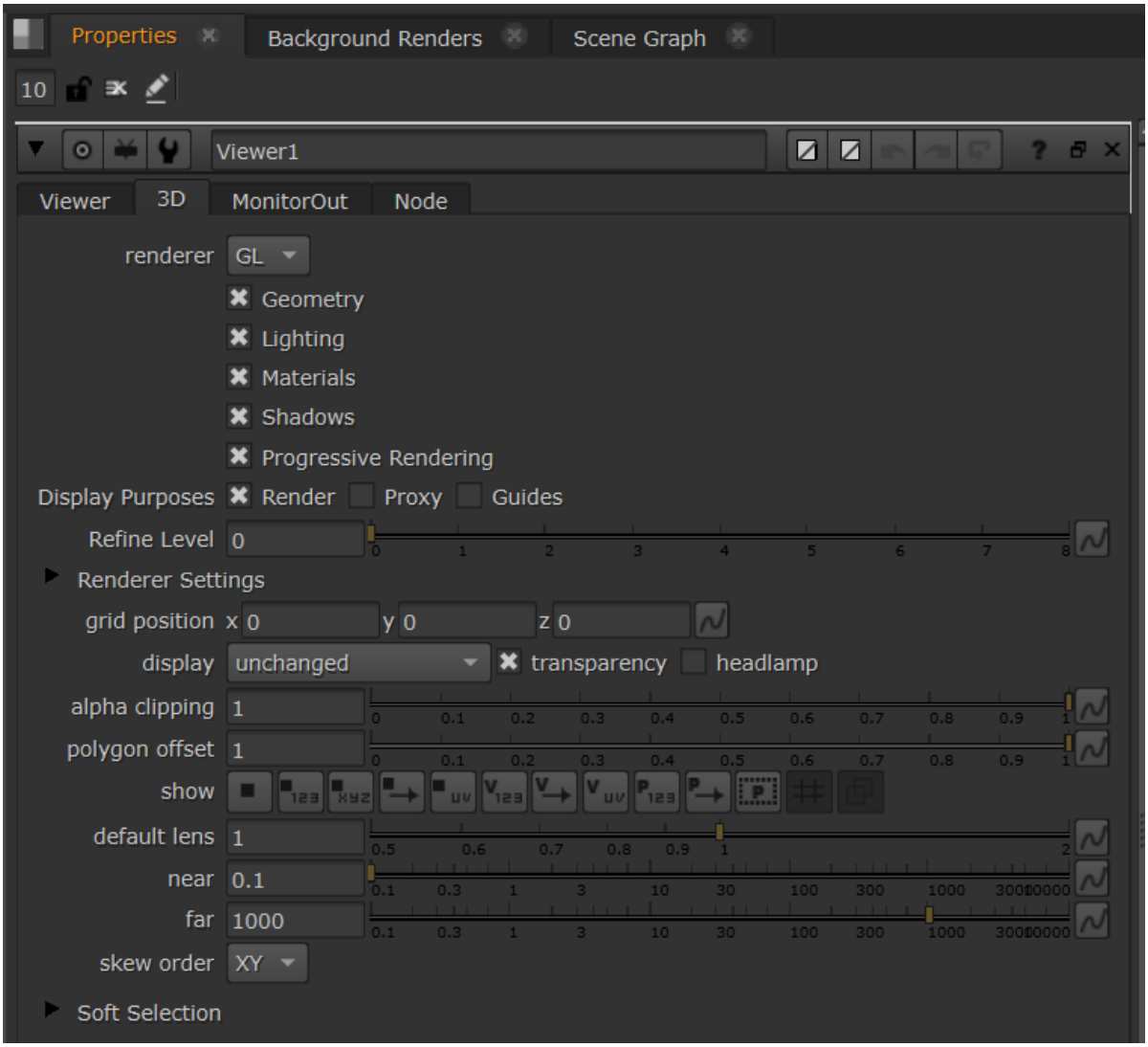


Nuke's **New 3D System** uses a Hydra renderer, and nodes in the **Classic 3D System** use a "Nuke" renderer.

As such, the Viewer's **renderer** knob only has one option: **GL**, which is both the Hydra and Nuke renderers, depending on if you are viewing New or Classic 3D nodes.

Note: Learn more about the New 3D System and USD at 3D Compositing with the New 3D System.

You can also select whether or not to display geometry, lights, materials and/or shadows in the 3D Viewer by toggling the relevant checkboxes.



The Viewer renderer option is also included in the **Preferences** under **Panels** > **Viewer** > **default renderer**. Changing the preference only affects new Viewer nodes.

Importing USD Items

Learn how to import mesh, point cloud, light, camera, and axis data into Nuke from a USD file.

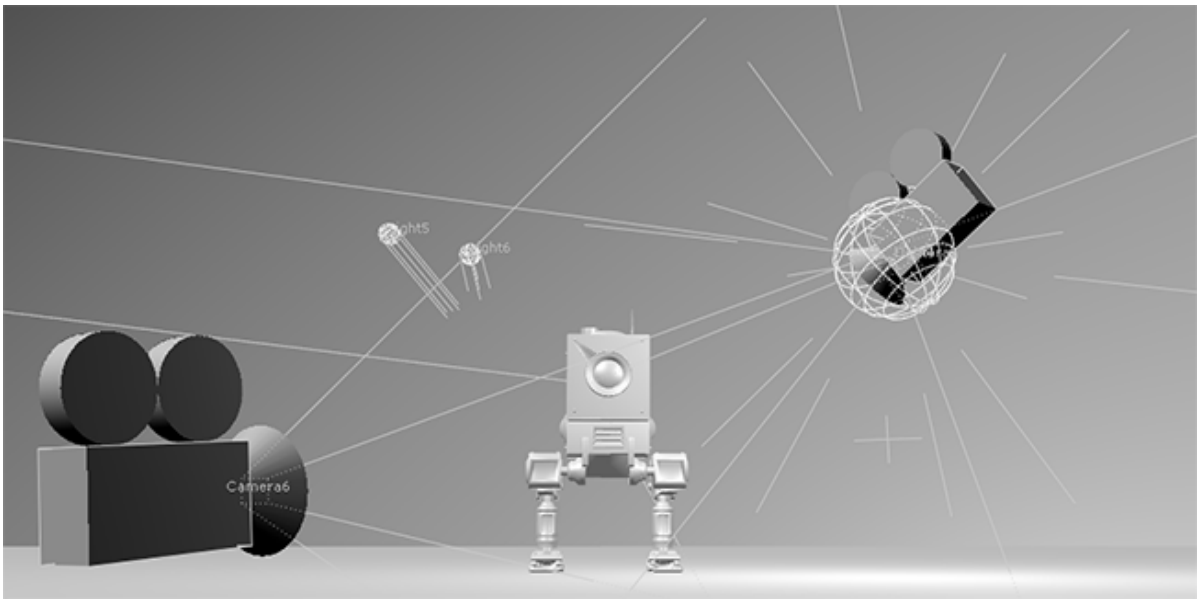
Manipulating USD Items

Learn how to use the scenegraph to show and hide loaded USD items, as well as transform, and add materials to USD items.

Importing USD Items

You can import meshes, point clouds, lights, cameras and axes from USD files (**.usd** file format) into a Nuke Classic 3D scene. During the import, you can control which items in the USD scene are loaded using the import dialog.

Note: To learn more about importing USD items using the **New 3D System**, see Importing with GeoImport.



USD geometry, lights and cameras loaded into Nuke

Meshes and Point Clouds

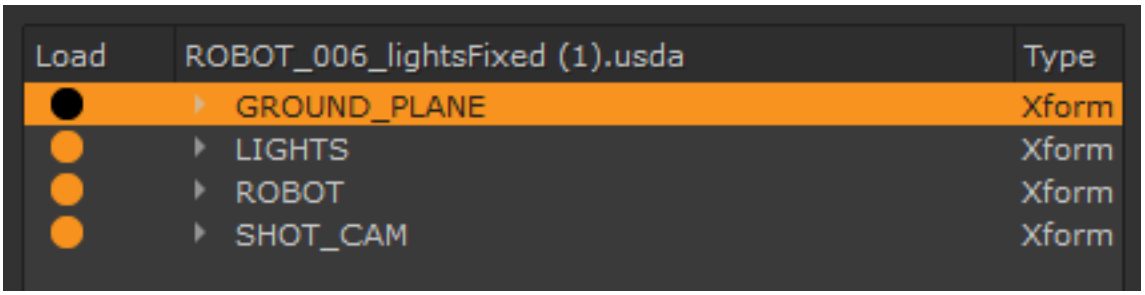
You can load all mesh and point cloud items into a single ReadGeo node or specify certain prims as parents, which load in separate ReadGeo nodes along with all their children. If there is only one prim in the USD file, it loads automatically.

Lights, Cameras and Axes

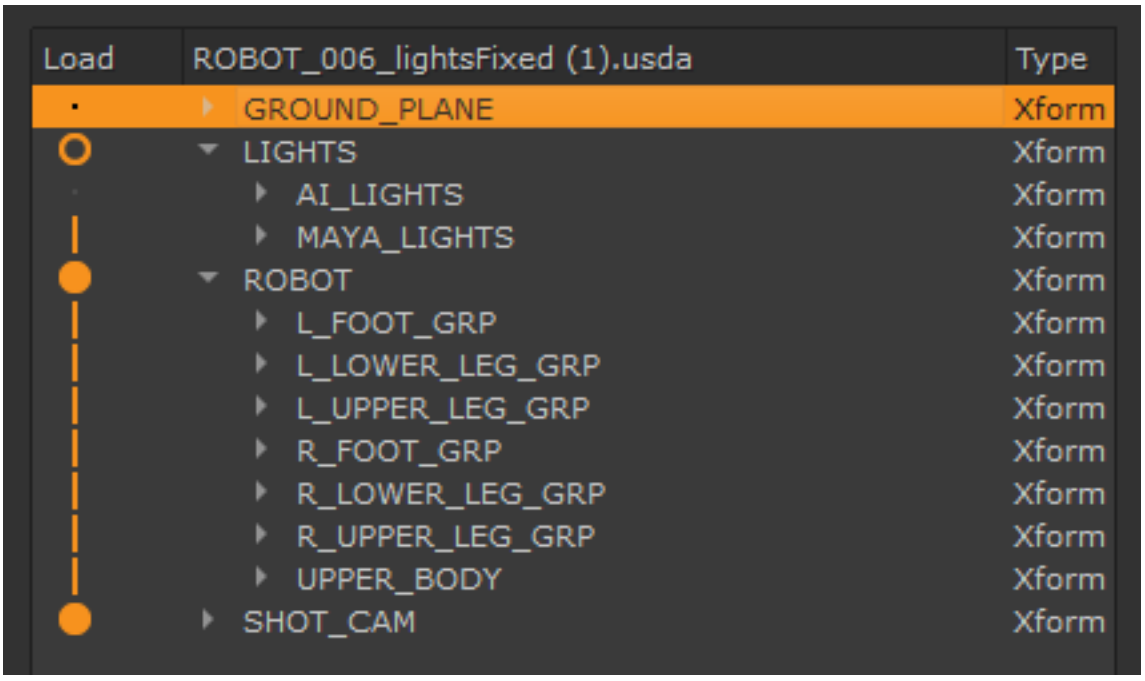
Light, camera, and axis items are loaded using the relevant native Nuke 3D nodes, Light, Camera and Axis. If there is only one item in the USD file, it loads automatically.

Importing Items from USD Files

1. Click **Image** > **Read** or press **R** on the Node Graph.
   The **Read File(s)** dialog displays.
2. Select the **.usd** file you want to import from the file location, then click **Open**.
   - Files containing only one item are loaded automatically into the relevant node in the Node Graph.
   - Files containing more than one item display a scenegraph load dialog that allows you to create separate nodes from parents in the scenegraph. By default, all items are selected in the scenegraph.



3. Click in the **Load** column to load or ignore the items you want to import.



Note: Unsupported items are imported along with supported items, but are grayed-out in the scenegraph.

	Parent full load	This prim and all its children are included in the relevant node.
	Parent partial load	This prim and selected children are included in the relevant node.
If you click **Create parents as separate nodes**, full load and partial load parents and their children are loaded in separate nodes.		
	Item full load	This prim and all its children are included in the relevant node.
	Item partial load	This prim and selected children are included in the relevant node.
	No load	This prim and its children are not included in the relevant node.

4. When importing mesh or axis items, you can also right-click on an item and choose:
    - **select as parent** - select this prim and make it a parent. This allows you to create a separate node for this item and any child items.
    - **select as child** - select this prim and make it a child to the nearest parent item up the tree.
    - **deselect** - deselect this prim so that it is not imported.
5. To import your selections, do one of the following:
    - Click **Create all-in-one node** to create one node for everything that's selected, regardless of whether the prims are selected as parent or child.
    - Click **Create parents as separate nodes** to create one node for each parent prim marked with an orange circle in the scenegraph. Each node also contains all child items.

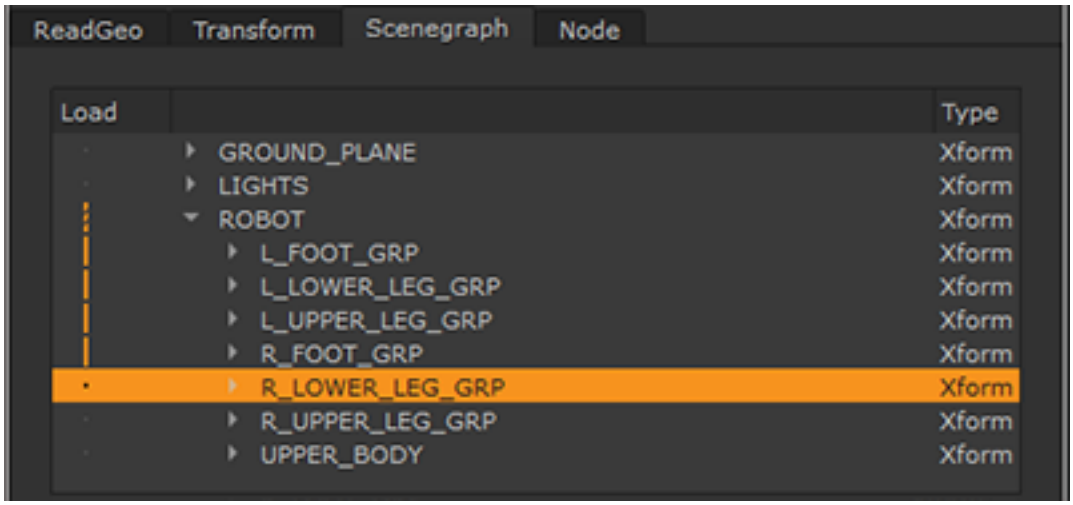
      If you want to transform parts of the **.usd** file independently, you need to set the objects as parents so that they load in their own node. See Transforming Mesh Items for more information.

    Nuke creates nodes for each parent you chose to import from the scene.

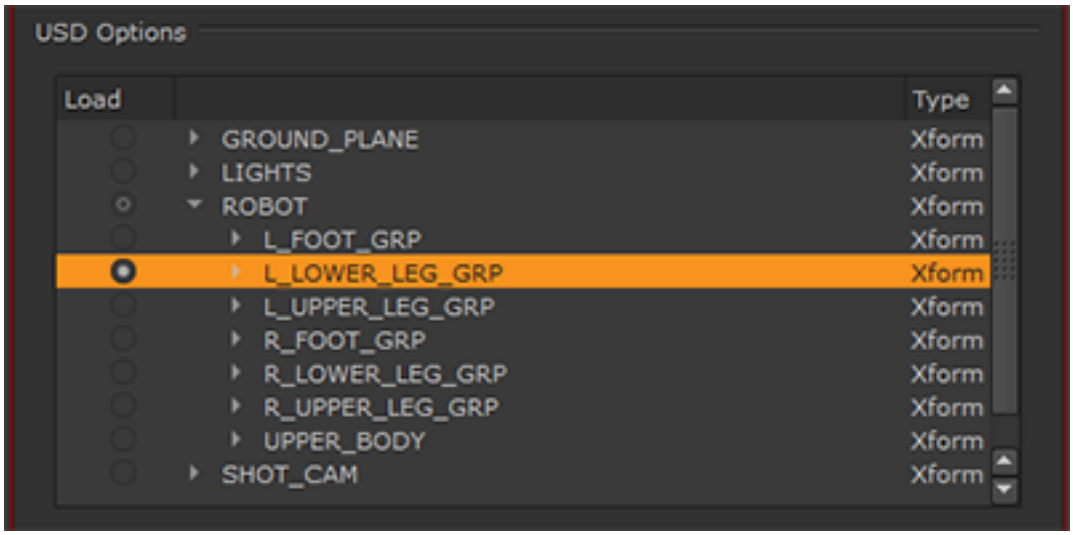
Manipulating USD Items

After importing items from a USD file into a Classic 3D scene, you can access the scenegraph through the node **Properties** and control which items are loaded.

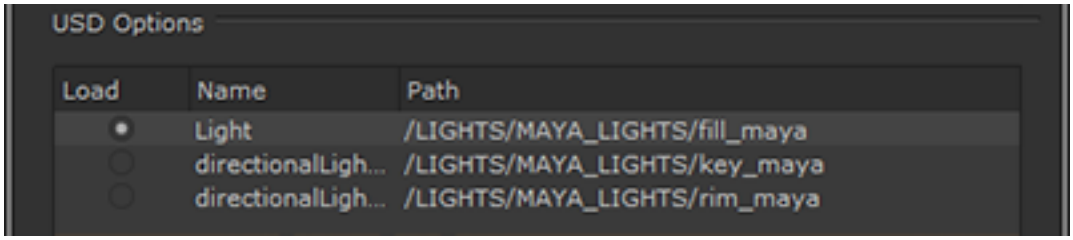
The ReadGeo node has a **Scenegraph** tab, the same as the scenegraph load dialog displayed when loading a **.usd** file containing multiple items.



The Axis node has a scenegraph view in the **File** tab under **USD Options** where you can make individual selections. This is because just one axis item can be loaded at a time.



Camera and Light nodes contain a list view rather than a scenegraph in the **File** tab under **USD Options**.



Note: The USD Options are only available when the **file** path is pointing to a valid USD file.

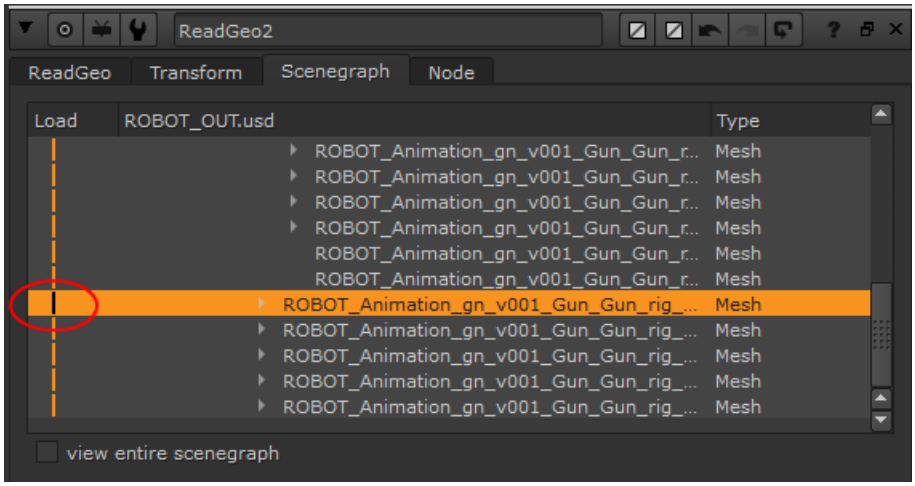
You can also manipulate USD items by transforming all, or parts of the scenegraph, or by applying materials to selected scenegraph locations.

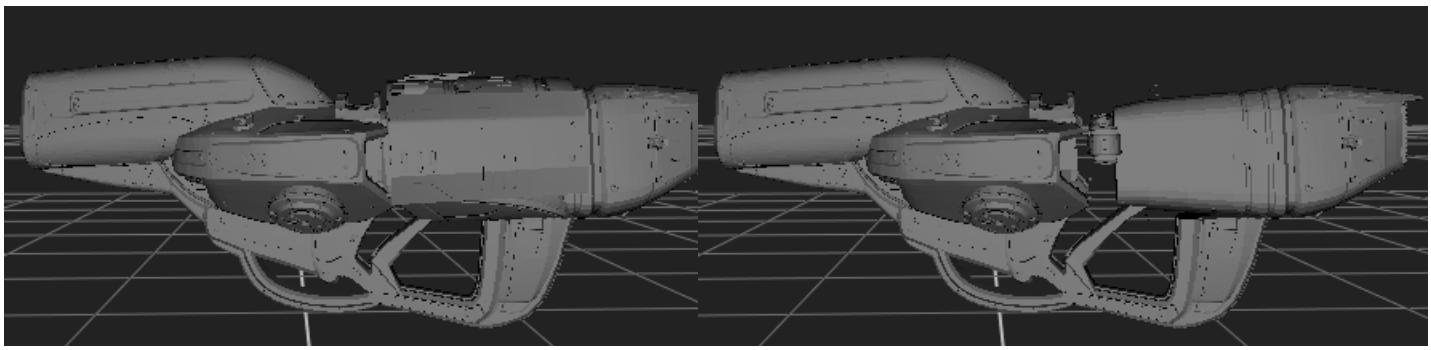
Note: To learn more about the updated Scene Graph in the **New 3D System**, see New 3D Scenes in the Scene Graph.

Showing and Hiding Loaded Meshes and Point Clouds

Loaded meshes and point clouds in Nuke use similar selection controls to those in the import dialog. You can show and hide supported items by clicking in the **Scenegraph** > **Load** column.

For example, clicking the following scenegraph mesh item on and off in the **Load** column affects part of the casing on the geometry.

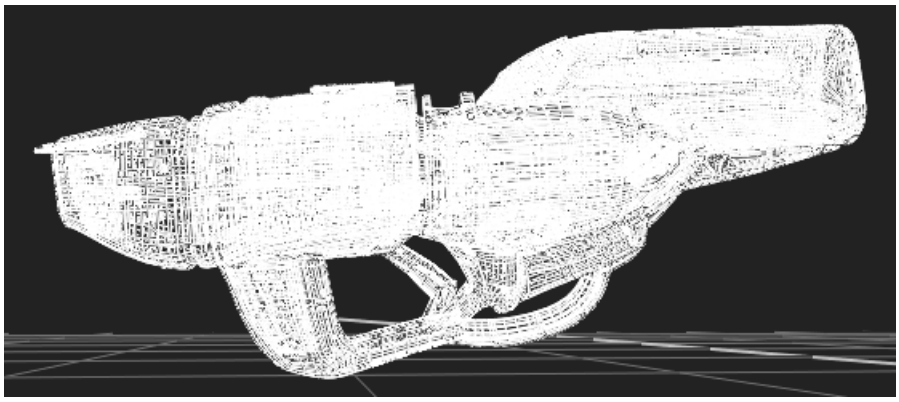




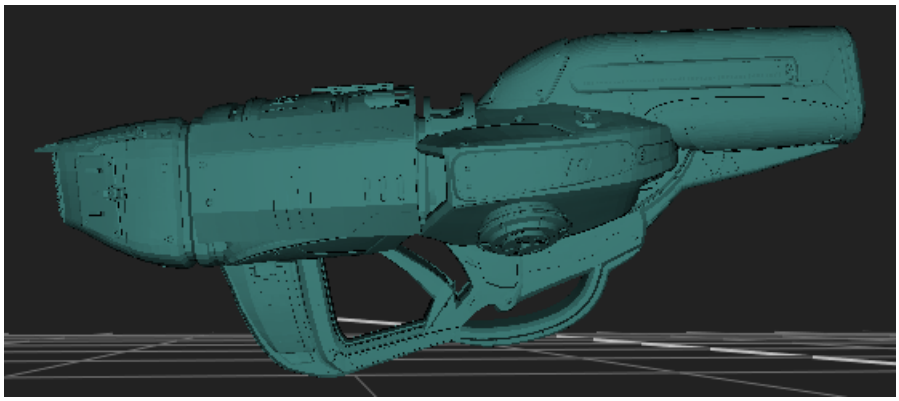
Note: If you want to import items that were not included in the original USD import, click **view entire scenegraph** and then use the **Load** column to import the required items.

You also have limited control over how items in the ReadGeo node appear in the 3D Viewer and in rendered output using the standard **display** and **render** dropdowns in the **Properties** panel:

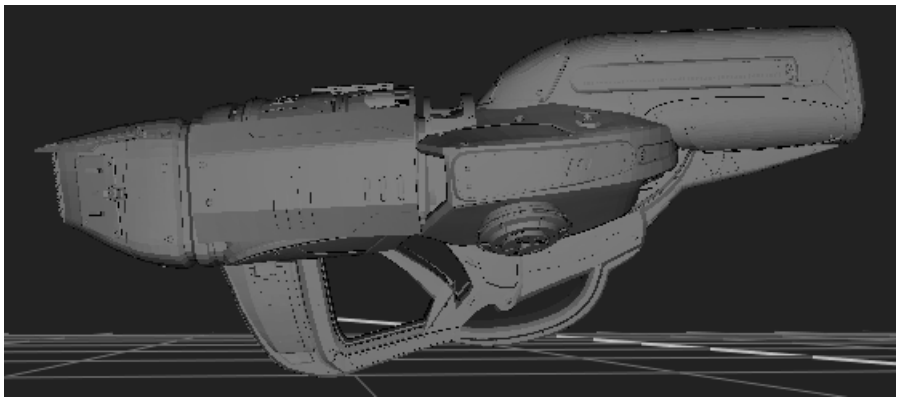
- **off** - hides the geometry.
- **wireframe** - displays/renders only the outlines of the item's geometry.



- **solid** - displays/renders all geometry with a solid color. Some **.usd** files include color information, which can help to pick out the different items in a scene.



- **solid+wireframe** - displays/renders the geometry as solid color with the item's geometry outlines.
- **textured** - displays/renders only the surface texture.



• **textured+wireframe** - displays/renders the wireframe plus the surface texture.

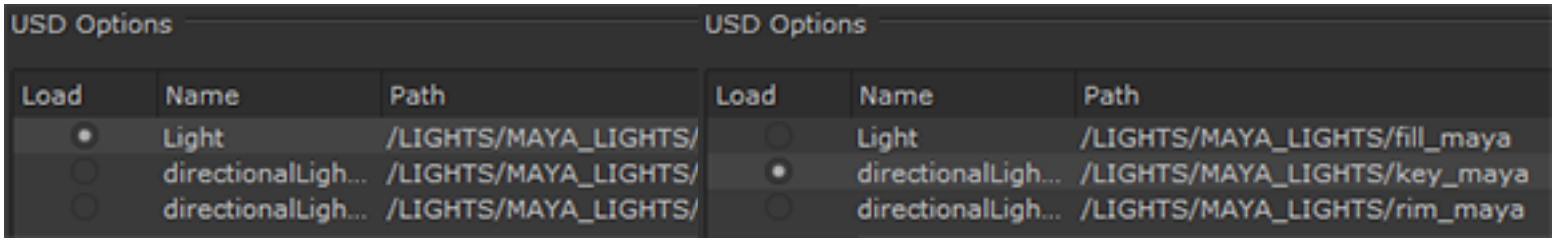
Note: The **display** dropdown does not affect the rendered output, only the item's appearance in the 3D Viewer. Similarly, the **render** dropdown does not affect the item's appearance in the 3D Viewer, only the rendered output.

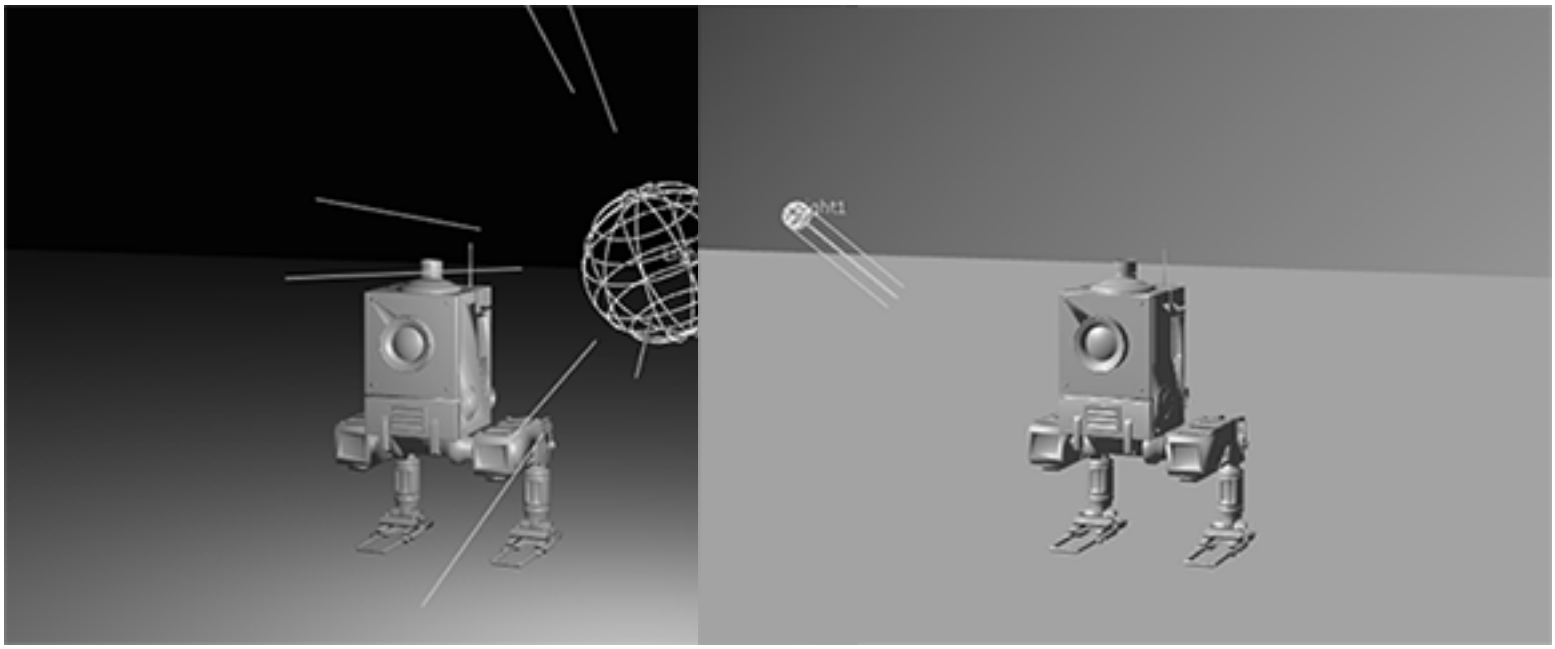
See Applying Materials to Mesh Items for more information on how to add materials to your items.

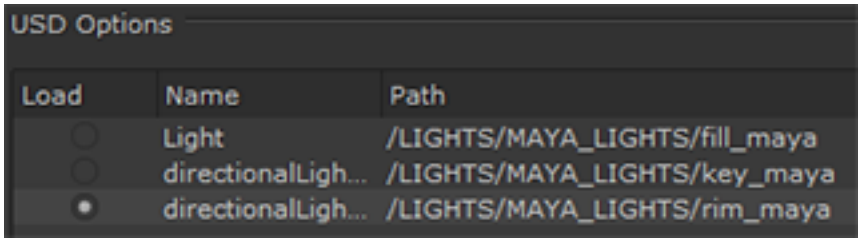
Selecting Loaded Lights, Cameras and Axes

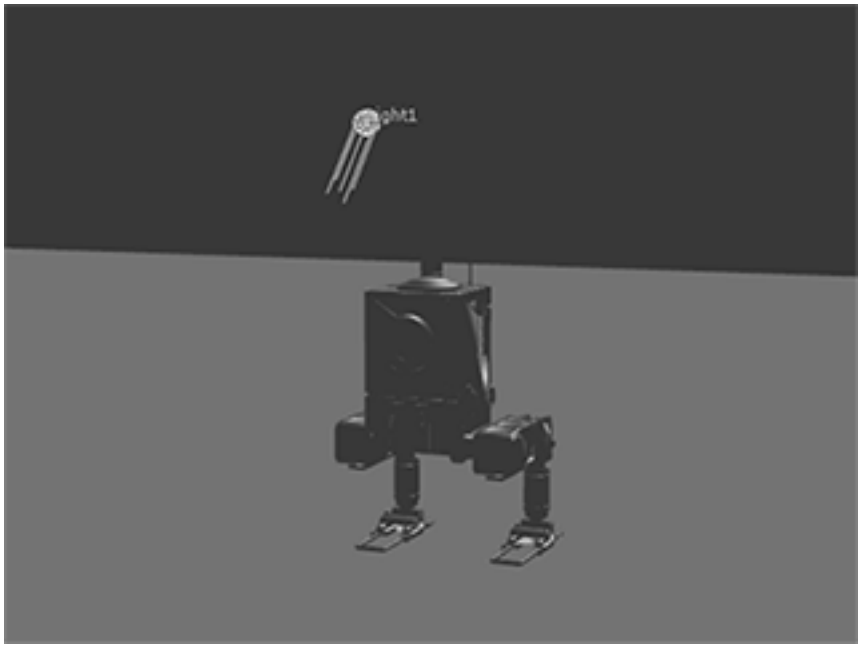
Lights, Cameras and Axes loaded from a USD file contain **USD options** in the relevant node **Properties** under the **File** tab. The USD Options allows you to click in the **Load** column to easily select which light, camera or axis you want to load.

For example, if your USD file has three lights, you can use the USD Options to switch between them. If you originally loaded just one of the three lights you are still able to select the others and switch between all three lights.



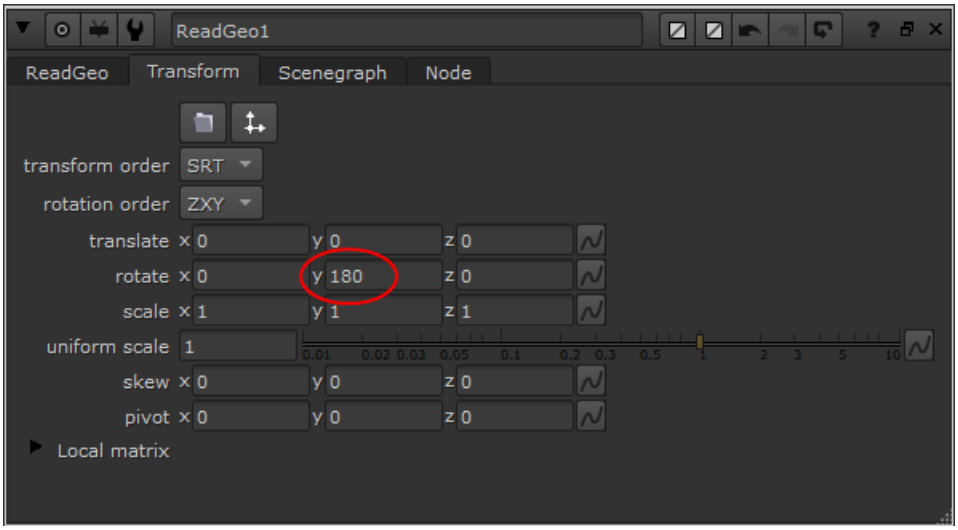


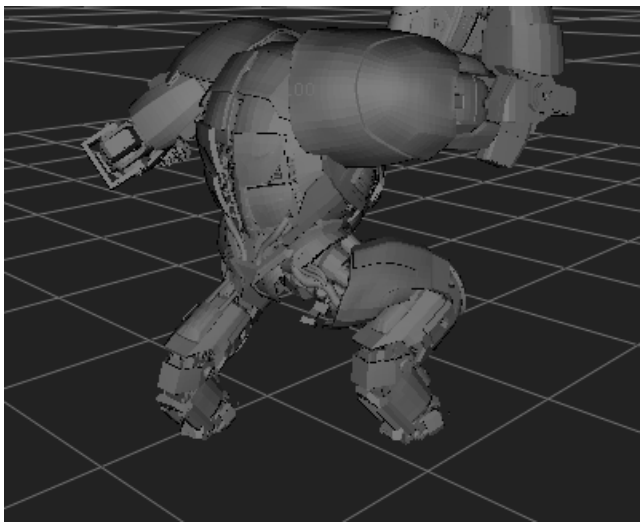

USD Options
Load
Name
Path
Light
/LIGHTS/MAYA_LIGHTS/fill_maya
directionalLigh...
/LIGHTS/MAYA_LIGHTS/key_maya
directionalLigh...
/LIGHTS/MAYA_LIGHTS/rim_maya

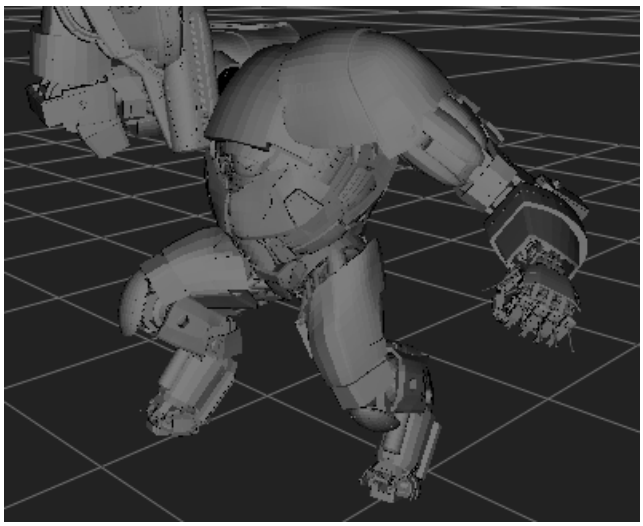


Transforming Mesh Items

The controls in the **ReadGeo** > **Transform** tab affect all the items in the scenegraph. You can use these controls to correct global transforms imported along with the USD items. For example, importing a **.usd** from Maya might place the scene facing away from Nuke's default camera.



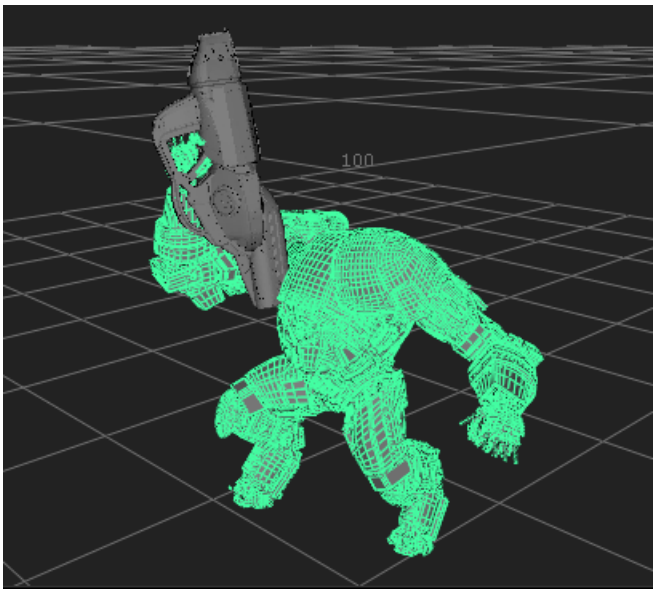


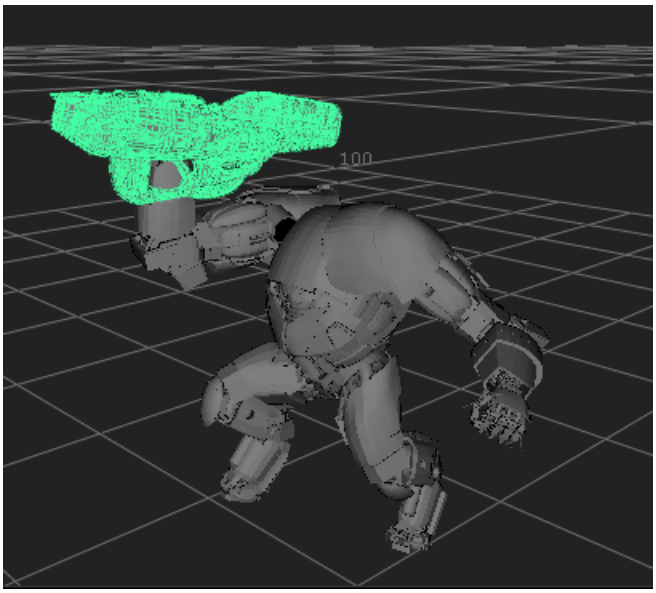


To transform parts of the scene individually, mark the items in the scenegraph as parents during import and then click **Create parents as separate nodes** to create one ReadGeo node for each parent item. See Manipulating USD Items for more information.

You can then transform each item separately using the ReadGeo **Transform** controls or a TransformGeo node.

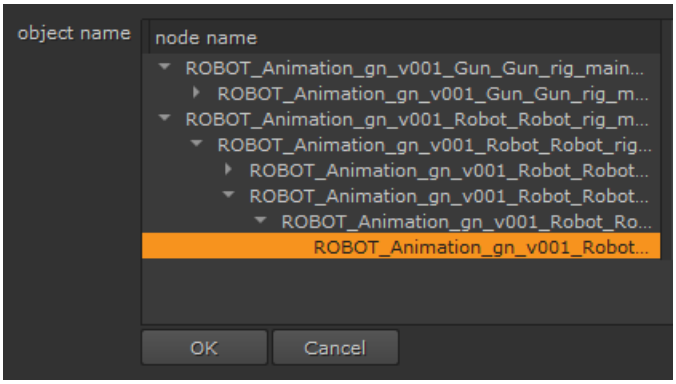
See Transforming Geometry, Cameras, and Lights for more information.



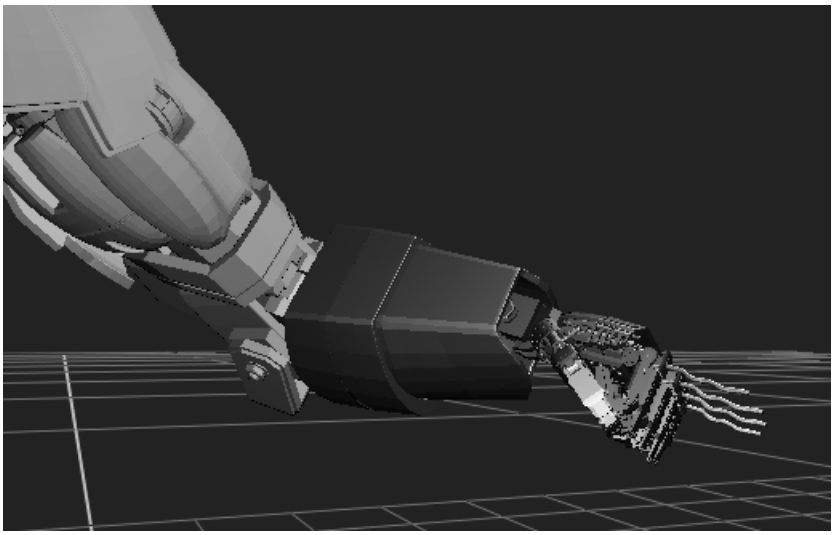


Applying Materials to Mesh Items

You can apply a material to a single scenegraph location using a downstream ApplyMaterial node. Enter a named location in the filter name control or click **choose** to navigate to the required scenegraph location.



Click **OK** to apply the material to the selected scenegraph location.

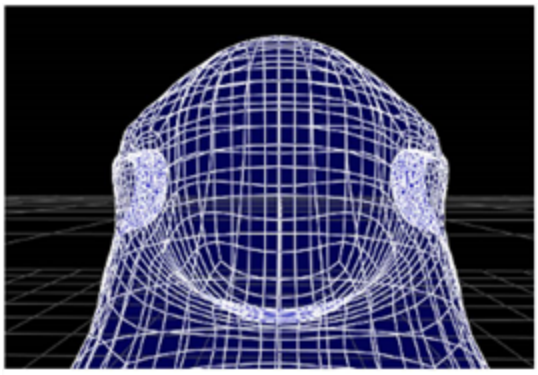


See Materials and Textures for more information.

Importing ABC Meshes and Point Clouds

You can import meshes (or NURBS curves/patch surfaces converted to meshes) and point clouds from Alembic files (**.abc** file format) into a Nuke Classic 3D scene. During the import, Nuke allows you to control which nodes in the Alembic scene get loaded by using an import dialog. If there is only one item in the Alembic file, it loads automatically. Below is an example of an imported octopus Alembic file, courtesy of Sony Pictures Imageworks.

In addition to meshes and point clouds, you can also import cameras and transforms from Alembic files. See Importing Cameras from Alembic Files and Importing Transforms from Alembic Files for more information.



For more information on Alembic, see http://code.google.com/p/alembic/

Note: To learn how to export files in the Alembic (**.abc**) format, refer to Exporting Geometry, Cameras, Lights, Axes, or Point Clouds for more information.

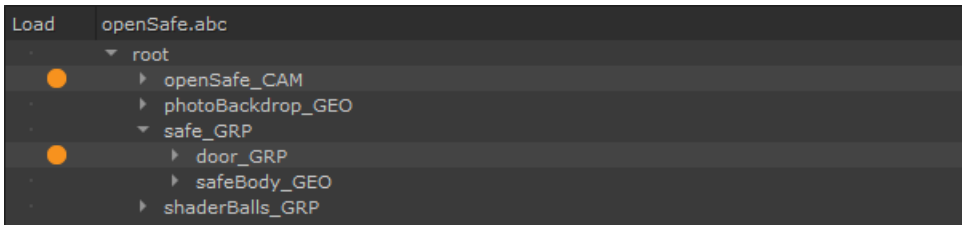
To Import Meshes and Point Clouds from an Alembic File

1. Click **Image** > **Read** or press **R** on the Node Graph.

   The **Read File(s)** dialog displays.
2. Select the **.abc** file you want to import from the file location, then click **Open**.
   - Files containing only one object are loaded automatically into a ReadGeo node in the Node Graph.
   - Files containing more than one object display a scenegraph load dialog that allows you to create separate ReadGeo nodes from parents in the scenegraph. By default, all items are selected in the scenegraph.



   Selected parent items are indicated by an orange circle, and selected child items by an orange bar (these turn gray when you select them in the list). Unselected items are indicated by a dot in the **Load** column.
3. To import specific items, deselect the root item by clicking on the orange circle. This de-selects the root and any child items.
4. Select the required items in the scenegraph by clicking on the dots against the items you want to import.



   Alternatively, you can right-click on an item and select:
   - **select as parent** - select this item and make it a parent. This allows you to create a separate ReadGeo node for this item and any child items.
   - **select as child** - select this item and make it a child to the nearest parent item up the tree.
   - **deselect** - deselect this item so that it is not imported.

   You can also select multiple items by pressing **Ctrl**/**Cmd** or **Shift** while clicking them.
5. Do one of the following:

- Click **Create all-in-one node** to create one ReadGeo node for everything that's selected, regardless of whether the items are selected as parent or child.
- Click **Create parents as separate nodes** to create one ReadGeo node for each parent item marked with an orange circle in the scenegraph. Each ReadGeo node also contains all child items.

Tip: If you always want **.abc** files to import all-in-one without displaying the import scenegraph, enable **Preferences** > **Behaviors** > **File Handling** > **always load abc files as all-in-one**.

Nuke creates ReadGeo, Camera, and Axis nodes as necessary, depending on what you chose to import from the scene.

6. In the ReadGeo node **Properties** panel, you can adjust the following:
   - If the objects you imported are animated, make sure **read on each frame** is checked. This preserves the animation. When **read on each frame** is disabled, use **lock frame** to select the frame the object is loaded at.
   - If you don't want to render motion blur, disable **sub frame** for faster UI interactions.
   - In the **frame rate** field, define a frame rate (frames per second) to sample the animation curves.
   - Use **render points as** to determine how point primitives are rendered: either as Nuke point clouds or Nuke particles.
   - Enable **use geometry colors** to apply geometry color attributes read from **.abc** files and apply them to the Nuke geometry.



Note: When disabled, this control can cause differences in rendered output when compared to previous versions of Nuke. If this occurs, enable **use geometry colors** in the ReadGeo **Properties** panel.

   - Any mesh items imported are listed on the **Scenegraph** tab of the ReadGeo node. To display all items that exist in the Alembic file in the list, check **view entire scenegraph**. This allows you to add items to the list or remove them from it by clicking on the indicators in the **Load** column.

Tip: To load specific items from an Alembic file, you can also create a ReadGeo, Camera, or Axis node, check **read from file**, and click the folder icon on the **File** tab to browse to an **.abc** file.

Creating Point Clouds and Geometry from Scratch

In addition to using the built-in primitives and imported geometry, Nuke also allows you to create dense point clouds and geometry from scratch.

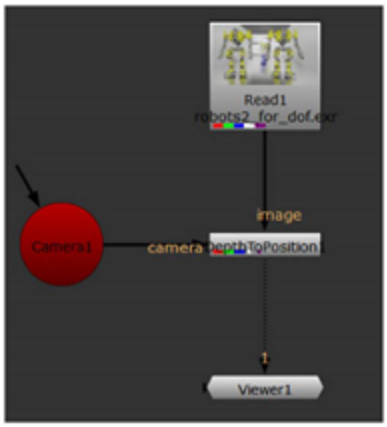
This section focuses on creating point clouds using the DepthToPosition, PositionToPoints, and DepthToPoints nodes available in Nuke. However, if you have a NukeX license, you can also use the CameraTracker and PointCloudGenerator nodes to create point clouds or the ModelBuilder node to build 3D models for 2D shots. See Creating Dense Point Clouds and Using ModelBuilder for more details.

Creating a Position Pass Using the DepthToPosition Node

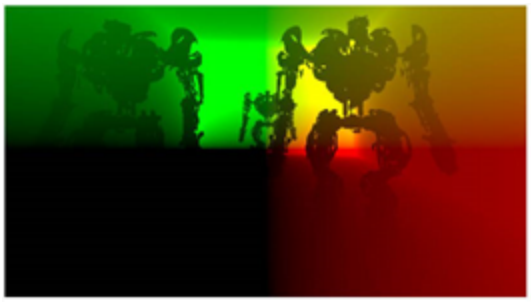
DepthToPosition takes depth data contained in an image file and the camera data to create a 2D position (xyz) pass. This pass is created by projecting the depth through the camera and recording the xyz positions of each projected point. The DepthToPosition node can be used together with the PositionToPoints node to create a point cloud, similar to the effect achieved with the DepthToPoints node.

To Create a Position Pass using DepthToPosition

1. From the **3D** > **Geometry** menu, select **DepthToPosition** to add the node to your script.
2. Read in an image with a depth pass and connect it to the **image** input of the node.
3. From the **3D** menu, select a **Camera** and connect it to the **camera** input of the DepthToPosition node.



4. In the **DepthToPosition** tab of the DepthToPosition node, select the depth channel from the **depth** dropdown menu. You should now see a position pass in the Viewer.



5. If you want to change the output channel, select the channel you would like, or create your own, from the **output** dropdown menu.
6. Set the **far** control value to specify at what distance from the camera the depth values are ignored. This prevents large depth values from creating unwanted banding caused by the precision of the floating math calculations.

The DepthToPosition node allows you to create a position pass from an image's depth pass, and feed that into PositionToPoints to ultimately create a point cloud. In fact the three nodes, DepthToPosition, PositionToPoints, and DepthToPoints can be used together, as DepthToPoints is a gizmo that contains the DepthToPosition and PositionToPoints nodes.

To learn more about generating depth maps, please refer to Generating Depth Maps.

Creating a Dense Point Cloud Using the PositionToPoints Node

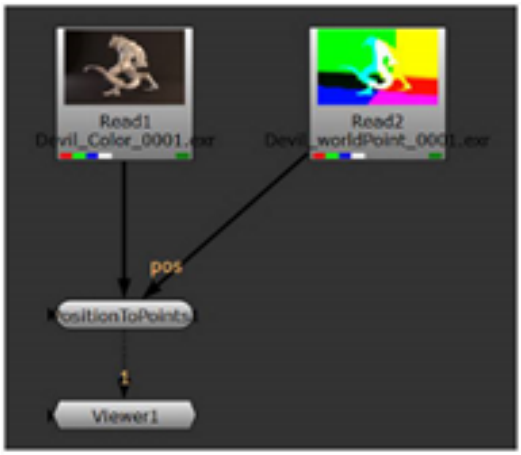
PositionToPoints takes position data contained in an image file (rendered from a 3D application) and recreates the image as a dense 3D point cloud in Nuke. The x, y, and z vertices in the position channel are used to define point positions in 3D space, the size and number of which can be adjusted using the **point size** and **point detail** controls.

To Create a Point Cloud Using PositionToPoints

1. From the **3D** > **Geometry** menu, select **PositionToPoints** to add the node to your script.
2. Read in a position pass and connect it to the node.

   If you have multiple images containing separate position and normal information, connect them to the **pos** and **norm** inputs, respectively.

   Note that the **pos** and **norm** inputs only appear once you've connected the unnamed input.



The normals are contained in the source image, while a second image contains the position pass.

3. Select the position channel from the **surface point** dropdown menu. If your image contains a normal channel, select this from the **surface normal** dropdown menu. You should now see the point cloud in the 3D Viewer.

If you supplied the node with separate position and normal information via the **pos** and **norm** inputs, the **surface point** and **surface normal** dropdowns are disabled.

4. To change the number of points, adjust the **point detail** slider. A value of 0 means no points are displayed. A value of 1 displays all available points.
5. To change the size of the points, adjust the **point size** slider.



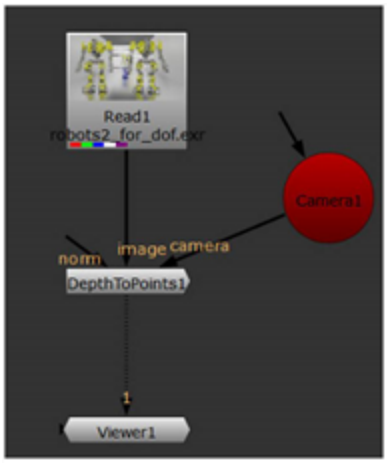
Point cloud created from
the PositionToPoints node.

Creating a Point Cloud Using the DepthToPoints Node

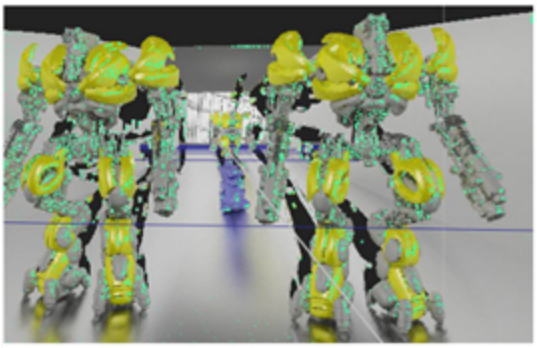
DepthToPoints is a gizmo containing the DepthToPosition and PositionToPoints nodes. It can be used to generate a 3D point cloud from a depth pass and 3D camera. DepthToPoints takes the depth data and color information contained in an image file and recreates the image as a 3D point cloud.

To Create a Point Cloud Using DepthToPoints

1. From the **3D** > **Geometry** menu, select **DepthToPoints** to add the gizmo to your script.
2. Read in an image with a depth pass and connect it to the **image** input of the gizmo.
3. From the **3D** menu, select a **Camera** and connect it to the **camera** input of DepthToPoints.



4. In the DepthToPoints **User** tab, select the depth channel from the **depth** dropdown menu. If your channel contains a normal channel, select this from the **surface normal** dropdown menu, or connect an image containing normal data to the **norm** input of the node. You should now see the point cloud in the 3D Viewer.
5. To change the number of points, adjust the **point detail** slider. A value of 0 means no points are displayed. A value of 1 displays all available points.
6. To change the size of the points, adjust the **point size** slider.



Tip: If you are having trouble viewing the image and point cloud, ensure that your image's alpha is not set to black.

Object Display Properties

You can adjust the display characteristics of all geometric objects in a scene. These settings don't affect the render output of the scene; these are for display purposes only in the 3D Viewer.

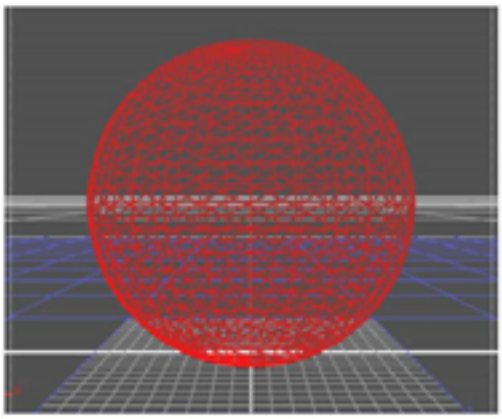
To Edit an Object's Display Attributes

1. Double click on the object's node to display its parameters.
2. From the **display** dropdown menu, select the display type that you want for the object.

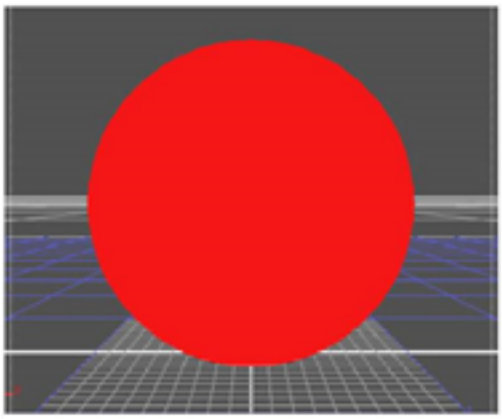


These are how each of the display options appear:

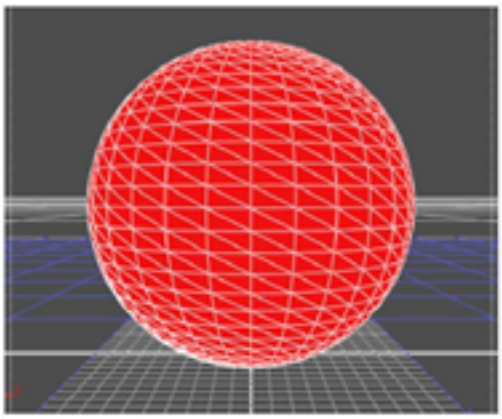
• **wireframe** displays only the outlines of the object's geometry.



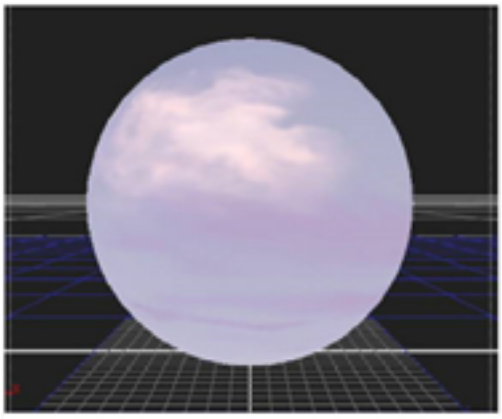
• **solid** displays all geometry with a solid color.



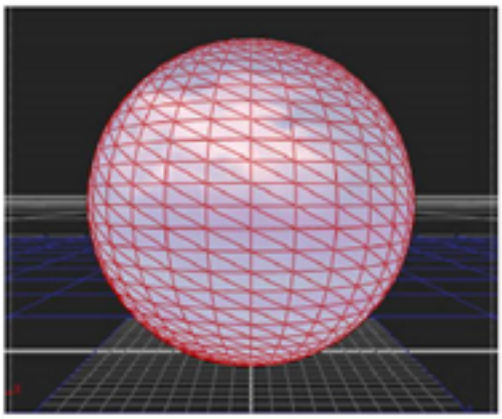
• **solid+wireframe** displays the geometry as solid color with the object's geometry outlines.



• **textured** displays only the surface texture.



• **textured+wireframe** displays the wireframe plus the surface texture.

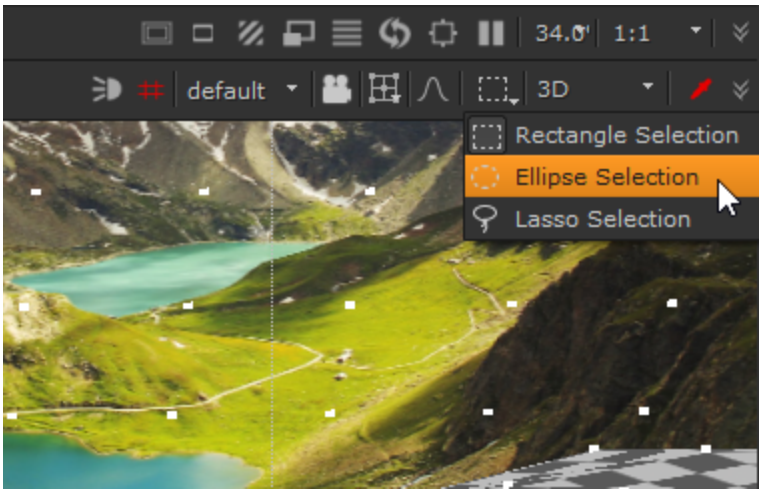


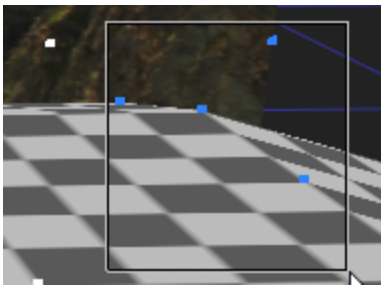
3D Selection Tools

Using the 3D selection tools in the top right corner of the Viewer, you can select nodes, vertices and faces on a 3D object. Additionally, you can select individual 3D objects in the 3D view, which is handy when you have more than one object in a node.

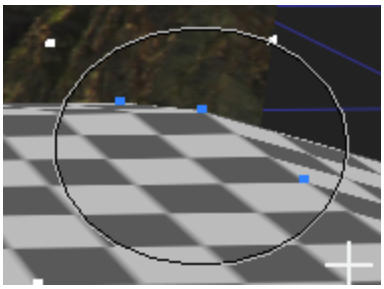
Selection Modes

Nuke features three selection modes, **Rectangle**, **Ellipse**, and **Lasso**. Selection modes work the same in all Viewer contexts, whether you're selecting vertices, faces, or objects. The selection mode dropdown is located above the Viewer at the top-right.

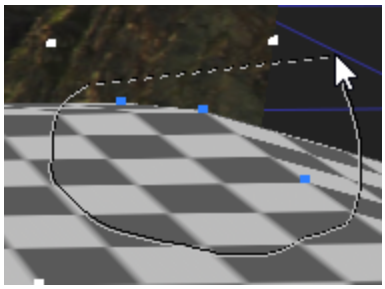




Rectangle



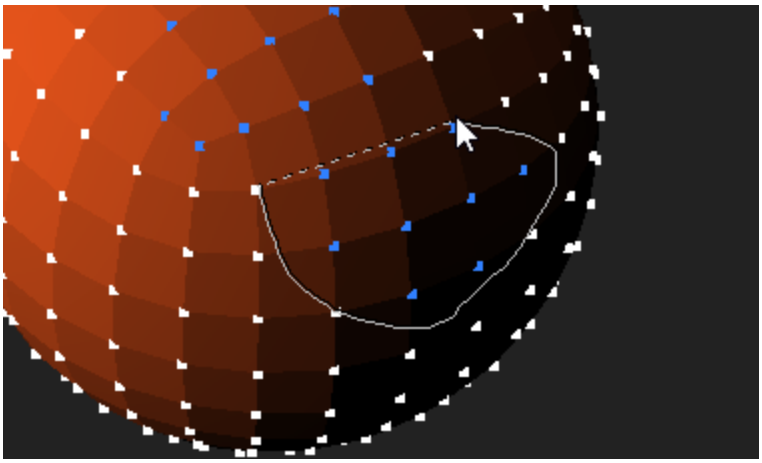
Ellipse



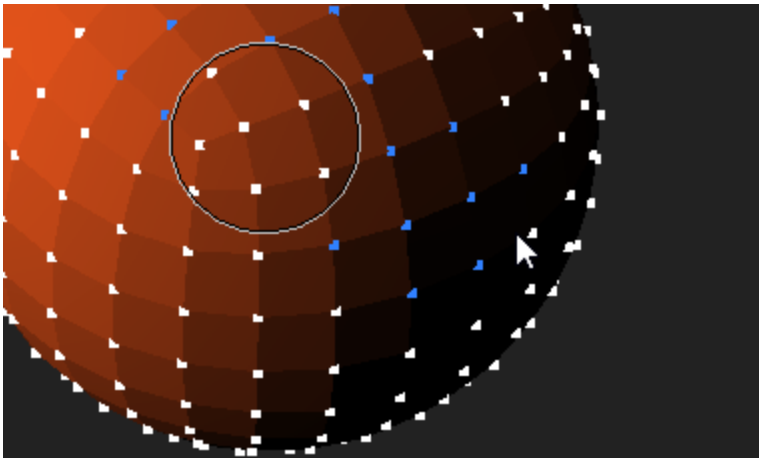
Lasso

Each mode uses the same selection modifiers:

- **Shift** - additive selection. Holding **Shift** and selecting vertices, faces, or objects adds the selection to any existing selection.



- **Alt** + **Shift** - subtractive selection. Holding **Alt + Shift** and selecting vertices, faces, or objects removes the selection from any existing selection.



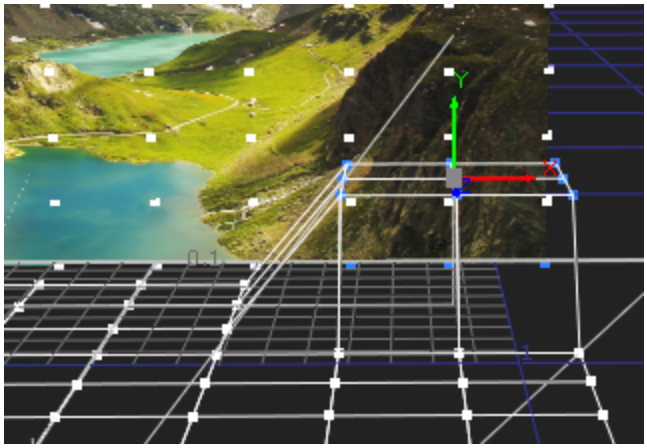
Soft Selection

Soft selection makes it easier to make subtle changes to geometry in the 3D Viewer, creating soft transitions in your geometry modifications using EditGeo or ModelBuilder. See Modifying Object Shapes and Using ModelBuilder for more information.

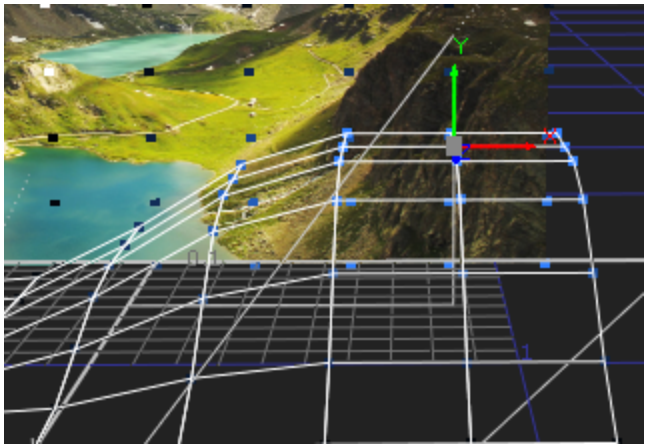
When enabled, selecting vertices or faces in the Viewer adds additional selections according to a **Falloff Radius** and **Falloff Curve** in the Viewer's node **Properties** > **3D** tab under **Soft Selection**. Soft selection includes preset falloff selection curves, such as **Linear** and **Steep In**, and allows you to create custom curves.

To enable soft selection, click  above the 3D Viewer and then select the target vertices or faces. Dragging the transform handle adjusts the selected vertices or faces along with the additional selections described by the falloff.

Tip: You can also enable soft selection by pressing **N** in the Viewer.



Falloff disabled on a Card



S-shape falloff enabled on a Card

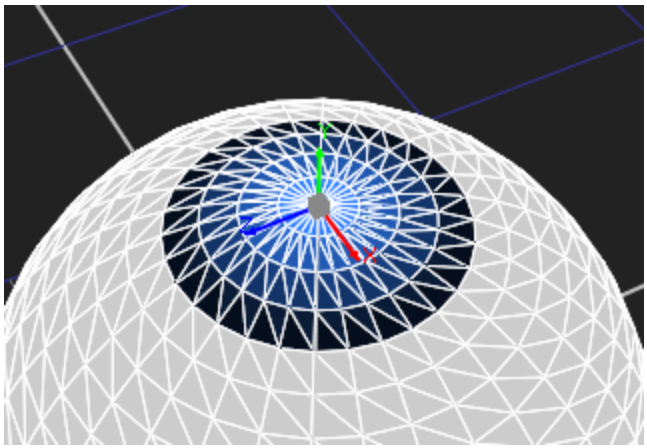
The Viewer **F** and **H** keyboard shortcuts also work with selections:

- Pressing **F** focuses the Viewer differently depending on selection:
  - If you have vertices or faces selected, pressing **F** focuses the Viewer on the selection.
  - With no selections in the Viewer, pressing **F** focuses on the whole scene.
- Pressing **H** focuses the Viewer differently depending on the view mode:
  - If you're using a custom camera, pressing **H** switches the view to the camera view.
  - In any view with the **default** camera, pressing **H** focuses on the whole scene, the same as pressing **F** with no selection.

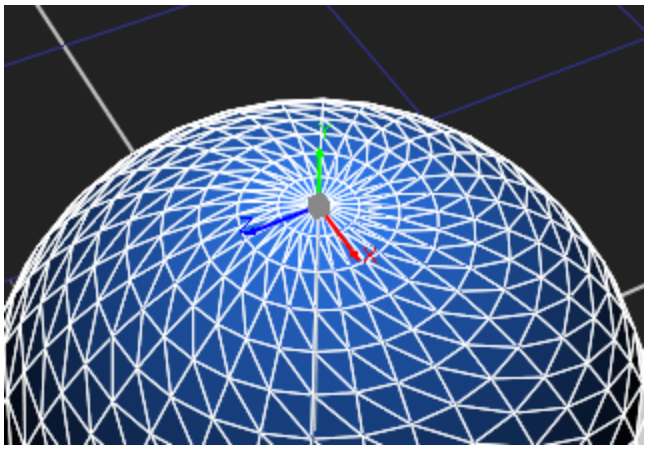
Controlling Selection Falloff

Selection falloff determines how vertices or faces farther away from your selection are affected by changes to the geometry. The greater the **Falloff Radius**, the more you affect remote vertices or faces. You can use one of the built-in **Falloff Curve** presets, such as **Linear** or **Steep In** to modify selection more accurately or create your own curve from scratch.

1. Add your selection to the Viewer using the EditGeo or ModelBuilder nodes.
2. Press **S** in the Viewer to display the Viewer node **Properties** panel.
3. Click the **3D** tab and then open the **Soft Selection** properties.
4. Select the required **Falloff Curve** and then adjust the **Falloff Radius** to update the selection falloff.



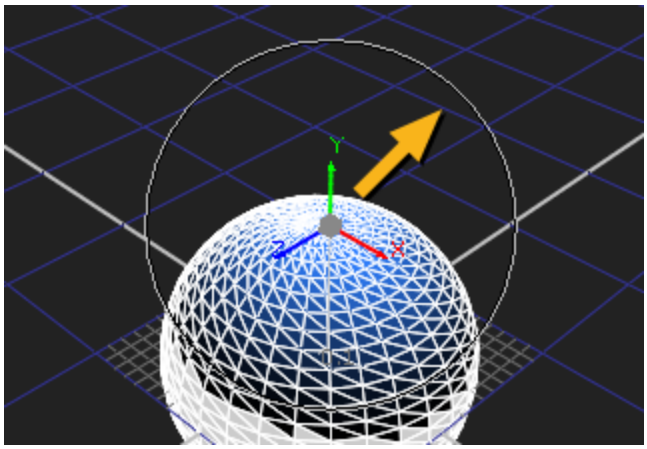
Low radius with a **Linear** curve

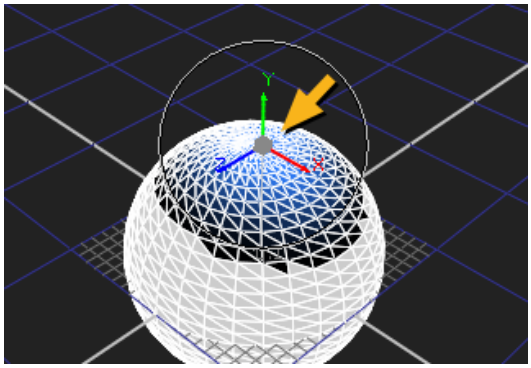


Default radius with a **Linear** curve

OR

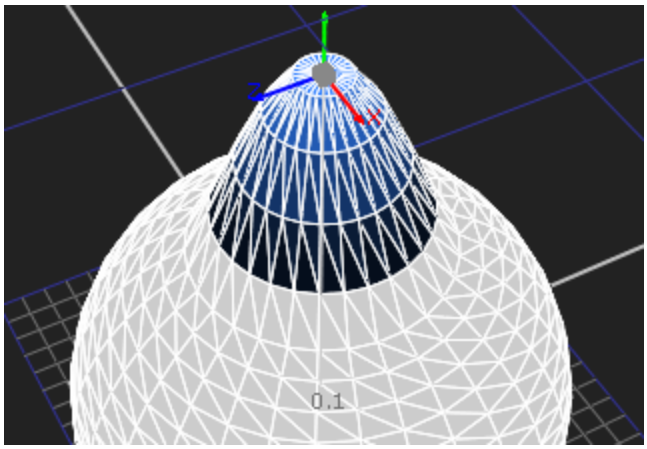
You can hold the **N** key over the Viewer and then left-click and drag or middle-click and drag to control the falloff.



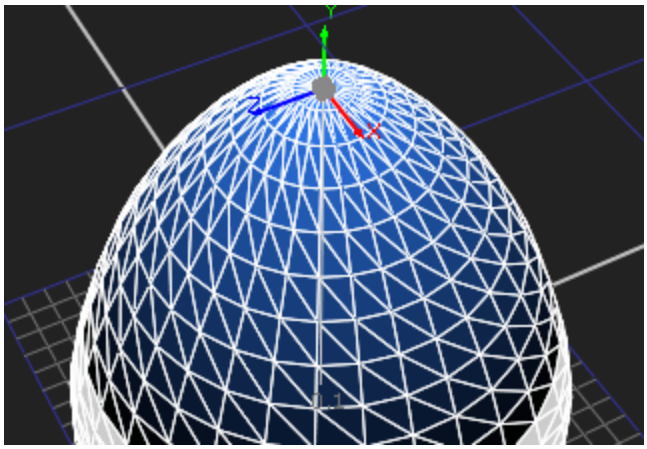


Note: Left-click and drag remembers the extent of the **Falloff Radius** between operations, but middle-click and drag always starts from a **Falloff Radius** of 0.

5. Adjust your geometry as required.



Low radius with a **Linear** curve



Default radius with a **Linear** curve

Selecting Nodes

You can select nodes with the Node selection tool . This is the default selection tool, and corresponds to the familiar way of selecting 3D object nodes in the Viewer.

Selecting Vertices on a 3D Object

You can select vertices on a 3D object using the Vertex selection tool. To save selections into your script, you need to use the GeoSelect node. You can find the 3D selection tools in the top right corner of the Viewer.



Note: The selection mode is set by the Viewer - changing geometry nodes translates your selections to the new object.

You can also use the Vertex selection tool to edit geometry using the EditGeo node. See Modifying Objects Using the EditGeo Node.

To select vertices in the Viewer

1. Attach a 3D object to the Viewer and activate the Viewer's 3D mode.
2. Click the **Vertex selection** tool  to activate it. It is disabled by default.
3. Select vertices on the object by dragging a marquee over the vertices you want to select.
4. By default, a new selection replaces an existing one, but you can also add and remove from an existing selection:

- To add to existing selection, hold down **Shift** while selecting.
- To remove from existing selection, hold down **Shift+Alt** while selecting.

5. You can also include hidden items by selecting the **occlusion test** button . Deselect it if you don't want to include occluded vertices in your selection. Select if you want to select both visible and occluded vertices.

To Save or Restore Selected Vertices Using the GeoSelect Node

With the default Viewer 3D selection tools you can only make a temporary selection. To save or restore a selection, you can use the same 3D selection tools in the Viewer with the GeoSelect node:

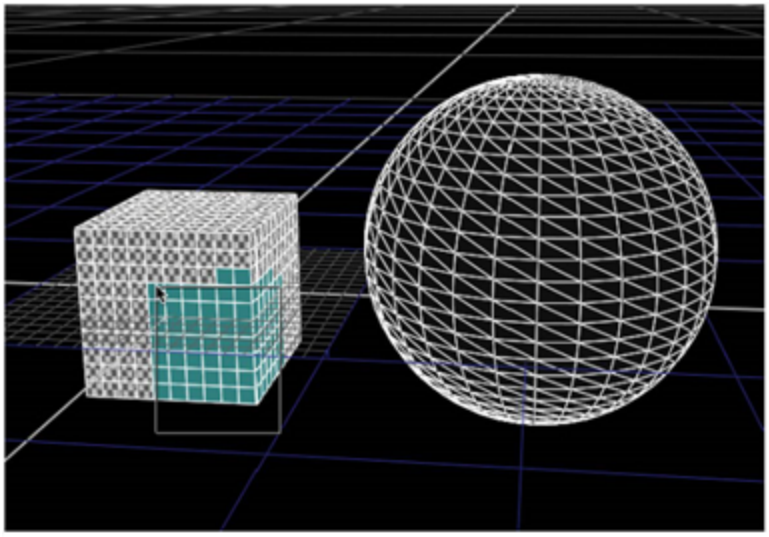
1. Create a GeoSelect node (**3D** > **Modify** > **GeoSelect**) and attach a 3D object to its input. The GeoSelect properties panel needs to be open in order for you to use the node for selecting vertices.
2. Click the **Vertex selection** tool  from the 3D selection panel.
3. Select vertices on the object by dragging a marquee over the vertices you want to select.
4. By default, a new selection replaces an existing one, but you can also add and remove from existing selection:
   - To add to existing selection, hold down **Shift** while selecting.
   - To remove from existing selection, hold down **Shift+Alt** while selecting.
5. In the GeoSelect properties panel, you can also uncheck the **selectable** box to freeze your selection so it can't be changed. Checking the box again enables selection again and you can continue selecting vertices as normal.
6. With the **display** and **render** controls you can select the display option you want for your object. For more information on these controls, see Object Display Properties.
7. When you're happy with your selection, click **save selection** to store the vertices in the GeoSelect node.
8. The vertices are saved by the GeoSelect node and can be restored by clicking **restore selection**.

Selecting Faces on a 3D Object

You can select faces on a 3D object using the **Face selection** tool . Each 3D object is divided into polygonal faces by wireframe lines, and with the Face selection tool you can select one or many of these.

You can also use the Face selection tool to edit geometry using the EditGeo node. See Modifying Objects Using the EditGeo Node.

1. Click the **Face selection** tool to activate it. It is disabled by default.
2. Select faces on the object by dragging a marquee over the faces you want to select.
3. By default, a new selection replaces an existing one, but you can also add and remove from existing selection:
   • To add to existing selection, hold down Shift while selecting.
   • To remove from existing selection, hold down **Shift+Alt** while selecting.



Marquee-selecting faces on a cube.

Selecting 3D Objects

When you have more than one 3D object in one node (such as in a case when you're reading in a .fbx file with multiple objects), you can use the Object selection tool  to select them. To select objects, activate the Object selection tool and click on the object you want to select in the 3D view.

Matching Position, Orientation, and Size to 3D Selection

You can match the position, orientation, and size of a 3D object, such as a Card node, to your selected vertices in another object. Do the following:

1. Click the **snap** dropdown menu  on the properties panel of the 3D object you want to move and select **Match selection position**. The object is now positioned to match the selected vertices.
2. If you select **snap** > **Match selection position, orientation**, your 3D object is positioned and aligned according to the 3D vertex selection.

3. If you select **snap** > **Match selection position, orientation, size**, your 3D object is positioned, aligned, and scaled according to the selected vertices.

Merging Objects

With the MergeGeo node, you can merge your 3D objects together to process all of them at the same time. For example, after merging your objects, you can use a Transform node to move the objects together, or add an ApplyMaterial node to apply a global material to them (note that this overrides any individual materials applied to the geometry before it was merged).

To Merge Your 3D Objects

1. Select **3D** > **Modify** > **MergeGeo** to insert a MergeGeo after the 3D objects in your script.
2. Connect the objects you want to merge to the MergeGeo node's inputs.

   You can now process all the objects you connected to the MergeGeo node together.

Modifying Object Shapes

Many nodes under the **Modify** menu let you modify the shape of an object as a whole. Modifying only selected portions of an object is also supported using the EditGeo node.

You can modify 3D objects by selecting vertices or faces in the Viewer using the EditGeo node or by using lookup curves, power functions, images, a Perlin noise function, a distortion function, or trilinear interpolation.

Modifying Objects Using the EditGeo Node

EditGeo allows you to directly modify an object's vertices or faces, depending on the Viewer selection mode currently active. You can also modify multiple objects simultaneously if the EditGeo is downstream of a MergeGeo node with multiple geometry inputs.



Tip: See 3D Selection Tools for more information on selecting objects in the Viewer.

To Modify Objects Using EditGeo

1. Select the 3D object or MergeGeo you want to modify in the Node Graph.
2. Navigate to **3D** > **Modify** > **EditGeo** to insert an EditGeo node.
3. In the node's controls, use the **display** dropdown menu to select how you want to view your object in the Viewer while making changes to it.

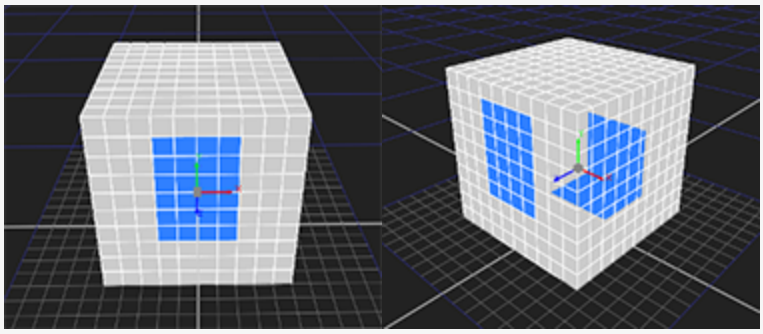
Tip: If you don't have an image or texture associated with your geometry, you may find that enabling the **headlamp** control in the Viewer properties improves visibility.

Press **S** in the Viewer to display the Viewer properties, then enable the **3D** > **headlamp** checkbox.

4. Set the Viewer selection mode using the 3D selection tools in the top right corner of the Viewer:
   - **Vertex selection** - allows you to select individual vertices on geometry, giving you fine control while editing. You can also marquee select multiple vertices or use additive selection by holding down **Shift**.
   - **Face selection** - allows you to select individual faces on geometry. You can also marquee select multiple faces or use additive selection by holding down **Shift**.

Note: You can turn off **occlusion testing** by clicking the disable icon  in the Viewer to allow you to select vertices and faces that are hidden from the view point.

For example, selecting faces on the "front" of a cube with occlusion testing disabled also selects faces on the opposite, hidden side of the cube.

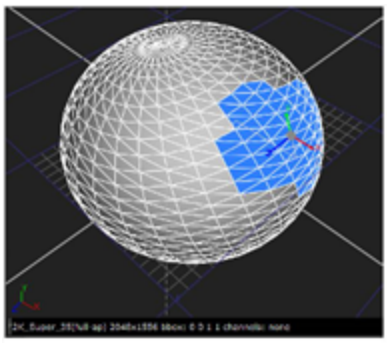


See 3D Selection Tools for more information.

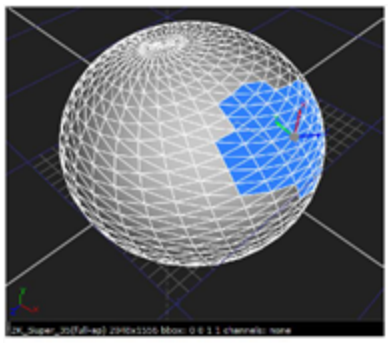
5. Set how the axis handles are aligned with your selection using the **axisalignment** dropdown:
   - **object** - the position of the xyz axis is determined by the average position of all vertices in the selection. The orientation of the axis is the same as the object's orientation.

- **average normal** - the position of the xyz axis is determined by the average position of all vertices in the selection. The orientation of the axis is aligned to the average of the current selection's normals.

  In this mode, the z axis always points directly away from the current selection.



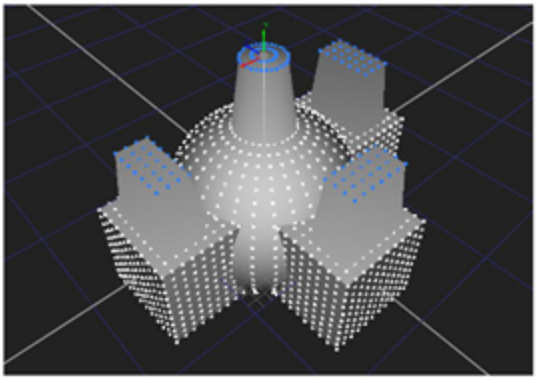
Object alignment



Average normal alignment

Tip: Holding **Ctrl**/**Cmd**+**Alt** allows you to drag the center of the axis across the surface of the geometry, with the orientation set to the nearest face's normal.

6. Make vertex or face selections in the Viewer as required. Hold **Shift** to add to a selection or **Alt**+**Shift** to remove selections.
7. When you're satisfied with your selections, drag the axis associated with the selection to a new position to edit the geometry.



Tip: You can create smoother, more organic edits using Nuke's soft selection tool above the Viewer. See Soft Selection for more information.

To Animate Edits

1. Set the Viewer selection mode using the 3D selection tools in the top right corner of the Viewer:
   - **Vertex selection** - allows you to select individual vertices on geometry, giving you fine control while editing. You can also marquee select multiple vertices or use additive selection by holding down **Shift**.
   - **Face selection** - allows you to select individual faces on geometry. You can also marquee select multiple faces or use additive selection by holding down **Shift**.
2. Make selections in the Viewer and then click  in the properties panel.
3. Scrub the playhead to the required frame and transform the geometry using the handles in the Viewer.
   A keyframe is added automatically on the new frame.
4. Repeat the process to produce the required animation.

Tip: Click  to remove a keyframe at the current frame.

5. Click **reset geometry** to return the vertices to their original positions.

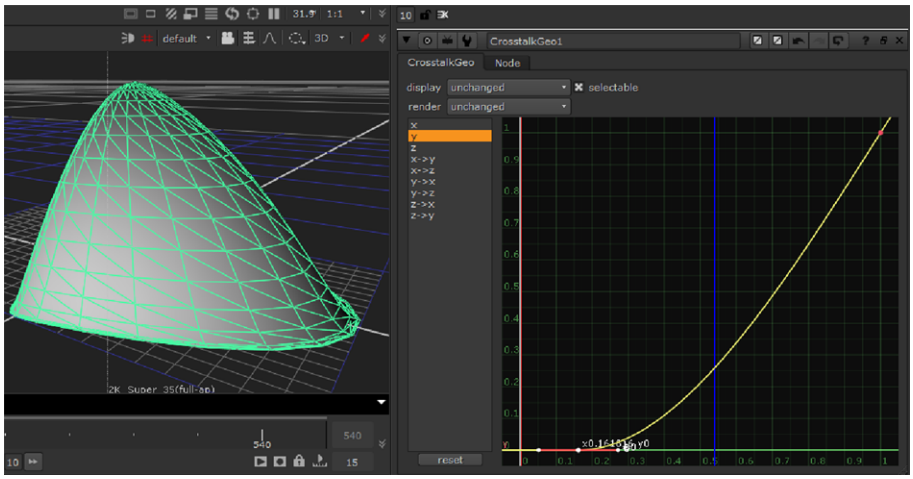
Modifying Objects Using Lookup Curves

The CrosstalkGeo and LookupGeo nodes offer you direct global control over each of the vertex x, y, and z values respectively. You can, for example, only modify all the y values without touching the x and z values.

You change the different vertex values (x, y, or z) by modifying their associated 2D curves in lookup tables (LUTs). The x axis in the LUT represents the current vertex value, and the y axis the new vertex value.

By default, the curve is a diagonal line where all the points in the curve have the same value on the y axis (the new value) as they do on the x axis (the current value). Because both x and y values are the same, there is no change in the object's shape.

By modifying, for example, the CrosstalkGeo node's **y** LUT the following way, you can set some of the vertex y values of a sphere to 0 to squash its bottom half:



Modifying the CrosstalkGeo node's LUT.

With the CrosstalkGeo node, you can also use one of the vertex x, y, and z values to evaluate the lookup curve and then add the result to another vertex value. For example, you could modify the x->y curve, using the vertex x value to find the new value on the curve, and then add that to the vertex y value. This way, you can modulate the y values by another channel.

By default, these curves are horizontal lines at y=0. They produce no change, because the value added to the vertex (the new value on the y axis) is 0.

To Modify Objects Using Lookup Curves

1. Select **3D** > **Modify** > **CrosstalkGeo** or **LookupGeo** to insert a CrosstalkGeo or LookupGeo node anywhere after the 3D object you want to modify.
2. Attach a Viewer to the node to see your changes.
3. In the node's controls, use the **display** dropdown menu to select how you want to view your object in the Viewer while making changes to it.
4. From the list on the left, select the curve you want to modify. For example, you'd select **z** to only modify the vertex z values.

   In the case of the CrosstalkGeo node, you can also select **y->x**, for example, to use the vertex y value to evaluate the curve and add the result to the vertex x value.
5. Adjust the curve as necessary. To insert points on the curve, **Ctrl/Cmd**+**Alt**+click on the curve.

Modifying Objects Using a Power Function

The LogGeo node lets you modify the shape of your 3D objects using a power function. Using this node, you can raise each of the vertex x, y, and z values to a power (X^x, Y^y, Z^z). This can have a different effect depending on whether you are dealing with negative or positive values.

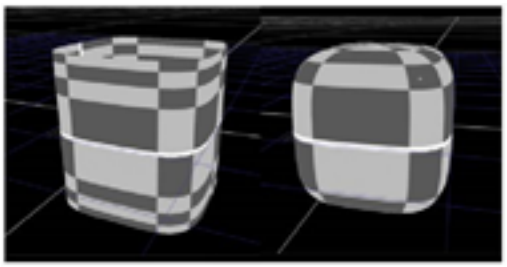
To Modify Objects Using a Power Function

1. Select **3D** > **Modify** > **LogGeo** to insert a LogGeo node anywhere after the 3D object you want to modify.
2. Attach a Viewer to the node to see your changes.
3. In the node's controls, use the **display** dropdown menu to select how you want to view your object in the Viewer while making changes to it. See Object Display Properties.
4. Check **swap**. This swaps the values and the powers they are raised to around (for example, changes 5^7 into 7^5).
5. In the **log x**, **y**, and **z** fields, enter the power you want to raise the respective vertex values to. For example, if you want to raise the vertex z values to the power of 20, enter 20 in the **z** field.

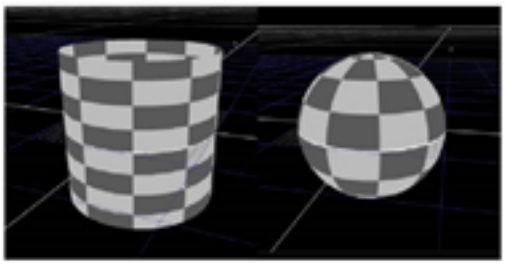
   Alternatively, you can adjust your 3D object in the Viewer by dragging the white control point to a new location. You can find the control point just outside the object.
6. To clamp the negative x, y, and z values to 0.0, check **clamp black**. This option is only valid if you have checked **swap**.

Tip: If you set the **log x**, **y**, and **z** values to 1 and check **swap**, the LogGeo node produces no change in the incoming geometry. If you want to try out how the node works, this is a good place to start as you can then gradually adjust the values from there.

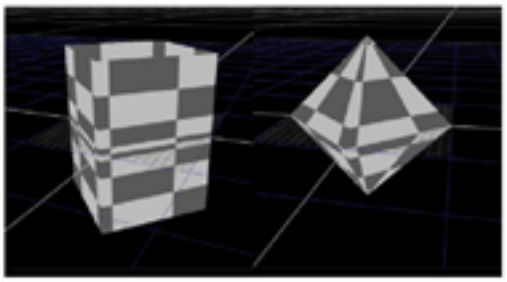
The following images illustrate the effect of the LogGeo node on the default Nuke cylinder and sphere when **swap** is checked in the LogGeo controls. Notice that if these objects were not positioned in the default location (centered around 0,0,0), the results would be different.



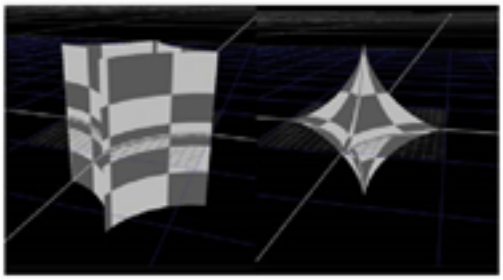
The LogGeo node applied to the default cylinder and sphere: Log x, y and z set to 0.5.



The LogGeo node applied to the default cylinder and sphere: Log x, y, and z set to 1 (no change).



The LogGeo node applied to the default cylinder and sphere: Log x, y, and z set to 2.

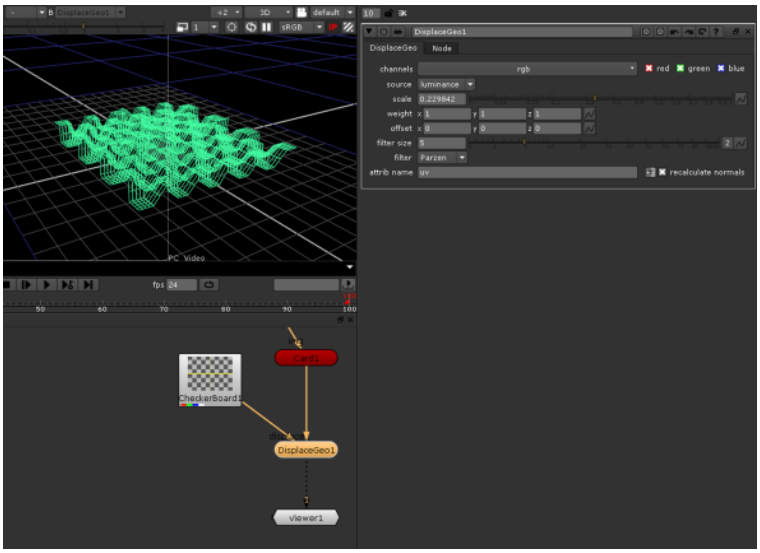


The LogGeo node applied to the default cylinder and sphere: Log x, y, and z set to 2.

Modifying Objects Using an Image - Method 1

With the DisplaceGeo node, you can modify geometry based on an image. When using the node, each vertex is displaced along its normal with a value corresponding to the image pixel the vertex's uv attribute points to. The higher the pixel value, the greater the displacement.

The following image illustrates the principle behind the DisplaceGeo node. A Card node is modified to resemble the pattern of a Checkerboard image.



Using the DisplaceGeo node to modify geometry.

To Modify Objects Using an Image

1. Select **3D** > **Modify** > **DisplaceGeo** to insert a DisplaceGeo node anywhere after the 3D object you want to modify.
2. Attach a Viewer to the node to see your changes.
3. In the node's controls, use the **display** dropdown menu to select how you want to view your object in the Viewer while making changes to it.
4. Read in your image map and connect it to the DisplaceGeo node's **displace** input.
5. Adjust the following controls:
   - From the **channels** dropdown menu and check boxes, select the channels to use for the displacement value.
   - From the **source** dropdown menu, select the source for the displace value. For example, if you selected **rgb** or **rgba** from the **channels** dropdown menu, you can use the red, green, blue, or alpha channel or the pixel luminance as the source. You can also select **rgb relative** to move the vertices on the x, y, and z axes by the amounts in rgb, or **rgb absolute** to move the vertices to the values in rgb.
   - To define the scale of the displacement, adjust the **scale** slider. The higher the value, the bigger the displacement.
   - To give x, y, and z different weightings, enter new weights the **weight** fields. By default, each weighting is set to 1. If you don't want to make changes to a value, set its weight to 0.
   - To offset x, y, and z values, enter the value by which you want to offset them in the **offset** fields. For example, if you enter 0.5 in the y offset field, 0.5 is added to the y value.
   - To change the size of the filtering applied to the image before the displacement, adjust the **filter size** slider.

- To select the filtering algorithm applied to the image before the displacement, select an algorithm from the **filter** dropdown menu. For more information, see Choosing a Filtering Algorithm.
- To change the name of the attribute that's used as the vertex's UV coordinates to find the image pixel, enter a name in the **attrib name** field.
- Usually, the normals aren't correct after the vertices have been moved. To recalculate them after the displacement, check **recalculate normals**.

Modifying Objects Using an Image - Method 2

Like the DisplaceGeo node, the Displacement shader node also performs displacement mapping and at first glance the nodes seem very similar. However, the approach they have on performing displacement mapping is different.

Displacement mapping is a technique for adding geometric detail to object surfaces as you render them. Unlike the DisplaceGeo node, The Displacement node does this on the fly, only displacing those parts of the geometry that are visible at any given moment. Displacement considers the point of view of camera to determine which parts of the displacement need rendering, thus saving render time. It's also possible to optimize the level of tessellation to be the level that you need for an object at a certain distance.

Connecting the Displacement Node

1. Create the Displacement node by clicking **3D** > **Shader** > **Displacement**.
2. Connect your geometry to the Displacement node's output. If you want, you can connect a texture to the Displacement node's input.
3. Connect the image you want to create the displacement from in the **displacement** input.
4. Optionally, you can use a separate map for calculating the normals. Connect this to the **normals** input.
5. Proceed to Adjusting the Displacement Controls below.

Adjusting the Displacement Controls

1. Use the **displacement channel** dropdown menu to select the channel from your **displacement** input that you want to use as the displacement map.
2. If you're using a **normals** input, uncheck **build normals** and set **normal expansion** to:

- **none** to use the normals as they are,
- **XY** to multiply them on x and y dimensions, and
- **XYZ** to multiply them in x, y and z dimensions.

3. Use **scale** to set the overall scale of the displacement.
4. Set **filter size** to the size of the filter you want to use when sampling the input image.
5. Use the **filter** dropdown menu to select a filtering algorithm. For more information, see Choosing a Filtering Algorithm.
6. If you want to automatically calculate normals after the displacement, check **build normals**. Uncheck this, if you want the normals calculated from the **normals** input.
7. Proceed to Adjusting Displacement Controls for Rendering below.

Adjusting Displacement Controls for Rendering

Before rendering a scene where you've used Displacement, adjust the controls on the **Tessellation** tab to gain speed and quality for the render process:

1. Use **max subdivision** to set the maximum number of iterations of polygon subdivision that occurs in tessellation.

   If you're used to working with DisplaceGeo or displacement in other applications, it's worth noting that you can get a similar amount of detail with lower geometry subdivisions using the Displacement node.

   For example, a 30x30 Card is constructed from the tessellation of 1800 triangles (30x30x2), but you can achieve a similar tessellation levels using the Displacement node with fewer card subdivisions:

Card subdivisions	max subdivisions	tessellation triangles
1x1 (2 triangles)	5	2048
2x2 (8 triangles)	4	2048
4x4 (32 triangles)	3	2048



Note: Using a high number of geometry **and** Displacement subdivisions is likely to slow down rendering so far it becomes unusable - A 10x10 card with **max subdivisions** set to 4 generates 51200 triangles!

2. Set the **mode** dropdown menu to the mode used in polygon subdivision:
   - **uniform** - uniform polygon tessellation. This is a good option mainly for testing your result, and only in rare occasions the best option to actually use for a displacement render.

- **screen** - tessellation is determined by the screen size. This is the default and often the best mode option. The tessellation is determined by the size of the tessellate polygons on the screen. This mode ensures that no new polygons are created once a particular polygon edge screen length is reached.
- **adaptive** - tessellation is determined by the complexity of the displacement. This option attempts to estimate flat areas in the image where tessellation is unnecessary. The calculation is based on the threshold controls which are only active if the **adaptive** mode is selected.

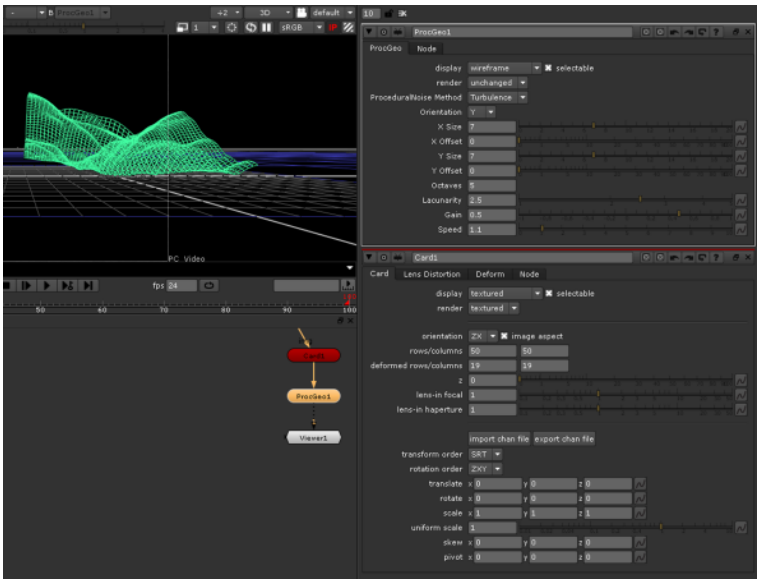
3. If you set **mode** to **screen** or **adaptive**, set **pixel edge length** to the maximum size of polygons used in tessellation.
4. If you set **mode** to **adaptive**, you can also adjust the following:
   - **edge threshold** - edges larger than this threshold get divided, whereas edges smaller than this are subdivided according to the normal and displace thresholds.
   - **normal threshold** - detects normal orientations to determine if the surface is flat or not. If the angle between adjacent normals is larger than this threshold, tessellation occurs.
   - **displace threshold** - compares the degrees to which two points on the surface are displaced, and if the results do not match, tessellation occurs.



Note: Keep in mind that applying the Displacement shader to a very complex high resolution geometry can be very slow.

Modifying Objects Using a Perlin Noise Function

The ProcGeo, or ProceduralNoise, node lets you modify your 3D objects using a Perlin noise function that creates *seemingly* random noise. For example, you could use the ProcGeo node to generate animated noise for rippling waves or clouds, or to create a terrain from a flat card, like in the following image:



Using the ProcGeo node to create a terrain from a card object.

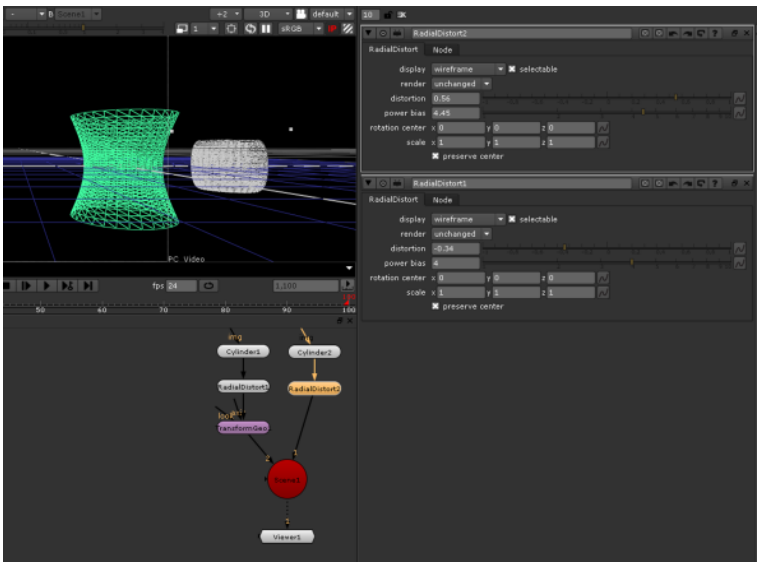
You can select the type of noise and control its look in the ProcGeo node's parameters.

To Modify Objects Using a Perlin Noise Function

1. Select **3D** > **Modify** > **ProceduralNoise** to insert a ProcGeo node anywhere after the 3D object you want to modify.
2. Attach a Viewer to the node to see your changes.
3. In the node's controls, use the **display** dropdown menu to select how you want to view your object in the Viewer while making changes to it.
4. From the **ProceduralNoise Method** dropdown menu, select the type of noise you want to use: **Turbulence** or **fBm** (Fractal Brownian Motion).
5. To select whether to modify the x, y, or z values or all of them, use the **Orientation** dropdown menu.
6. To change the look of the noise, adjust the rest of the parameters. For example, to control the amount of detail of the noise, adjust **Octaves**.

Modifying Objects Using a Distortion Function

The RadialDistort node is a non-linear transformation of the vertices along directions from the object center, giving either a barrel or pin-cushion distortion. In the following image, two cylinders have been distorted using the RadialDistort node.



Barrel and pin-cushion distortions.

To Modify Objects Using a Distortion Function

1. Select **3D** > **Modify** > **Radial Distort** to insert a RadialDistort node anywhere after the 3D object you want to modify.
2. Attach a Viewer to the node to see your changes.
3. In the node's controls, use the **display** dropdown menu to select how you want to view your object in the Viewer while making changes to it.
4. To select whether the distortion is a barrel or pin-cushion, adjust the **distortion** slider. Values below 0 produce a barrel distortion, whereas values above 0 produce a pin-cushion distortion. If you set the value to 0, the 3D object is not distorted.
5. To control the magnitude of the distortion, adjust the **power bias** slider. The higher the value, the more distorted the object becomes.
6. To move the center point of the distortion, enter new coordinates in the **rotation center** fields.

7. To control the amount of distortion in each of the x, y, or z directions, adjust the values in the **scale** fields.
8. To keep the object's center in its original place in the 3D space, check **preserve center**.

Modifying Objects Using a Trilinear Interpolation

With the Trilinear node, you can warp the object as a whole by using a trilinear interpolation to warp the object's bounding box. For example, you can use this node to create animated object deformations, such as the squish/squash of a bouncing ball.

To Modify Objects Using a Trilinear Interpolation

1. Select **3D** > **Modify** > **Trilinear** to insert a Trilinear node anywhere after the 3D object you want to modify.
2. Attach a Viewer to the node to see your changes.
3. In the node's controls, use the **display** dropdown menu to select how you want to view your object in the Viewer while making changes to it.
4. To move each corner of the bounding box, enter new coordinates in the **p0**, **p1**, **p2**...**p7** fields. To cancel your changes and reset the box, select **reset shape to input**.
5. To not use the object's bounding box but define a box yourself, go to the **Source box** tab and uncheck **use incoming bounding box**. Adjust the **src0** and **scr1** coordinates define the box. To change the color of the box, click the **box** button.

Materials and Textures

This section teaches you how to:

- use the nodes in the **3D** > **Shader** menu to control what material your objects seem to be made of. See Object Material Properties.
- merge two **Shader** nodes together. See Merging Two Shader Nodes.
- merge a material with the objects rendered behind it in the 3D scene. See Merging a Material with the Objects Behind.

- replace selected material channels with a constant color to make one object hold out the others. See Replacing Material Channels with a Constant Color.
- project texture images onto your 3D objects. See Projecting Textures onto Objects.
- import a set of texture patches following the UDIM scheme and apply them to the surface of a 3D object. See Importing UDIM Patches.

Choosing the 3D Viewer Type

Nuke 13.0 onwards includes a new Hydra 3D viewport, which uses hdStorm as a new renderer. The Hydra Viewer is more consistent with other applications in your pipeline, such as Katana, Solaris, or USDView, and the represents the output of the ScanlineRender node.



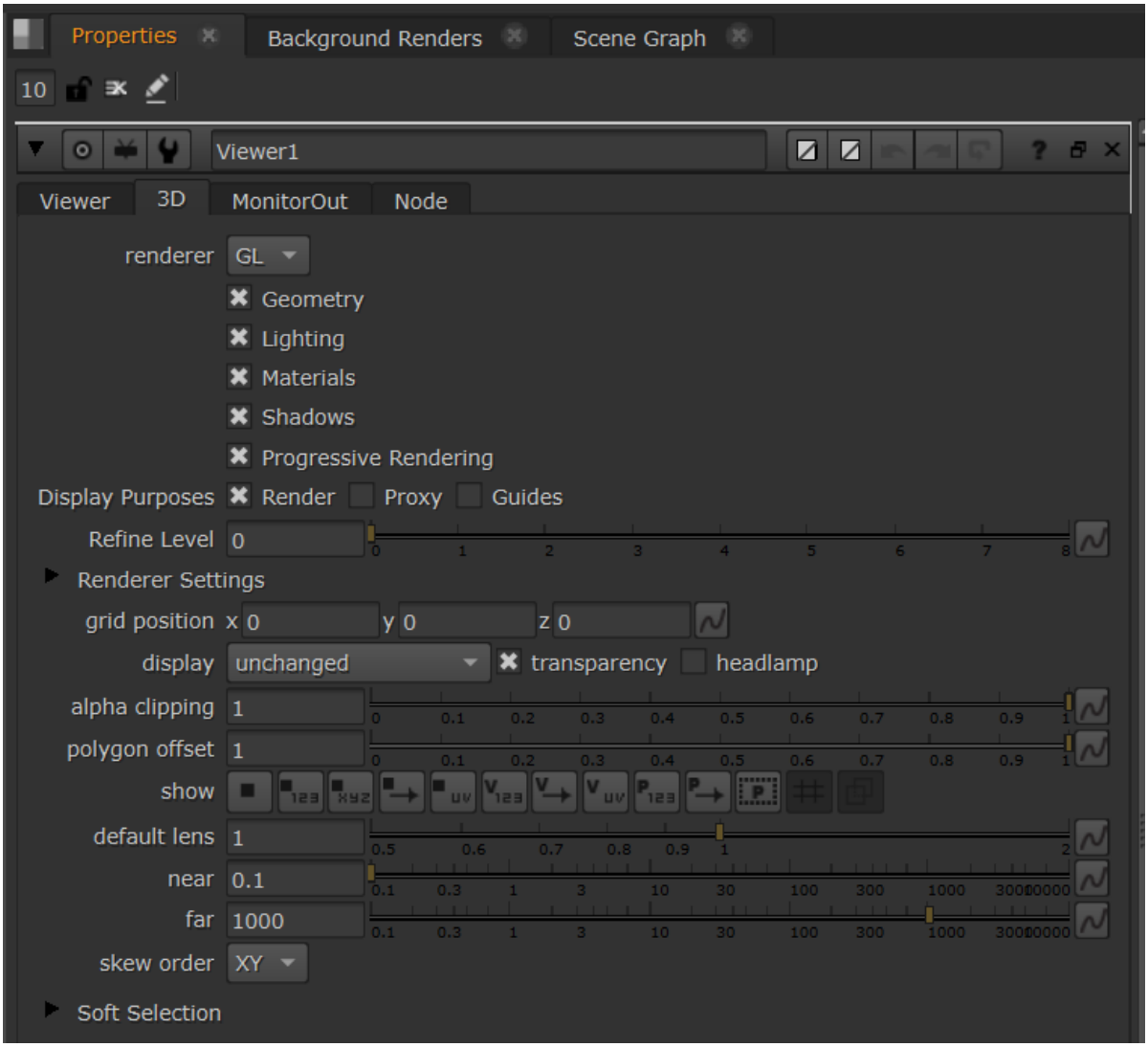
Nuke Viewer | Hydra Viewer | Scanline Render

Nuke's **New 3D System** uses a Hydra renderer, and nodes in the **Classic 3D System** use a "Nuke" renderer.

As such, the Viewer's **renderer** knob only has one option: **GL**, which is both the Hydra and Nuke renderers, depending on if you are viewing New or Classic 3D nodes.

Note: Learn more about the New 3D System and USD at 3D Compositing with the New 3D System.

You can also select whether or not to display geometry, lights, materials and/or shadows in the 3D Viewer by toggling the relevant checkboxes.



The Viewer renderer option is also included in the **Preferences** under **Panels** > **Viewer** > **default renderer**. Changing the preference only affects new Viewer nodes.

Object Material Properties

The nodes under the **Shader** menu let you define the material attributes of geometric objects in a scene, including the quality of light reflected back to the camera from an object's surface. Using these nodes, you can control what material your objects seem to be made of.

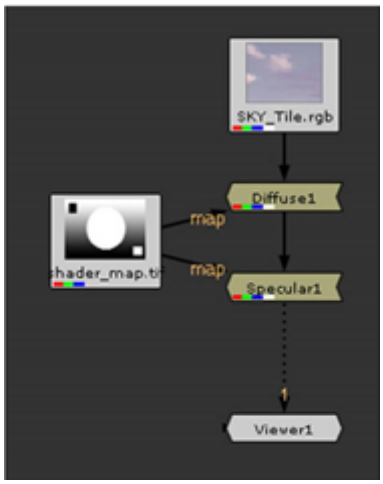
You can also add several Shader nodes one after the other to produce more complex effects. For this, you should use the unlabeled inputs on the Shader nodes.

The material property settings you apply affect the render output of the scene.

You can insert 3D shader nodes in the following places in your scripts:

- between the 2D image you're using for the surface texture and the 3D object node that creates the surface, or
- after the 3D object nodes using the ApplyMaterial node. This is a good way to apply a global material to all objects. See Applying a Material Using the ApplyMaterial Node.

You can use the **map** connectors to input a mask image to limit the effect of the material change.



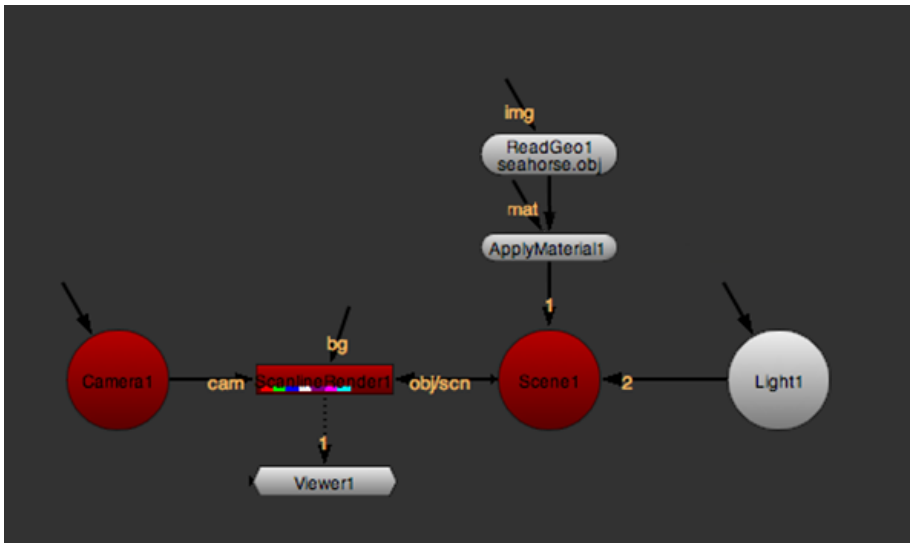
Diffuse and Specular nodes.

Note: You can only see the effect of your changes to an object's material properties in the 2D view.

Applying a Material Using the ApplyMaterial Node

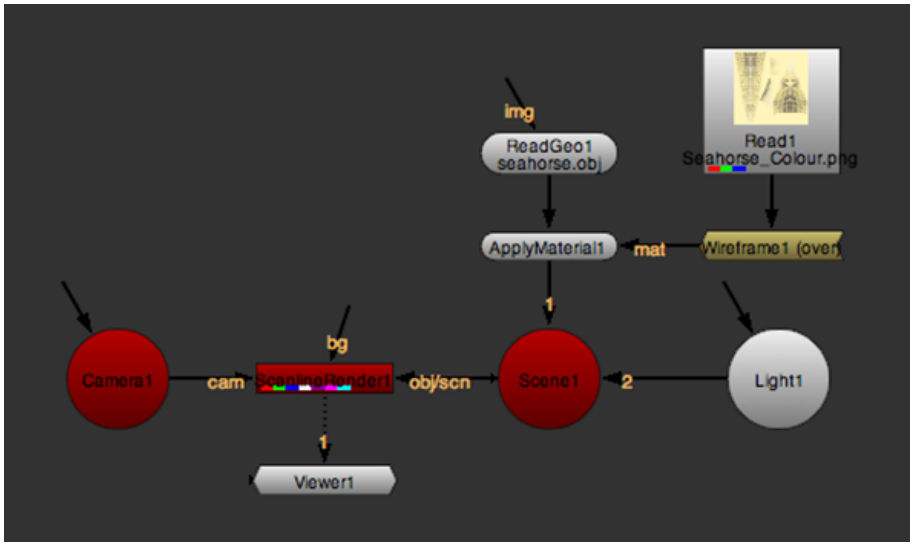
The ApplyMaterial node applies a material from the **mat** input to your 3D object(s).

1. Select **3D** > **Shader** > **ApplyMaterial** to insert an ApplyMaterial node into your script.
2. Connect the unnamed input of the ApplyMaterial node to your geometry (for example, a Sphere, ReadGeo, or ModelBuilder node).



Tip: If you want to apply a global material to several objects, you can also connect the unnamed input to a MergeGeo node. This overrides any materials applied to the individual geometry nodes before they were merged.

3. Connect your materials (for example, a 2D texture image, a BasicMaterial node, or a Wireframe node) to the ApplyMaterial node's **mat** input.



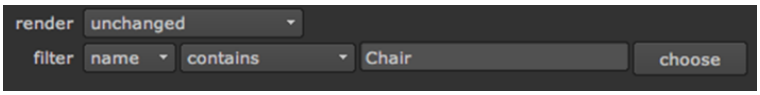
By default, ApplyMaterial applies the material from the **mat** input onto all incoming geometry objects.

4. If you created your geometry using ModelBuilder or imported an Alembic file using ReadGeo, you can choose to only apply the material onto a particular object in the incoming geometry. To do so, open the ApplyMaterial properties and set **filter** to **name**. This allows you to tell ApplyMaterial to ignore any geometry that doesn't match the filter on the right.

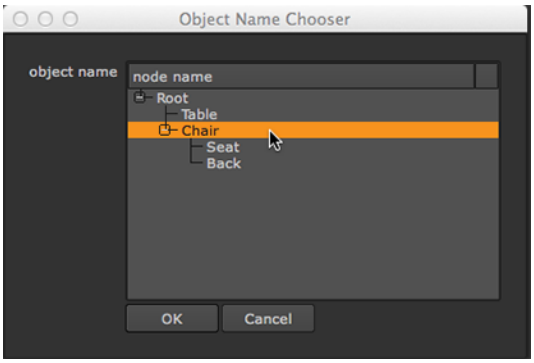
Note: You can also limit a material to a particular object if your geometry was created or imported using a third-party plug-in that adds a name attribute for the geometry objects.

5. To set how to filter the incoming geometry objects, set the dropdown menu next to **name** to:
   - **equals** - set the material on any objects whose name matches the string in the filter name field exactly.
   - **doesn't equal** - set the material on any objects whose name does not match the string in the filter name field exactly.
   - **contains** - set the material for any objects whose name contains the string in the filter name field.

     This can be useful when you have some structure to your object names. For example, if you have objects like /Root/Chair/Seat, /Root/Chair/Back, and /Root/Table, you can select **contains** and set the filter name field to **Chair** to apply the material to all parts of the chair while leaving the table alone.



   - **doesn't contain** - set the material for any objects whose name does not contain the string in the filter name field.
6. To set the filter name, type the name directly into the text entry field or use the **choose** button to open the **Object Name Chooser** dialog and select a filter name from a list of incoming geometry objects.



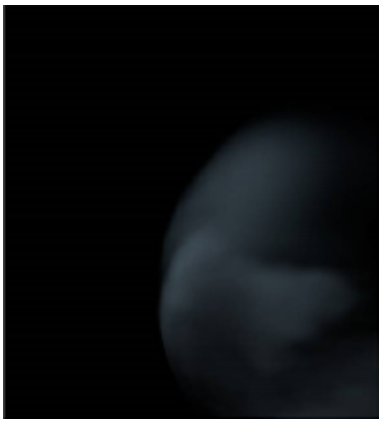
Tip: You can also **Ctrl/Cmd**+click or **Shift**+click in the **Object Name Chooser** dialog to select multiple objects.

Adjusting the Diffuse Color

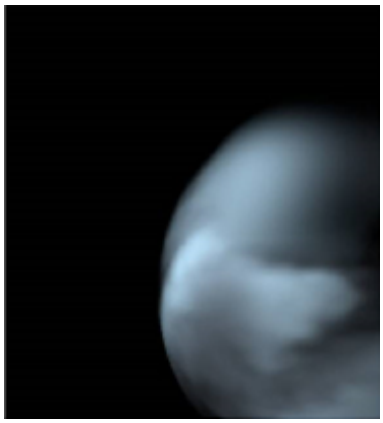
The Diffuse node lets you adjust the color of the material when illuminated. The material appears darker as the surface points away from the light, as the light is not falling on it.

1. Select **3D** > **Shader** > **Diffuse** to insert a Diffuse node into your script.

2. Place the Diffuse node between your 2D texture image and your 3D object node, or connect it to an ApplyMaterial node's **mat** input.
3. In the Diffuse properties, use the **channels** dropdown menu to select the channels you wish to process.
4. Adjust the **white** slider to control the diffuse color. By default, this is in grayscale, but you can also adjust the individual r, g, and b values. The higher the value, the brighter the material.



Diffuse: Low **white** value.

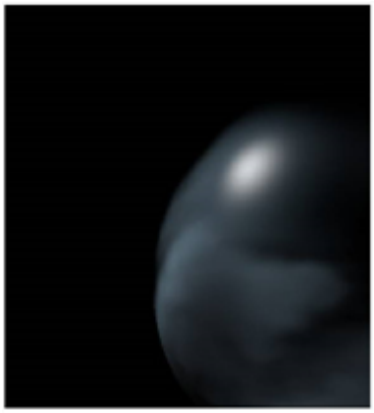


Diffuse: High **white** value.

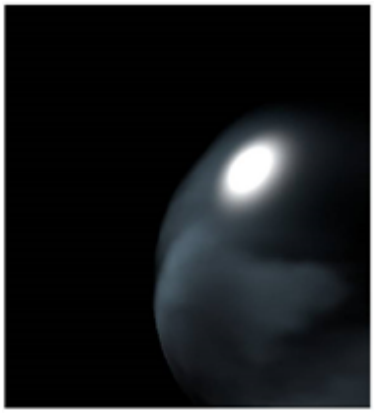
Adjusting Specular Highlights

You can use the Specular node to control how bright and wide the highlights on the material seem. The location of the viewpoint is significant: the specular highlights are the brightest along the direct angle of reflection.

1. Select **3D** > **Shader** > **Specular** to insert a Specular node into your script.
2. Place the Specular node between your 2D texture image and your 3D object node, or connect it to an ApplyMaterial node's **mat** input.
3. In the Specular properties, use the **channels** dropdown menu to select the channels you wish to process.
4. Adjust the **white** slider to control the brightness of the specular highlight. The higher the value, the shinier the material seems.

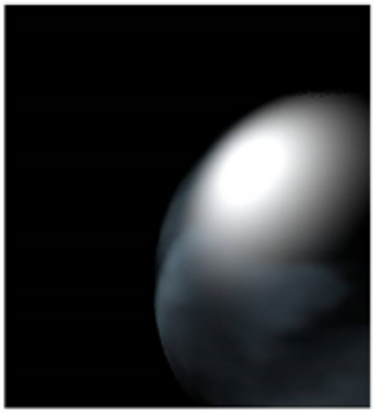


Low **white** value.

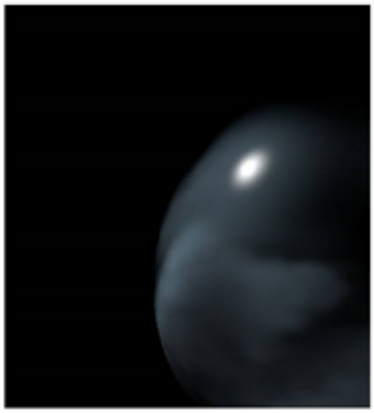


High **white** value.

5. To control the width of the highlights, adjust the **min shininess** and **max shininess** sliders.



Low **shininess** value.



High **shininess** value.

6. If necessary, adjust **shininess channel** to control how the input channels are used to map the black and white values to the **min shininess** and **max shininess** parameters when a **mapSh** input is connected. Select **red** to use the red channel for the mapping, **green** to use the green channel, **blue** to use the blue channel, **luminance** to use the luminance, or **average rgb** to use the average of the red, green, and blue channels.

Simulating Materials That Emit Light

You can use the Emission node to simulate lamps or other sources that emit light.

1. Select **3D** > **Shader** > **Emission** to insert an Emission node into your script.
2. In the Emission properties, use the **channels** dropdown menu to select the channels you wish to process.
3. Adjust the **emission** slider to change the brightness of non-illuminated areas for the surface.
   The higher the value, the more light the material seems to emit and the brighter it appears.

Adjusting Diffuse, Specular, and Emission Using a Single Node

The BasicMaterial node is a combination of the Diffuse, Specular, and Emission nodes, allowing you to control all three aspects of the material from a single properties panel.

1. Select **3D** > **Shader** > **BasicMaterial** to insert a BasicMaterial node into your script.
2. Place the BasicMaterial node between your 2D texture image and your 3D object node, or connect it to an ApplyMaterial node's **mat** input.
3. The BasicMaterial node has several map inputs you can use to mask the effect of the node. Use:
   - **mapD** to modulate the diffuse component,
   - **mapS** to modulate the specular component,
   - **mapE** to modulate the emission component, and
   - **mapSh** to modulate the shininess value.
4. In the BasicMaterial properties, use the **channels** dropdown menu to select the channels you wish to process.
5. Adjust **emission** to change the color of the light the material emits. Note that when you have an image connected to the unlabeled input of the BasicMaterial node and adjust this value, you need to look at the rendered 2D image to see the effect of your changes. Changing the emission value does not have any effect in the 3D Viewer.
6. Adjust **diffuse** to control the color of the material when illuminated.
7. Use **specular** to control how bright the highlights on the material seem.
8. Adjust **min shininess** and **max shininess** to set the minimum and maximum shininess values. If you haven't connected an image to the **mapSh** input of the node, the average of these values is used as the shininess value for the material.
9. Select a **shininess channel** to control how the input channels are used to map the black and white values to the minShininess and maxShininess parameters when a **mapSh** input is connected. Select **red** to use the red channel for the mapping, **green** to use the green channel, **blue** to use the blue channel, **luminance** to use the luminance, or **average rgb** to use the average of the red, green, and blue channels.

Simulating Smooth, Regular Surfaces

The Phong node uses the Phong algorithm to smooth edges between faces. It provides realistic shading and highlights for smooth materials, such as skin and other organic surfaces.

1. Select **3D** > **Shader** > **Phong** to insert a Phong node into your script.

2. Place the Phong node between your 2D texture image and your 3D object node, or connect it to an ApplyMaterial node's **mat** input.
3. The Phong node has several **map** inputs you can use to mask the effect of the node. Use:
   - **mapD** to modulate the diffuse component,
   - **mapS** to modulate the specular component,
   - **mapE** to modulate the emission component, and
   - **mapSh** to modulate the shininess value.
4. In the Phong properties, use the **channels** dropdown menu to select the channels you wish to process.
5. Adjust **color** to change the material color.
6. Adjust **emission** change the color of the light the material emits.
7. To control the color of the material when illuminated, adjust **diffuse**.
8. To control how bright the highlights on the material seem, adjust **specular.**
9. To control how shiny the material appears, adjust **shininess**.
10. To set the minimum and maximum shininess values, adjust **min shininess** and **max shininess**. If you haven't connected an image to the **mapSh** input of the node, the average of these values is used as the shininess value for the material.
11. Select a **shininess channel** to control how the input channels are used to map the black and white values to the **minShininess** and **maxShininess** parameters when a **mapSh** input is connected. Select **red** to use the red channel for the mapping, **green** to use the green channel, **blue** to use the blue channel, **luminance** to use the luminance, or **average rgb** to use the average of the red, green, and blue channels.

Rendering a Wireframe Overlay on Your Geometry

The Wireframe node allows you to render a wireframe overlay on the surface of your geometry object or particle simulation. This can be useful, for example, if you want to:

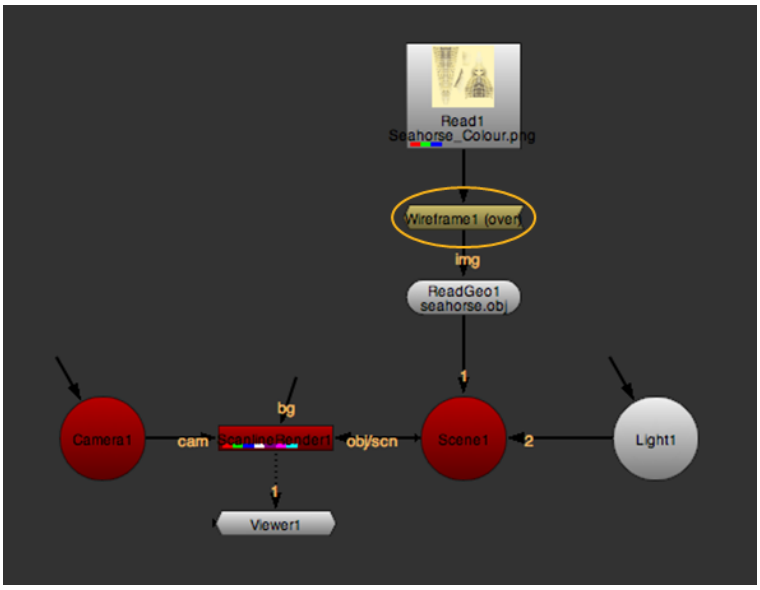
- check that your texture projection correctly lines up with your geometry.
- create a quick render of your 3D scene to check the positioning of objects.
- create motion graphics.
- create "making of" videos.

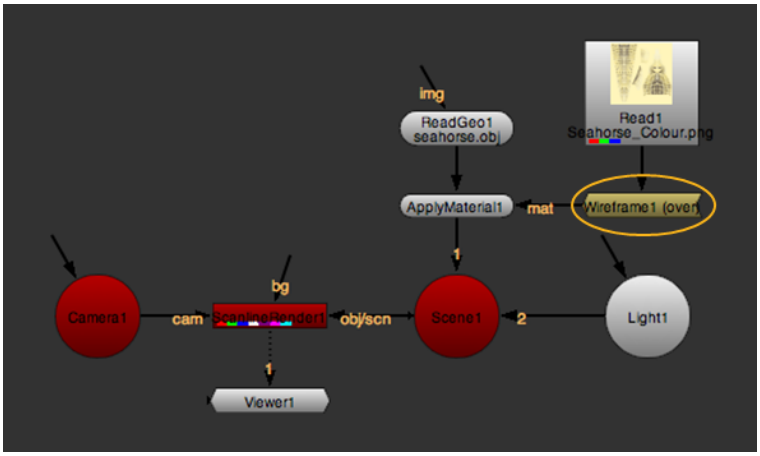


Note: The Wireframe node currently only works if you are rendering your 3D scene using ScanlineRender, RayRender and PrmanRender do not support the Wireframe shader.

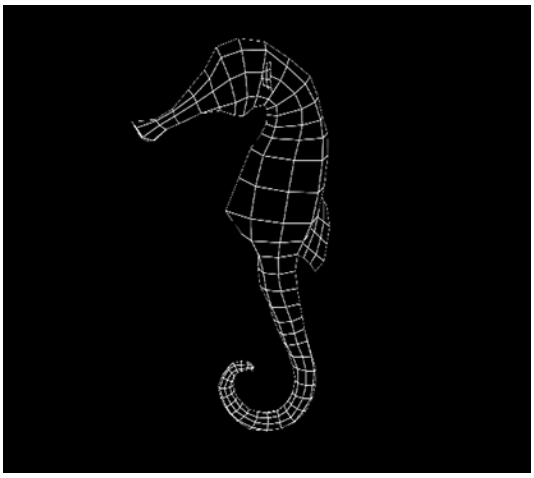
1. Select **3D** > **Shader** > **Wireframe** to insert a Wireframe node into your script.

2. Place the Wireframe node between your 2D texture image and your 3D object node, or connect it to an ApplyMaterial node's **mat** input.

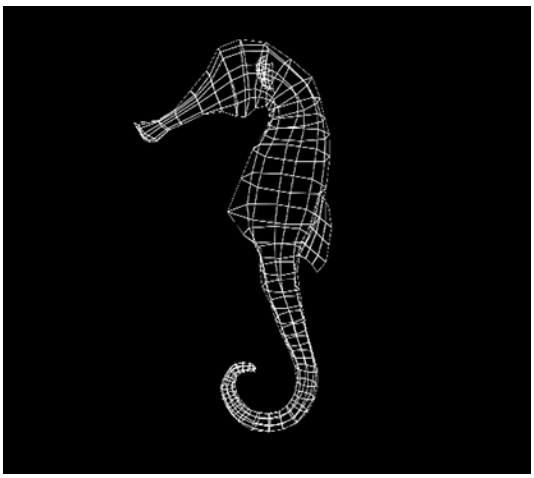




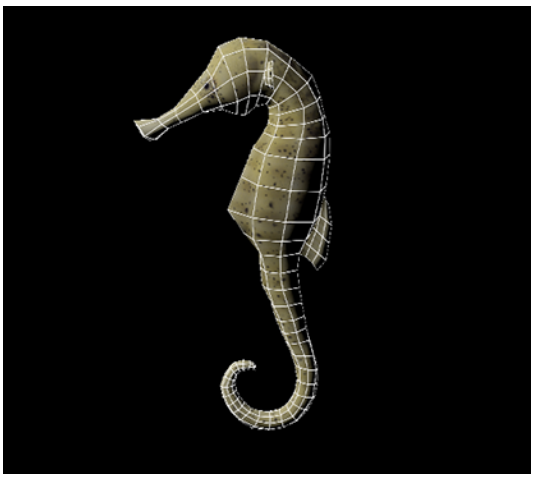
3. In the Wireframe properties, use the **channels** dropdown menu to select the channels you wish to process.
4. From the **operation** dropdown, select how to apply the wireframe overlay to your geometry:
   • **opaque** - display the wireframe on fully opaque black input geometry.



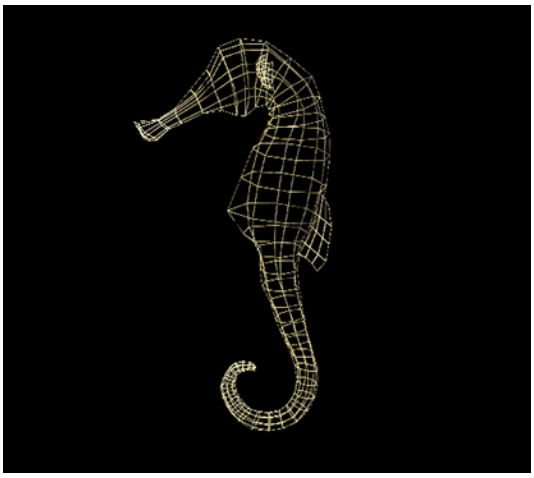
• **see through** - display the wireframe on fully transparent geometry.



• **over** - display the wireframe on top of the input shader or texture.



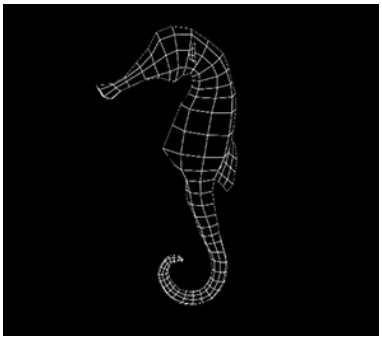
• **multiply** - multiply the wireframe by the input shader or texture and display it on fully transparent geometry.



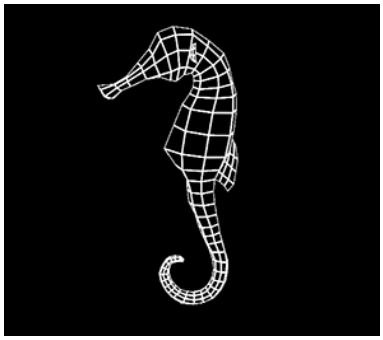
- **modulate** - apply standard diffuse shading to the wireframe and display it on top of the input shader or texture. This takes into account any lights in the scene.



5. To set the width of the wireframe lines (in pixels), adjust **line width**.

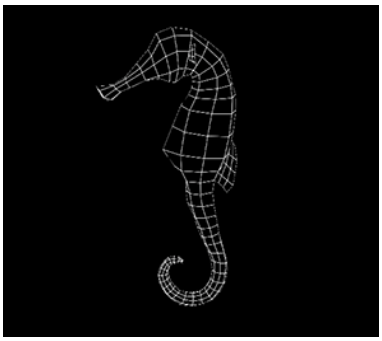


Line width set to 0.5.

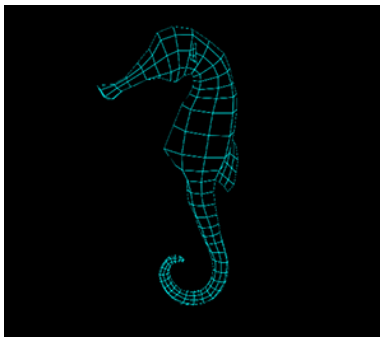


Line width set to 3.

6. To set the color and transparency of the wireframe lines, adjust **line color**.



Line color set to white.



Line color set to cyan.

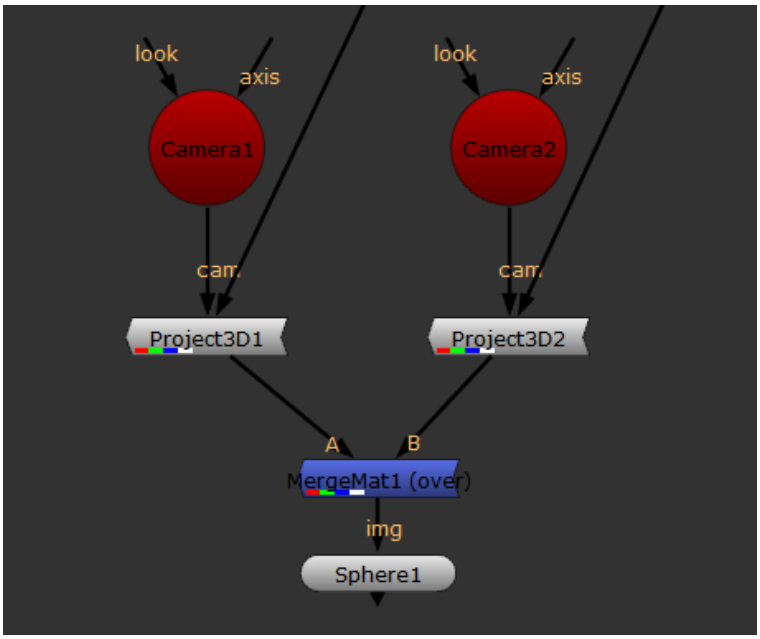
Merging Two Shader Nodes

With the **Shader** menu's MergeMat node, you can combine two shader nodes together, using compositing algorithms like **none**, **replace**, **over**, and **stencil**. The MergeMat node is particularly useful for combining multiple Project3D nodes, allowing you to composite 2D images projected onto the 3D geometry atop each other.

To Merge Two Shaders

1. Select **3D** > **Shader** > **MergeMat** to add a MergeMat (over) node after the two shader nodes you want to combine.
2. Connect the MergeMat node to the **img** input of the 3D object you want to project the images on.
3. Connect the shader nodes to the MergeMat node's **A** and **B** inputs. **A** refers to the foreground element, and **B** to the background element.

   For example, if you wanted to combine two Project3D nodes and composite their results onto a sphere, your node tree would look something like the following:



4. From the **operation** dropdown menu, select how you want to composite the results of the two shader nodes together:
   - to only use input **B** in the composite, select **none**.
   - to only use input **A** in the composite, select **replace**.
   - to composite input **A** over input **B** using a mask, select **over**.
   - to use input **B** outside the mask area, select **stencil**.
   - to use input **B** inside the mask area, select **mask**.
   - to add input **B** to input **A**, select **plus**.
   - to use input **A** if it is greater than input **B** or else use input **B**, select **max**.
   - to use input **A** if it is less than input **B** or else use input **B**, select **min**.
5. For operations (such as **over**) that need an alpha channel (mask), select which channel to use for the alpha from the alpha **channels** dropdown menu.

Merging a Material with the Objects Behind

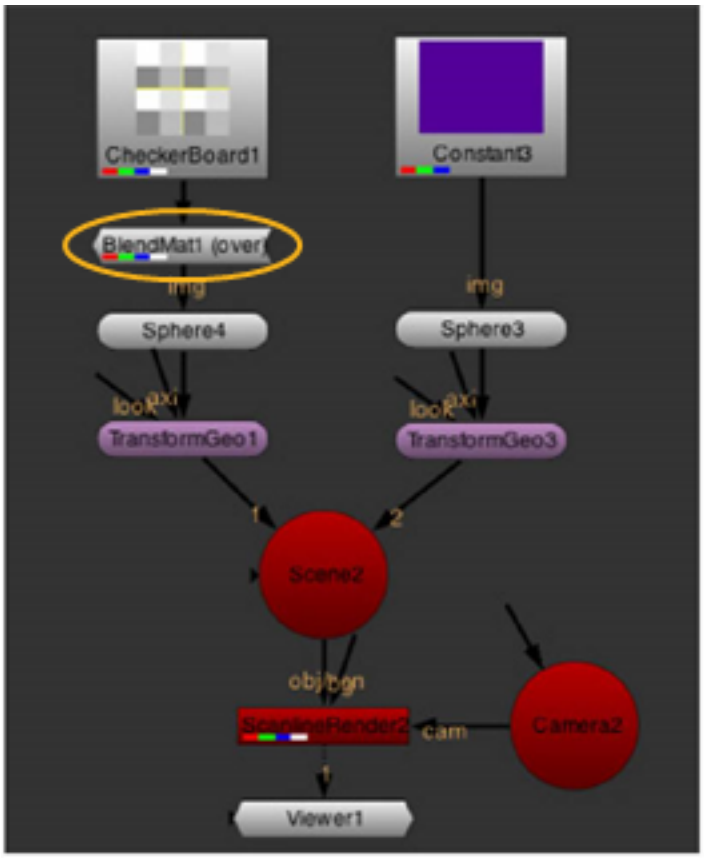
The **Shader** menu's BlendMat node sets how the pixels colored by the material is applied to combine with the pixels from objects behind. It is like the MergeMat node, but instead of blending with another material, it blends with whatever is rendered behind in the 3D scene.



The right-hand image shows the BlendMat node applied to the checkered sphere (that has a checkered alpha channel) and the BlendMat **operation** set to **stencil**.

To Merge a Material with the Objects Behind

1. Select **3D** > **Shader** > **BlendMat** to add a BlendMat node after the material you want to merge with the background pixels.
2. Connect the BlendMat node to the **img** input of the 3D object you want to project the material on.



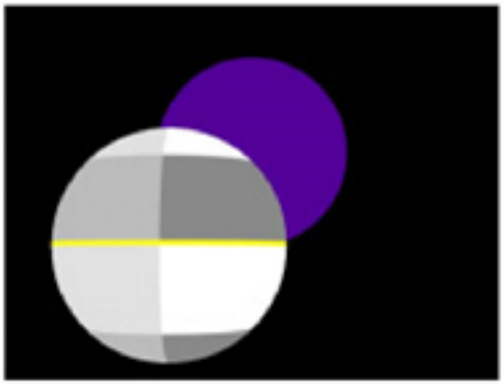
3. From the **channels** dropdown menu, select the channels you want to affect.

4. From the **operation** dropdown menu, select how you want to composite the BlendMat node's input material and the background pixels together:

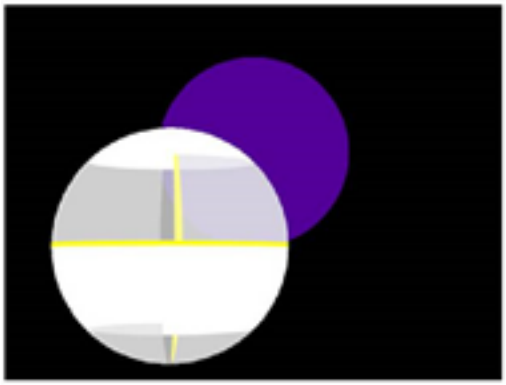
   • to set the material to black, select **none**.



   • to show the material where the material and the background overlap, select **replace**.



   • to composite the material over the background pixels according to the material's alpha, select **over**.



   • to show the background pixels where the material's alpha is black, select **stencil**. Where the material's alpha is white, the material is set to black.

     For this to work, the BlendMat node needs to process the alpha channel, so set **channels** to **rgba**.

     This operation is the opposite of **mask**.



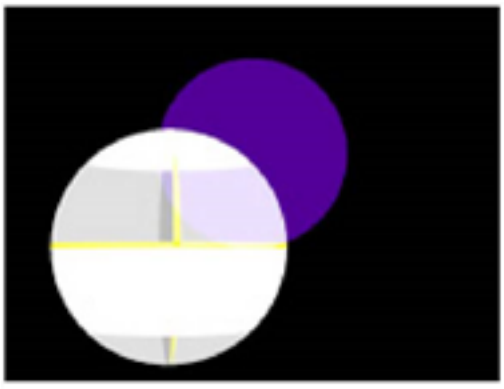
- to show the background pixels where the material's alpha is white, select **mask**. Where the material's alpha is black, the material is also set to black.

  For this to work, the BlendMat node needs to process the alpha channel, so set **channels** to **rgba**.

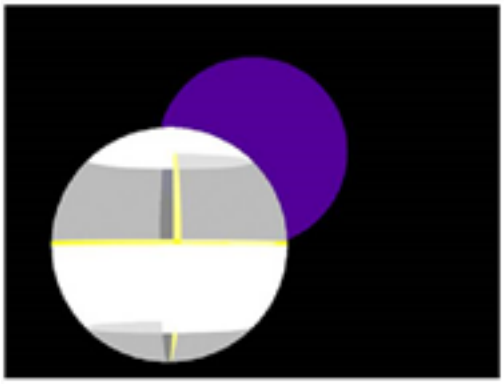
  This operation is the opposite of **stencil**.



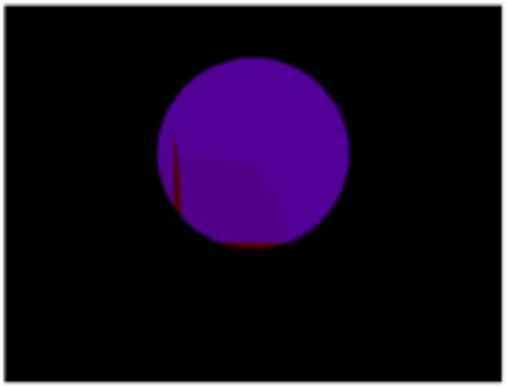
- to add the background pixels to the material, select **plus**.



- to use the material if its pixel values are greater than the background pixels or else use the background pixels, select **max**.

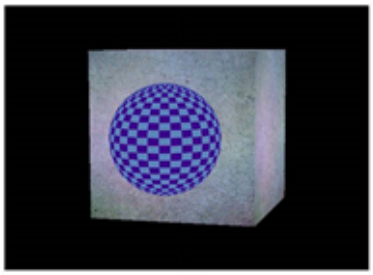


- to use the material if its pixel values are less than the background pixels or else use the background pixels, select **min**.

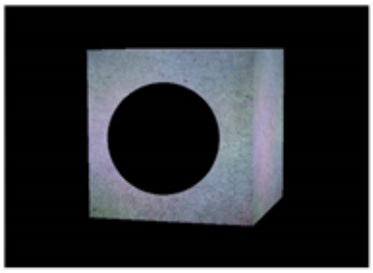


Replacing Material Channels with a Constant Color

The FillMat node lets you replace selected material channels with a constant color. Typically, you would use this node to make one object hold out the others. When you set the FillMat color to 0, it acts as a "3D cookie cutter" and makes a black hole where the material would otherwise be.



A sphere in front of a cube.



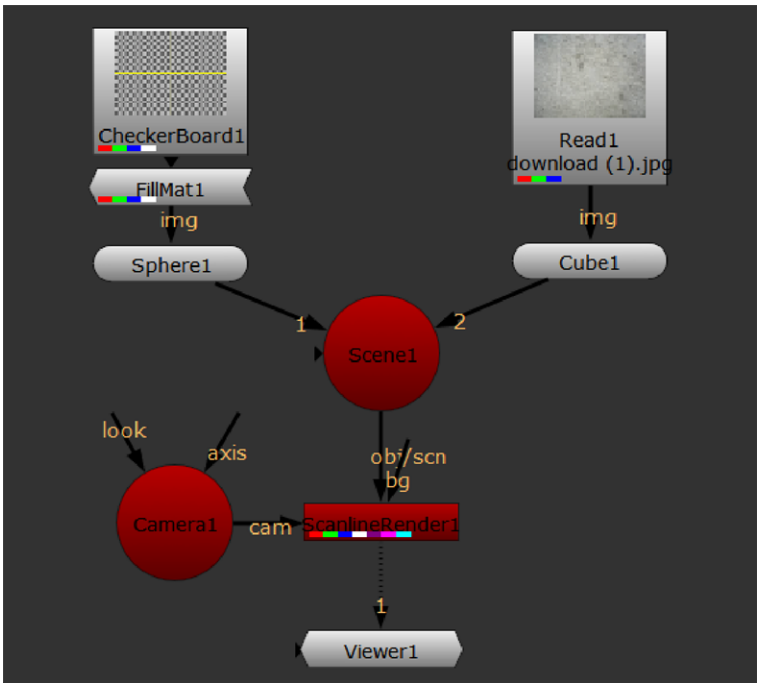
The same scene with the sphere material's rgba channels set to black using the FillMat node.



The alpha channel after applying the FillMat node.

This is similar to using a black Constant node as the input texture. However, the advantage of using the FillMat node is that you can easily apply it to the alpha channel in addition to the rgb channels. Another advantage is that the FillMat node doesn't break the shading sequence, so you can insert it after other material nodes in your node tree.

1. Select **3D** > **Shader** > **FillMat** to insert a FillMat node between the 2D image you're using for the surface texture and the 3D object node that creates the surface.



2. In the FillMat controls, use the **channels** controls to select the channels you want to replace with a constant color.
3. Use the **color** control to select the constant color. By default, this is set to black (0).

Projecting Textures onto Objects

You can use the UVProject and the Project3D nodes to project texture images onto your 3D objects. This way, you can add detail, surface texture, or color to your geometry, making the geometry more realistic and interesting.

The UVProject node changes the uv values of the vertices whereas the Project3D node is a material shader.

Projecting Textures with the UVProject Node

The UVProject node sets the uv coordinates for the object, allowing you to project a texture image onto the object. If the object already has uv coordinates, this node replaces them.

1. Select **3D** > **Modify** > **UVProject** to insert a UVProject node anywhere after the 3D object you want to modify.
2. Attach a Viewer to the node to see your changes.
3. In the node's controls, use the **display** dropdown menu to select how you want to view your object in the Viewer while making changes to it.
4. Connect an Axis or a Camera node to the UVProject node's axis/cam input. If you connect an Axis node, project the texture UV coordinates onto the object using the axis transform values (that is, translation, rotation, scale, etc.). If you connect a Camera node, do a similar projection as with the axis but also use the camera lens information, such as the aperture.
5. Adjust the following parameters:
   - From the **projection** dropdown menu, select the projection type. Usually, it's best to select a type that's close to the object's surface shape. For example, if your object is a sphere, like a football or a planet, select **spherical**.
   - From the **plane** dropdown menu, select the projection direction: **XY**, **YZ**, or **ZX** to project the texture image along the z, x, or y axis. This dropdown menu is only available if you selected **planar** as the projection type.
   - From the **projecton** dropdown menu, select **both**, **front** or **back** depending on whether you want to project the texture on the front face of the object, its back face or both. The front face of an object is the one facing the camera and similarly the back face is the one furthest away from the camera.
   - Check **view frustum culling** if you want the UVProject node to affect only the vertices inside the camera view frustum. Any vertices outside the view frustum are not affected and they still keep their original UV coordinates. Uncheck if you want the node to affect all vertices.
   - To mirror the texture UV coordinates in the horizontal direction, check **invert u**. To mirror them in the vertical direction, check **invert v**.

- To scale (stretch or squash) the texture UV coordinates in the horizontal direction, adjust the **u scale** slider. To scale them in the vertical direction, adjust the **v scale** slider. The higher the value, the more the texture is stretched.
- To change the name of the attribute that's used as the vertex's UV coordinates to find the image pixel, enter a name in the **attrib name** field. If your geometry has more than one set of UV coordinates, you can choose the attribute that suits the projection type.

Projecting Textures with the Project3D Node

The Project3D node projects an input image through a camera onto the 3D object.

1. Select **3D** > **Shader** > **Project3D** to insert a Project3D node after the image you want to project. Connect a Camera node to the Project3D node's cam input.
2. Insert a 3D geometry node (a Sphere for example) after the Project3D node.
3. Attach a Viewer to the 3D geometry node to see your changes.
4. In the node's controls, use the **display** dropdown menu to select how you want to view your object in the Viewer while making changes to it.
5. From the **project on** dropdown menu, select to project the image on either the front facing, back facing, or both polygons.
6. To extend the input image at its edges with black, check **crop**. To extend the image with the edge colors, uncheck **crop**.
7. If you want to use ray casting to test the projection and find out which parts of it are occluded, you can use the occlusion mode dropdown. Select:
   - **none** - to disable occlusion testing.
   - **self** - to tell Project3D that only the geometry connected to it can cause occlusion.
   - **world** - to tell Project3D that other objects in the scene can cause occlusion.

Importing UDIM Patches

When applying textures to models that use regions of UV space outside the standard (0,0) - (1,1) range, it's common to use one texture for each 1x1 square. These textures can be numbered in a variety of ways. UDIM is a numbering scheme that identifies the first texture that's applied to the (0,0) - (1,1) region as 1001, with numbers increasing by one for each texture in the U direction, and by ten for each texture in the V direction.

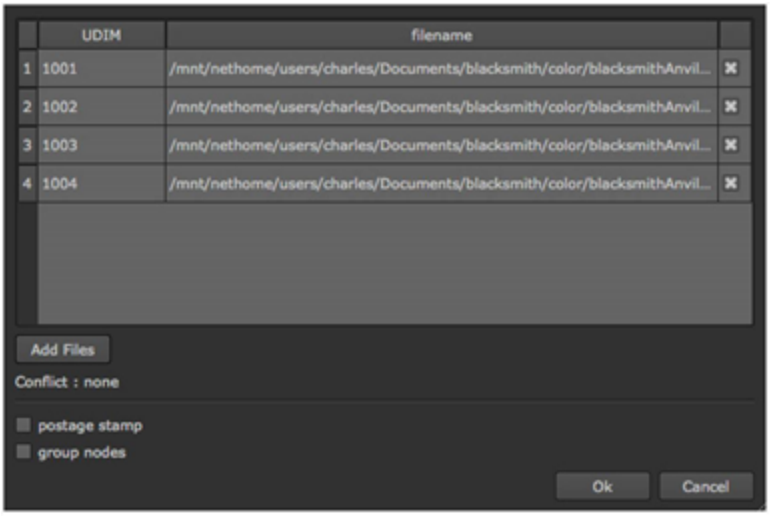
For example, the texture for the region (1,0) - (2,1) would have UDIM 1002, and the texture for (0,1) - (1,2) would have UDIM 1011. A set of patches following the UDIM scheme might look like this:

- color.1001.tif

- color.1002.tif
- color.1003.tif
- color.1004.tif
- color.1005.tif

Where color is the channel name and 1001, 1002, ... is the UDIM value for each patch in UDIM space. In Nuke, you can import a set of patches following the UDIM scheme and quickly apply them to the surface of a 3D object. To do this:

1. Select **Image** > **UDIM Import** from the Toolbar.
2. Select the UDIM image or sequence to import and click **Open**.



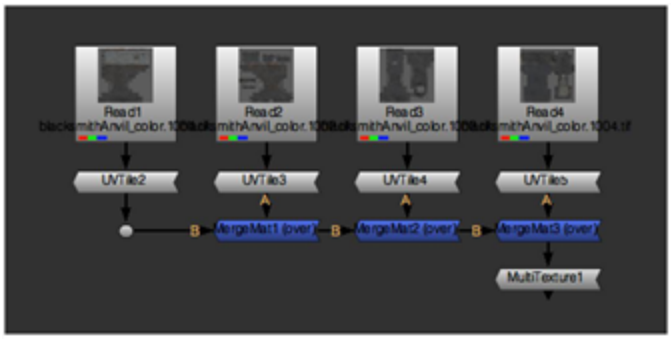
3. To ignore an individual patch, disable the checkbox next to it.
4. To add additional patches, click **Add Files**.



Note: If there's a conflict between the patches that you've added, a notification appears underneath the **Add Files** button describing the conflict.
A typical conflict would be if you attempted to import multiple files sharing the same UDIM value.

5. Check the **postage stamp** checkbox to enable a thumbnail view of the patch on the Read node.
6. Check the **group nodes** checkbox to place the Read nodes for each individual patch into a Group node. The name of the Group node is based on the name of the image or sequence.
7. Click **Ok** to import the files.

This creates a Read node for every patch in the sequence and appends a UVTile node (which allows you to modify a patch's coordinates in UV space) to each one.



Note: If you checked **group nodes** in the UDIM import dialog, highlight the Group node and press **Ctrl**+**Return** (**Cmd**+**Return** on a Mac) to view the contents.

The UDIM value for each patch is inserted into the **udim** field in the UVTile node. This value is important as it offsets every Read node (and therefore, patch) to the correct UDIM space. A number of MergeMat nodes are used to combine the Read/UVTile node combinations.

Note: The MultiTexture node at the end of the tree optimizes rendering when multiple MergeMat nodes are chained together. It uses the default Iop vertex shader and only handles multi-texturing. This node has no fields or knobs and is intended for use only with the **UDIM Import** operation.

To apply these patches to a 3D object in your scene, simply connect the output of the last node in the tree to the **img** input of a ReadGeo node.

To move a patch to a different point in UV space, in the UVTile control panel enter a different UDIM value or disable the **udim** field and enter values in the **u** and **v** fields.

Note: By default the **UDIM Import** operation parses the file name based on the UDIM number. It is possible, however, to create a custom parsing function to work with your own naming conventions. Please refer to the Python Developers Guide for details on how to do this.

Lighting

The nodes under the **Lights** menu let you control the lighting in your scene. Using these nodes, you can bring objects out or push them back, create an illusion of depth, simulate the conditions in the real world, or simply alter the feeling of the scene.

Nuke features four types of light you can use in your 3D scenes: direct light, point light, spot light, and environment light. You can add these using the DirectLight, Point, Spotlight, and Environment nodes. In addition to these, there is a Light node, which lets you create direct, point, and spot lights, as well as read in lights from **.fbx** files. For more information, see Working with Lights.

The Light, DirectLight, Point, and Spotlight nodes all have controls that you can use to adjust how the lights cast shadows in your 3D scene. See Casting Shadows.

Lighting is also affected by the 3D object normals, which are used to determine how the light should bounce off a surface at any particular point. In Nuke, the Normals node allows you to manipulate object normals in order to control the diffuse and specular light contributions. See Manipulating Object Normals.

There's also a Relight node, which takes a 2D image containing normal and point position passes and lets you relight it using 3D lights. See Relighting a 2D Image Using 3D Lights.

Choosing the 3D Viewer Type

Nuke 13.0 onwards includes a new Hydra 3D viewport, which uses hdStorm as a new renderer. The Hydra Viewer is more consistent with other applications in your pipeline, such as Katana, Solaris, or USDView, and the represents the output of the ScanlineRender node.



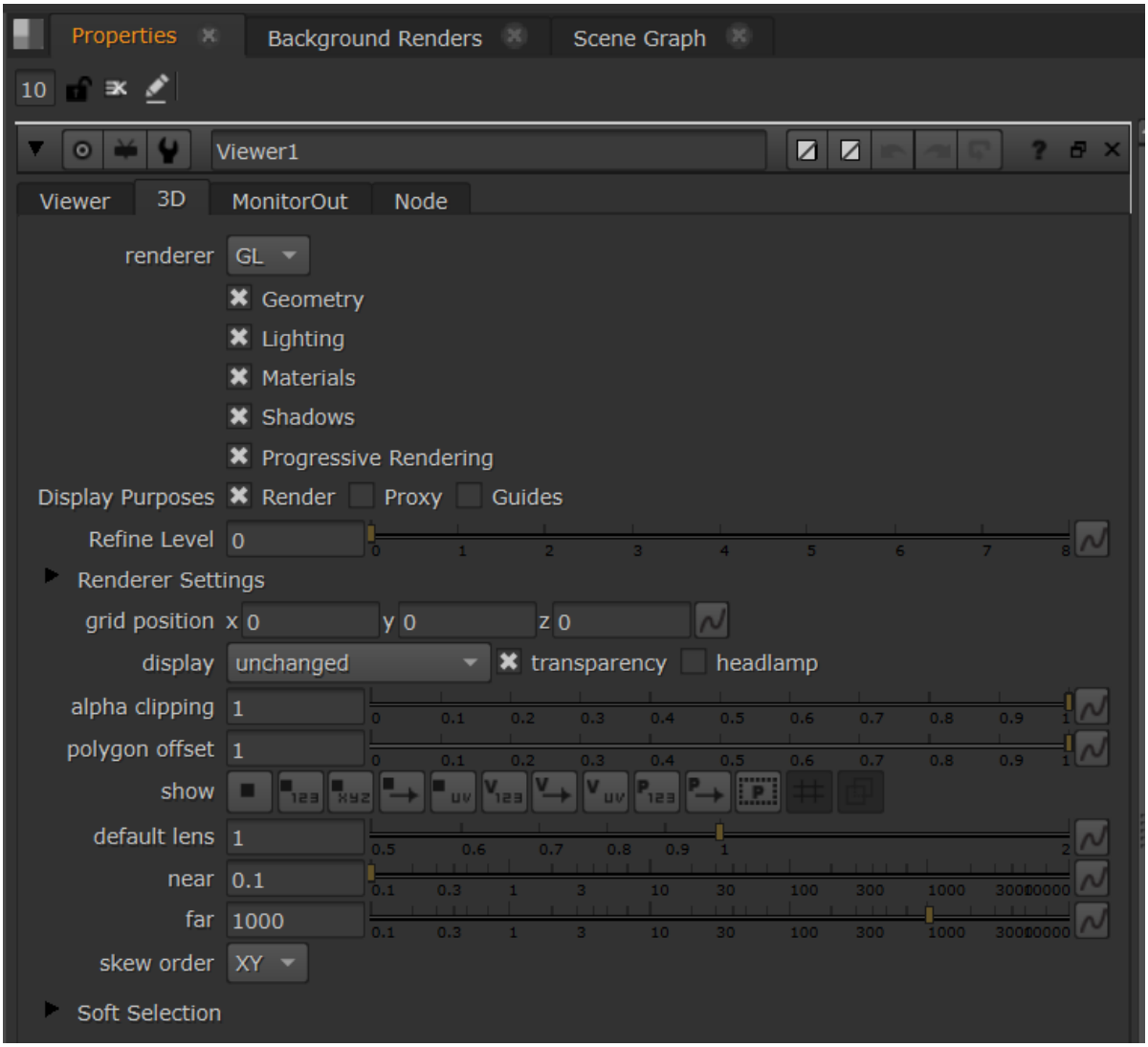
Nuke Viewer | Hydra Viewer | Scanline Render

Nuke's **New 3D System** uses a Hydra renderer, and nodes in the **Classic 3D System** use a "Nuke" renderer.

As such, the Viewer's **renderer** knob only has one option: **GL**, which is both the Hydra and Nuke renderers, depending on if you are viewing New or Classic 3D nodes.

Note: Learn more about the New 3D System and USD at 3D Compositing with the New 3D System.

You can also select whether or not to display geometry, lights, materials and/or shadows in the 3D Viewer by toggling the relevant checkboxes.



The Viewer renderer option is also included in the **Preferences** under **Panels** > **Viewer** > **default renderer**. Changing the preference only affects new Viewer nodes.

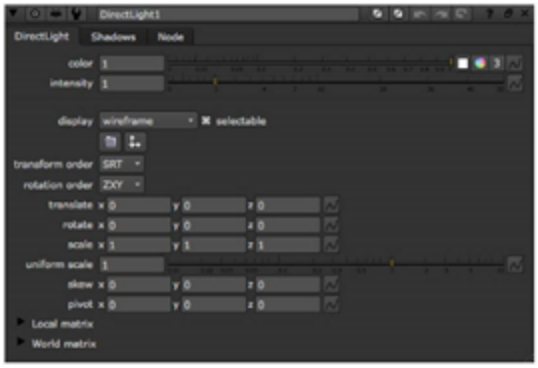
Working with Lights

There are several types of lights that you can add and manipulate in your Classic 3D scene to create the required effect.

Inserting Direct Lights

A direct light is a light that emits parallel light in one direction. It appears to illuminate all objects with equal intensity, as if it was coming from a far away source. Being at an infinite distance from the objects, direct light has orientation, but no position. A real world example of a direct light is the sun. You can use direct light to simulate sunlight and moonlight, for example.

1. Select **3D** > **Lights** > **Direct** to insert a DirectLight node in your script.
2. Connect the DirectLight node to the Scene node.
3. In the DirectLight node's controls, adjust the following:
   - Drag the **color** slider to change the light color.
   - Drag the **intensity** slider to change the brightness of the light.
   - To control the direction of the light, enter values in the **rotate** fields.



   - To adjust the settings for shadows, change values for the controls on the **Shadows** tab. For more information on these controls, see Casting Shadows.

Inserting Point Lights

A point light is a point in 3D space that emits light in every direction. A real world example of a point light is a light bulb. You can use point light to simulate light bulbs, lamps, and candles, for example.

1. Select **3D** > **Lights** > **Point** to insert a Point node in your script.
2. Connect the Point node to the Scene node.
3. In the Point node's controls, adjust the following:
   - Drag the **color** slider to change the light color.
   - Drag the **intensity** slider to change the brightness of the light.
   - To control how much light the object gets from the light source (based on the distance between the object and the light source), use the **falloff type** dropdown menu. A **Linear** type diminishes the light at

a fixed rate as it travels from the object, whereas **Quadratic** and **Cubic** types diminish the light at an exponential rate. If you select **No Falloff**, the distance between the light source and the object does not affect the lighting.

- To control the position of the light in the 3D space, enter values in the **translate** fields.
- To adjust the settings for shadows, change values for the controls on the **Shadows** tab. Note, however, that the Point light doesn't cast shadows if you're using ScanlineRender. For more information on these controls, see Casting Shadows.

Inserting Spot Lights

A spot light is a point in 3D space that emits a cone-shaped light in a given direction. A real world example of a spot light is a desk lamp.

1. Select **3D** > **Lights** > **Spot** to insert a Spotlight node in your script.
2. In the node's controls, adjust the following:
   - Drag the **color** slider to change the light color.
   - Drag the **intensity** slider to change the brightness of the light.
   - Drag the **cone angle** slider to control the spread of the light (how wide or narrow the beam is) in degrees from 0 to 180.
   - Drag the **cone penumbra angle** slider to control the softness along the edge of the area of illumination. A negative value fades inward from the circle's edge. A positive value fades outward from the circle's edge. The **cone falloff** should be set to zero or a low value in order to see the softness. This feature is only visible in the rendered objects and not in the 3D OpenGL Viewer.
   - Drag the **cone falloff** slider to control how concentrated the light is (that is, how much the light diminishes from the center of the circular region out to the edge). The higher the value, the more focused the light becomes. The falloff is independent of the **falloff type**.
   - To control how much light the object gets from the light source (based on the distance between the object and the light source), use the **falloff type** dropdown menu. A **Linear** type diminishes the light at a fixed rate as it travels from the object, whereas **Quadratic** and **Cubic** types diminish the light at an exponential rate. If you select **No Falloff**, the distance between the light source and the object does not affect the lighting.
   - To control the direction of the light, enter values in the **rotate** fields.
   - To control the position of the light in the 3D space, enter values in the **translate** fields.
   - To adjust the settings for shadows, change values for the controls on the **Shadows** tab. For more information on these controls, see Casting Shadows.

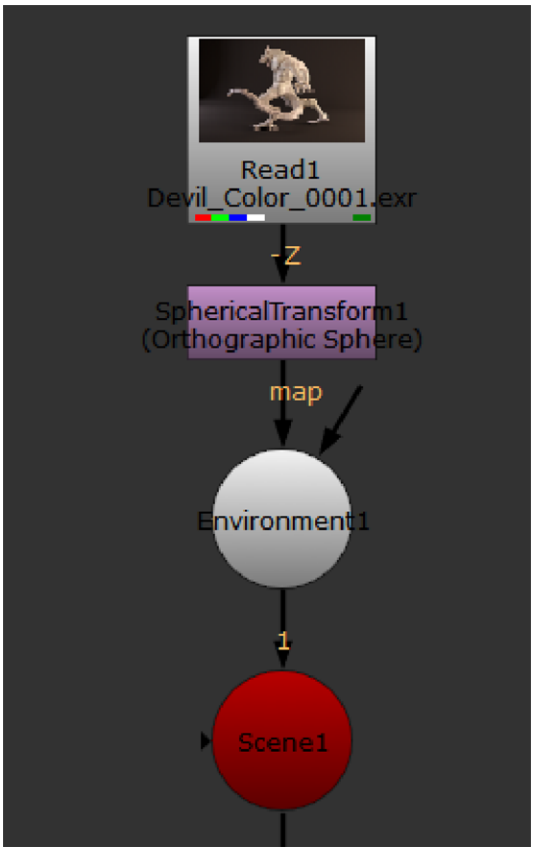
Inserting Environment Lights

An environment light is a light that illuminates the objects using an image of light from a real-world environment. This image-based lighting is generated using High Dynamic Range Images (HDRI). When HDR images are created, several differently exposed images are combined to produce a single image of the surrounding environment. As a result, HDR images have a wide range of values between light and dark areas, and represent the lighting conditions of the real world more accurately.

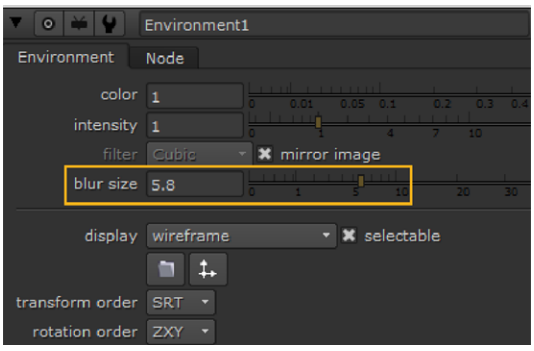
To use environment light, you first need to shoot a real life environment as an HDR image. Using the SphericalTransform node, you then convert this image into a spherical mapped image. The sphere is used to surround the 3D objects, so that the mapped image color illuminates them.

Environment light only works with shiny object materials that can reflect the mapped image. It results in a very realistic lighting that makes it easier to integrate the objects into the environment.

1. Read an HDR image of the environment into your script.
2. Select **Transform** > **SphericalTransform** to insert a SphericalTransform node after the HDR image. You use this node to convert the HDR image into a spherical mapped image. In the node's controls, select the **Input Type** and the **Output Type** (in this case, **Sphere**).
3. Select **3D** > **Lights** > **Environment** to insert an Environment node in your script. Connect the SphericalTransform node to the Environment node's **map** input, and the Environment node to the Scene node.



4. In the Environment node's controls, adjust the following:
   - Drag the **color** slider to change the light color.
   - Drag the **intensity** slider to change the brightness of the light.
   - From the **filter** dropdown menu, select a filtering algorithm for the map image. For more information, see Choosing a Filtering Algorithm.
   - To change the blur size of the map image, adjust the **blur size** slider.

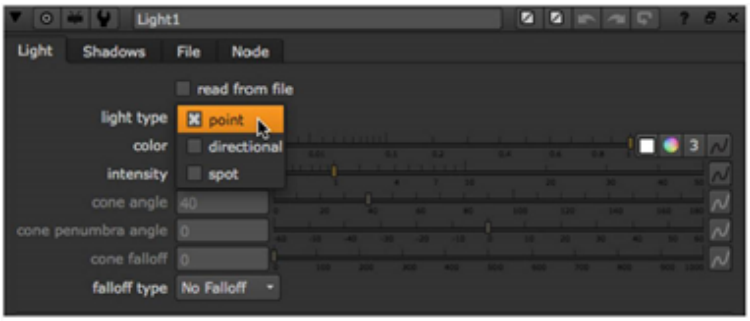


Inserting Direct Lights, Point Lights, or Spot Lights

The Light node includes the DirectLight, Point, and Spotlight nodes, so you can set it to act as any of these three nodes. The advantage of using a Light node in this way is that if you want to change the light type later, you can do so without setting up a new node. For example, you might insert a direct light, but then realize that what you actually need is a spot light. If you inserted the direct light using a DirectLight node, you need to delete this node and insert a SpotLight node instead. However, if you inserted the direct light using a Light node, you can simply change the light **type** from **directional** to **spot** in the Light controls.

Tip: The node can also be used to import lights from **.fbx** files. This is described under Importing Lights from FBX Files.

1. Select **3D** > **Lights** > **Light** to insert a Light node into your script.
2. In the Light controls, select the **light type** you want to use: **point**, **directional**, or **spot**. The controls are enabled and disabled according to the light type you select. For example, if you chose directional light, you get the same controls that appear on the DirectLight node.



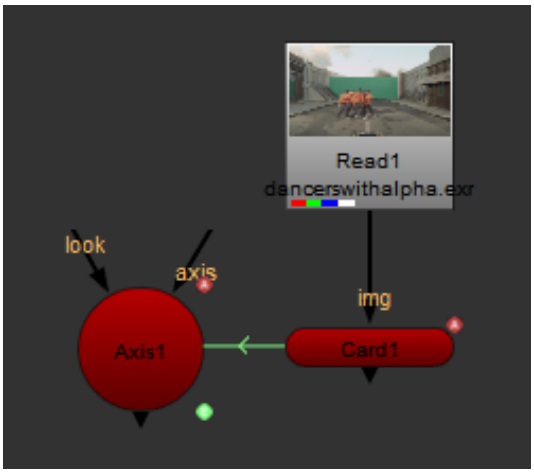
3. Adjust the controls as necessary. For information on the functions of the controls, refer to the following:
   - If you selected point as the light type, see Inserting Point Lights.
   - If you selected directional as the light type, see Working with Lights.
   - If you selected spot as the light type, see Inserting Spot Lights.

Using the Look Input

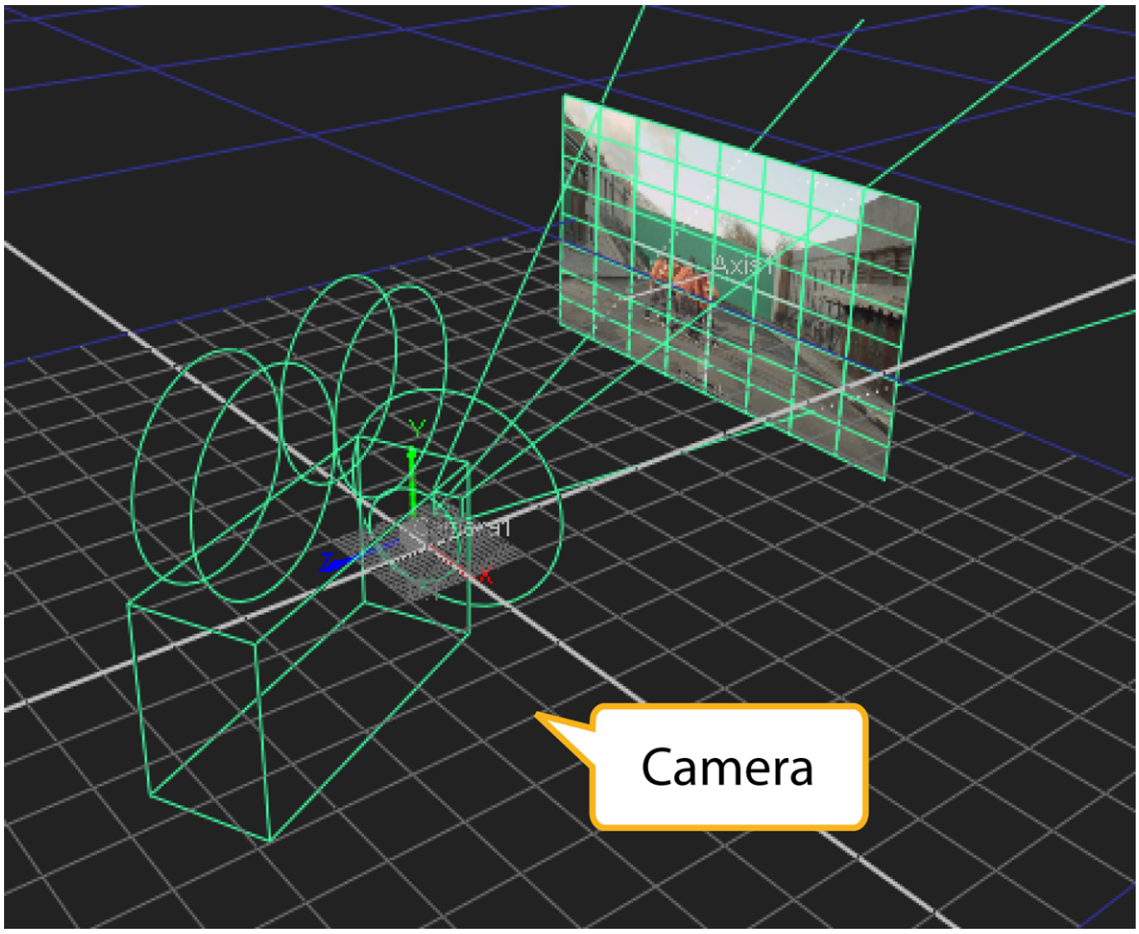
You can use the optional **look** input of the Light node, so that the light automatically rotates to face towards the connected input. You can attach a Camera, Light, or Axis node to the **look** input. For example, you can connect an Axis node to the **look** input so that the light rotates to face the axis, wherever it is moved.

If you animate a card to move along the x axis, you can attach a Camera and a Light node with the **look** inputs so that they automatically rotate and face the card as it moves. To do this, complete the following steps:

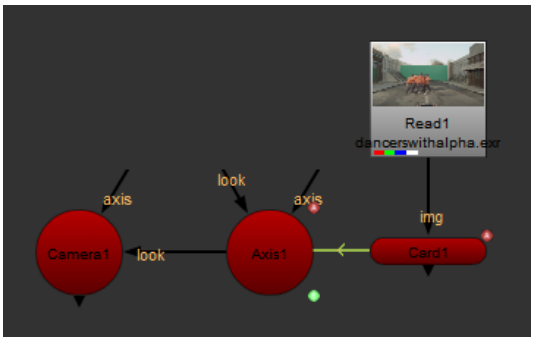
1. After animating your card, insert an Axis node.
2. Expression link the Axis node to the Card node by holding **Ctrl**, clicking on the **translate** animation button in the Card node's properties, and dragging it to the **translate** animation button in the Axis node's properties. The expression link is displayed as a green line with an arrow denoting the direction of the expression. See Linking Expressions for more information.



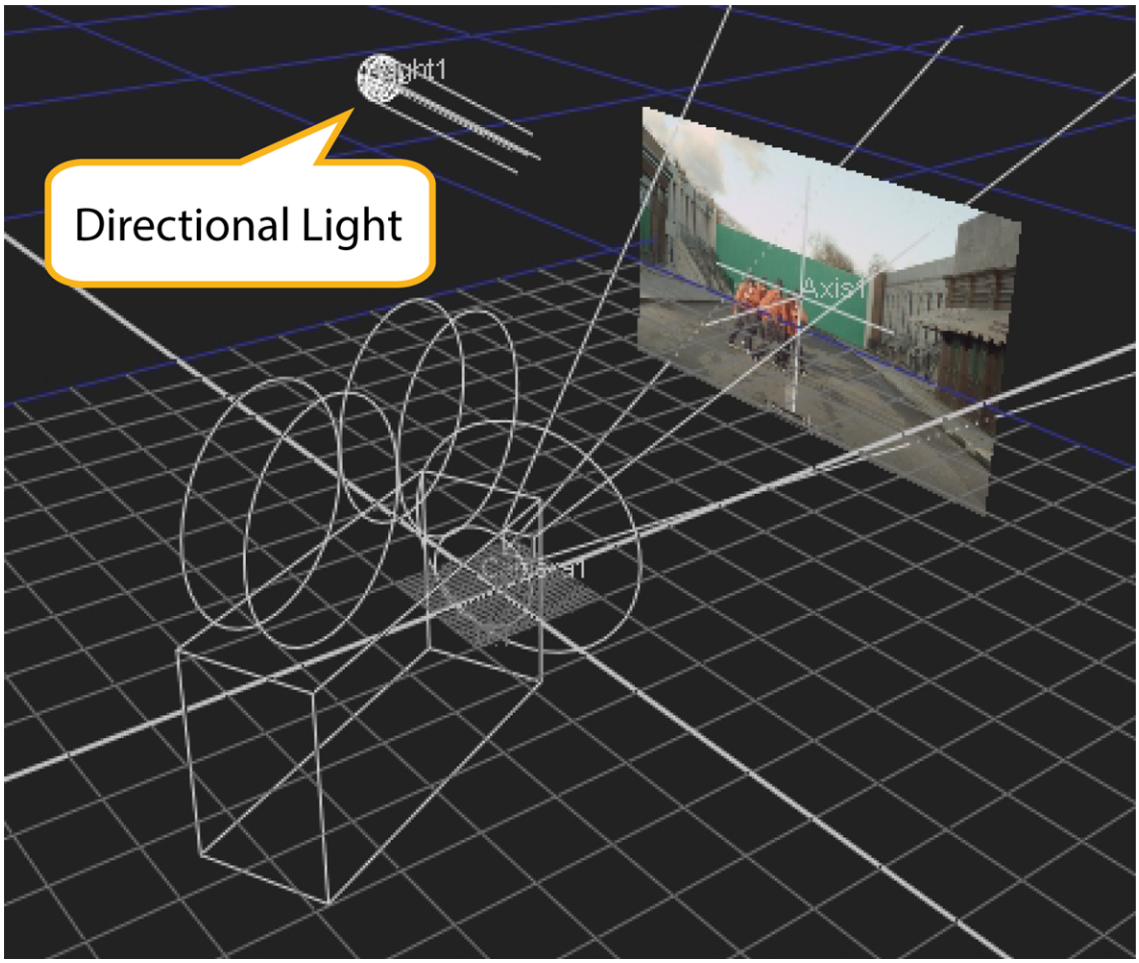
3. Insert a Camera node and either drag the camera to the required position in the Viewer, or use the Camera node properties to adjust the position of the camera.



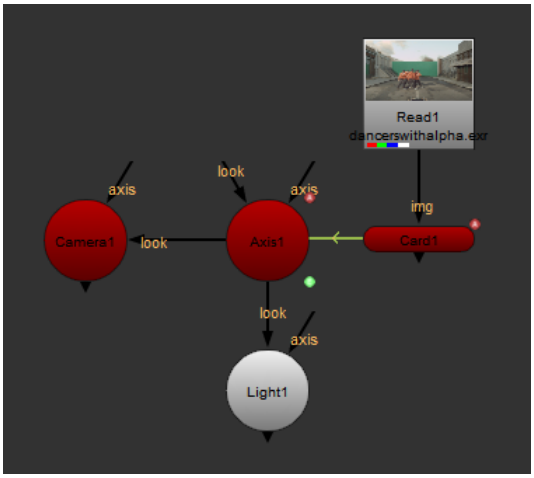
4. Connect the Camera's **look** input to the Axis node.



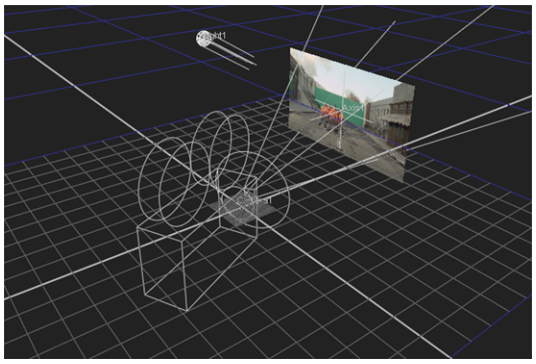
5. Insert a Light node and drag the light to required position in the Viewer, or use the Light node properties to adjust the position of the camera.

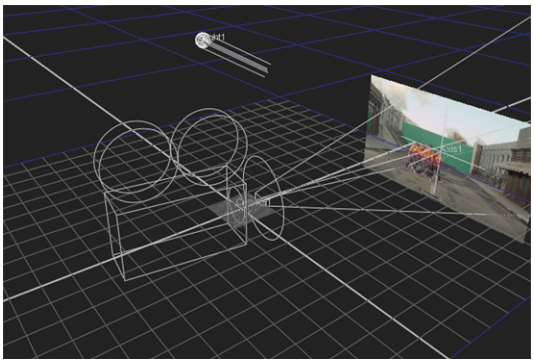


6. Connect the Light node's **look** input to the Axis node



7. Playback your animated card and notice that the Camera and Light follow the animated card.





Importing Lights from FBX Files

FBX is a standard 3D file format that gives you access to 3D scenes created in other applications supporting the same format. What you generally have in an **.fbx** file is an entire 3D scene containing cameras, lights, meshes, non-uniform rational B-spline (NURBS) curves, transformation, materials, and so on. From this scene, you can extract cameras, lights, transforms, and meshes into Nuke. This way, you can, for example, create a light in Maya, export it in a **.fbx** file, and use the same light again in Nuke.

Note: For the FBX SDK version used in Nuke, see Third-Party Library Versions.

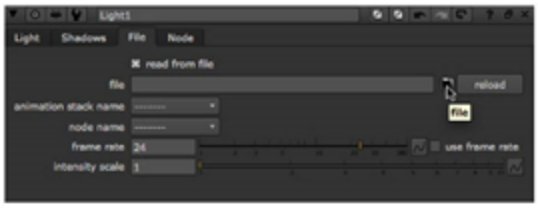
Tip: If you have trouble with **.fbx** files, it may be because they were written with an older version of FBX. If they load very slowly, it is also possible that they are ASCII rather than binary. To get around these problems, you can use the FBX converter on the Autodesk website (http://usa.autodesk.com/fbx/download/). It converts between various different formats, including older FBX versions, ASCII, and binary, and is available on Windows, Mac, and Linux.

You can use the Light node to read in directional, point, and spot lights from FBX scene files (for more information on these three light types, refer to Lighting). One Light node only reads in one light. Therefore, if your **.fbx** file contains three lights and you want to import all of them into Nuke, you need to use three Light nodes.

To import a light from an .fbx file:

1. Select **3D** > **Lights** > **Light** to insert a Light node in the place where you want to add the light in your script.
2. In the Light controls, check **read from file**. This enables the controls on the **File** tab, allowing you to read in lights from an **.fbx** file. It also disables all controls whose values are filled from the **.fbx** file. You can still view these values and use them in expressions, but you cannot modify them, because they are read from the **.fbx** file. Any changes you make in the **.fbx** file are reflected in these values of the Light node.

3. On the **File** tab, click the folder icon and browse to the **.fbx** file that contains the light you want to use. Click **Open**.



4. From the **animation stack** dropdown menu, select the take you want to use from the **.fbx** file. FBX files support multiple takes in the same file. One of the takes is usually a default take without any animation.
5. From the **node name** dropdown menu, select the light node you want to import from the **.fbx** file.
6. If you want to override the frame rate used in the **.fbx** file to sample the animation curves, enter a new frame rate (frames per second) in the **frame rate** field. Check **use frame rate** to use the rate you entered (rather than the one in the **.fbx** file).
7. To scale the intensity channel values read from the **.fbx** file, adjust the **intensity scale** slider. If the light is too dark, increase this value.
8. If you want to modify the light properties imported from the **.fbx** file, uncheck **read from file** on the **Light** tab and make the necessary modifications. As long as **read from file** is unchecked, your changes are kept.
9. To reload the light properties from the **.fbx** file, make sure **read from file** is checked and click the **reload** button on the **File** tab.

Casting Shadows

The Light, Point, DirectLight, and Spotlight nodes all have controls that you can use to adjust how the lights cast shadows in your 3D scene. The following geometry nodes also have controls that allow you to select whether they receive shadows cast by the lights and whether they themselves cast shadows on other objects:

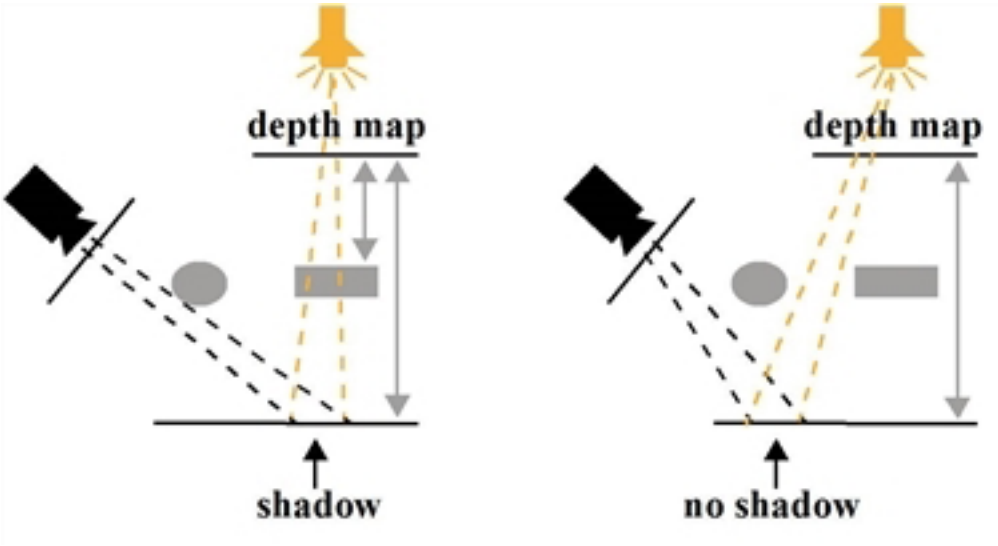
- Card
- Cube
- Cylinder
- Sphere
- MergeGeo
- ModelBuilder
- PointCloudGenerator
- PositionToPoints

- ReadGeo
- Scene

Note: The method used to create shadows varies between different render nodes:

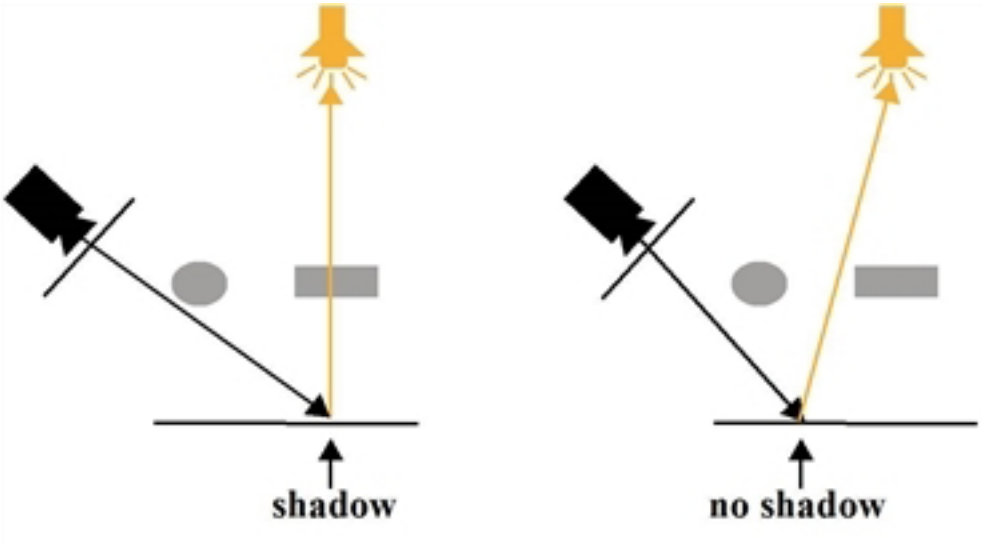
- **ScanlineRender uses depth mapping to create shadows.** It first renders a depth map for each light that casts a shadow. The depth map is rendered from the light's point of view, and each pixel in the depth map represents the distance from the light to the nearest surface the light illuminates in a specific direction. The depth map is then compared to the render from the point of view of the camera. If a point is farther away in the image that the camera sees than in the depth map, then that point is considered to be in shadow.

  Depth map shadows are often faster to render than raytraced shadows, but may not appear as realistic.



- **PrmanRender creates shadows through raytracing.** It fires individual light rays from the camera into the scene for each pixel. When a light ray hits a surface in the scene, PrmanRender traces so called shadow rays between the intersection point and every light source in the scene. If there are obstacles between the intersection point and the light sources, the intersection point is considered to be in shadow.

  The advantage of raytracing is that it can be used to create more accurate shadows and shadows that have soft edges, much like those in the real world. However, compared to creating shadows using depth mapping, raytracing can take much longer to render.



To Cast Shadows:

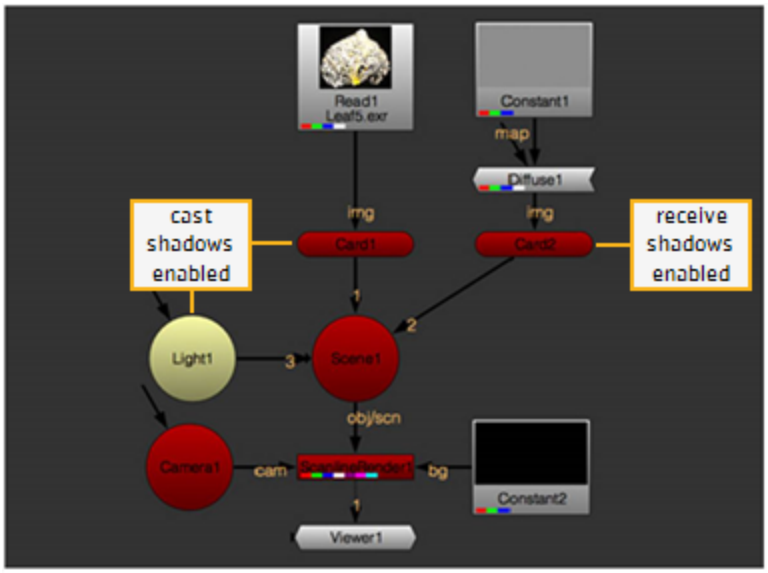
1. Open the properties panels for the 3D objects in your scene.
2. Check the **cast shadow** or **receive shadow** box, or both.

   Cast shadow tells objects in a light's path to cast shadows onto other objects in the scene.

   Receive shadow tells objects to display shadows cast by other objects in the scene. This is useful for showing occlusion in a scene.

   You can also use a Scene node to define these settings for all 3D objects connected to it. In the Scene properties, set **shadow** to **override inputs** and use the **cast shadow** and **receive shadow** controls to override the corresponding controls on the individual 3D objects.
3. Make sure you've got a Shader node attached to any 3D objects that you want to receive shadows.
4. Attach a light to your scene, and check the **cast shadows** box on the **Shadows** tab.

   In our example node tree below, we are using a ScanlineRender node, but you could use a PrmanRender node instead.



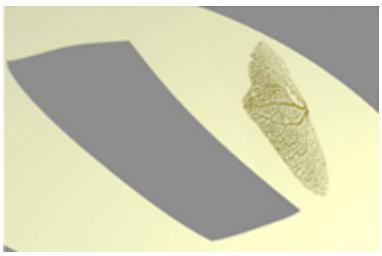
Note: If you are using a point light, casting shadows is not supported in the ScanlineRender node.

5. Depending on the render node you are using, proceed to either:
   - Adjusting Shadows When Using ScanlineRender below, or
   - Adjusting Shadows When Using PrmanRender.

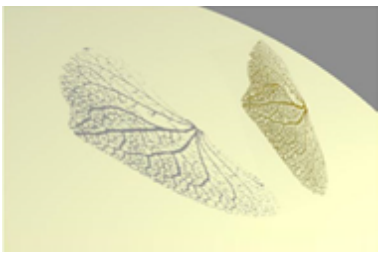
Adjusting Shadows When Using ScanlineRender

1. On the **Shadows** tab of the light node properties, set **shadow mode** to:
   - **solid** - Objects seen from the light that cast shadows are considered completely solid.
   - **clipped alpha** - objects that cast shadows are considered to be transparent if the object's alpha is below the light's clipping threshold control. All other alpha values fully occlude the light.
   - **full alpha** - Shadows are calculated based on how light is reduced when it passes through non-opaque occluders.

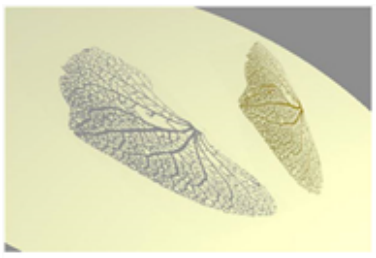
   This affects shadows cast by objects, based on the objects' opacity.



Solid. (The shadow is



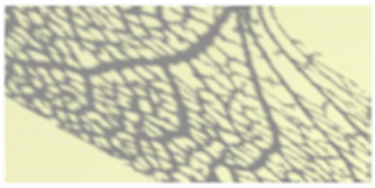
Clipped alpha.



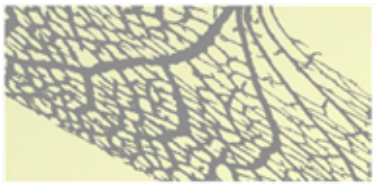
Full alpha.

rectangular because the leaf image is applied on a Card object.)

2. If you want the light node to to output the shadow map, set **output mask** to the channel where you want to store this. You can enable this even if **cast shadows** is disabled.
3. If you set **shadow mode** to **solid** or **clipped alpha**, you can adjust the following:
   - **depthmap resolution** - this sets the resolution of the depth map. Larger values result in shadows with less crunchy edges and less artifacts, but require more time to process.



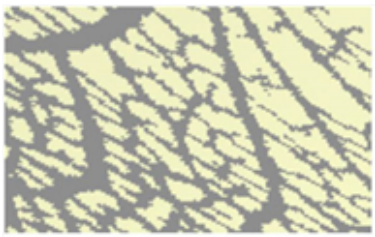
Depthmap resolution set to 800.



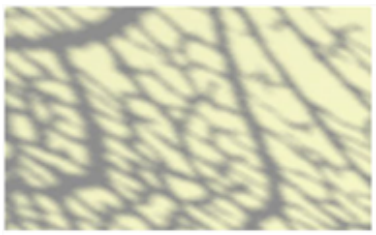
Depthmap resolution set to 7400.

Note: You can also fix crunchy edges by increasing the number of **samples** instead of increasing the depth map resolution.

   - **samples** - this sets the number of samples for the light when generating soft shadows. If soft shadows in your scene appear dotty or noisy, try increasing this value. The higher the value, the smoother the soft the shadows become.



Samples set to 1.

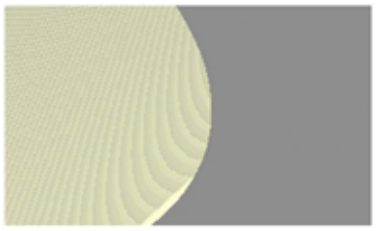


Samples set to 50.

   - **jitter scale** - the amount of jitter used when doing percentage-closer filtering (PCF) for soft shadows. A larger **jitter scale** value results in softer, more perceptually accurate shadows.

     PCF works by sampling the depth map at many different positions around the same spot. The final shadow value for that spot is an average of how many of the samples were occluded or visible from point of view of the light.
   - **bias** - this is a constant offset that moves the surface sample point away from surface, towards the light that is casting the shadow. If self-shadowing artifacts appear in the image, you may want to increase

this value. Note, however, that if you increase the value too much, some shadows may start moving away from the base of the objects that cast them.



Shadow artifacts.



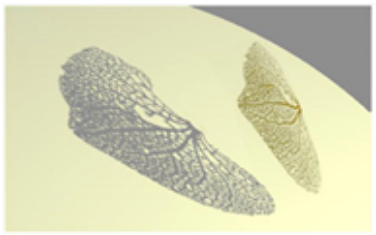
Increasing **bias** can help reduce artifacts.

- **slope bias** - this is like **bias**, but the offset is proportional to the slope of the depth map. This allows you to give a different offset to each value in the depth map, depending on the surface's slope relative to the light. For example, if the surface's slope relative to the light is shallow, the value in the depth map may be a correct approximation and no offset (or only a very small offset) is needed. If the surface's slope relative to the light is steep, the values in the depth map are less likely to be a correct approximation and a larger offset is needed.

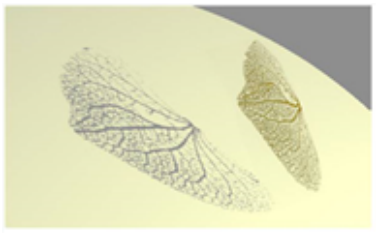
  If increasing **bias** reduced the existing self-shadowing artifacts but introduced more artifacts in other areas of the image, you may want to bring **bias** down a little and increase **slope bias** instead. Then, tweak both values until you're happy with the result.

In the **clipped alpha** mode, you can also adjust:

- **filter** - the type of filter to use. For more information on the available filtering algorithms, see Choosing a Filtering Algorithm.
- **clipping threshold** - any surface samples with alpha values below this threshold are considered transparent. The higher the value, the more areas on objects that cast shadows are considered transparent and pass through some light.



Clipping threshold set to 0.1.



Clipping threshold set to 0.9.

4. If you set **shadow mode** to **full alpha**, you can adjust the following:
   - **filter** - the type of filter to use. For more information on the available filtering algorithms, see Choosing a Filtering Algorithm.
   - **scene epsilon** - an offset that moves the sampling point away from the geometry surface, towards the light that is casting the shadow. Increasing this value can reduce self-shadowing artifacts.

Tip: To generate accurate shadows from a Direct light, view the scene through the light (using the Viewer's camera dropdown menu) and adjust the Direct light's **scale** control so that the part of the scene that should cast shadows fits within the view. This ensures that none of the shadow-casting geometry is missed by the depth map.

Adjusting Shadows When Using PrmanRender

1. Open the PrmanRender properties and make sure **shadows** is enabled.
2. Open the light node properties and go to the **Shadows** tab.
3. Make sure **shadow mode** is set to **solid**. The other two modes aren't relevant when using PrmanRender.
4. If you want to cast soft shadows from the light, increase **sample width**. This value multiplies the width of the soft area around the egde of a shadow. The higher the value, the larger the soft area.

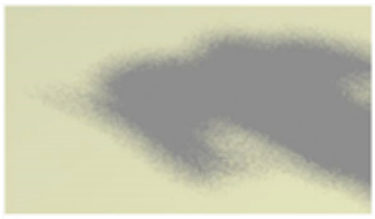


Sample width set to 1.



Sample width set to 15.

5. If you increased **sample width** in the previous step and the resulting soft shadows in your scene appear dotty or noisy, increase **samples**. This sets the number of samples for the light when generating soft shadows. The higher the value, the smoother the soft shadows become.

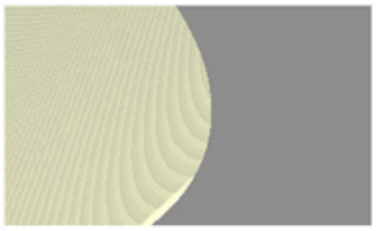


Samples set to 10.



Samples set to 80.

6. If self-shadowing artifacts appear in the image, increase the **bias** value. This moves the surface sample point away from surface. Note, however, that if you increase the value too much, some shadows may start moving away from the base of the objects that cast them.



Shadow artifacts.



Increasing **bias** can help reduce artifacts.

Manipulating Object Normals

Object normals are vectors that are perpendicular to the surface. They are used in lighting calculations to determine how the light should bounce off a surface at any particular point. By manipulating them, you can control the diffuse and specular light contributions.

To Manipulate Object Normals

1. Select **3D** > **Modify** > **Normals** to insert a Normals node anywhere after the 3D object whose lighting you want to adjust.
2. Connect a Camera, Axis, or light node to the Normals node's **lookat** input.
3. In the Normals controls, open the **action** dropdown menu and select:
   - **unchanged** to make no changes.
   - **set** to assign the normals value to the normal x, y, and z fields.
   - **build** to rebuild each normal based on the surrounding vertices. Adjust the **threshold angle** slider to determine the break angle where two faces no longer constitute a smooth surface. An angle of 0 means all faces are flat, whereas 180 means all faces are smooth. A good average setting is 60.
   - **lookat** to point all normals towards the Normals node's **lookat** input.
   - **delete** to remove the named attribute from the object. For example, if you remove the N attribute, the object has no normals.

Relighting a 2D Image Using 3D Lights

The Relight node takes a 2D image containing normal and point position passes and lets you relight it using 3D point lights. Essentially bypassing the need to return to a 3D application and re-render the lighting, Relight provides a quick and interactive way to relight a 3D scene in a 2D environment.

Relight works by applying a 3D shader to a 2D image using the normal and point position passes stored in separate image channels, and lets you attach and manipulate a 3D point light (or multiple lights via a Scene node).

Note: Relight only works with Light nodes set to **light type** > **point**.

To Relight a 2D Image Using Relight

1. From the **3D** > **Lights** menu, select **Relight** to add the node to your script.
2. Read in a 2D image containing normal and point position passes and connect it to the **color** input of the node.

Tip: If the position pass and normal vectors are contained in separate images, they can be combined using a Shuffle node, which connects to Relight through the **color** input.

   You can create normal and point position passes using the DepthGenerator node in NukeX, for example. See Generating Depth Maps.
3. In the Relight properties panel, select the channel containing the normal data from the **normal vectors** dropdown menu.
4. Select the channel containing the point position information from the **point positions** dropdown menu.



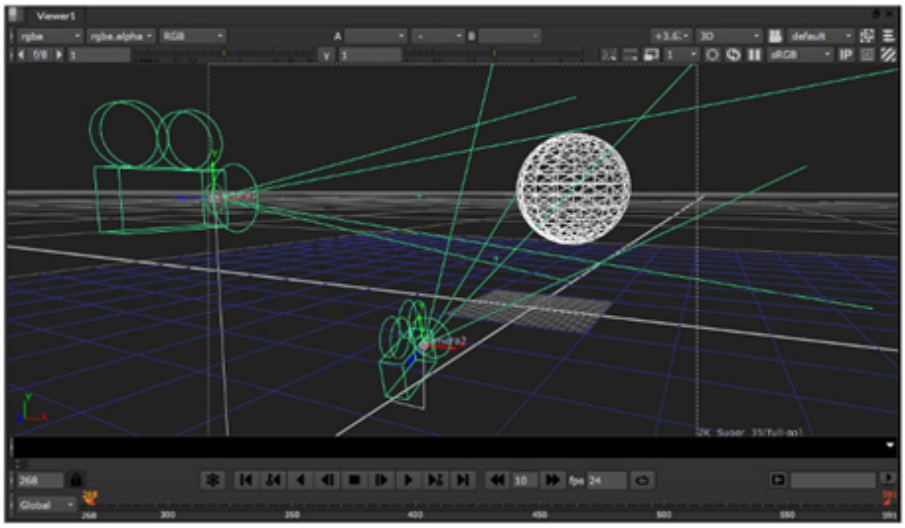
5. Connect a Light node with **light type** > **point** to the **lights** input, or multiple lights via a Scene node.
6. Connect the camera that was used to render the original scene to the **cam** input.
7. Attach a shader (a Phong node, for example) to the **material** input. Depending on the type of shader you attach, ensure that you've defined the necessary properties for it. For information on defining material properties, refer to Object Material Properties.

Note: The camera input does not appear until the light inputs have been connected to a light or scene node, and the material inputs do not appear until the camera input has been connected.

8. Switch to the 3D Viewer and position your lights.
9. If the image in the **color** input contains an alpha channel and you'd like to use this as a mask to limit the effects of Relight, check **use alpha**.
10. If required, adjust the **ambient** slider to set the global ambient light level for the scene.
11. To combine the lighting information from the Relight node with the original 2D image, use a Merge node with **operation** set to **multiply**. Connect the Relight node to the Merge node's **A** input and your 2D image to its **B** input.

Cameras

Nuke's Classic 3D System supports multiple cameras in a scene, with each providing a unique perspective.



Cameras in a scene.

For details on how to add a camera, look through it, and edit its lens characteristics, see Working with Cameras.

In addition to using cameras to view and render 3D scenes, you can also set up cameras that project 2D images onto geometry in your scene. For more information, see Projection Cameras.

If you have created a camera in a third-party 3D application (such as Maya) and want to use it in Nuke, you can export it from your 3D application in the FBX or Alembic format and then import it into Nuke. See Importing Cameras from FBX Files and Importing Cameras from Alembic Files.

If your camera was created in Boujou, you can also use the **import_boujou.tcl** script to import it into Nuke. See Importing Cameras from Boujou.

You can extrapolate a 3D point's position from imported camera data using the PointsTo3D node, which can help you position geometry in a scene. See <xref> for more information.

Tip: NukeX and Nuke Studio include CameraTracker which creates animated cameras without resorting to third-party applications. See Camera Tracking for more information.

Working with Cameras

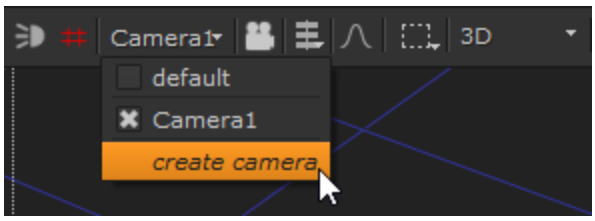
This section explains how to add a camera to your script, look through it, lock it, and edit its lens characteristics.

To Add a Camera

1. Click **3D > Camera** to insert a Camera node.

   OR

   In the 3D Viewer, select **create camera** to place a new camera at the current position and orientation in 3D space.



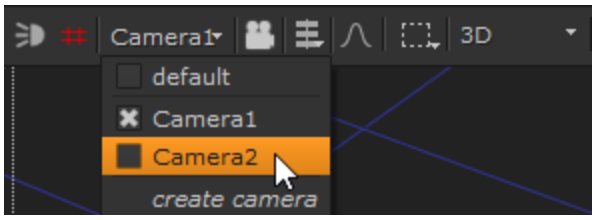
2. To setup the rendering camera, drag a connector from the new Camera node to the ScanlineRender node.

   OR

   To setup an additional scene camera for viewing, drag a connector from the new Camera node to the Scene node.

To Look Through a Camera

1. Press **V** to make sure you are looking through the 3D perspective view, and not one of the orthographic views.
2. From the 3D Viewer window, select the camera from the dropdown menu in the top right corner.



Note: This selection does not change the camera used for rendering. This changes only the camera to "look through" for the current 3D Viewer.

Cameras in the current data stream automatically appear in the dropdown menu of cameras you can select. To select a camera that doesn't appear in menu, double-click the Camera node to open its panel and add it to the menu.

To Lock the 3D Camera View

You can lock the 3D view to the selected camera or light. You can toggle between the unlocked and locked modes by clicking the **3D view lock** button, or by pressing **Ctrl/Cmd**+**L**.

- **unlocked** - move freely in the 3D view without restrictions. The 3D view lock button is gray.
- locked - lock your movement to the camera or light you've selected in the dropdown menu on the right side of the 3D view lock button. The 3D view lock button is red.

To Use the Interactive 3D Camera View Mode

With the interactive 3D camera view mode you can change the camera or light values according to your movement in the Viewer. You can activate the interactive mode by **Ctrl/Cmd**+clicking the **3D view lock** button. When the interactive mode is on, the **3D view lock** button turns green. In order to activate the interactive mode, you need to have a Camera or a Light node selected in the dropdown on the right side of the **3D view lock** button.

When the interactive mode is on, you can use the plus (**+**) and minus (**-**) keys to change the translate values of the camera or light you've selected. When the interactive mode is off, these keys zoom your view in and out.

To Edit a Camera's Lens Characteristics

1. If necessary, double-click on the Camera node to display its parameters.
2. Click the **Projection** tab.
3. Drag the **focal length** slider to adjust the camera's level of magnification.
4. Drag the **near** slide to edit the position of the camera's forward clipping plane. Objects closer to the camera than this plane are not rendered.

Note: The value for the **near** clipping plane must always be positive to produce a sensible result.

5. Drag the **far** slider to edit the position of the camera's rearward clipping plane. Objects farther from the camera than this plane are not rendered.

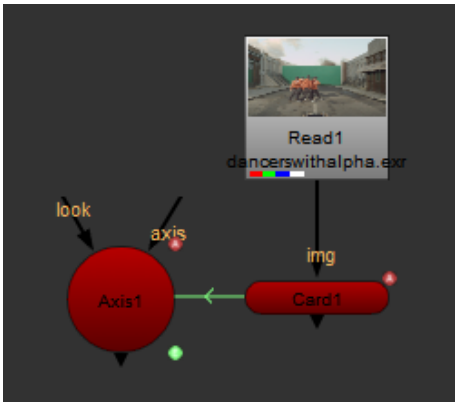
6. Increment the **window translate u** (horizontal axis) and **v** (vertical axis) sliders to translate the camera's output along either axis.
7. Increment the **window scale u** (horizontal axis) and **v** (vertical axis) sliders to scale the camera's output on either axis.
8. Drag the **window roll** slider to rotate the camera's output on z.

Using the Look Input

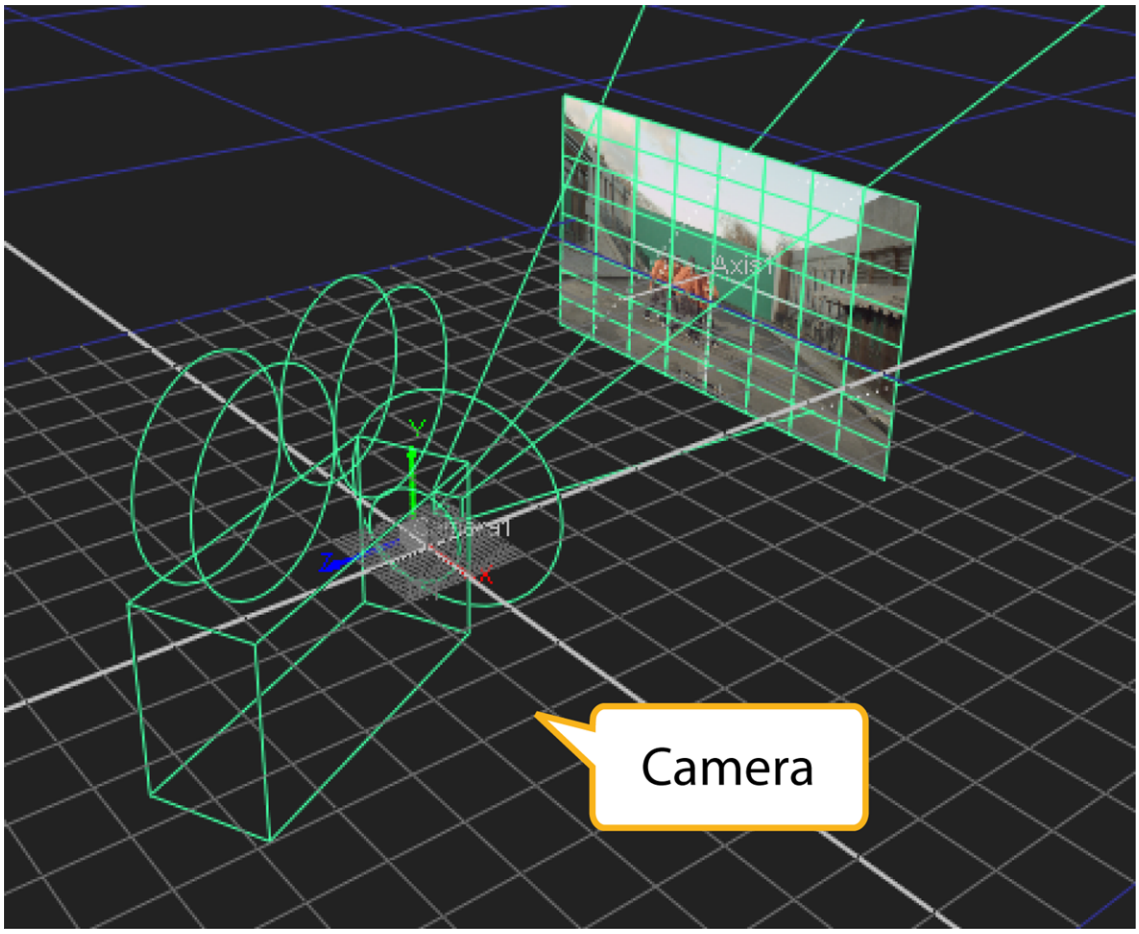
You can use the optional **look** input of the Camera node, so that the camera automatically rotates to face towards the connected input. You can attach a Camera, Light, or Axis node to the **look** input. For example, you can connect an Axis node to the **look** input so that the camera rotates to face the axis, wherever it is moved.

If you animate a card to move along the x axis, you can attach a Camera and a Light node with the **look** inputs so that they automatically rotate and face the card as it moves. To do this, complete the following steps:

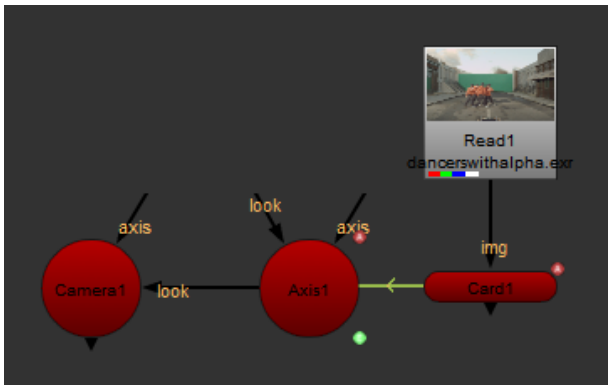
1. After animating your card, insert an Axis node.
2. Expression link the Axis node to the Card node by holding **Ctrl**, clicking on the **translate** animation button in the Card node's properties, and dragging it to the **translate** animation button in the Axis node's properties. The expression link is displayed as a green line with an arrow denoting the direction of the expression. See Linking Expressions for more information.



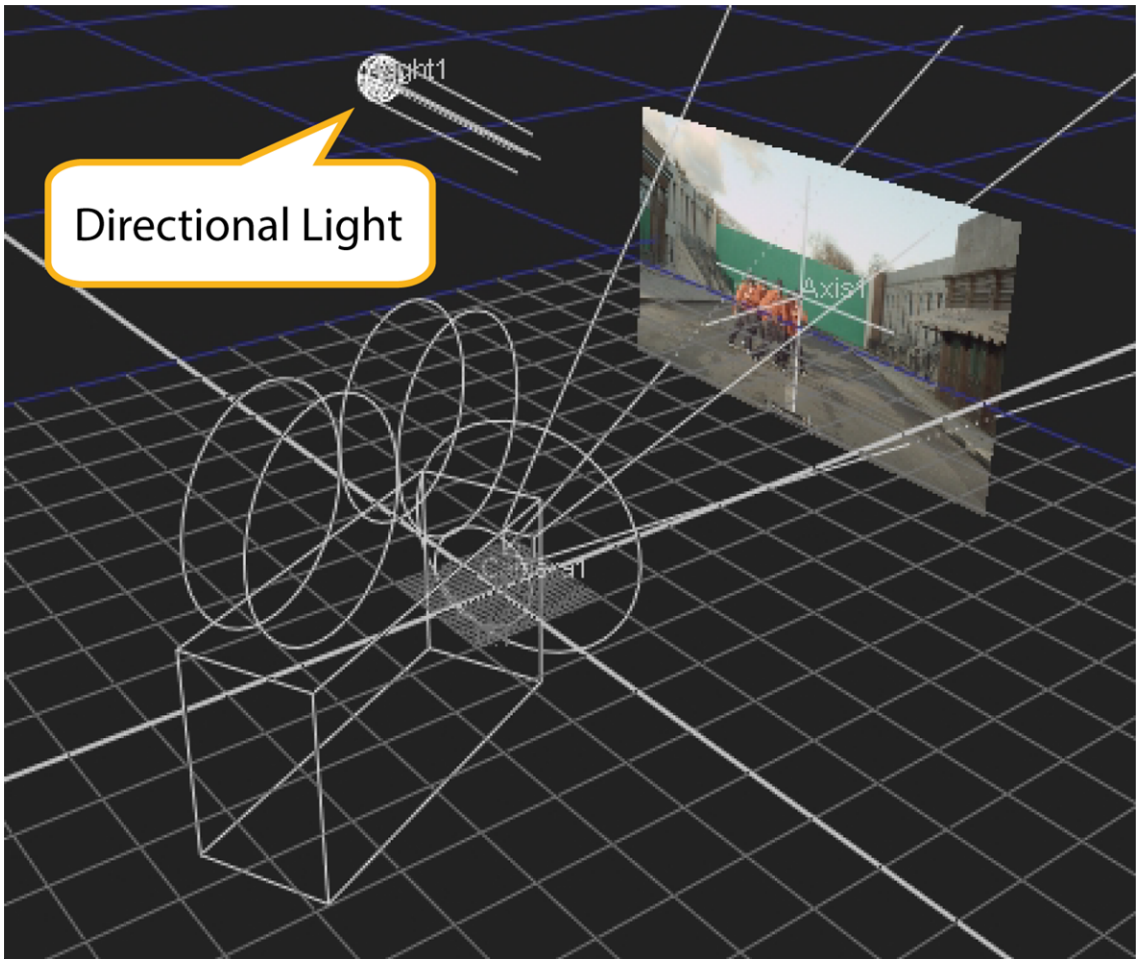
3. Insert a Camera node and either drag the camera to the required position in the Viewer, or use the Camera node properties to adjust the position of the camera.



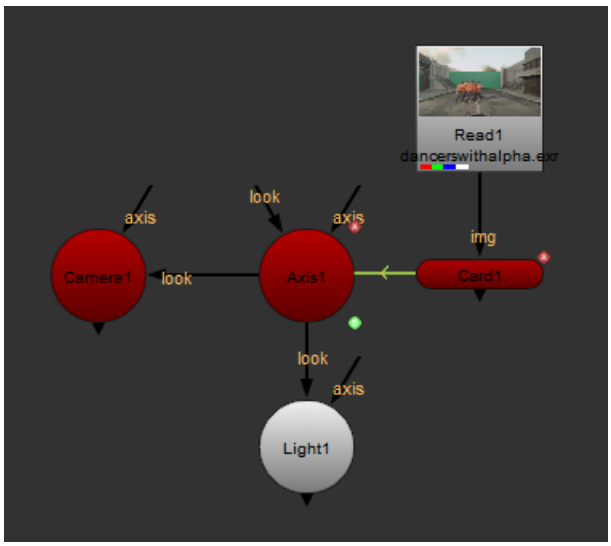
4. Connect the Camera's **look** input to the Axis node.



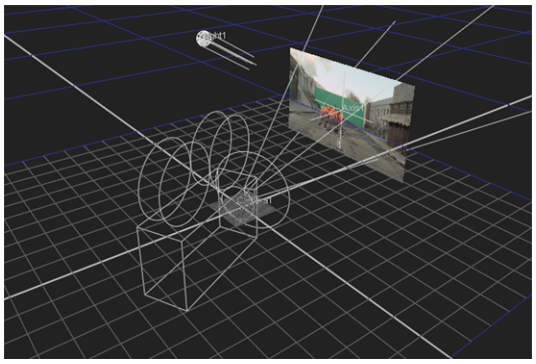
5. Insert a Light node and drag the light to required position in the Viewer, or use the Light node properties to adjust the position of the camera.

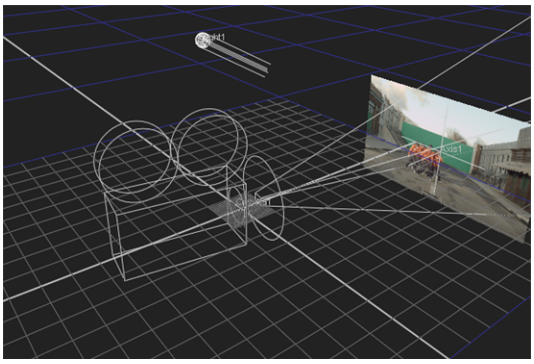


6. Connect the Light node's **look** input to the Axis node



7. Playback your animated card and notice that the Camera and Light follow the animated card.





Projection Cameras

In addition to viewing and rendering a 3D scene, cameras can also project a 2D still image or image sequence onto geometry in the scene. This is similar to the front-projection systems used in practical photography, where a background image or other element is projected onto the stage and photographed with other elements.

In Nuke, a projection camera can receive camera data tracked from the original shot - or another shot - to setup a projection that is match-moved to another source.

This setup requires these nodes: a projection camera, a Scene node, a Project3D node, a geometry object node (what you'll be projecting onto), and a 2D node with the image that you want to project.

First a Little Math...

When you create a projection camera, you need to gather some information and do a few small calculations to make sure the projection works. Here are the bits of information you need:

- Focal length of the lens that photographed the projection image.
- Resolution of scanned image.
- Scanner pitch of the film scanning device.

After you have this information, you need to do these calculations to get the horizontal and vertical aperture settings for the projection setup:

```
horiz. res. / scanner pitch = horizontal aperture
vertical res. / scanner pitch = vertical aperture
```

So, for example, if your image resolution is 720 x 486 and the scanner pitch is 20, then these are the results:

```
720 / 20 = horizontal aperture = 36
486 / 20 = vertical aperture = 24.3
```

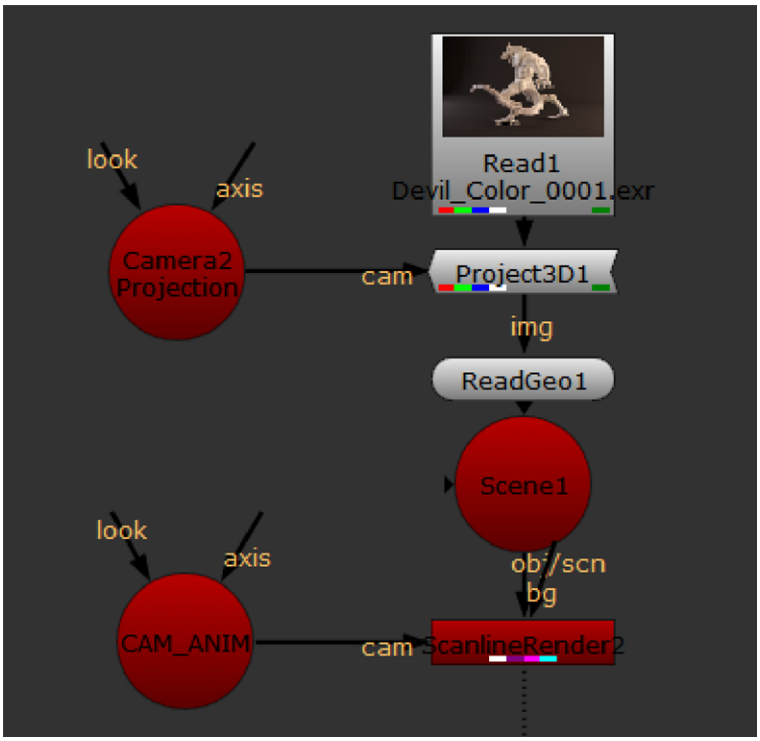
Generally, for most professional projects, you can get the lens focal length from the camera report for the shot(s). If that is not available, you may be able to extrapolate lens information by running the shot through a 3D tracking application, such as Boujou, Syntheyes, or RealViz.

Setting Up the Projection Camera Script

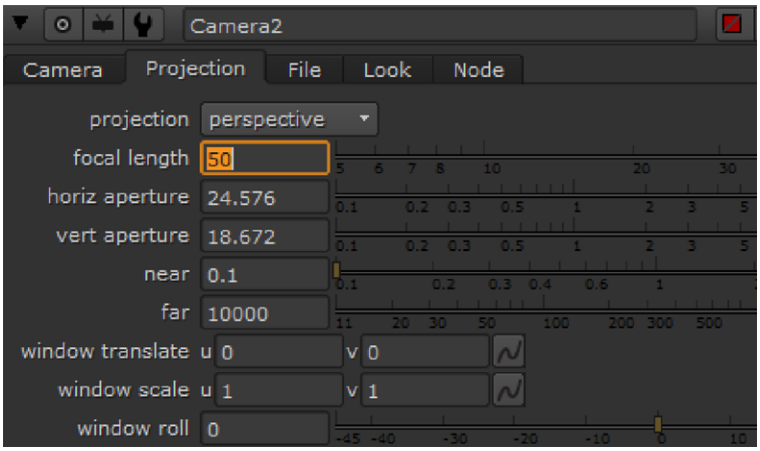
Once you have the horizontal and vertical aperture and the lens focal length for the image you want to project, you can complete the projection camera setup.

To Add a Projection Camera

1. Select **3D > Camera** to add a new camera to your script and rename the node to identify it as a projection camera.
2. Select **3D > Shader > Project3D** to add a Project3D node to the script.
3. Connect the 2D image (i.e., Read node) to the Project3D node.
4. Connect the projection camera to the Project3D node.
5. Connect the Project3D node to the geometry node that should receive the 3D projection.
6. Double-click the projection camera node to load its parameters.



7. Click the **Projection** tab in the camera's panel and then enter the information you gathered for **focal length**, **horiz aperture**, and **vert aperture**.

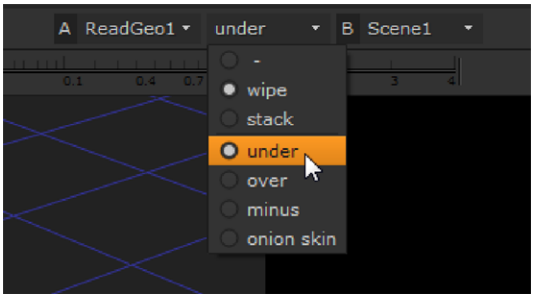


When you are finished, view the 3D scene to check the placement of the projection. The next section explains how to preview the 2D and 3D elements together to check the results of the composite.

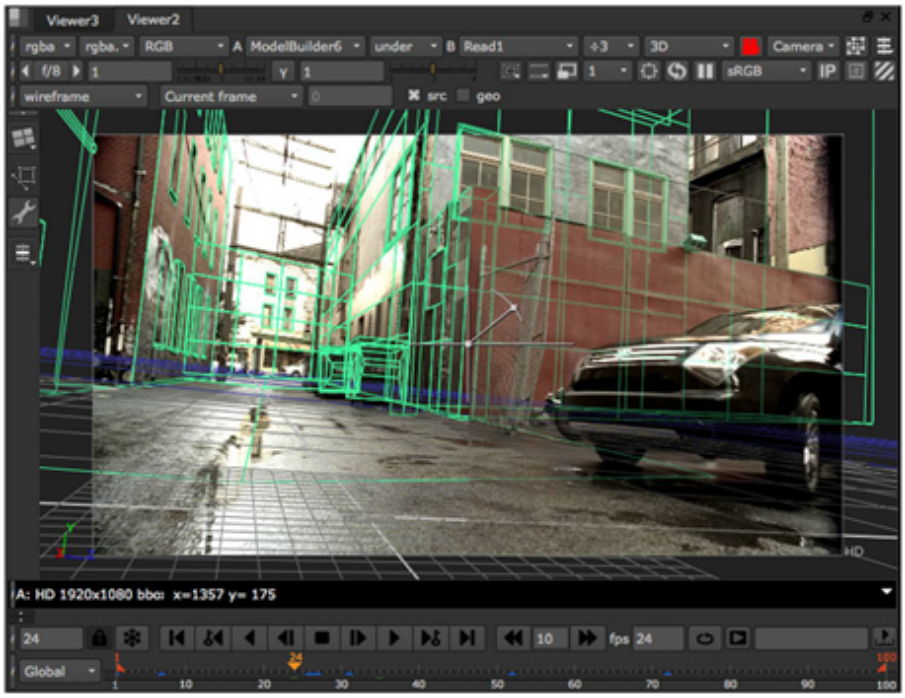
To View a 3D Scene over a 2D Background Image

1. Select the Scene node and press **1** to display its output to the Viewer.
2. If necessary, press **Tab** to toggle the Viewer to 3D mode.
3. Select the rendering camera object or node and press **H** to look through it.
4. Select the node with the 2D image you want to see in the Viewer, and then press **Shift+2**.

   The **Shift**+**2** keystroke connects the image to the Viewer (assigning the next available connection, number 2), and also sets up the compare wipe.
5. Select the desired option from the Viewer composite dropdown menu, such as **over**, **under**, **minus**, and **wipe**).



This last step superimposes the two elements in the Viewer. The crosshair (shown below) is the control that lets you adjust the location and angle of the wipe for the comparison.



Comparing a 3D scene over a 2D image.

Importing Cameras from FBX Files

FBX is a standard 3D file format that gives you access to 3D scenes created in other applications supporting the same format. What you generally have in an **.fbx** file is an entire 3D scene containing cameras, lights, meshes, non-uniform rational B-spline (NURBS) curves, transformation, materials, and so on. From this scene, you can extract cameras, lights, transforms, and meshes into Nuke. This way, you can, for example, create a camera in Maya, export it in a **.fbx** file, and use the same camera again in Nuke.

Note: For the FBX SDK version used in Nuke, see Third-Party Library Versions.

Tip: If you have trouble with **.fbx** files, it may be because they were written with an older version of FBX. If they load very slowly, it is also possible that they are ASCII rather than binary. To get around these problems, you can use the FBX converter on the Autodesk website (http://usa.autodesk.com/fbx/download/). It converts between various different formats, including older FBX versions, ASCII, and binary, and is available on Windows, Mac, and Linux.

Importing Cameras from an FBX File

The Camera node lets you read in the standard FBX cameras (Producer Perspective, Producer Top, Producer Bottom, Producer Right, Producer Left, Producer Front, Producer Back) and any other cameras.

Using one Camera node, you can only import one camera from a **.fbx** file. If you need to import several cameras, you need to use one Camera node per camera.

1. Select **3D** > **Camera** to insert a Camera node in the place where you want to add the camera in your script.
2. In the Camera controls, check **read from file**. When this is checked, the controls on the **File** tab are enabled, and you can use them to read in a camera from an **.fbx** file. Any controls whose values are read in from the **.fbx** file are disabled. You can still view these values and use them in expressions but, as long as **read from file** is checked, you cannot modify them. Modifying the values in the **.fbx** file, however, affect the disabled values in the Camera controls, because these are reloaded from the **.fbx** file every time the node is instantiated.
3. To read in a camera from an **.fbx** file, click the folder icon on the **File** tab. Navigate to the **.fbx** file and select **Open**.



4. From the **animation stack** dropdown menu, select the take you want to use from the **.fbx** file. FBX files support multiple takes in one file. Usually, one of the takes is a default take with no animation.
5. From the **node name** dropdown menu, select the camera node you want to import from the **.fbx** file.
6. In the **frame rate** field, define a frame rate (frames per second) to sample the animation curves. To use this rate rather than the one defined in the **.fbx** file, check **use frame rate**.
7. To have the camera rotation values calculated using the look up vector and look at position, check **compute rotation**. If you don't check this, Nuke uses the rotation channel from the **.fbx** file instead of computing a new one. The rotation values are always computed when there is a look at target.
8. If you want to modify the camera properties imported from the **.fbx** file, uncheck **read from file** on the **Camera** tab and make the necessary modifications. As long as **read from file** is unchecked, your changes are kept.
9. To reload the camera properties from the **.fbx** file, make sure **read from file** is checked and click the **reload** button on the **File** tab.

Importing Cameras from Alembic Files

You can import cameras from Alembic files (**.abc** file format) into a Nuke scene. During the import, Nuke allows you to control which nodes in the Alembic scene get loaded by using an import dialog. If there is only one item in the Alembic file, it loads automatically.

For more information on Alembic, see http://code.google.com/p/alembic/

Tip: In addition to cameras, you can also import meshes (or NURBS curves/patch surfaces converted to meshes), point clouds, and transforms from Alembic files.

To learn how to export files in the Alembic (**.abc**) format, refer to Exporting Geometry, Cameras, Lights, Axes, or Point Clouds for more information.

To Import Cameras from an Alembic File:

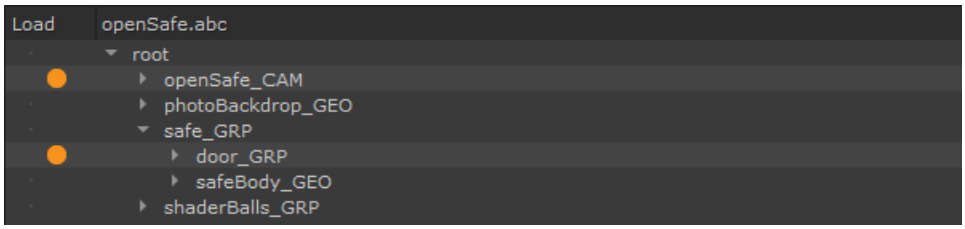
1. Click **Image** > **Read** or press **R** on the Node Graph.

   The **Read File(s)** dialog displays.
2. Select the Alembic file you want to import from the file location, then click **Open**.

   The Alembic import dialog displays. By default, all items are selected in the Scene Graph when the import dialog is opened, as shown below:



   Selected parent items are indicated by a yellow circle, and selected child items by a yellow bar (these turn orange when you select them in the list). Unselected items do not have an indicator next to them.
3. To import specific items, you must first deselect the root item by clicking on the yellow circle. This de-selects the root and any child items. Then, select specific items in the scenegraph by clicking on the blank space where the circles had been, as shown below:



Alternatively, you can right-click on an item and select:

- **Select as parent** - to select this item and make it a parent to other items. This allows you to create a separate Nuke node for this item (and any child items underneath it) in the next step.
- **Select as child** - to select this item and make it a child to the nearest parent item up the tree.
- **Deselect** - to deselect this item (that is, not import it from the scene).

You can also select multiple items by pressing **Ctrl/Cmd** or **Shift** while clicking them.

4. Do one of the following:
   - Click **Create all-in-one node** to create one Nuke node for everything that's selected, regardless of whether the items are selected as parent or child.
   - Click **Create parents as separate nodes** to create one Nuke node for each parent item (yellow circle) in the tree. This node contains all the child items (yellow bars) under the parent.

   Nuke creates ReadGeo, Camera, and Axis nodes as necessary, depending on what you selected to import from the scene.
5. On the Camera nodes, you can adjust the following:
   - From the **animation stack** dropdown menu, select the take you want to use from the **.abc** file. Alembic files support multiple takes in one file.
   - From the **node name** dropdown menu, select the camera you want to import from the **.abc** file.
   - In the **frame rate** field, define a frame rate (frames per second) to sample the animation. To use this rate rather than the one defined in the **.abc** file, check **use frame rate**.
   - If you want to modify the camera properties imported from the **.abc** file, uncheck **read from file** on the **Camera** tab and make the necessary modifications. As long as **read from file** is unchecked, your changes are kept.
   - To reload the camera properties from the **.abc** file, make sure **read from file** is checked and click the **reload** button on the **File** tab.
   - To have the camera rotation values calculated using the look up vector and look at position, check **compute rotation**. If you don't check this, Nuke uses the rotation channel from the **.abc** file instead of computing a new one. The rotation values are always computed when there is a look at target.

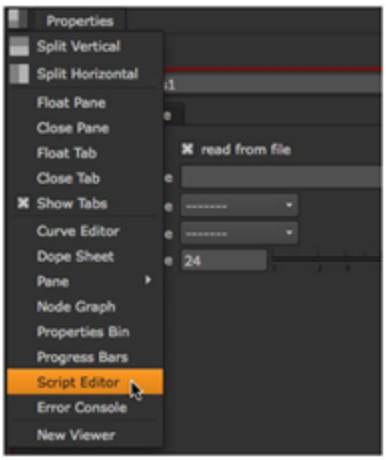
Tip: To load specific items from an Alembic file, you can also create a ReadGeo, Camera, or Axis node, check **read from file**, and click the folder icon on the **File** tab to browse to an **.abc** file.

Importing Cameras from Boujou

Nuke is shipped with a script called import_boujou.tcl, which lets you load in cameras created with Boujou.

To Import a Camera from Boujou:

1. Save the Boujou camera solve as a **.txt** file.
2. In Nuke, click on a content menu button and select Script Editor. The Script Editor opens in the pane whose content menu you used.



3. In the input pane of the Script Editor (that is, the lower pane), enter **nuke.tcl("import_boujou")**. Click the **Run the current script** button on the top of the Editor, or press **Ctrl**+**Return** (**Cmd**+**Return** on a Mac).
4. In the File Browser that opens, navigate to the **.txt** file you saved in step 1.

   A Camera, a ScanlineRender, and a Group node are loaded into Nuke. The Group node contains cylinders to represent points from Boujou.

Tip: You can also open Nuke's Boujou Text File Browser by doing the following:
1. Press **x** on the Node Graph to open the Nuke script command dialog.
2. In the dialog, check **Tcl** (if it's not already checked).
3. In the command field, enter **import_boujou**.
4. Click **OK**.

These steps can be used to replace the first three steps in the above instructions.

Locating a 3D Point from an Animated Camera

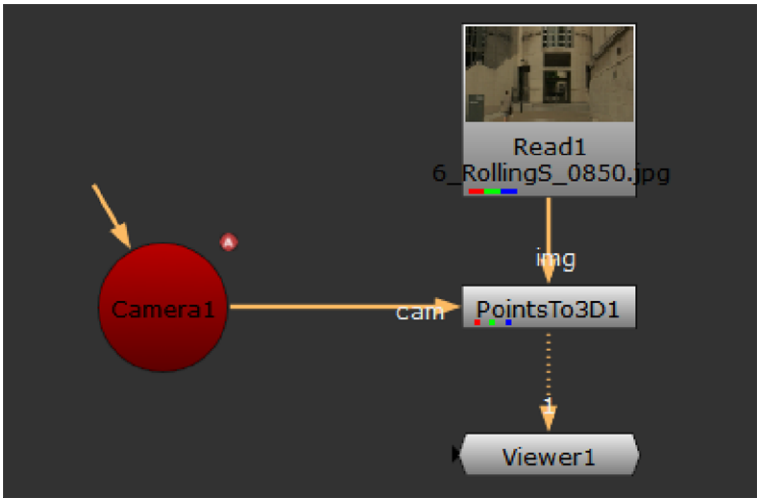
Animated cameras created in third-party applications, such as Maya, or Nuke's CameraTracker contain enough parallax information to locate a 3D point in a 2D scene using the PointsTo3D node. PointsTo3D uses three reference frames in an image sequence to extrapolate the location of a 2D point in 3D space.

Connecting the PointsTo3D Node

1. Read in your image sequence. See Loading Image Sequences for more information.
2. Import your animated camera as described under Cameras.

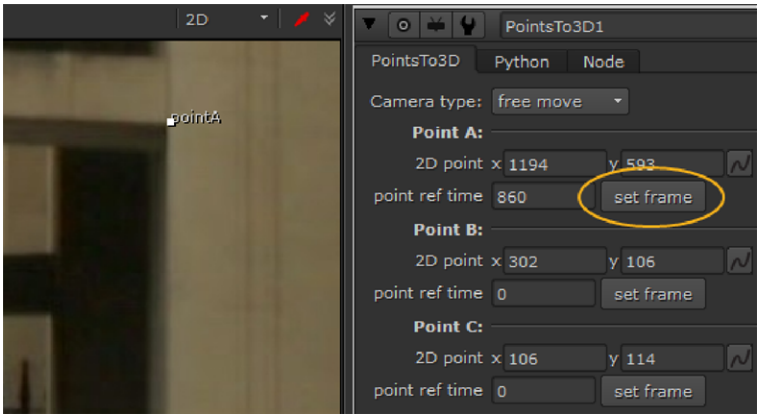
Tip: NukeX and Nuke Studio include CameraTracker, which creates animated cameras without resorting to third-party applications. See Camera Tracking for more information.

3. In the toolbar on the left of the interface, navigate to **Transform** > **PointsTo3D** to add a PointsTo3D node.
4. Connect the image sequence and Camera to the PointsTo3D node and then connect the Viewer.



Setting Reference Frames

1. In the **Properties** panel, set **Camera type** to **free move**.
2. Scrub through the sequence and pick out three frames where the feature is clearly visible.
3. Scrub to the first reference frame where your feature is visible and then drag the **pointA** Viewer widget from the bottom-left of the Viewer to the feature's location.
4. Click **set frame**.



The 2D coordinates and reference frame number are inserted into the **Point A** controls.

Tip: You can adjust the 2D point using the **x** and **y** controls in the properties for pixel-perfect positioning.

5. Scrub to the second reference frame where your feature is visible and then drag the **pointB** Viewer widget from the bottom-left of the Viewer to the feature's location.
6. Click **set frame**.
7. Scrub to the third reference frame where your feature is visible and then drag the **pointC** Viewer widget from the bottom-left of the Viewer to the feature's location.
8. Click **set frame**.

   You can now see coordinate and frame information for all three reference frames in the properties.

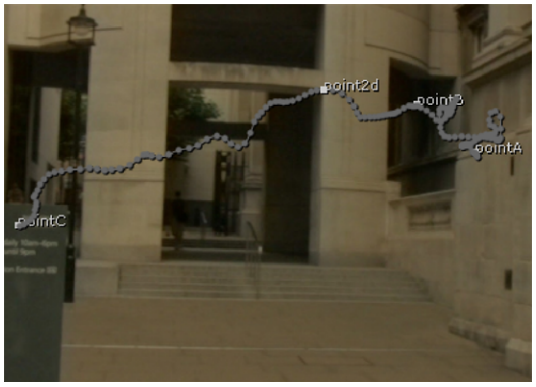
Calculating the 3D Position

1. Click **Calculate** to display the range dialog.
2. Click the **Frame range** dropdown to select the range bounds and the frames to calculate, if required.
3. To calculate the position from the image sequence using a lower resolution, enable **Use proxy**. This can speed up the calculation, but the results may not be as accurate.

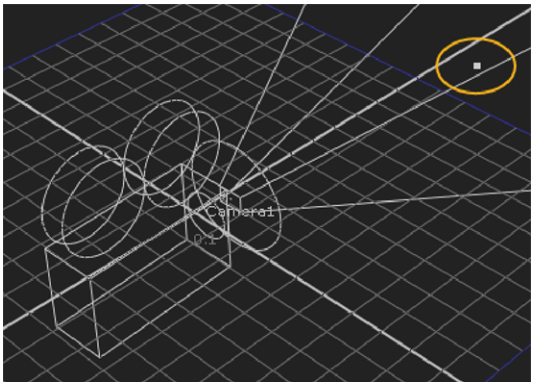
4. To stop the calculation if an error occurs during the process, disable **Continue on error**.
5. Click **OK**.

   Nuke runs through the sequence, tracking the 2D position against the three reference frames to determine the corresponding 3D point.

   Once the calculation is complete, the **point2d** Viewer widget is placed on the tracked point in 2D space.
6. Press **Tab** in the Viewer to switch to the 3D perspective view. The calculated 3D point is highlighted in 3D space.



The 2D points used to calculate the 3D position.



The resulting position in the 3D Viewer.

You can use the calculated 3D point to help you place elements in the scene, or click **generate axis** to add an Axis node to control all elements in the scene together. See Parenting to Axis Objects for more information on axes.

Transforming Geometry, Cameras, and Lights

Transform operations include moving, scaling, rotating the objects in your 3D scene and can be in either Object, Screen, or World space. When an object node is active, you can enter specific transform settings in the node parameters (see Transforming from the Node Properties Panel), or directly manipulate the object with the transform handles displayed in the 3D Viewer (see Using the Transform Handles). Transformations can also occur from the location of the object's pivot point (see Transformations and the Pivot Point).

Nuke allows you to transform several objects together or have one object always face another (see Parenting to Axis Objects and Using the TransformGeo Node).

You can also link transform parameters to imported track or camera data (see Applying Tracks to an Object), or control the transforms with animation curves.

Using FBX, Alembic, and USD files, you can import transforms from third-party 3D applications, such as Maya (see Importing Transforms from FBX Files, Importing Transforms from Alembic Files and Importing USD Items).

Using the Transform Handles

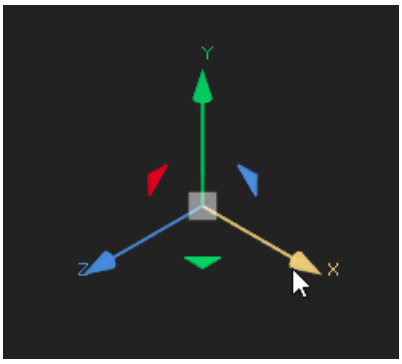
Transform handles appear when a 3D object with transform capabilities is loaded into the **Properties** panel. The colors of the handles correspond to the axes available in 3D space: red transforms the x-axis, green transforms the y-axis, and blue transforms the z-axis.

Tip: If you prefer to work using Nuke's classic transform tools, go to the **Preferences** under **Panels** > **Viewer Handles** and set **3D Hotkeys** to **Nuke Classic**. You can also choose different navigation set ups using this preference, such as the Houdini and Blender configurations.

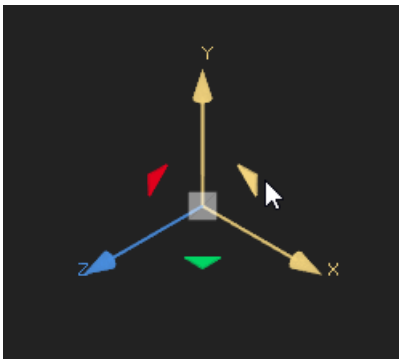
To Translate an Object in the Viewer

Enable the Translate  tool to the right of the Viewer, or press **W** on the keyboard with the object selected, and then:

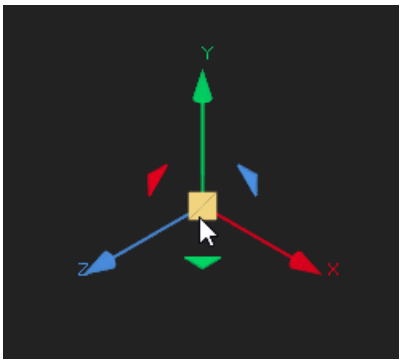
- Drag the object's X, Y, or Z handles to move it on a single axis.
- Drag the object's XY, YZ, or ZX dual-axis handles to move it on two axes simultaneously.
- Drag the center gray square to move it on any axes free-style.



Single axis translate.



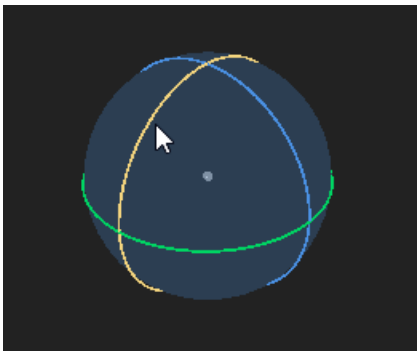
Dual-axes translate.



Free-style translate.

To Rotate an Object in the Viewer

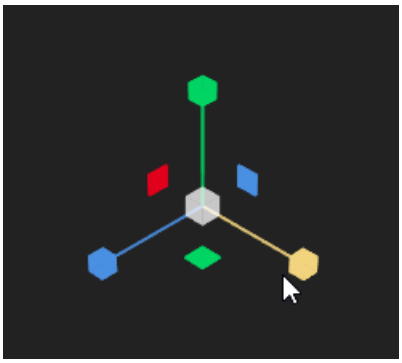
Enable the Rotate tool to the right of the Viewer, or press **E** on the keyboard with the object selected, and then drag the object's X, Y, or Z handles to rotate it on a single axis.



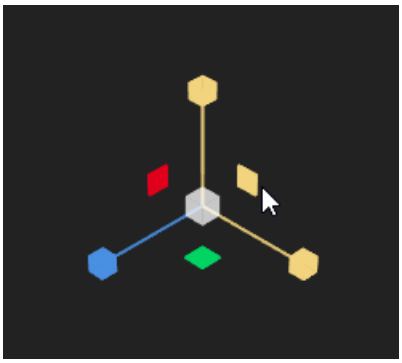
To Scale an Object in the Viewer

Enable the Scale tool to the right of the Viewer, or press **R** on the keyboard with the object selected, and then:

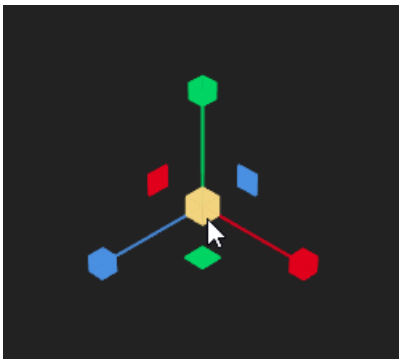
- Drag the object's X, Y, or Z handles to scale it on a single axis.
- Drag the object's XY, YZ, or ZX dual-axis handles to scale it on two axes simultaneously.
- Drag the center gray square to scale it uniformly on all axes.



Single axis scale.



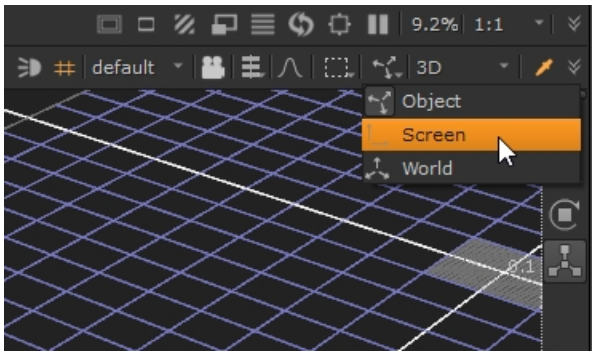
Dual-axes scale.



Uniform scale.

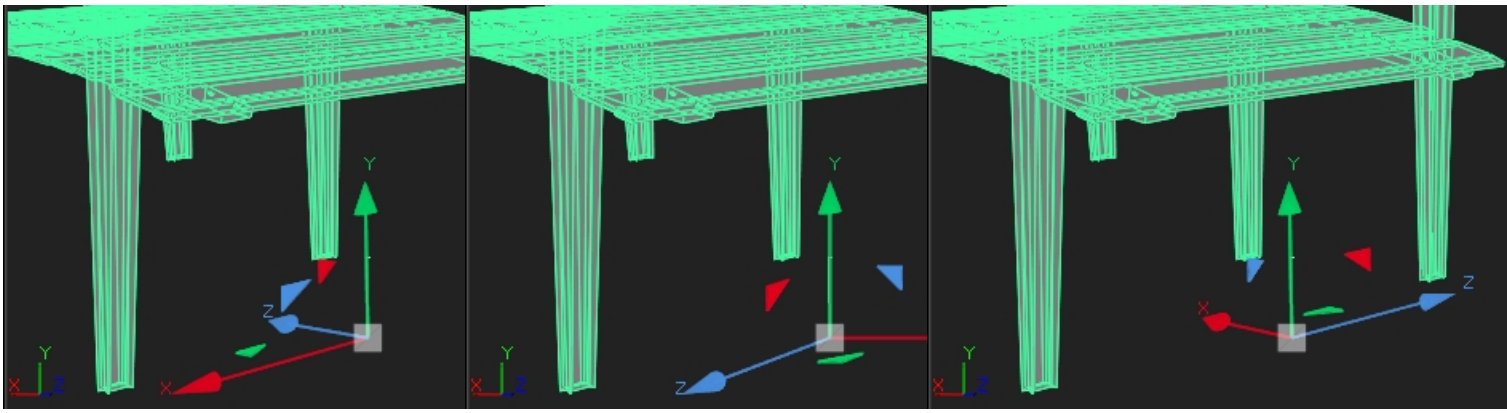
Transforming in Object, Screen, and World Space

Transforming objects in the Viewer is available in three modes, **object**, **screen**, and **world** space. You can select which transform space you're using by clicking the  icon above the Viewer.



Transforms behave differently depending on which transform space you're using:

- **object space** - transforms in this space are relative to the object itself and its orientation. For example, translating on the X axis in **object** space might shift its position on the Z axis in **world** space.
- **screen space** - transforms in this space are relative to the screen, the flat plane of your monitor, and its orientation. You can only translate on the X and Y axes, the Z axis is disabled.
- **world space** - transforms in this space are defined by the XYZ axes of the scene in Nuke's 3D Viewer. The mini axes drawn at the bottom left of the Viewer always show world space no matter what space you're working in.



The translate handles for the same geometry in **object**, **screen**, and **world** space.

Transforming from the Node Properties Panel

The transform handles (see Using the Transform Handles) are a convenient way to move objects around in the 3D workspace, but when you want more precision, you should enter values directly into the object's node panel. The panel also includes transform and rotation order options, which are not available within the 3D Viewer.

The following assumes you've already loaded the object's parameters into the Properties Bin.

To Set Transformation Options

- From the **transform order** dropdown menu, select an option to define the order by which transformations are executed (**s** signifies scale, **r**, rotation; and **t**, translation).
- From the **rotation order** dropdown menu, select an option to define the axial order by which rotations are executed.

To Transform an Object from Its Panel

- To move the object along one or more axes, increment or decrement the **translate x**, **y**, and **z** fields.
- To rotate the object, increment or decrement the **rotate x**, **y**, and **z** fields.
- To scale the object on all axes simultaneously, increment or decrement the **uniform scale** field.
- To scale the object asymmetrically (on x, y, or z), increment or decrement the **scale x**, **y**, and **z** fields.

- To skew the object (warp it by rotating its local axes), increment or decrement the **skew x**, **y**,and **z** fields to rotate the corresponding axis (and associated object vertices).

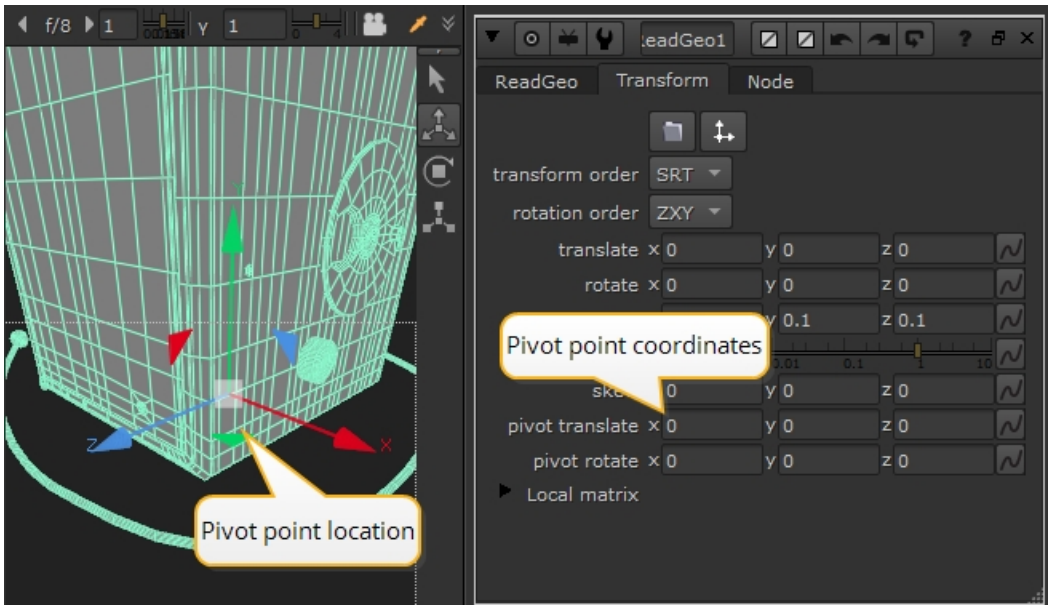
Transformations and the Pivot Point

When you make changes to an object's position, scaling and rotation, these occur from the location of the object's origin point or *pivot*. By default, the pivot point is located at the intersection of the object's local axes.

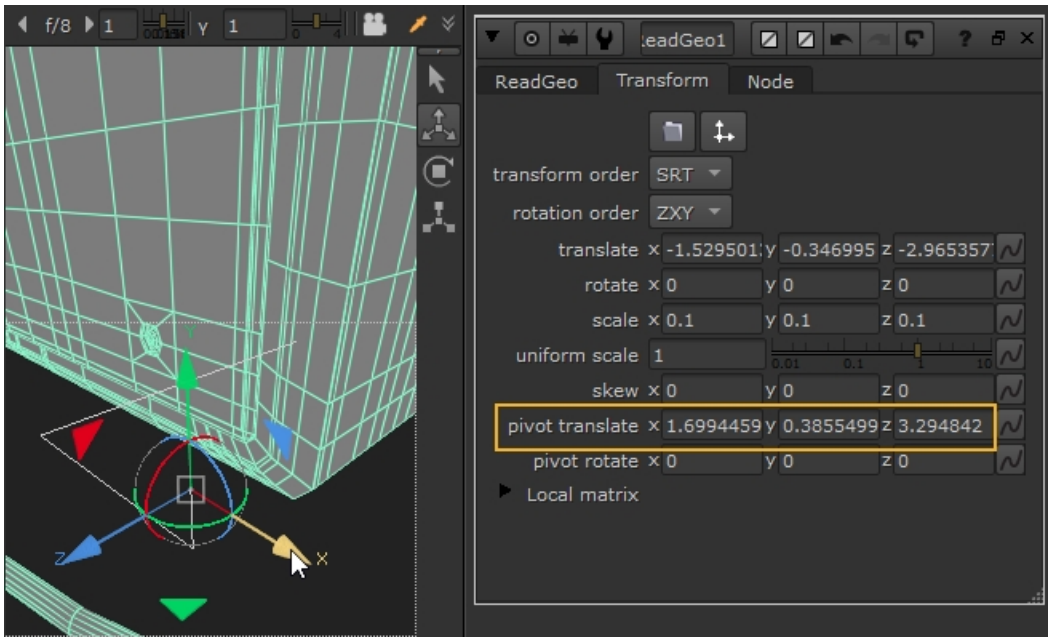
You can offset the pivot point and move it anywhere you like - you can even move it outside of the object. Subsequent local transformations then occur relative to the new pivot point location.

To Move the Pivot Point

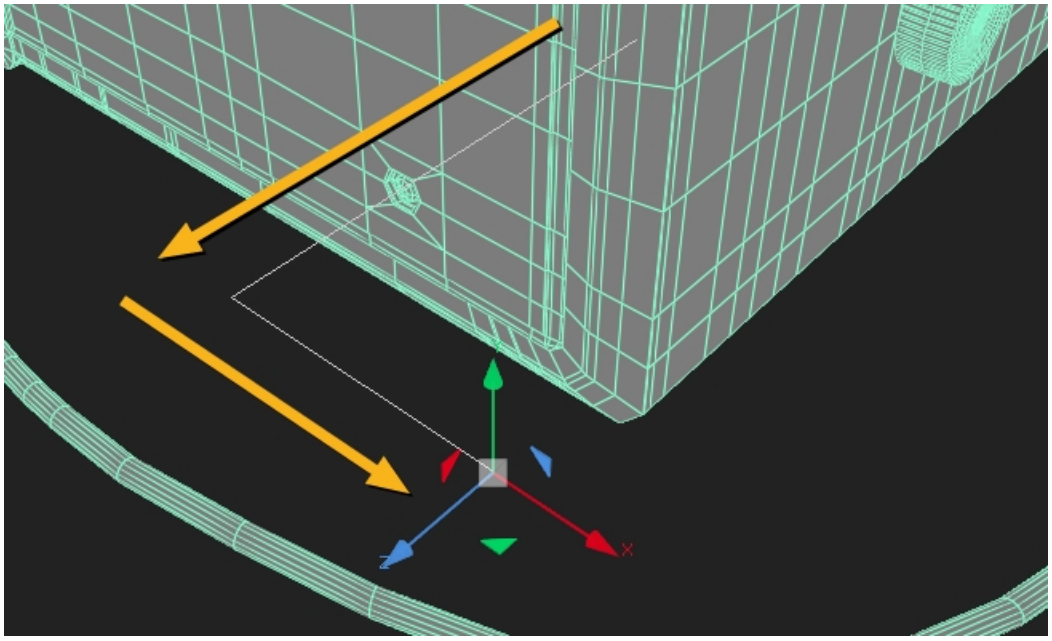
1. Double-click on the object node to display its parameters.
2. Change the **pivot translate xyz** fields to move the local axis in any direction.



3. Or you can hold **Ctrl**/**Cmd** and drag the pivot point to translate it on the required axis. The **pivot translate** fields update as you drag the pivot point around the Viewer.



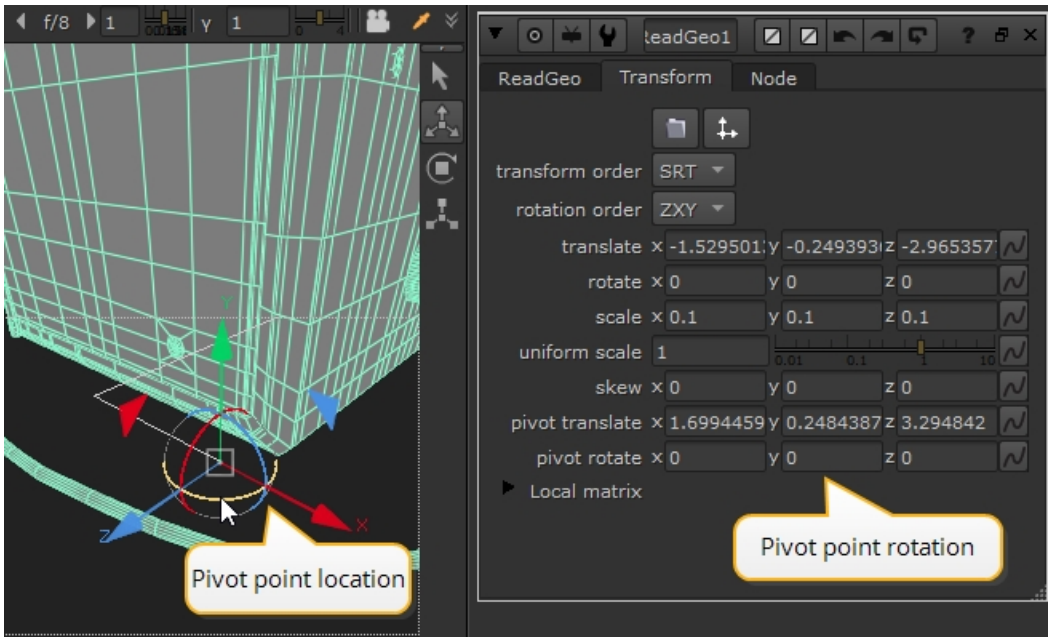
The object's graphical overlay points to the location of the pivot point with a parent line. All subsequent local transformations occur relative to this pivot point.



Once you've defined the location of an object's pivot point, you can use the object's transform parameters to translate, rotate, scale, and skew the object relative to the pivot point.

Rotating the Pivot Point

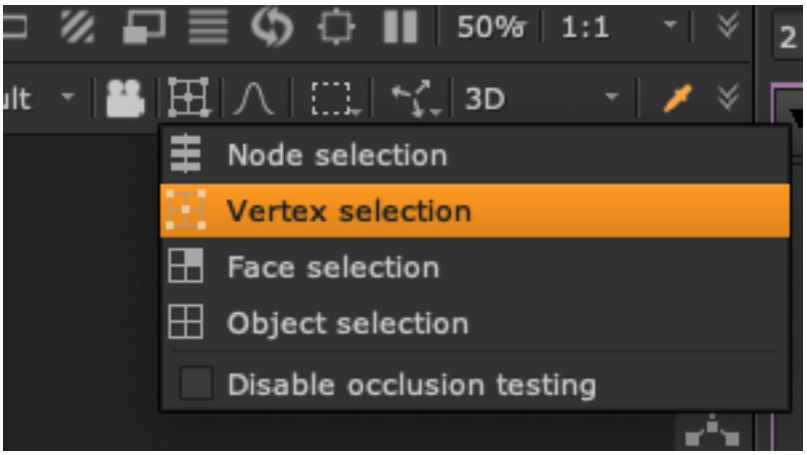
You can rotate the pivot point in the **Properties** panel or directly in the Viewer by holding **Ctrl/Cmd** and dragging the rotation handles, so you can see the true orientation of the pivot point.



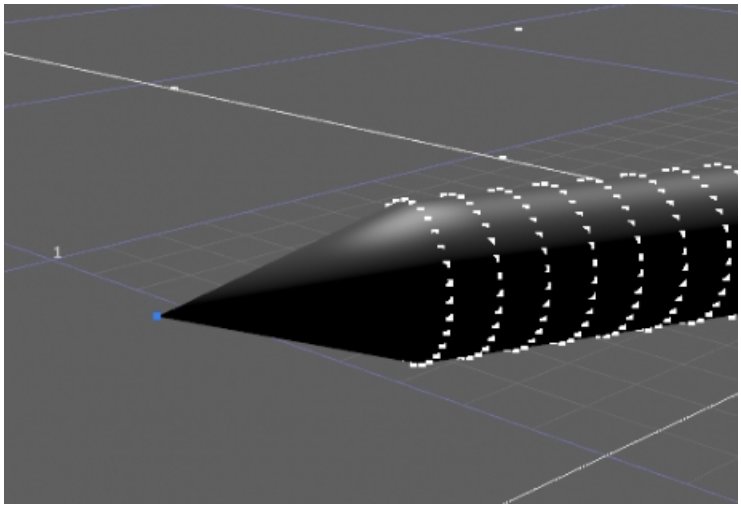
Matching Your Pivot Point to Vertices for Accurate Rotation

You can match the pivot point position to selected vertices, making it much easier to precisely place pivot points and move objects more accurately.

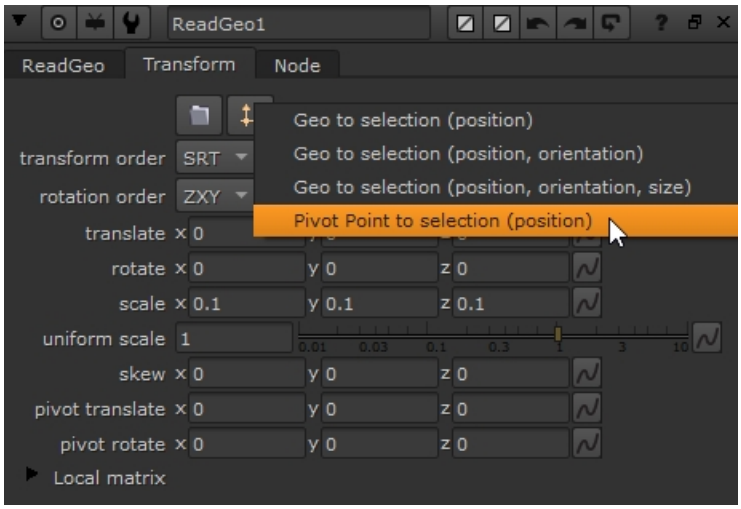
1. Switch the Viewer selection mode to **Vertex selection**.



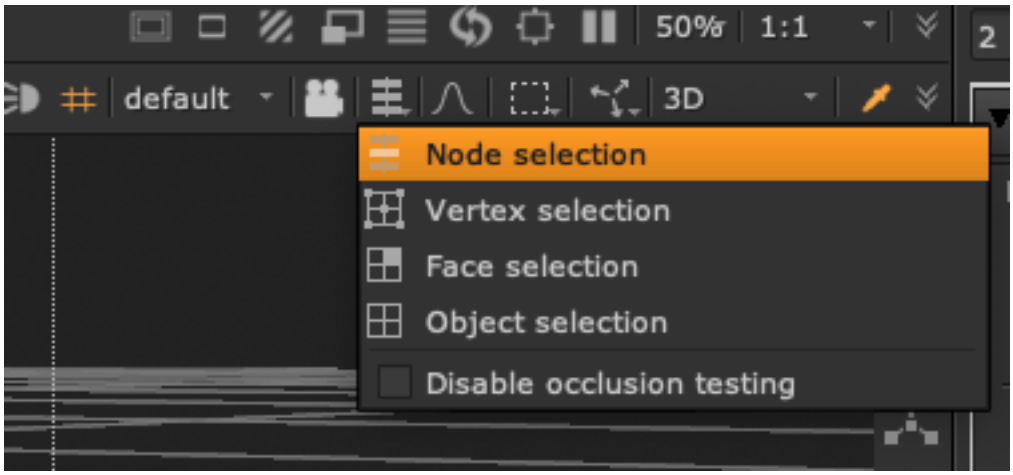
2. Select the vertices to which the pivot point is matched. In this case, the tip of the pen.



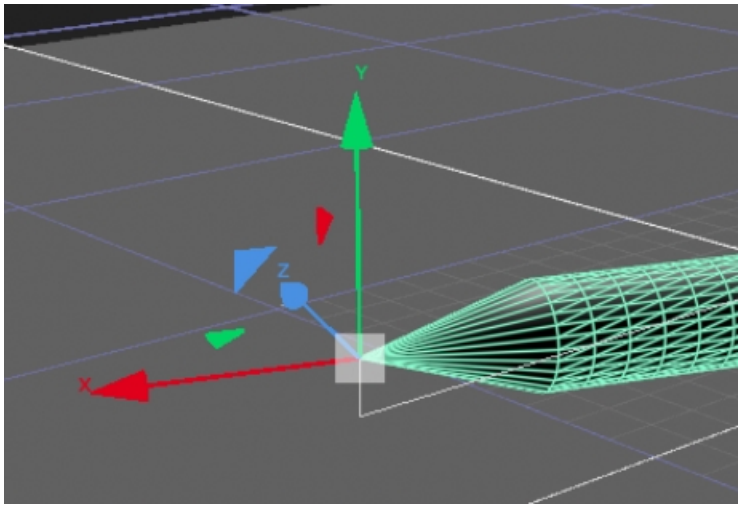
3. In the node's **Properties** panel, click the snap button and then select **Pivot Point to selection (position)** from the dropdown.



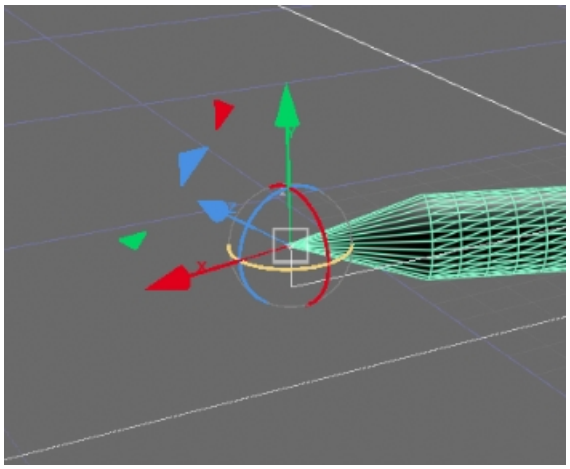
4. Switch the Viewer selection mode to **Node selection** or **Object selection** in the dropdown.



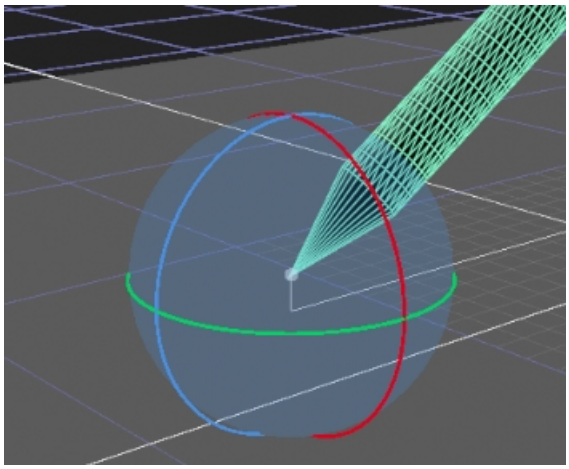
The pivot point snaps to the selected vertex.



You can now control your object from the new pivot position, including rotating the pivot point.



Transform geometry from the pivot point.

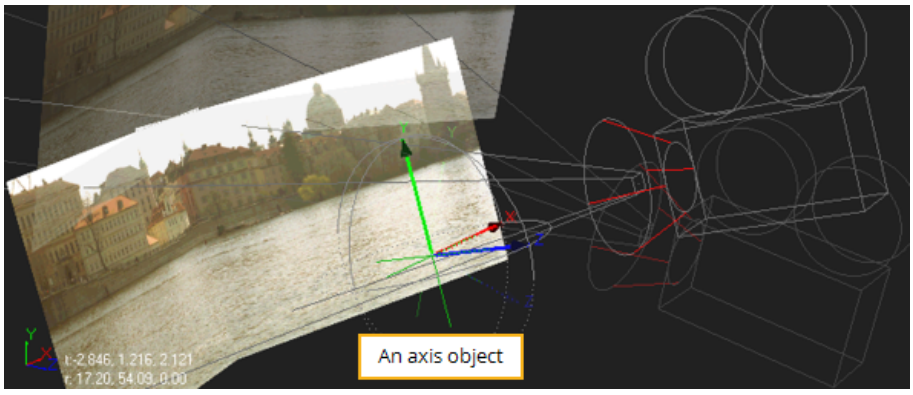


Rotate geometry from the pivot point.

Parenting to Axis Objects

An axis object works as a null object by adding a new transformational axis to which other objects may be parented. Even when objects already have their own internal axes, it's sometimes useful to parent in a separate axis.

For example, the Axis node has been parented to the other objects in the scenes (the two image planes and the camera). The result is an axis object that globally controls the scene. Rotating it, for example, rotates all objects in the scene, as the figure below shows.

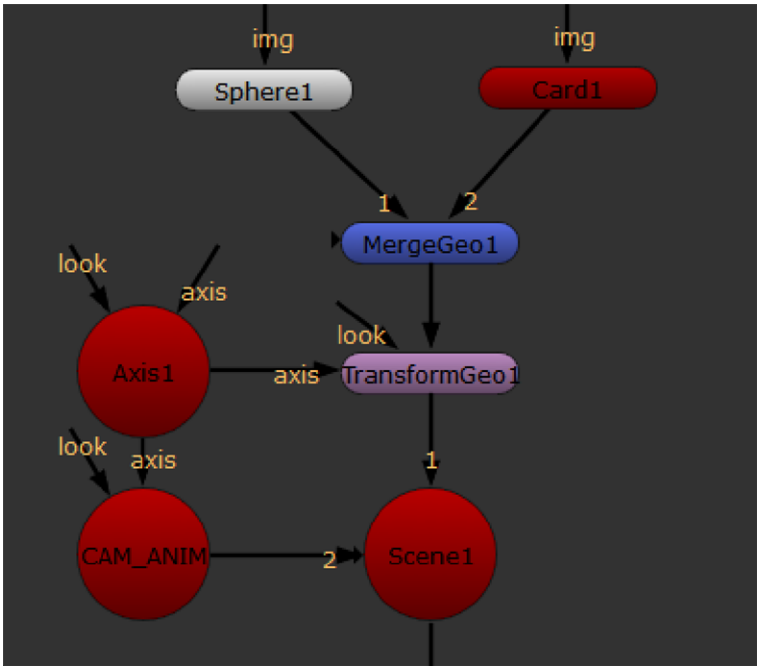


Rotating an entire scene with an axis object.

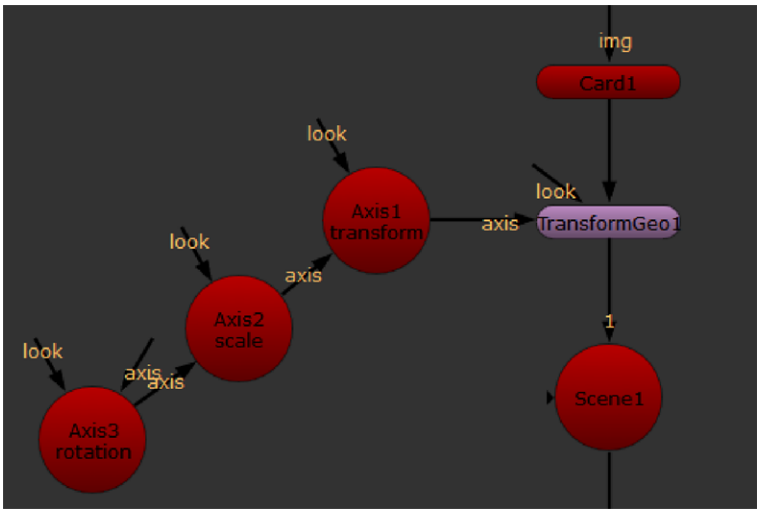
Tip: To move several objects together, you can also merge them using a MergeGeo node (see Merging Objects) and then control them using a TransformGeo node (see Using the TransformGeo Node).

To Add an Axis Object

1. Click **3D** > **Axis** to insert an Axis node.
2. To create the parent relationships, connect the Axis node to all object nodes you wish to control with the Axis transformation controls.



To create a nested transformational hierarchy, chain additional Axis nodes to the first one you inserted. For example, you could create a hierarchy of three Axis nodes to control rotation, scale, and transform.



Creating a nested transformational hierarchy.

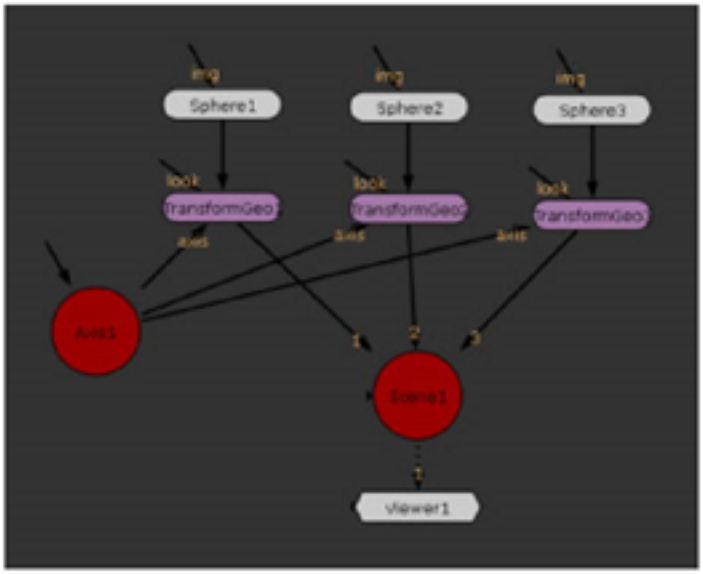
In the above example, Axis3 (rotation) is connected to Axis2 (scale). Axis2 is connected to Axis1 (transform), and Axis1 is connected to the TransformGeo node(s) that you want to affect. With the Axis nodes connected in this manner, their transformation data ripples down the chain and is added to the settings of the TransformGeo node.

Using the Look Input

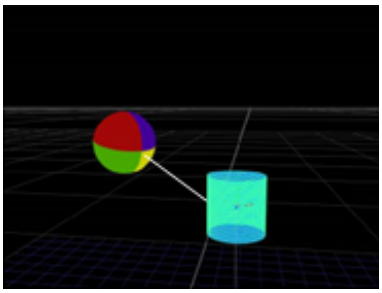
You can use the optional **look** input of the Axis node, so that the axis automatically rotates to face towards the connected input. You can attach a Camera, Light, or Axis node to the **look** input. For example, you can connect an Camera node to the **look** input so that the axis rotates to face the camera, wherever it is moved. See Working with Lights or Working with Cameras for an example.

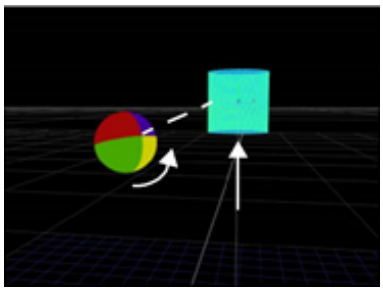
Using the TransformGeo Node

The TransformGeo node allows you to move, rotate, scale, and perform other transformations on several objects merged together with a MergeGeo node. It also lets you connect geometry objects to an Axis node. By doing so, you can move all the connected objects together by using the Axis transformation controls. All you need to do is insert a TransformGeo after each geometry object, connect the Axis node to the TransformGeo nodes' **axis** input, and adjust the transform controls of the Axis node. For more information, see Parenting to Axis Objects.



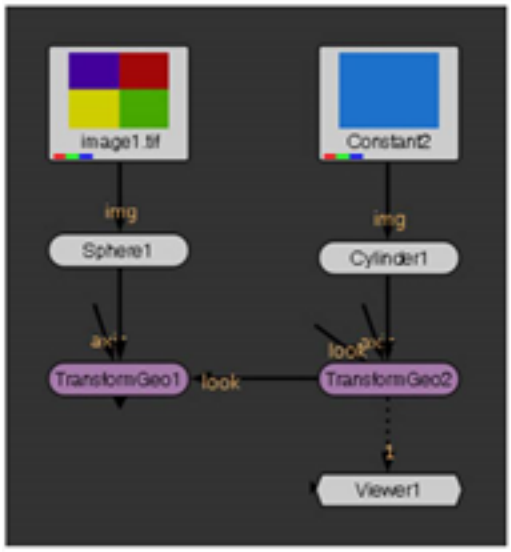
Another use of the TransformGeo node is to have the rotation of one object depend on the position of another so that the first object is always rotated to face or "look at" the second one. For example, you can have a sphere object always facing a cylinder object, regardless of the cylinder's position. When the cylinder is moved to a new position, the sphere is automatically rotated to face it.



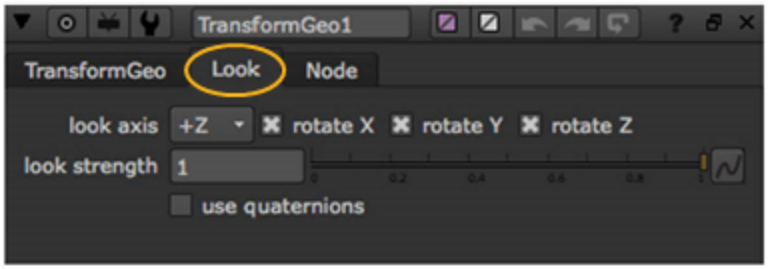


To Have One 3D Object Always Face Another

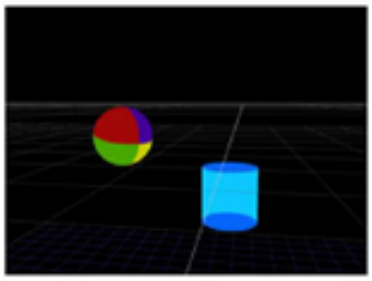
1. Select the 3D object node (for example, a sphere) that you want to face another object.
2. Select **3D** > **Modify** > **TransformGeo** to insert a TransformGeo node.
3. Select the object you want the first object to face (for example, a cylinder), and insert a TransformGeo after this node, too.
4. Connect the first TransformGeo node into the **look** input of the second TransformGeo node.



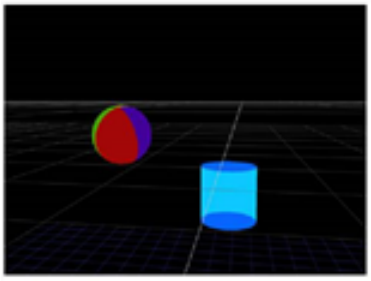
5. Open the controls of the first TransformGeo node and go to the **Look** tab.



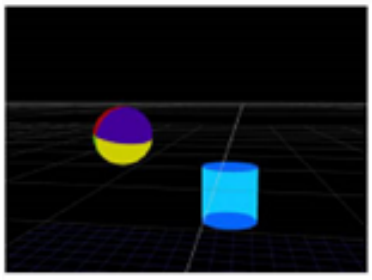
6. From the **look axis** dropdown menu, select the axis around which the object rotates to face the other object:

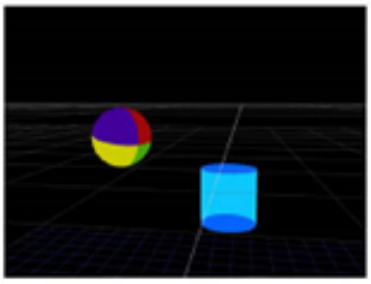


+Z

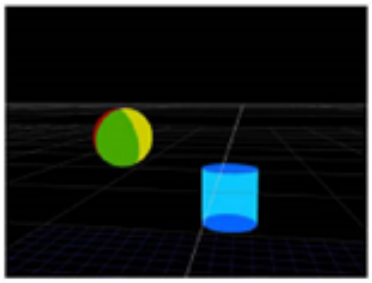


+Y

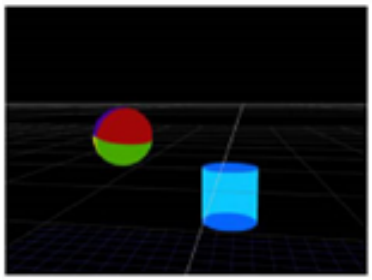




+X



-Z



-Y

-X

7. Use the **rotate X**, **rotate Y**, and **rotate Z** check boxes to select the axes the object rotates around. For the first object to truly face the second, you need to check all three check boxes.
8. Adjust the **look strength** slider to define the extend of the rotation. The smaller the value, the less the object is rotated. Setting the value to 0 produces no rotation.
9. If you want to use an alternate scheme to calculate the rotation, check **use quaternions**. This may be useful for smoothing out erratic rotations along the selected **look axis**.

If you now adjust the second TransformGeo node's transform controls, you'll notice that the first object automatically rotates to face the second object. For more information on how to adjust the transform controls, see Using the Transform Handles and Transforming from the Node Properties Panel.

Applying Tracks to an Object

Nuke can import channel files and apply the motion data to the transformation parameters of any camera or object. The most common purpose for this is to simulate a practical camera move or move objects along a defined path.

Channel files contain a set of Cartesian coordinates for every frame of animation in a given shot. This information is calculated by 3D tracking software, such as 3D-Equalizer, Maya, or Boujou, and then exported as channel files.

To apply a channel file to an object

1. Double-click on an object or camera node to display its parameters.
2. Click **import chan file**. The file navigation dialog appears.
3. Navigate to the channel file, then click **OK**.
4. Nuke reads in the channel data and displays a status message about the number of data frames imported. You'll also notice the object's translation parameters turn green to indicate these parameters

are now controlled by animation data. Scrub the Viewer and you'll notice the object or camera now moves according to the transformation data imported from the channel file.

Note: You can use the **export chan file** button to export as a chan file any animated translation parameters which you've applied to given object. This is a useful method of sharing setups between artists.

Tip: You can also use channel files to import cameras created in other applications into Nuke. However, as the chan file format is not a standard file format, you may need a file format converter to export chan files from other applications.

Importing Transforms from FBX Files

FBX is a standard 3D file format that gives you access to 3D scenes created in other applications supporting the same format. What you generally have in an **.fbx** file is an entire 3D scene containing cameras, lights, meshes, non-uniform rational B-spline (NURBS) curves, transformation, materials, and so on. From this scene, you can extract cameras, lights, transforms, and meshes into Nuke. This way, you can, for example, create a transform in Maya, export it in a **.fbx** file, and use the same transform again in Nuke.

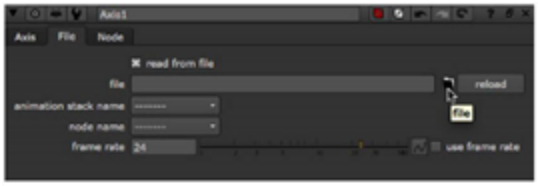
Note: For the FBX SDK version used in Nuke, see Third-Party Library Versions on page 1.

Tip: If you have trouble with **.fbx** files, it may be because they were written with an older version of FBX. If they load very slowly, it is also possible that they are ASCII rather than binary. To get around these problems, you can use the FBX converter on the Autodesk website (http://usa.autodesk.com/fbx/download/). It converts between various different formats, including older FBX versions, ASCII, and binary, and is available on Windows, Mac, and Linux.

Importing Transforms from an FBX File

The Axis node reads in transforms, markers and nulls (locators) from FBX files. You can use it to import one transform, marker, or null per Axis node.

1. Select **3D** > **Axis** to insert an Axis node in your script. Connect the Axis node to a Scene node.
2. In the Axis controls, check **read from file**. This enables the controls on the **File** tab, allowing you to import transforms from an **.fbx** file. It also disables controls whose values are filled in from the **.fbx** file. As long as **read from file** is checked, you cannot modify these values. You can, however, view them and use them in expressions. The values are reloaded from the **.fbx** file every time the node is instantiated, so any changes you make in the **.fbx** file's values are reflected in the Axis controls.



3. On the **File** tab, click the folder icon to open the File Browser. Navigate to the **.fbx** file that contains the transform you want to use. Click **Open**.
4. From the **animation stack** dropdown menu, select the take you want to use from the **.fbx** file. FBX files support multiple takes, one of which is usually a default take with no animation.
5. From the **node name** dropdown menu, select the transform, marker, or null you want to import from the **.fbx** file.
6. If you do not want to use the frame rate from the **.fbx** file for sampling the animation curves, in the **frame rate** field, enter a new value (frames per second). To override the frame rate defined in the **.fbx** file and use the one you defined here, check **use frame rate**.
7. If you want to modify the transform properties imported from the .fbx file, uncheck **read from file** on the **Axis** tab and make the necessary modifications. As long as **read from file** is unchecked, your changes are kept.
8. To reload the transform properties from the **.fbx** file, make sure **read from file** is checked and click the **reload** button on the **File** tab.

Importing Transforms from Alembic Files

You can import transforms from Alembic files (**.abc** file format) into a Nuke scene. During the import, Nuke allows you to control which nodes in the Alembic scene get loaded by using an import dialog. If there is only one item in the Alembic file, it loads automatically.

For more information on Alembic, see http://code.google.com/p/alembic/

Tip: In addition to transforms, you can also import meshes (or NURBS curves/patch surfaces converted to meshes), point clouds, and cameras from Alembic files.

To learn how to export files in the Alembic (**.abc**) format, refer to Exporting Geometry, Cameras, Lights, Axes, or Point Clouds for more information.

To Import Transforms from an Alembic file:

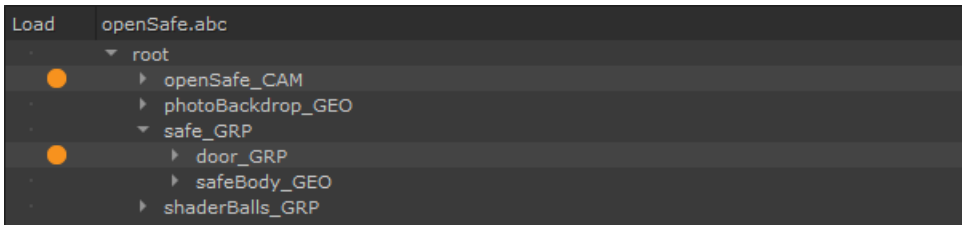
1. Click **Image** > **Read** or press **R** on the Node Graph.

   The **Read File(s)** dialog displays.
2. Select the Alembic file you want to import from the file location, then click **Open**.

   The Alembic import dialog displays. By default, all items are selected in the Scene Graph when the import dialog is opened, as shown below:



   Selected parent items are indicated by a yellow circle, and selected child items by a yellow bar (these turn orange when you select them in the list). Unselected items do not have an indicator next to them.
3. To import specific items, you must first deselect the root item by clicking on the yellow circle. This de-selects the root and any child items. Then, select specific items in the scenegraph by clicking on the blank space where the circles had been, as shown below:



Alternatively, you can right-click on an item and select:

- **Select as parent** - to select this item and make it a parent to other items. This allows you to create a separate Nuke node for this item (and any child items underneath it) in the next step.
- **Select as child** - to select this item and make it a child to the nearest parent item up the tree.
- **Deselect** - to deselect this item (that is, not import it from the scene).

You can also select multiple items by pressing **Ctrl/Cmd** or **Shift** while clicking them.

4. Do one of the following:
   - Click **Create all-in-one node** to create one Nuke node for everything that's selected, regardless of whether the items are selected as parent or child.
   - Click **Create parents as separate nodes** to create one Nuke node for each parent item (yellow circle) in the tree. This node contains all the child items (yellow bars) under the parent.

   Nuke creates ReadGeo, Camera, and Axis nodes as necessary, depending on what you selected to import from the scene.
5. On Axis nodes, you can adjust the following:
   - From the **animation stack** dropdown menu, select the take you want to use from the **.abc** file. Alembic files support multiple takes in one file.
   - From the **node name** dropdown menu, select the transform, marker, or null you want to import from the **.abc** file.
   - In the **frame rate** field, define a frame rate (frames per second) to sample the animation. To use this rate rather than the one defined in the **.abc** file, check **use frame rate**.
   - If you want to modify the transform properties imported from the **.abc** file, uncheck **read from file** on the **Axis** tab and make the necessary modifications. As long as **read from file** is unchecked, your changes are kept.
   - To reload the transform properties from the **.abc** file, make sure **read from file** is checked and click the **reload** button on the **File** tab.

Tip: To load specific items from an Alembic file, you can also create a ReadGeo, Camera, or Axis node, check **read from file**, and click the folder icon on the **File** tab to browse to an **.abc** file.

Transforming and Projecting with SphericalTransform

SphericalTransform converts images between different projections, including 360 work, and takes advantage of Blink GPU acceleration. These view projections can be divided into two broad categories:

- full frame, such as **Latlong**, encompassing the entire 360 world around a single point, and
- partial frame, such as the **Rectilinear** view that Nuke was designed to work in.

The SphericalTransform node can be used for common Nuke operations on 360 material, such a rotoing, comping, and tracking. SphericalTransform allows you to configure the input and output projections for the desired conversion. For partial frame projections, additional projection space parameters are enabled on the **Input** and **Output** tab for the specific camera parameters, such as **focal length**, **sensor size**, and so on.

Use the Rotation controls to adjust the **Input** and **Output** from a single point governing **Look** position, two points going **From/To**, **Pan/Tilt/Roll**, or **Full Rotation** angles with control over rotation order.

The **Output** rotation is also controllable using an in-viewer control system. Hold down **Ctrl**/**Cmd**+**Alt** and left-click and drag to move the image around, setting the pan and tilt setting. Add **Shift** to lock into a single dimension for the movement. In a partial frame projection, use the right mouse button to set the **focal length**, in essence zooming in and out.

SphericalTransform can convert between the following projection modes:

Projection Name	Example	Description
Latlong		**Latlong**, or equirectangular, projections are the most common full 360 frame projection. Many VR pipelines use them for both ingest and export due to their simplicity and wide use. Working in latlong space can be problematic due to its unfamiliar mapping, that many compression techniques were not designed to handle, and inefficiencies towards the poles, where many pixels represent a single pixel in the output.

Projection Name	Example	Description
Cubemap		**Cubemap** projections are another full 360 projection. Each of the six faces is essentially rectilinear, so the data can be more familiar to work in. Faces can be packed in a number of ways: • **Image** - all faces are placed in a single image and packed according to the **Packing** control. The default is **LL-Cross** as shown in the example. • **Views** - each face creates a view in the stream. The view is named using the following convention: `cubemap_<direction><axis>` Where **<direction>** can be **pos** or **neg**, for positive and negative, and **<axis>** can be **x**, **y**, or **z**. • **Faces** - each face is a placed in a separate image stream, but due to Nuke's limitations, there can only be a single face output in this mode. You can choose the face to output using the **Face** control. When in input mode, you get six separate inputs, labeled according to the face they represent.
Rectilinear		**Rectilinear** projection is a partial frame, standard projection you're most likely familiar with. **Tip:** Remember when you're going to or from rectilinear, you're only able to cover part of the frame (up to virtually 180 degrees). As a partial frame projection this enables extra parameters on the **Input** and **Output** tabs (depending on if you've picked it as an input or output projection). These govern the camera parameters, such as **focal length**, **sensor size**, and so on that are not applicable in full frame projections.

Projection Name	Example	Description
Fisheye		**Fisheye** covers a number of projections, all of which emulate common optical models used in fisheye lenses. These are all partial frame projections, so enable specific camera parameters similar to the **Rectilinear** projection type. Additionally, you can select the particular model to use: • **Stereographic** is not widely used in optics, Samyang Optics being one of the few to employ it. The **Stereographic** model forms the basis of the **little planet** projection, where the center of the projection is the nadir. SphericalTransform ships with a **LittlePlanet** properties preset, which applies such a look to a latlong with a horizon line centered vertically. • **Equidistant** is the default setting and matches the zeroed model employed in the fisheye distortion estimation employed by CaraVR's C_CameraSolver node. **Equidistant** is often considered the ideal model, as its response is a balance between the curves of various models. Other tools without this level of control most likely employ this model. • **Equisolid** is the most frequently found model in practical optics. • **Orthographic** is a classical 'perfect' response model that sees little use in practical optics. It does, however, match the fisheye model in Nuke.
Mirrorball		Mirrorball produces an image of a reflective ball, scaled up so that the ±XY of the image are at the edge of the format.

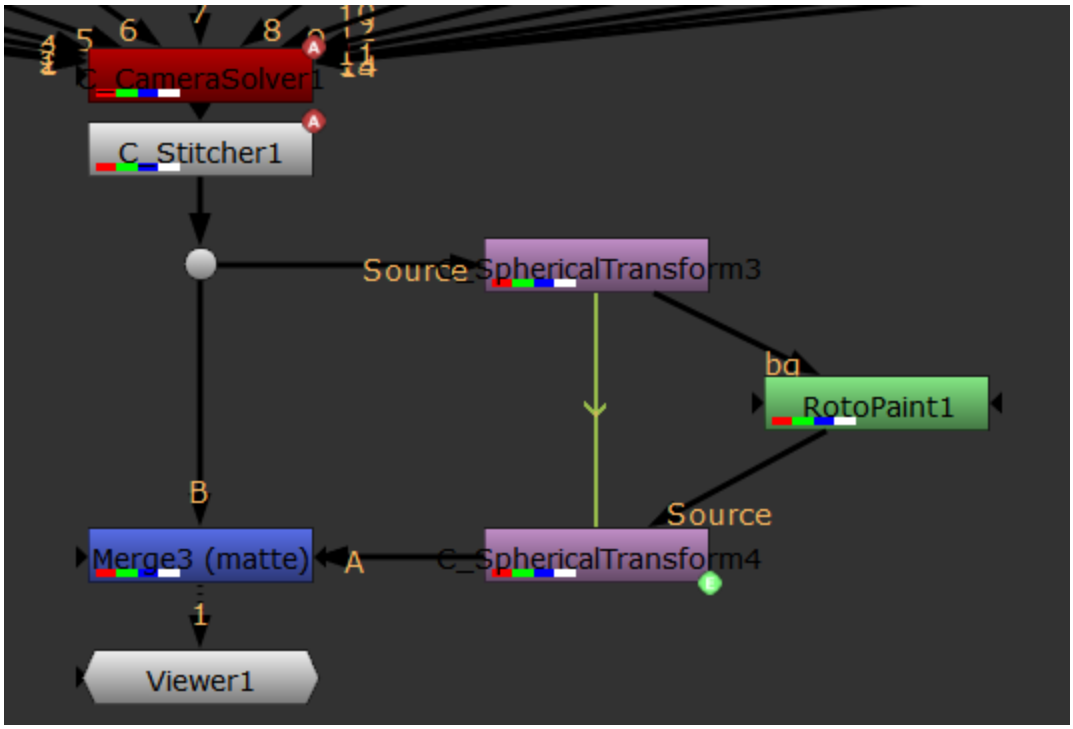
Compositing in 360 Footage

Nuke nodes often work under the assumption they are applied to rectilinear shots, and as an artist, you might prefer working with material in rectilinear space, for example when rotoing. SphericalTransform allows you to work on 360 material quickly and easily, taking care of the required projection changes for you.

Painting in Latlong Space

CaraVR, with NukeX or Nuke Studio licenses, ships with a toolset to make these conversions quick to perform, but you can recreate the node tree shown manually.

1. Select the point in the node tree you want to apply the paint to, and navigate to **CaraVR** > **ToolSets** > **Latlong_RotoPaint**.
2. The toolset is added to the Node Graph at the specified node.



The toolset contains two SphericalTransforms converting the image into a rectilinear preview, so you can add your paint, and then back into latlong space. Apply any paint required in the RotoPaint node and it'll be mapped back to 360, and then merged back into the original image.

3. Connect the Viewer to the first SphericalTransform node and open its **Properties** panel to activate the **directionInput** Viewer widget.
4. Move the **directionInput** widget to the area you're interested in to move around the image.

Tip: You can Adjust the **Angle** control to rotate the image relative to the **directionInput** widget to get the view you need.

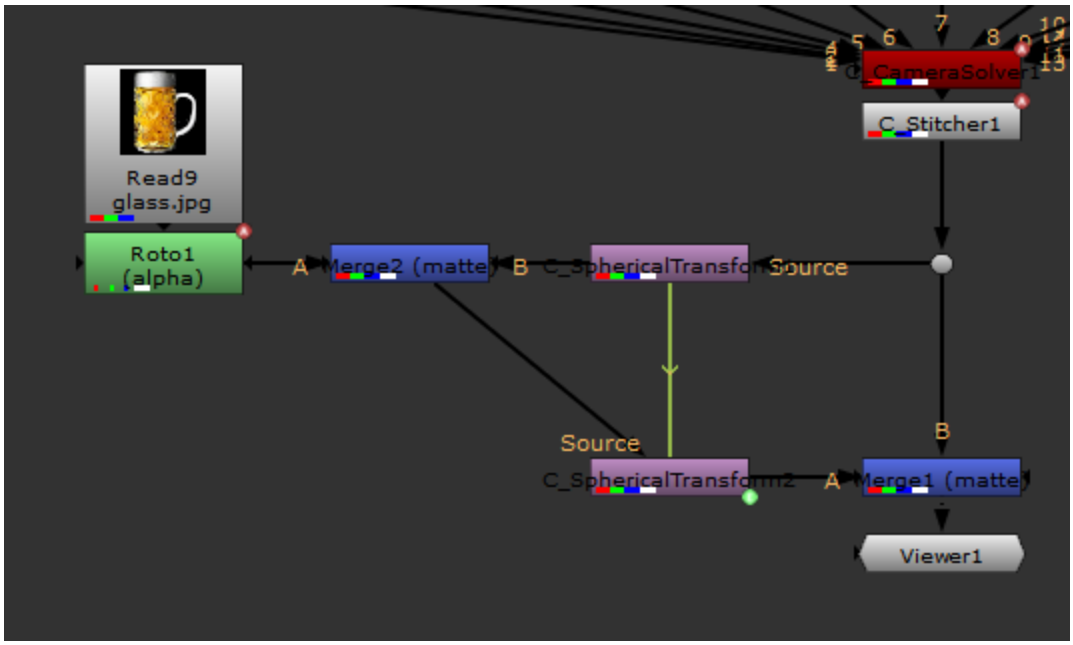


5. If necessary, you can increase the field of view of the rectilinear projection by decreasing the output **focal length** of the first SphericalTransform node.
6. Connect the Viewer to the RotoPaint node and paint as required.
7. To view you paint in latlong space, connect the Viewer to the second SphericalTransform node.

   The Merge node in the toolset uses the alpha channel you painted into to merge your paint back over the source 360 shots, so you get fewer filtering hits for the majority of the image.

Comping in Latlong Space

Similar to the RotoPaint toolset, the **Latlong_Comp** toolset in CaraVR, allows you to merge a rectilinear object into a 360 environment.



Note: Ensure the alpha channel of the rectilinear object carried through to the final merge, so that there's minimal filtering hits on the other parts of the frame.

Reviewing 360 Footage

SphericalTransform is invaluable for reviewing 360 material without the use of a headset. Setting the output to **Rectilinear** allows you to pan around the scene to get a feel for how the final render might look.

1. Add a SphericalTransform downstream of your stitch.
2. Navigate to the **Output** tab in the **Properties** panel.
3. Set the **Focal** control in the range 15-50 mm or use the **Review** preset that is included with the node.
4. Click the **Rotation** dropdown, and then select **Look**.
5. Use the **directionInput** Viewer widget to pan around the scene.



Tip: If you disconnect the SphericalTransform from the node tree and rename it to VIEWER_INPUT using the **Properties** panel, you can toggle the rectilinear projection on and off using the **IP** button (or **Alt**+**P** keyboard shortcut) above the Viewer.
If you leave the SphericalTransform **Properties** panel open, you can pan around in the lat-long projection as well.

Adding Motion Blur to the 3D Scene

To create more realism for a Classic 3D scene, you'll want to add motion blur to it based on the movement of your 3D camera. This can be done in two ways:

- Adjust the **samples** value in the ScanlineRender or RayRender node's Properties panel. The image is sampled multiple times over the shutter period. This way is the most accurate, but also the slowest, because the full rendering calculation is done multiple times for each pixel. See Adding Motion Blur Using a Renderer.
- Use the VectorBlur node to generate motion blur based on motion vectors. This way is faster to render, as it is only affected by the pixel resolution of the final image rather than the complexity of your 3D scene. See Adding Motion Blur Using VectorBlur.

Adding Motion Blur Using a Renderer

To add motion blur using ScanlineRender or RayRender:

1. In the node's controls, go to the **MultiSample** tab.
2. Increase the **samples** value to sample the image multiple times over the shutter time. The higher the value, the smoother the result, but the longer the render time.



Samples set to 5.



Samples set to 15.

3. In the **shutter** field, enter the number of frames the shutter stays open when motion blurring. For example, a value of **0.5** would correspond to half a frame. Increasing the value produces more blur, and decreasing the value less.



Shutter set to 20.



Shutter set to 40.

4. Adjust the shutter offset using the **shutter offset** dropdown menu. The different options control when the shutter opens and closes in relation to the current frame value. Select:
   - **centered** - to center the shutter around the current frame. For example, if you set the shutter value to **1** and your current frame is 30, the shutter stays open from frame 29.5 to 30.5.

- **start** - to open the shutter at the current frame. For example, if you set the shutter value to **1** and your current frame is 30, the shutter stays open from frame 30 to 31.
- **end** - to close the shutter at the current frame. For example, if you set the shutter value to **1** and your current frame is 30, the shutter stays open from frame 29 to 30.
- **custom** - to open the shutter at the time you specify. In the field next to the dropdown menu, enter a value (in frames) you want to add to the current frame. To open the shutter before the current frame, enter a negative value. For example, a value of **-0.5** would open the shutter half a frame before the current frame.

5. If you're using ScanlineRender, you can add randomness to the distribution of samples in time so they don't produce regularly spaced images, by adjusting **randomize time**. This affects all samples except the first and the last one. The larger the value, the larger the time difference between the samples.
6. If you're using ScanlineRender, set **sample diameter** to the diameter of the circle that the samples for each pixel are placed in for antialiasing. The larger the value, the more pixels are jittered.
7. If you're using ScanlineRender, adjust the **focus diameter** to randomly orbit the camera about a point at the **focal distance** in front of it for each sample to produce depth-of-field effects from multiple samples.



Note: The **focal distance** is set in the Camera node's controls, in the **Projection** tab.

8. If rendering has become very slow, you can approximate multi-sample rendering and reduce render times by using **stochastic samples**. This sets the number of samples, per pixel, to use in stochastic estimation (0 is disabled). Lower values result in faster renders, while higher values improve the quality of the final image.

   Note that rather than setting **stochastic samples** to a very large value, you may want to leave it at a relatively low value and increase **samples** instead.

   Stochastic sampling is based on Robert L. Cook's *Stochastic Sampling in Computer Graphics*, available in *ACM Transactions on Graphics*, Volume 6, Number 1, January 1996.



Stochastic samples set to 0.



Stochastic samples set to 5.

9. To distribute the samples uniformly over the shutter time, enable **uniform distribution**. This gives the same importance to all samples over the shutter time, and produces more accurate results for stochastic multi-sampling.

For more information on the ScanlineRender and RayRender nodes, see Rendering a 3D Scene.

Adding Motion Blur Using VectorBlur

Nuke's VectorBlur node generates motion blur by blurring each pixel into a straight line, using the values from the motion vector channels (u and v channels) to determine the direction of the blur. Compared to generating motion blur using the ScanlineRender node's **MultiSample** controls, this is less accurate but faster to render.

You can create the necessary motion vectors for use with VectorBlur in several ways:

- Use the ScanlineRender or RayRender nodes. This is usually more accurate than using a MotionBlur3D node (described below), and works well with both static and moving cameras as well as both linear and non-linear camera movement. See To use VectorBlur with ScanlineRender or To use VectorBlur with RayRender.
- Alternatively, if your 3D scene is static or nearly so and the camera movement over the shutter time is nearly linear, you can also use the **Filter** > **MotionBlur3D** node. However, note that MotionBlur3D only uses camera information to produce motion vectors. If you have moving objects in your scene but the camera is static, MotionBlur3D isn't able to produce any output. See To use VectorBlur with MotionBlur3D.
- Several third-party 3D applications can also produce motion vector information as two-channel, floating point images that you can use with VectorBlur. If possible, you should unpremultiply these images. See To use VectorBlur with Third-Party Motion Vectors.

To use VectorBlur with ScanlineRender

1. In your 3D scene, open the ScanlineRender properties and go to the **Shader** tab.
2. Make sure **motion vectors** is set to one of the following (not **off**):
   - **classic** - Render motion vectors the classic (pre-Nuke 6.1) way. This option is only provided for backwards compatibility, and isn't always accurate.
   - **velocity** - Store the velocity of every single pixel in the motion vector channels (pre-Nuke 7.0 way). This option is only provided for backwards compatibility. In order to have the same behavior as Nuke 6.3, set **samples** to 1.
   - **distance** - For every pixel, store the distance (in pixels) between samples in the motion vector channels. This is the recommended option that usually produces the best results. It also allows the VectorBlur

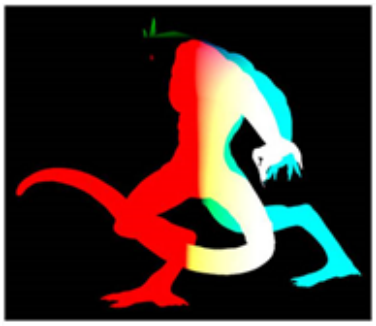
node to produce the impression of curved motion blur by interpolating multiple samples of linear motion.

3. Use **motion vector channels** to select where to store the generated motion vectors.

   If you view the channels you selected, you should see they now contain motion vectors.

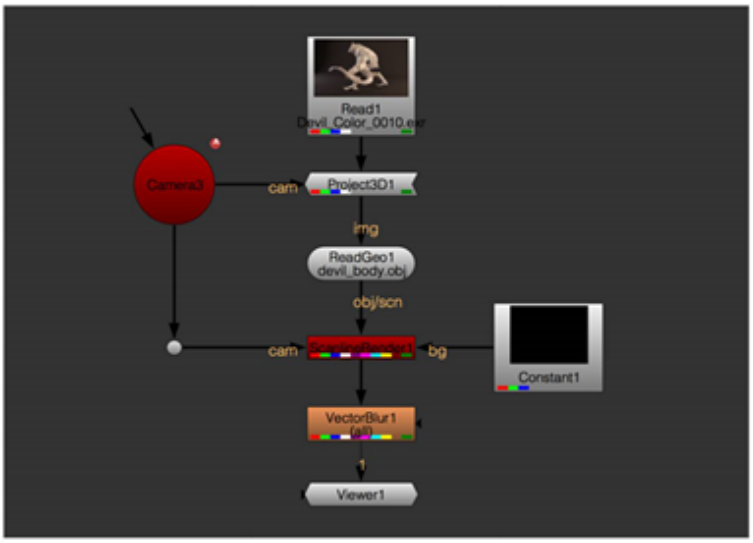


A render of a 3D scene.



Motion vectors produced by ScanlineRender.

4. On the **MultiSample** tab, leave **samples** set to 1. You can increase this value later if non-linear movement cannot be approximated sufficiently well using VectorBlur's linear approach.
5. Select **Filter** > **VectorBlur** to insert a VectorBlur node after ScanlineRender.



A node tree with ScanlineRender and VectorBlur.

6. In the VectorBlur properties, use the **channels** dropdown menu to select the channels to blur.
7. Select the motion vector channels from the **uv channels** dropdown menu. These should be the same channels as the channels you chose to create in step 3.
8. Set the **mv presets** dropdown to set which renderer format to apply to the motion vectors. Each preset contains a scale and offset specific to the renderer, so that the vectors are in the format expected by Nuke.

9. Use the **blur uv** dropdown to blur the motion vectors themselves, before blurring the image, to smooth the results between regions with very different motion. Choose either:
   - **uniform** - applies a small uniform blur to soften edges in the UV map, or
   - **linear** - applies a linearly-weighted blur to blend between regions with different motion. This has a stronger effect than the **uniform** option.



Tip: The **uniform** option can be used to create blur similar to pre-Nuke 10 results.

10. Set **shutter offset** to 0. This means the shutter opens at the current frame.
11. Select the **blur type** using the dropdown.
12. To adjust the amount of blur, adjust the **motion blur** controls.



The original image.



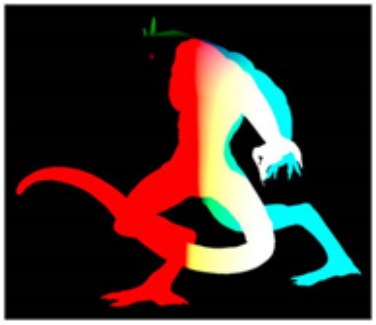
Motion blur produced by VectorBlur.

To use VectorBlur with RayRender

1. In your 3D scene, open the RayRender properties and go to the **AOVs** tab.
2. Enable **output AOV** and then use the **motion vector** dropdown to select where to store the generated motion vectors.

   If you view the channels you selected, you should see they now contain motion vectors.

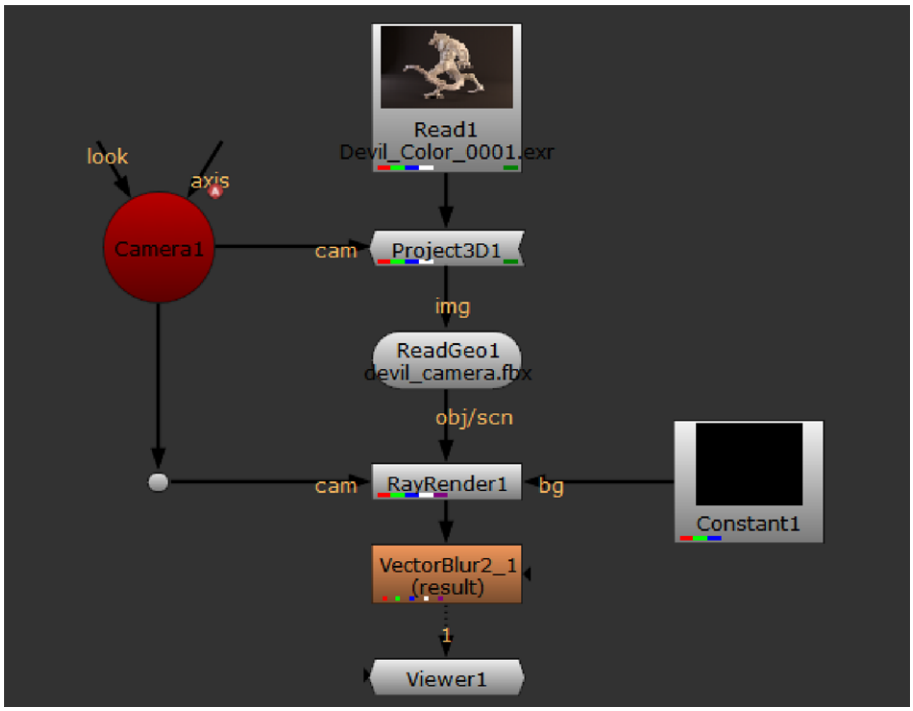




A render of a 3D scene.

Motion vectors produced by ScanlineRender.

3. On the **MultiSample** tab, leave **samples** set to 1. You can increase this value later if non-linear movement cannot be approximated sufficiently well using VectorBlur's linear approach.
4. Select **Filter** > **VectorBlur** to insert a VectorBlur node after RayRender.



A node tree with RayRender and VectorBlur.

5. In the VectorBlur properties, use the **channels** dropdown menu to select the channels to blur.
6. Select the motion vector channels from the **uv channels** dropdown menu. These should be the same channels as the channels you chose to create in step 2.
7. Set the **mv presets** dropdown to set which renderer format to apply to the motion vectors. Each preset contains a scale and offset specific to the renderer, so that the vectors are in the format expected by Nuke.
8. Use the **blur uv** dropdown to blur the motion vectors themselves, before blurring the image, to smooth the results between regions with very different motion. Choose either:
   - **uniform** - applies a small uniform blur to soften edges in the UV map, or
   - **linear** - applies a linearly-weighted blur to blend between regions with different motion. This has a stronger effect than the **uniform** option.

Tip: The **uniform** option can be used to create blur similar to pre-Nuke 10 results.

9. Set **shutter offset** to 0. This means the shutter opens at the current frame.

10. Select the **blur type** using the dropdown.
11. To adjust the amount of blur, adjust the **motion blur** controls.



The original image.



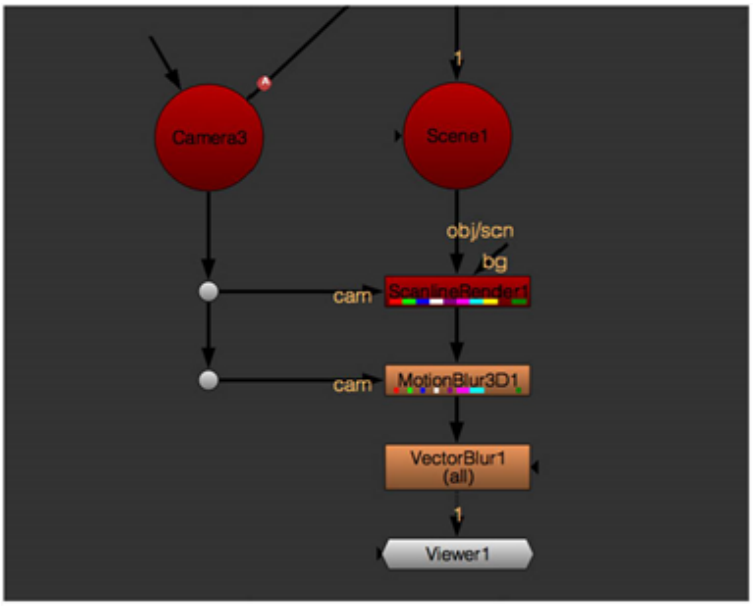
Motion blur produced by VectorBlur.

To use VectorBlur with MotionBlur3D

1. In your 3D scene (which should be static or nearly static), open the ScanlineRender properties, go to the **Shader** tab, and set **motion vectors** to **off**.

   This tells ScanlineRender not to produce motion vectors, which is what you want if you are going to use MotionBlur3D to generate the vectors.
2. Select **Filter** > **MotionBlur3D** to insert this node and connect it to the ScanlineRender node's output.
3. Connect the rendering camera to the **cam** input of the MotionBlur3D node. Note that this has to be a moving camera. The camera movement over the shutter time should be linear or nearly linear.
4. In the MotionBlur3D properties, use **Output UV** to select where to store the generated motion vectors.

   If you view the channels you selected, you should see they now contain motion vectors.
5. Select **Filter** > **VectorBlur** to insert this node and connect it to the MotionBlur3D node.



A node tree with MotionBlur3D and VectorBlur.

6. In the VectorBlur properties, select the motion layer from the **uv channels** dropdown menu.
7. Use the **blur uv** dropdown to blur the motion vectors themselves, before blurring the image, to smooth the results between regions with very different motion. Choose either:
   - **uniform** - applies a small uniform blur to soften edges in the UV map, or
   - **linear** - applies a linearly-weighted blur to blend between regions with different motion. This has a stronger effect than the **uniform** option.

Tip: The **uniform** option can be used to create blur similar to pre-Nuke 10 results.

8. Select the **blur type** using the dropdown.
9. To adjust the amount of blur, adjust the **motion blur** controls.
10. To adjust the length of the blur, adjust the **Shutter** setting in the MotionBlur3D **Properties** panel.



Low **Shutter** value.



High **Shutter** value.

To use VectorBlur with Third-Party Motion Vectors

1. Select **Filter** > **VectorBlur** to insert a VectorBlur node after a 2D image that contains motion vector channels.
2. In the VectorBlur properties, use the **channels** dropdown menu to select the channels to blur.
3. Select the motion vector channels from the **uv channels** dropdown menu.
4. Set the **mv presets** dropdown to apply the presets associated with the selected renderer's default settings, if present.
5. If your motion vectors have been premultiplied, check **uv alpha** and specify the channel used to premultiply the image.
6. Select the **blur type** and the required amount of blur using the **motion blur** controls.

Exporting Geometry, Cameras, Lights, Axes, or Point Clouds

In a Classic 3D scene, you can export geometry, cameras, light, axes and point clouds into an FBX (**.fbx**) or Alembic (**.abc**) file using the WriteGeo node.

1. Create a WriteGeo node and attach it to a Scene node.
2. In the WriteGeo controls, select **fbx** or **abc** in the **file type** dropdown, and check the objects to export:
   - **geometries** - to write the scene geometries into the file.
   - **cameras** - to write the scene cameras to the file.
   - **lights** - to write the scene lights into the file (**.fbx** only).
   - **axes** - to write the scene axes into the file.
   - **point clouds** - to write the scene point clouds into the file.
3. If you're writing FBX files:
   - Check **ascii file format** if you don't want to write out a binary **.fbx** file.
   - Check **animate mesh vertices** if you want the mesh vertices animated and keyframes created at every frame. The animated meshes use vertex point caches for the data. A directory with **_fpc** appended to the file name is created to contain the point caches. By default the animated meshes check box is off so the point cache directory is not created.
4. If you're writing Alembic files, select the **storage format** to use when writing the file:
   - **HDF** - A storage format that maintains backwards compatibility.

- **Ogawa** - A storage format that offers faster file reading and smaller files.

5. Browse to the destination you want to save the **.fbx** or **.abc** file in the **file** field and give it a name.
6. Check the **limit to range** box if you want to disable the node when executed outside of a specified frame range. In the **frame range** fields, enter the range of frames you want to make executable with this Write node.
7. Select the views to export, such as left and right for stereoscopic projects.
8. Click **Execute**.
9. If necessary, you can change the frame range you want to include in the file using the pop-up dialog and then click **OK**.

   A progress bar displays, and the selected file is saved to the specified directory.

Rendering a 3D Scene

The 3D Viewer displays the scene using an OpenGL hardware render. When you build a scene, Nuke renders high-quality output from the perspective of the camera connected to the render node. The rendered 2D image is then passed along to the next node in the compositing tree, and you can use the result as an input to other nodes in the script.

Choosing a Render Node

Nuke ships with two render nodes, ScanlineRender and RayRender, which are suited to different tasks. ScanlineRender, as the name suggests, renders row-by-row advancing down the picture. RayRender traces a path from the camera, or virtual eye, to the light source pixel-by-pixel.



Note: NukeX and Nuke Studio also include Pixar's PhotoRealistic RenderMan®, another ray renderer. See PrmanRender for more information.

The ScanlineRender and RayRender nodes have the same inputs and share some of the same controls in their respective **Properties** panels, but they both have different strengths and weaknesses:

- ScanlineRender generally produces faster results, but is less accurate with reflection and refraction.
- ScanlineRender supports Deep workflows downstream by injecting a **deep** channel into the node tree. See Using ScanlineRender to Generate Deep Data.
- RayRender generally produces very accurate reflection, but at the cost of processing time.

- RayRender does not currently support Deep workflows or render sprite-based particles from NukeX's Particles system.

As a general rule, if you can afford to wait for a render and don't need Deep data or particles that rely on sprites, use RayRender.

Rendering Out a Scene

To render a 3D scene into 2D space you need three things: your scene, a render camera, and a render node. When set up correctly, the output of the render node is passed down the node tree for compositing as normal.

To render out a scene:

1. Set up your 3D scene as normal, including geometry, materials, lights, etc.
2. Add a render node and connect the scene to the **obj/scn** input.
3. Make sure the render camera is connected to the render node's **cam** input. See Cameras for more information.
4. Add a background to the **bg** input, if required. You can use the **bg** input to composite a background image into the scene and to determine the output resolution.



Note: If the **bg** input is not used, the render node output defaults to the **root.format** or **root.proxy_format** defined in the **Project Settings**.

5. Toggle the Viewer back to 2D.
6. Connect the output of the render node to the appropriate 2D nodes in your script.

   See the *Nuke Reference Guide* for more information on the different render node controls.

To Add Motion Blur to the 3D Scene

You can use the properties on the **MultiSample** tab to add motion blur to your 3D scene. For more information, see Adding Motion Blur to the 3D Scene.

Stereoscopic Scripts

The title of this chapter is slightly misleading, as Nuke isn't actually limited to stereoscopic views, but rather provides multi-view support for as many views as you need. The views do not have to be stereo pairs, but since that is the most obvious application, this chapter mainly deals with stereoscopic projects.

Quick Start

In many ways, Nuke lets you work on stereoscopic material just like you would on any other images. However, there are also a few stereo-specific settings and nodes that you need to be aware of when compositing stereoscopic material. The following teaches you how to set up your stereo project, read in and view your images, use the stereo nodes, and render the final output.

Here's a quick overview of the workflow:

1. The first step in working on stereo footage in Nuke is to set up views for them in your project settings (you can open up the project settings by pressing **S** over the Node Graph). Press **Set up views for stereo** on the **Views** tab to do this. For more information, see Setting Up Views for the Script.
2. You can then load your stereo footage into Nuke, either by entering a variable in the file name field or by using a JoinViews node. For more information, see Loading Multi-View Images.
3. In the Viewer, you can select which view to display with the **views** buttons. You can also display the views side by side or mix them together with the SideBySide and MixViews nodes. For more information, see Displaying Views in the Viewer.
4. Sometimes you need to make changes to one view, while the other one remains as it is. In these cases you need to either split off a view using the **View menu** or, in case of RotoPaint, use the right-click options in the stroke/shape list to split off views. For more information, see Selecting Which Views Display Changes.

   If you need to perform totally different actions on the two views, you can separate one view for processing with the OneView node. See Performing Different Actions on Different Views.
5. If you have a disparity field available, you can also use the correlate options in the RotoPaint node and the **View menu** to have changes made to one view automatically reproduced in the other. For more information, see Reproducing Changes Made to One View.
6. With the ShuffleViews node you can rearrange your views, and with the Anaglyph node you can convert your footage into a red and cyan anaglyph footage. For more information, see Swapping Views and Converting Images into Anaglyph.

7. Sometimes you need to re-adjust the convergence, or the inward rotation of your left and right view cameras. This changes the point in the image that appears at screen depth when viewed with 3D glasses. For more information, see Changing Convergence.
8. Finally, you can preview a view of your stereo project by flipbooking it and rendering it out. For more information, see Previewing Stereoscopic Images.

Setting Up Views for the Script

You can import your footage and let Nuke create the views automatically or set up views in advance in the project settings. This allows you to process the individual views separately or both views together, and see the effect of your changes on each view.

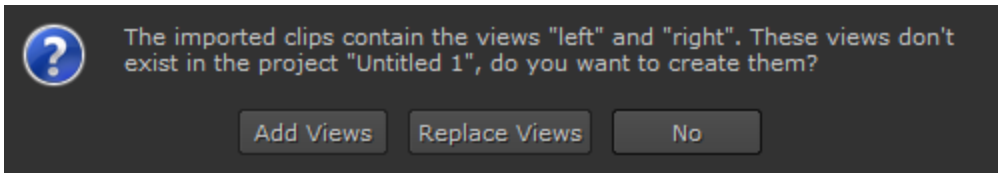
If you are likely to need the same views in several projects, you may want to save the views you created in a **template.nk** script file. For more information on how to do this, see Template Scripts.

Creating Views Automatically

Note: Automatic view creation is not implemented for multi-view **.mov** files. See Creating and Managing Views Manually for information on how to create the views.

1. Read in your multi-view files as normal.

   A warning dialog is displayed.

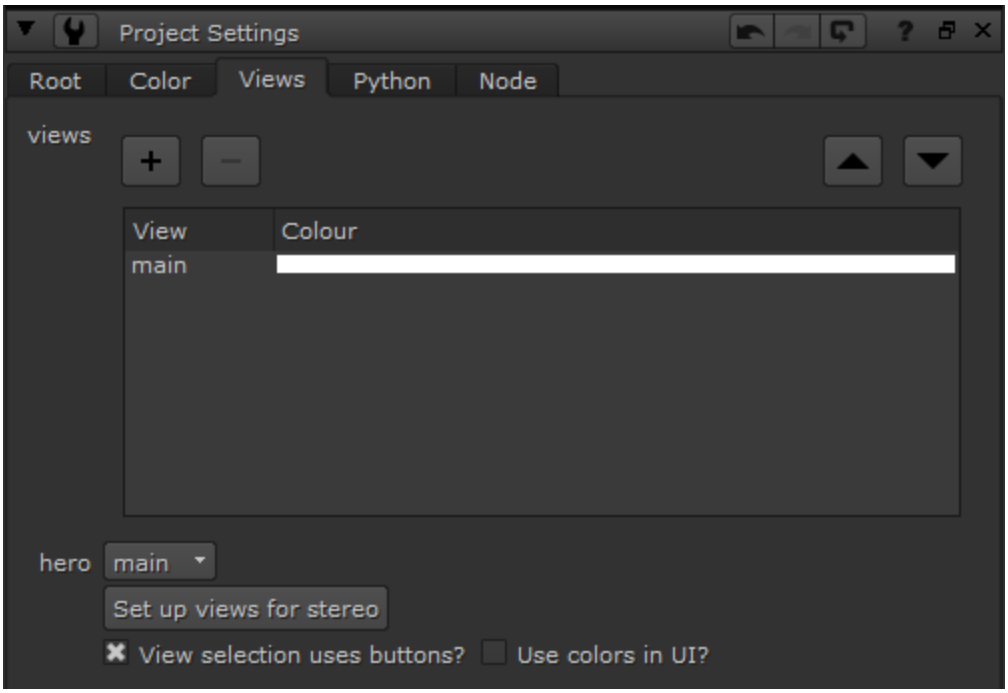


2. Click **Add Views**, **Replace Views**, or **No**:
   - **Add Views** - add the views in the incoming clip to those that exist in the project.
   - **Replace Views** - replace all existing project views with those in the incoming clip.
   - **No** - import the clip and display only the first view in the file, retaining any existing views in the project.

   You can now access the views in your project from the **view** dropdown menu of certain nodes' controls. You'll also notice that each view has its own button in the Viewer controls.

Creating and Managing Views Manually

1. Select **Edit** > **Project Settings** or pres **S** in the Node Graph.
2. Go to the **Views** tab. The exiting views are listed in the **views** field.



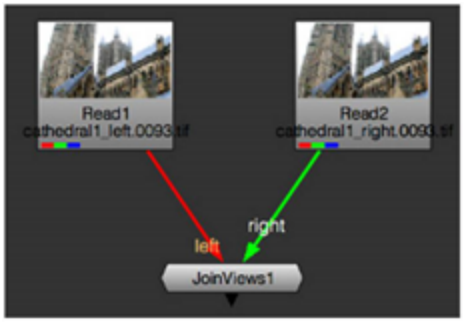
3. If you want to remove the view called **main** and add views called **left** and **right**, click the **Set up views for stereo** button.

Note: If the script is already set up for multiple views, **Set up view for mono** is displayed. Clicking **Set up view for mono** deletes all existing views and adds a single **main** view.

The two new views are assigned colors, red and green by default. To change the colors, double-click on the color field and select another color from the color picker.

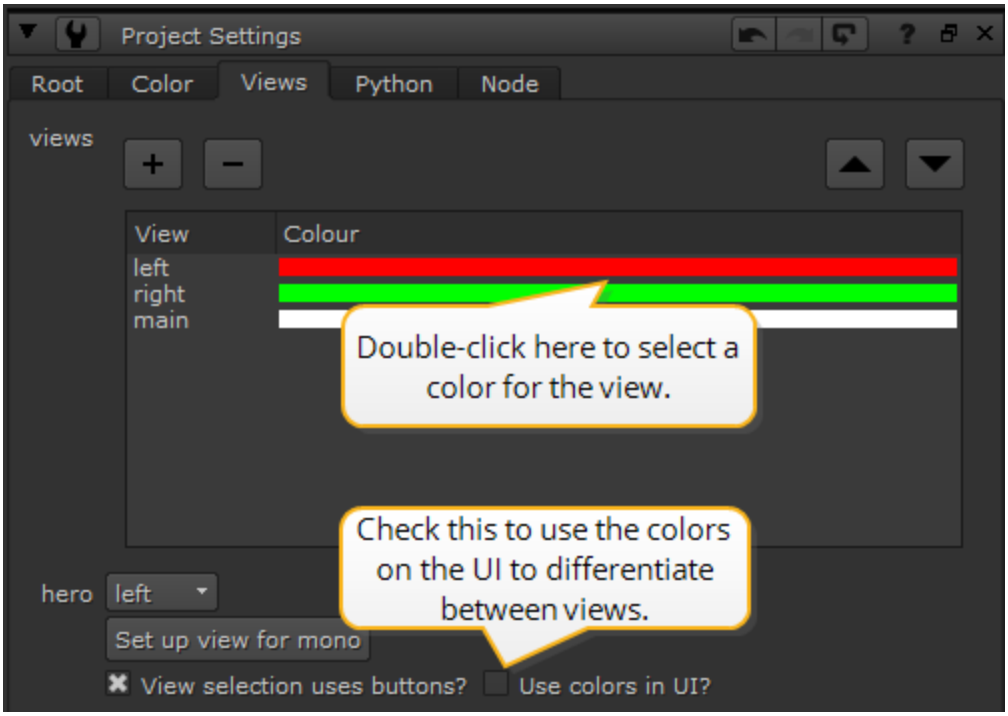


If you check **Use colors in UI?**, these colors are used in Node Graph connections, split views indicators on nodes, and Viewer and ShuffleViews node controls to make it easier to differentiate between views.



4. If you want to add other new views, click the **+** button.

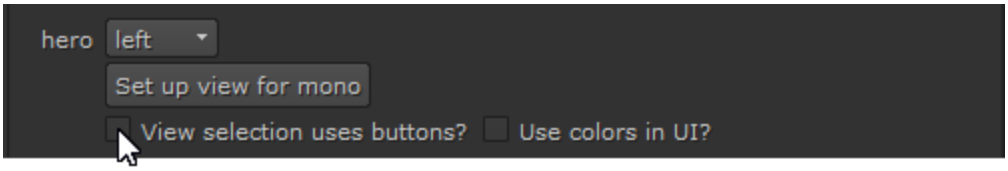
   A new view is added automatically.
5. To rename the view, double-click its name in the view list and enter the new name.
6. Repeat these steps as necessary until you've got the views you want. You can assign colors to all views by double-clicking the area on the right of the view name.



7. To delete an unnecessary view, select the view from the list and click the **-** button.
8. To move a view around in the list of views, click the up and down arrows above the views panel.
9. To select the view that's displayed whenever you load the project, set the **hero** dropdown to the appropriate view.

   You can now access the views in your project from the **view** dropdown menu of certain nodes' controls. You'll also notice that each view has its own button in the Viewer controls.

If you created many views, you may want them to appear in a dropdown menu rather than as separate buttons in the Viewer and node controls. To switch to using dropdown menus, disable **View selection uses buttons?** on the **Views** tab of the **Project Settings** panel.



Loading Multi-View Images

Once you have set up the views, you are ready to read your images into Nuke. To make things easier, the images you read in should have the view name or the first letter of the view name in the filename, for

example **filename.left.0001.exr**, **filename.l.exr**, or **lefteyefilename.0001.cin**.

If you are using **.exr** files, your files can contain both the input for the left eye and the input for the right eye, as **.exr** files support multiple views in a single file. With any other file types, you need to have separate files for the left and right inputs.

To Read Images in

1. Select **Image** > **Read.**
2. Navigate to the files containing the images intended for either the left or right eye (or in the case of **.exr** and **.mov** images, both eyes), and select **Open**.
3. Do one of the following:
   - If the images you want to read in contain a view name or the initial letter of one (for example, **left**, **right**, **l** or **r**) in their file names, replace this with the variable **%V** or **%v** in the **file** field of the Read node's controls. Use **%V** to replace an entire view name (for example, **left** or **right**), and **%v** to replace an initial letter (for example, **l** or **r**). When a variable is used, Nuke reads in the missing inputs and combines all inputs into a single output.

     For example, if you read in **image.left.cin** and changed the name to **image.%V.cin**, Nuke would read in both **image.left.cin** and **image.right.cin** with the same Read node, provided that views called **left** and **right** existed in your project settings. Both input images would be combined into a single output.



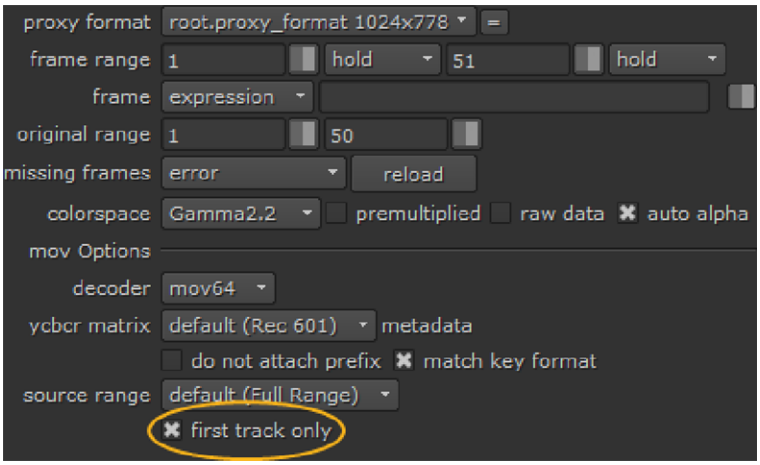
Note: Mac and Linux operating systems can be case-sensitive or case-insensitive. If your OS is case-sensitive, you'll need to make sure you use the correct case when naming your left and right views, as the **%v** variable can only retrieve the case used in the view name.

     You can also use the **%V** and **%v** variables at a directory level. For example, let's say you have set up views called **testleft**, **testmiddle** and **testright**, and you have the following directories and files:

```
mydirectory/testleft/image.testleft.cin
mydirectory/testmiddle/image.testmiddle.cin
mydirectory/testright/image.testright.cin
```

     If you now read in image.testleft.cin and changed the pathname to **mydirectory/%V/image.%V.cin**, all three inputs would be read in with the same Read node.
   - If the images you want to read in do NOT contain a view name or the initial letter of one (for example, **left**, **right**, **l** or **r**) in the file names and are not stereo **.exr** files, insert a Read node for each input and combine them into a single output using the JoinViews node (see below for instructions on how to do that).
   - If the images you want to read in are in the stereo **.exr** file format, you do not need to do anything. However, remember that not all **.exr** files contain multiple views. If you are using files that are not, follow the instructions in the first two points.

- If the images you want to read in are in the multi-view **.mov** file format, you need to open the Read node's properties and disable **First track only**.

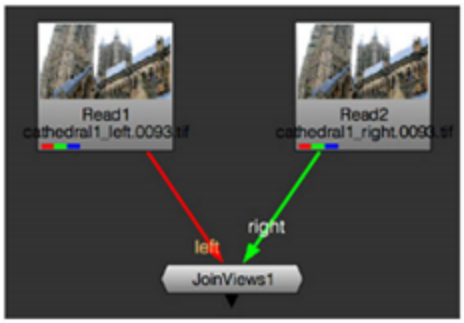


You'll notice that the **.mov** Read node in the Node Graph is now marked with V to denote multiple views. You can now switch between views using the buttons above the compositing Viewer.

Combining Views from Different Files into a Single Output

1. Select **Image** > **Read** to read in your image sequences containing the different views.
2. To insert a JoinViews node, select **Views** > **JoinViews**.
3. Connect the inputs of the JoinViews node into the appropriate Read nodes. There should be an input for each view you have created in the project settings. The inputs are labeled with the name of the view.

   If you have assigned colors to the views and checked **Use colors in UI?** on the **Views** tab of your project settings, the connecting arrows reflect the view colors. If this does not happen and the arrows are black, you may have connected the inputs the wrong way around. Check that you have connected each Read node to the correct input of the JoinViews node.



Nuke combines the inputs into a single output.

Displaying Views in the Viewer

You can only display the views that exist in your project settings. To see a list of these views, or to add or delete views, select **Edit** > **Project Settings** and go to the **Views** tab. For more information, see Setting Up Views for the Script.

To Display a Particular View

1. Add a Viewer into your script if you haven't already done so.
2. On top of the Viewer controls, do one of the following:
   - If you have checked **View selection uses buttons?** in the project settings, click the button of the view you want to display. For example, click the **right** button (assuming you have a view called **right** in your script).



   - If you haven't checked **View selection uses buttons?** in the project settings, select the view you want to display from the dropdown menu.



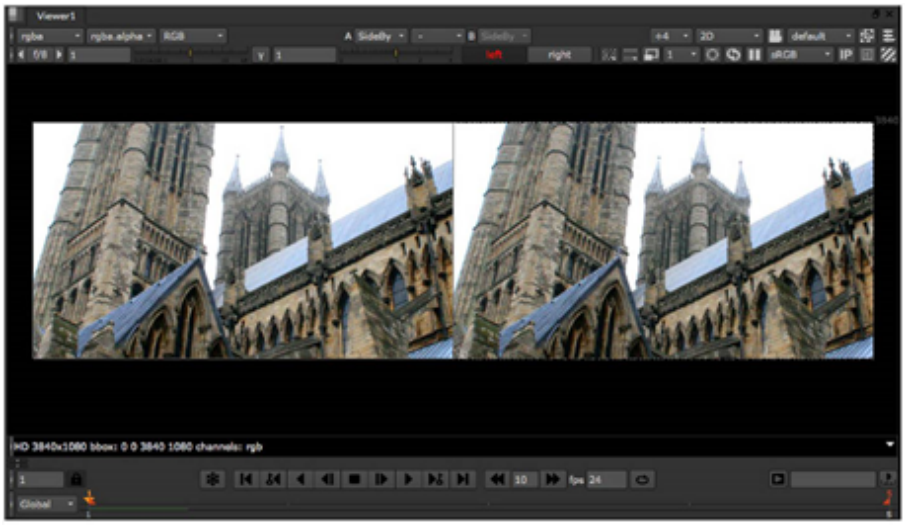
Note: Nuke lists the views in **.exr** files in the order they appear in the clip's header, so a view named '**left**' may not always be the first view displayed above the Viewer.
If your views do not appear in the correct order, you can rearrange them in the **Project Settings** > **Views** tab. See Setting Up Views for the Script for more information.

Tip: You can also press the **;** (semicolon) and **'** (forward single quote) keys to move between different views in the Viewer.

To Display Two Views Next to Each Other

1. Add a Viewer into your script if you haven't already done so.
2. If necessary, combine your views into a single output using the JoinViews node. For more information on how to do this, see Loading Multi-View Images.
3. Select **Views** > **Stereo** > **SideBySide** to insert a SideBySide node in an appropriate place in your script.
4. In the SideBySide node's controls, select the two views you want to display from the **view1** and **view2** dropdown menus. View1 is displayed on the left and view2 on the right.
5. If you want to display one view on top of another rather than next to it, check **vertical**. View1 is displayed above view2.
6. If you want to swap the views around in the Viewer, click the **swap** button.

   The Viewer displays the two selected views simultaneously, so you can easily compare them.



To Display a Blend Between Two Views

1. Add a Viewer into your script if you haven't already done so.
2. If necessary, combine your views into a single output using the JoinViews node. For more information on how to do this, see Loading Multi-View Images.
3. Select **Views** > **Stereo** > **MixViews** to insert a MixViews node into your script. This node displays a blend between two views in the Viewer, allowing you to check how elements in these views are aligned.

4. In the MixViews controls, use the **views** buttons or dropdown menus to select the two views to blend between.
5. To control the blend between the views, adjust the **mix** slider. Setting the slider to 0 or 1 displays only one of the views. Values between 0 and 1 produce different blends between the views.



Selecting Which Views Display Changes

By default, Nuke applies any changes you make to all views of the processed node. To apply changes to a particular view only (for example, the left view but not the right), you must first do one of the following:

- In the case of most nodes, split the view off in the node's controls.
- In the case of RotoPaint nodes, select the view you want to process from the **view** dropdown menu in the node's controls.

These methods are useful, for example, when you want to perform the same operation on both views but use different values for each.

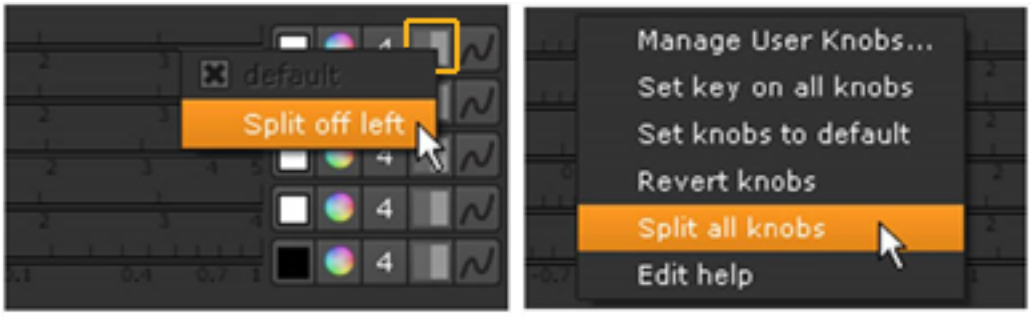
Splitting Views Off

Nuke allows you to split views off in order to apply changes to the existing views separately.

1. Insert a process node (for example, ColorCorrect) in the appropriate place in your script.
2. If you haven't already done so, attach a Viewer to the node. From the Viewer's controls, select the view you want to make changes to.



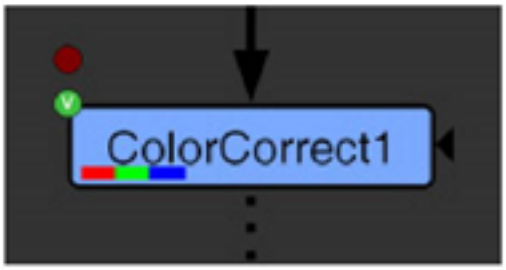
3. Open the node's controls.
4. Click the view button  next to the control you want to adjust. From the menu that opens, select **Split off [view name]**. For example, to apply changes to a view called **left**, select **Split off left**. You can also split all the node's controls by selecting **Split all knobs** from the right-click menu.



An eye appears on the view button  and the node gets a small green dot on it in the Node Graph to indicate that views have been split off.



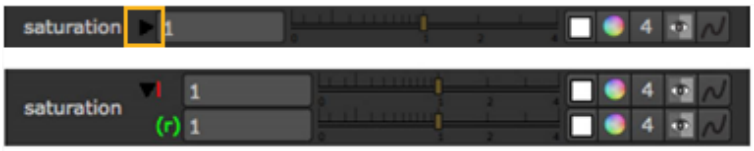
If you have assigned colors to the views and checked **Use colors in UI?** in your project settings, dots also appear on the node to indicate which views have been split off. For example, if you are using red for the left view and split off that view, a red dot appears on the node.



Any changes you now make using the control in question are only applied to the view you chose to split off. Changes to controls that have not been split off are still applied to all views.

To Show Separate Values for Each View

Once you have split off a view, you can apply changes to the existing views separately. Simply click on the small arrow on the left side of a control you have split off. This divides the control so that you can see the values for each view.



Adjusting the split control for only the current view and for all views separately.

To Unsplit Views

1. In the node's controls, click the **view** button .
2. From the menu that opens, select **Unsplit [view]**. For example, to unsplit a view called **left**, you'd select **Unsplit left**.
3. Repeat step 2 for all views you want to unsplit.

The view is unsplit, and all changes you made after splitting it off are lost.

Selecting the View to Process When Using the RotoPaint Node

To select the view to process:

1. Open the RotoPaint node's controls.
2. From the **view** dropdown menu, select the view you want to process. To apply changes to all views at the same time, select all the views separately.
3. If you selected to process just one view, make sure you are viewing the selected view in the Viewer when making your changes.

Performing Different Actions on Different Views

In case you need to perform totally different actions on the two views, you can add a OneView node to separate one view for processing.

To Extract a View for Processing

1. Select **Views** > **OneView** to insert a OneView node in an appropriate place in your script.
2. In the OneView node's controls, select the view you want to make changes to from the **view** dropdown menu.

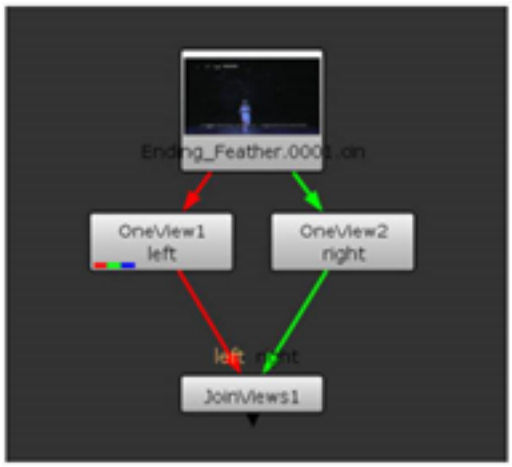
All views are extracted, and any changes you make are only applied to the view you selected (regardless of which view you are displaying in the Viewer).

To make changes to a different view, select it from the OneView node's **view** dropdown menu.

To merge views from two separate streams, select **Views** > **JoinViews** to combine the views (or delete the OneView node from your script).

If you need to extract all views, process them individually, and then merge them together, use the **Split and Join** menu item. This menu item is actually a combination of the OneView and JoinViews nodes. It first extracts all the views you have set up in your project settings and then merges them back together. It's no different to using several OneView nodes together with a JoinViews node, but speeds up work, because you don't need to add each node in a separate go. To use the menu item, select **Views** > **Split and Join**.

For example, if you have created views called **left** and **right** in your project settings and use a **Split and Join** menu item after your Read node, you get the following node tree:



You can then add any necessary nodes, such as color corrections, between the OneView and JoinViews nodes.

Reproducing Changes Made to One View

When rotoscoping, creating paint effects, or doing other operations dependent on image locality, you can have changes made to one view automatically reproduced in the other. This applies to the RotoPaint node, Roto node and any nodes, groups, or gizmos that have controls for x and y coordinates.

To reproduce changes made with the above nodes, groups, or gizmos, you need a disparity field that maps the location of a pixel in one view to the location of its corresponding pixel in the other view. You can create a disparity field using the O_DisparityGenerator plug-in, which is included in the Ocula plug-in set, or a 3D application. Once you have the disparity field, you can store it in the channels of an **.exr** file or use the Shuffle node to add the disparity channels in the data stream where you need them.

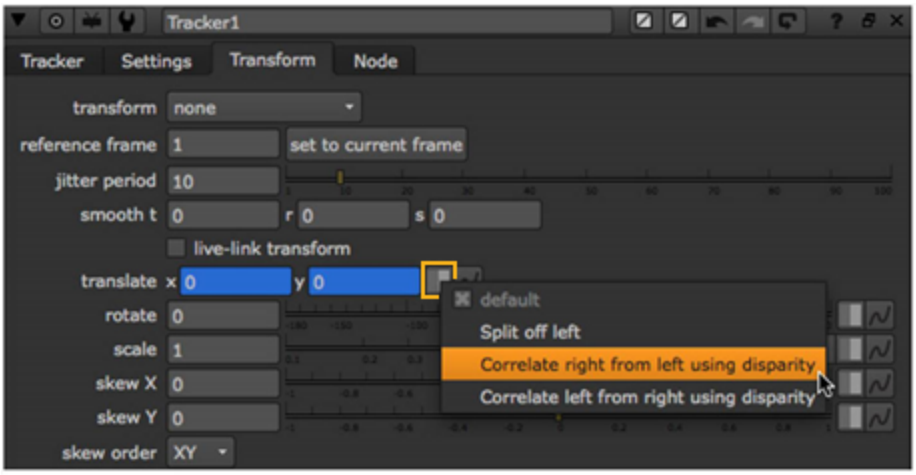
For more information on reproducing paint strokes, Bezier shapes, or B-spline shapes, see Reproducing Strokes/Shapes in Other Views.

Reproducing X and Y Values

Whenever there are values in any x or y control in Nuke for one view, you can automatically generate the corresponding values for the other view. This is true for both nodes and gizmos. For example, you can use a

Tracker node to track something in one view, and then have the track's x and y position generated for the other view automatically.

1. Make sure there is a disparity field upstream from the image sequence you are manipulating. If the image sequence is an **.exr** file, the disparity field can be included in its channels. Otherwise, you can use a Shuffle node or Ocula's O_DisparityGenerator plug-in to add it in the data stream.
2. Insert a node that has an x and y control after the image sequence you are manipulating.
3. Attach a Viewer to the node you added in the previous step, and make your changes in one view.
4. From the View menu next to the **x** and **y** controls, select **Correlate [view] from [view] using disparity**, for example **Correlate right from left using disparity**. This generates the corresponding x and y values for the other view.



If you have Foundry's Ocula plug-ins installed, you can also select **Correlate [view] from [view] with Ocula**. This way, extra refinements are made when creating the corresponding x and y values, and the results may be more accurate.

5. If you want to adjust the x and y values further, you need to adjust both views independently. Adjustments you make to one view are not automatically generated for the other.

Swapping Views

You can rearrange the views in your script using the ShuffleViews node. For example, you can swap the left and right views around in the pipeline, so that Nuke uses the left input for the right eye and vice versa.

To Rearrange Views

1. Select **Views** > **ShuffleViews** to insert a ShuffleViews node in an appropriate place in your script.
2. In the ShuffleViews controls, click **add** as necessary.

3. Use the buttons or dropdown menus to select which view to replace with which. For example, to swap the left and right views around, you need to make the following selections:
   - On one row, select **left** under **get**, and **right** under **from** ("get left from right"). The left view is now replaced with the right view.
   - On another row, select **right** under **get**, and **left** under **from** ("get right from left").

     The right view is replaced with the left view.



   If there aren't enough rows of buttons or dropdown menus on the ShuffleViews node's properties panel, click the **add** button to add a row.

   To remove unnecessary rows in the ShuffleViews node's controls, click the **delete** button next to the row you want to remove.

Converting Images into Anaglyph

You can use the Anaglyph node to convert your inputs into anaglyph images, which produce a 3D effect when viewed with 2-color anaglyph glasses.

To Convert Your Images into Anaglyph

1. Select **Views** > **Stereo** > **Anaglyph** to insert an Anaglyph node in an appropriate place in your script.
2. Use the **views** controls in the Anaglyph properties panel to select which views you want to use for the left and the right eye.

   Nuke converts the input images into grayscale anaglyph images. The left input is filtered to remove blue and green, and the right view to remove red.



3. To add color into the images, drag right on the **amtcolor** slider, or insert a value between 0 (grayscale) and 1 (colored) into the **amtcolor** input field.



If the images include areas that are very red, green, or blue, adding more color into them may not produce the best possible results.



4. To invert the colors and use the red channel from the right input and the blue and green channels from the left, check the **(right=red)** box.

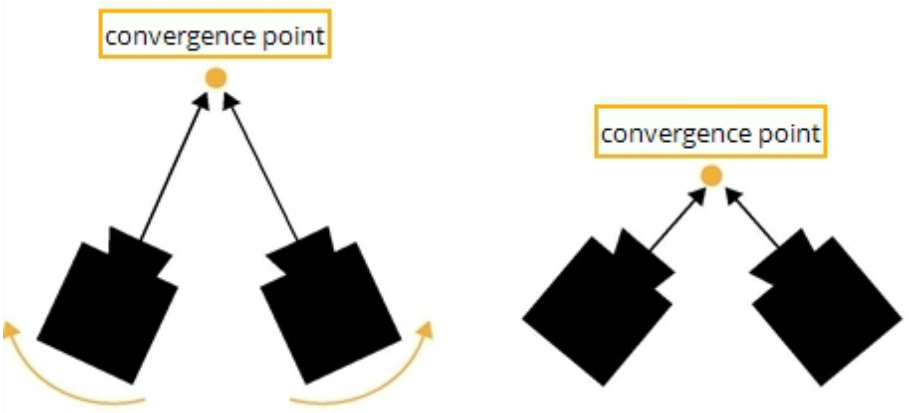


5. To control where the images appear in relation to the screen when viewed with anaglyph glasses, enter a value in the **horizontal offset** input field. To have the images appear in front of the screen, you would usually enter a negative value. To have the images appear further away, you would usually enter a positive value. (This is not the case if you have swapped the left and right views around.)

Tip: If you like, you can register the Anaglyph node as a Viewer Process. This way, you always have it as a viewing option in the Viewer's Viewer Process dropdown menu and can apply it to the current Viewer without having to insert the node in the Node Graph. Do the following:
1. Create a file called **menu.py** in your plug-in path directory if one doesn't already exist. For more information on plug-in path directories, see Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts.
2. To register the Anaglyph node as a Viewer Process, save the following in your **menu.py**:
```
nuke.ViewerProcess.register("Anaglyph", nuke.createNode, ("Anaglyph", ""))
```
3. Restart Nuke.
4. To apply the Anaglyph Viewer Process, select it from the Viewer Process dropdown menu in the Viewer controls.
5. To adjust the Anaglyph Viewer Process controls, select **show panel** from the Viewer Process dropdown menu.
For more information on Viewer Processes, see Using the Viewer Controls.

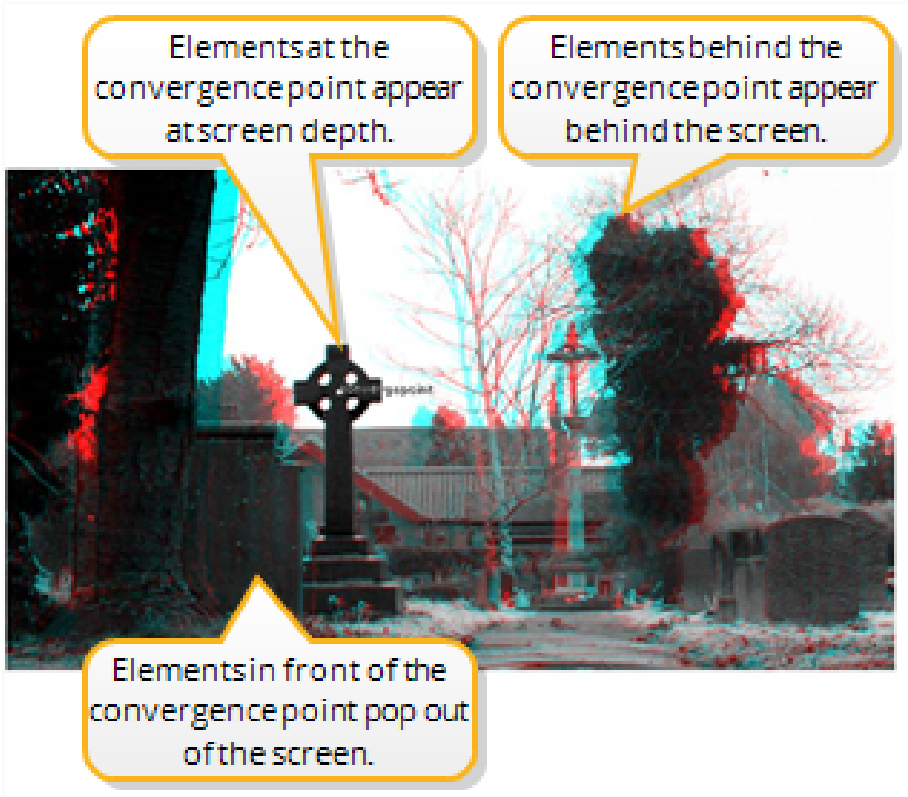
Changing Convergence

The ReConverge node lets you shift *convergence* (the inward rotation of the eyes or cameras) so that any selected point in the image appears at screen depth when viewed with 3D glasses. This point is called the *convergence point*. It is the point where the lines of sight from the two cameras meet.



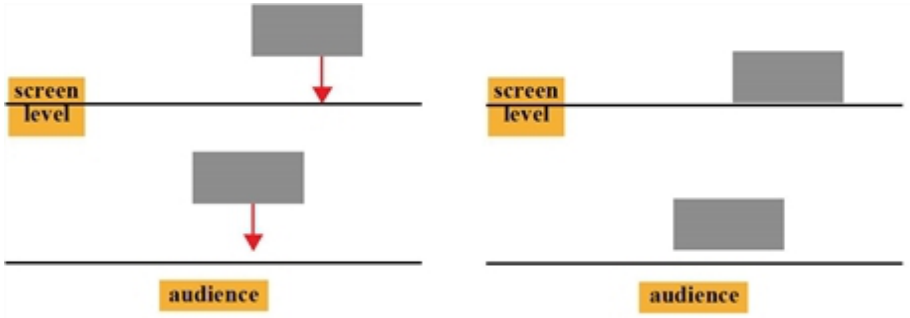
Changing convergence moves the point where the lines of sight from the two cameras meet.

At the convergence point, the different views in the image are aligned and appear at screen depth when viewed with 3D glasses. Anything behind the convergence point appears behind the screen, while anything in front of it seems to pop out of the screen. This is illustrated in the figure below.



Convergence controls where elements in the image appear in relation to the screen when viewed with 3D glasses.

Changing convergence changes the perceived depth of the images. It moves all the elements in the image backwards or forwards a fixed distance while keeping the distance between them the same. This is illustrated in the figure below, where the gray rectangles represent elements depicted in a stereo image.



Changing convergence changes the perceived depth of the images.

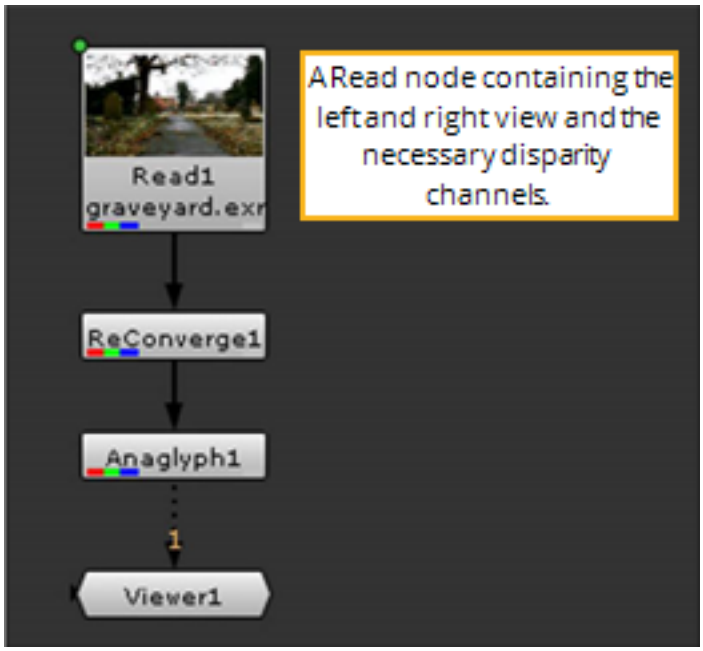
Often, the element of an image that appears closest to the audience is used as the convergence point. However, to make an element in your image jump out of the screen, you need to converge on something behind this element.

To calculate the convergence shift, the ReConverge node needs a disparity field that maps the location of a pixel in one view to the location of its corresponding pixel in the other view. To create the disparity field, you can use the O_DisparityGenerator plug-in, which is part of the Ocula plug-in set. Alternatively, you can create the disparity field in a 3D application. Once you have the disparity field, you can store it in the channels of an **.exr** file or use the Shuffle node to add the disparity channels in the data stream where you need them.

Note that the ReConverge node only shifts views horizontally, not vertically.

To Change the Convergence Point of a Stereo Image

1. Make sure there is a disparity field upstream from the image sequence whose convergence you want to change. If the image sequence is an **.exr** file, the disparity field can be included in its channels. Otherwise, you can use a Shuffle node or Ocula's O_DisparityGenerator plug-in to add it in the data stream.
2. From the Toolbar, select **Views** > **Stereo** > **ReConverge** to insert a ReConverge node after the image sequence whose convergence you want to adjust.
3. Attach a Viewer to the ReConverge node.
4. To better view the effect of the ReConverge node, insert an Anaglyph node (**Views** > **Stereo** > **Anaglyph**) between the ReConverge node and the Viewer.



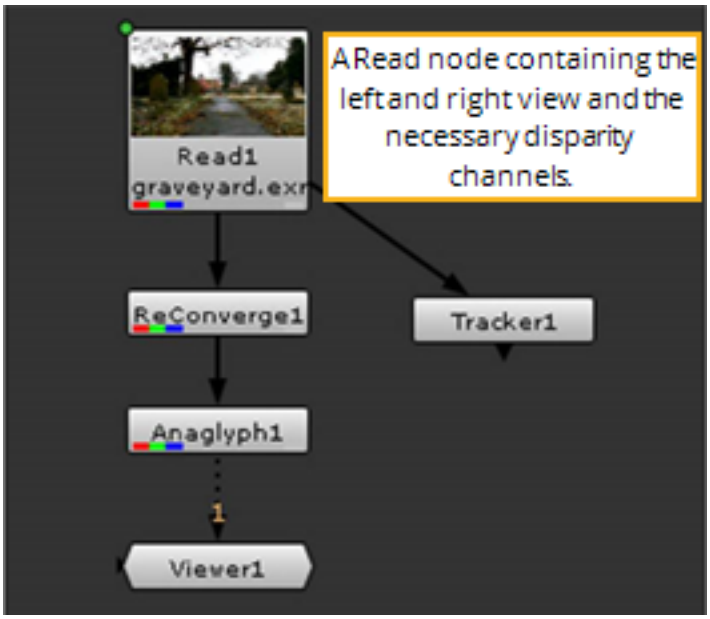
5. Make sure the ReConverge properties panel is open. You should see the convergence point overlay in the Viewer. Drag the point on top of the point you want to appear at screen level when viewed with 3D glasses. The convergence shifts to this location.

   You can also move the convergence point by entering the point's x and y coordinates in the **Convergence upon** fields.
6. By default, the ReConverge node moves the right view to achieve the convergence shift. However, if you like, you can use the **Mode** dropdown menu in the ReConverge controls to move the left view instead (select **shift left**) or move both views equally (select **shift both**).
7. If necessary, adjust the offset for convergence (in pixels) in the ReConverge controls. To bring all elements of your image forward from the screen level, enter a positive value in the **Convergence offset** field. To move all elements further away, enter a negative value.

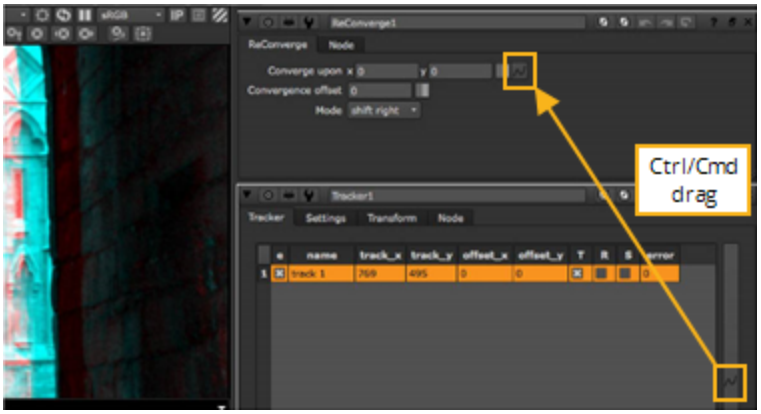
It is also possible to use the same element as the convergence point throughout the image sequence. You can, for example, have the same actor always appear at screen depth. To converge on the same element throughout the sequence, link the ReConverge node with a Tracker node.

Using the Same Convergence Point Throughout a Sequence

1. Insert a Tracker node after the image sequence whose convergence you want to adjust.



2. Track the point that you want to appear at screen level throughout the sequence. For more information on how to use the Tracker node, refer to Tracking and Stabilizing .
3. When you have the track animation data, apply it to the ReConverge node's **Converge upon** control via a linking expression. The easiest way to do this is to **Ctrl**/**Cmd**+drag the animation button next to the **tracks** list to the animation button next to the **Converge upon** control.



Previewing Stereoscopic Images

You can preview stereo images using the Viewer stereo modes or by flipbooking the sequence using Nuke's default flipbook.

Previewing Using the Viewer Stereo Modes

The Viewer stereo modes allow you to see both views at once, in either anaglyph or interlaced per scanline. To enable of Viewer stereo mode, do the following:

1. Right-click in the Viewer to display the context-sensitive menu.
2. Navigate to **Stereo Modes** and select the required mode:
   - **Single** - the default mode, shows only the view selected using the buttons above the Viewer.
   - **Anaglyph** - produces a 3D effect when viewed with 2-color anaglyph glasses.
   - **OpenGL Stereo** - displays both views at once on a 3D monitor for review purposes. See Displaying OpenGL Stereo in Comp Viewers for more information.
   - **Interlace** - shows both views, but alternately per scanline.



Anaglyph Stereo Mode



Interlaced Stereo Mode

Displaying OpenGL Stereo in Comp Viewers

The Viewer **OpenGL Stereo** mode allows you to see both views at once on a 3D monitor for review purposes. OpenGL Stereo is only supported on NVIDIA Quadro series GPUs and AMD Radeon Pro series GPUs on Windows and Linux OS.

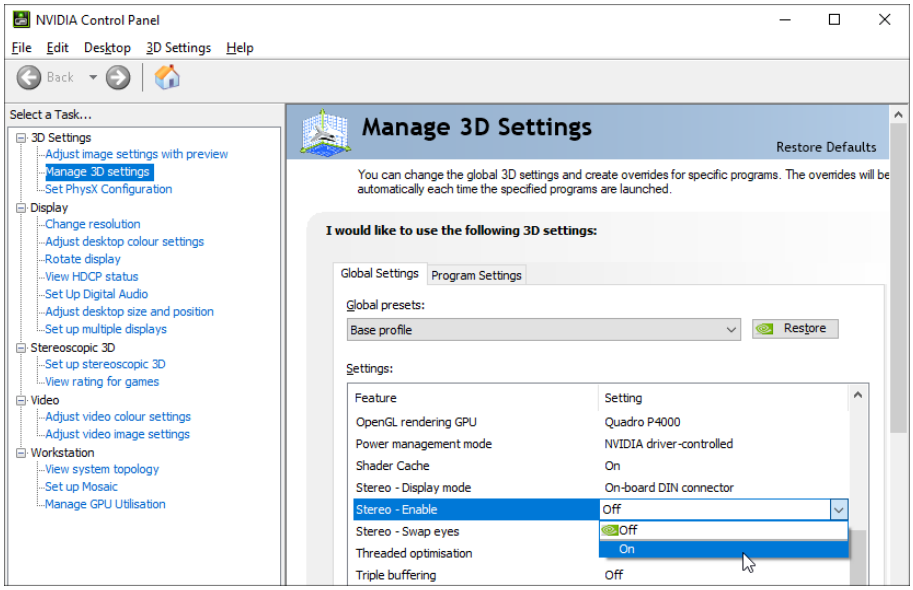
Note: OpenGL Stereo mode is not supported on Mac due to limitations in macOS and not supported with AMD GPUs on Linux.

Enabling OpenGL Stereo Output

Windows

To enable NVIDIA GPU stereo output:

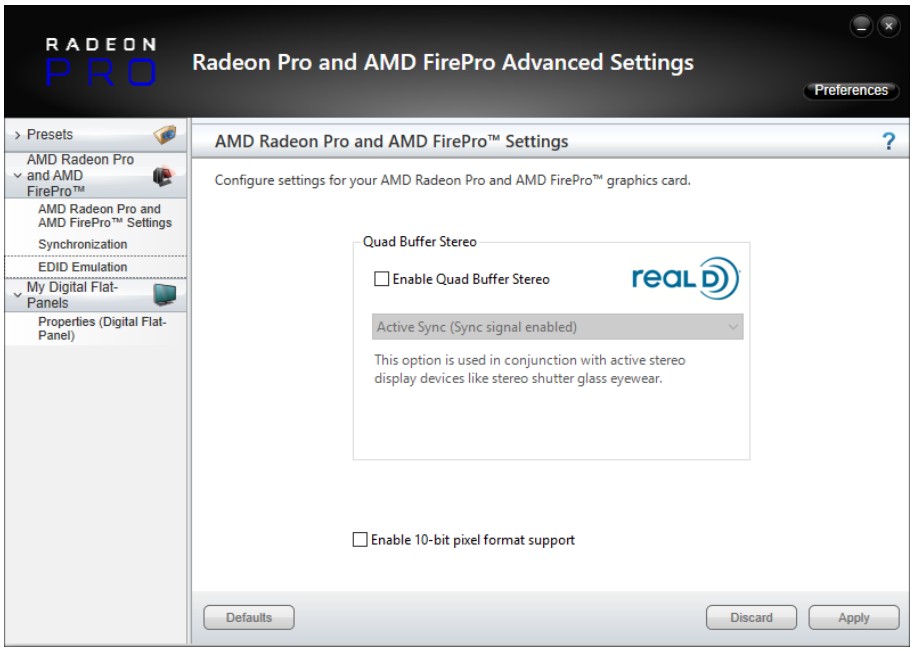
1. Right-click on the desktop and select NVIDIA Control Panel.
2. Navigate to **3D Settings** > **Manage 3D Settings** > **Stereo - Enable**.



3. Proceed to Switching to OpenGL Stereo Output.

To enable AMD GPU stereo output:

1. Double-click the AMD taskbar icon and select **Advanced Settings**.
2. Navigate to the AMD Pro Settings and check **Enable Quad Buffer Stereo**.



3. Select either **Auto-Stereo (Horizontal Interleaved)** or **Auto-Stereo (Vertical Interleaved)** and click **Apply**.
4. Proceed to Switching to OpenGL Stereo Output.

Linux

To enable NVIDIA GPU stereo output:

1. Open a command prompt and enter:
   `nvidia-xconfig --stereo=3`

Tip: For more information on the **nvidia-xconfig** utility, please see the **man** page: `man nvidia-xconfig`

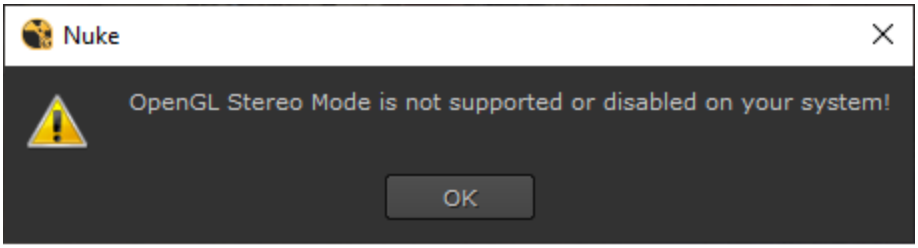
2. Proceed to Switching to OpenGL Stereo Output.

Note: **OpenGL Stereo** mode is not supported with AMD GPUs on Linux.

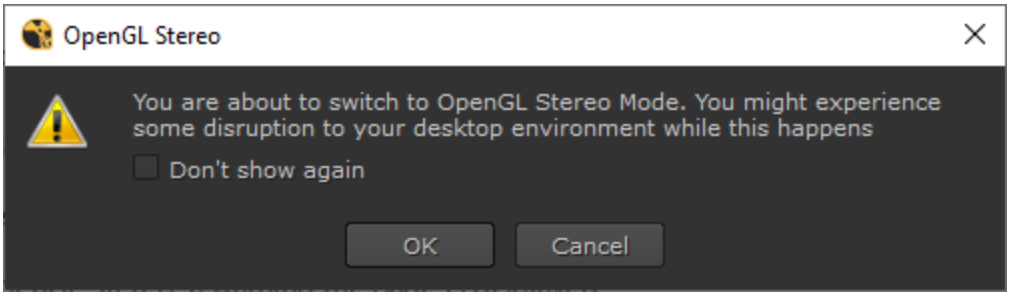
Switching to OpenGL Stereo Output

1. Right-click in the Viewer to display the context-sensitive menu.
2. Navigate to **Stereo Modes** and select **OpenGL Stereo**.

If you select **OpenGL Stereo** mode before enabling your GPU settings, a warning is displayed.



The first time you select OpenGL Stereo, a warning message is displayed.



Tip: You can disable the warning by enabling **Don't show again** and clicking **OK**.

OpenGL Stereo is displayed in the Comp Viewer.

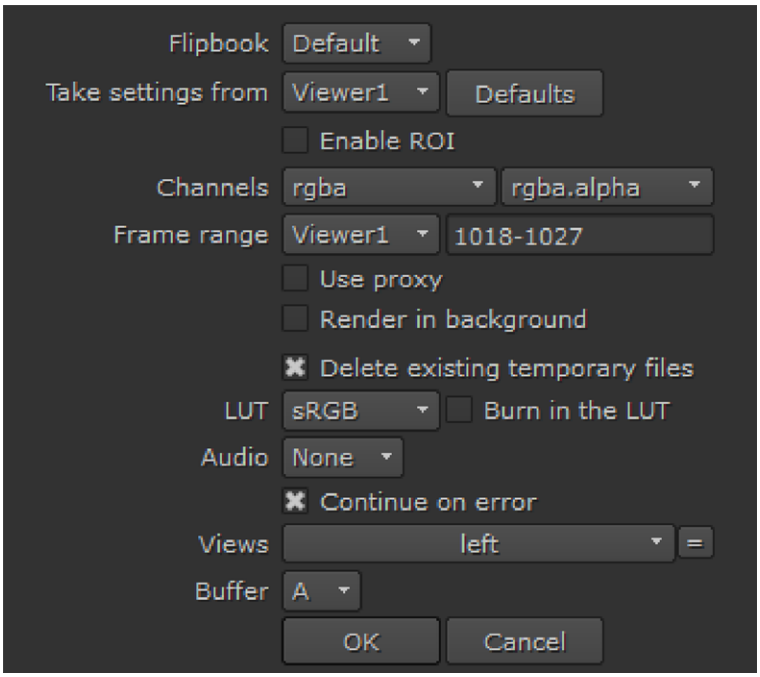
Flipbooking Stereo Sequences

To flipbook stereo images, do the following:

1. Select the node that you want to flipbook.
2. Select **Render** > **Flipbook Selected** from the main menu bar (or press **Alt**+**F**).

   You can also press the **Flipbook this Viewer** button at the bottom-right of the Viewer. This flipbooks the nodes that are connected to the Viewer.

   A dialog opens.



3. Check that your settings are correct in the dialog. The settings in the dialog are almost identical to the settings when you flipbook non-stereo sequences. For more information about these settings, see Flipbooking Sequences.
4. In the **frame range** field, enter the frame range you want to preview (for example, 1-35 or 1-8 10 12-15).
5. Using the **View** control, select the view you want to flipbook, either **left** or **right**.
6. Click **OK**.

   Nuke renders the selected view as a temporary sequence using the frame range and resolution defined in the script's settings. This may take a few moments.
7. After the render is complete, Nuke launches **Flipbook** Viewer and loads in the temporary sequence.

   You can play it back and view it using **Flipbook** Viewer controls.

Rendering Stereoscopic Images

You can render several views using a single Write node into separate files per view. If you're using **.exr** files, you can choose to write all views to a single file or multiple files per view.

Rendering EXR Files

To render views to a single **.exr** file:

1. Select **Image** > **Write** to insert a Write node in an appropriate place in your script.

2. In the Write node's controls, select **exr** from the **file type** dropdown menu.
3. From the **views** dropdown menu, select the view(s) you want to render, for example **left, right**.
4. Adjust any other Write controls as necessary and click **Render**. Nuke prompts you for the frames to render.

   Nuke writes the selected views into a single file.

Rendering Other File Formats

To render other file extensions:



Note: You can also render **.exr** files to a separate file per view using this process.

1. Select **Image** > **Write** to insert a Write node in an appropriate place in your script.
2. In the Write nodes' controls, select the file type of your images from the **file type** dropdown menu.
3. When entering names for the rendered image sequences, you can use the variable **%V** (with a capital V) to represent the words **left** and **right** (or any other full view names) in the file names, for example **filename.%V.####.exr**. To represent the letters **l** and **r** (or the first letters of any views), use the variable **%v** (with a lower-case v) instead. When rendering, Nuke then fills this in with left, right, l, or r, and renders all views you specify in the next step.
4. Adjust any other Write controls as necessary and click **Render**. Nuke prompts you for the frames to render as well as the views to execute (assuming you have set up several views in the project settings).

Nuke renders several views, but writes them into separate files. If you did not specify a view in the file names (using either the name of the view, its first letter, or a variable), you can only render one view.



Note: For command line renders, you can pass the -**view** argument with a list of view names to render, separated by a comma. If you do not specify a -**view** argument, Nuke renders all views.

Working with Graph Scope Variables

We have developed a native variable-enabled workflow that supports Multishot compositing, adding ways to make workflows more efficient, reliable and scalable.

The links below take you to pages outlining each of these features, and explaining how they can help with workflows managing multiple shots at once.

Variable features:

- Using Graph Scope Variables in Your Workflows
  How Variables can unlock more flexible workflows across multiple shots.
- Configuring Visibility with the Variables Panel
  Easily view and manage Variables in different configurations.
- Using the VariableSwitch Node
  Switch between Variables seamlessly, enabling and disabling them across multiple shots.
- Using the VariableGroup Node
  A way to group Variables, organising, and speeding up Variable usage.
- Using Graph Scope Variables in the New 3D System
  Use Variables in the new 3D system to unlock more flexible workflows.
- Using Python with Graph Scope Variables
  Use Python to edit Variable values and trigger custom behavior.
- Using Variables with the Root Node
  Use Variables to control parameters that affect the entire Nuke script,
- Specifying Variables for Command Line Render
  Specify Graph Scope Variables via the command line for rendering.

Other features you may find useful for multishot workflows:

- Viewing Grouped Nodes with Group View
  View and edit nodes contained within Groups in the context of the main node graph.
- Using Link Nodes
  A quick way to edit two or more of the same node at once, useful in large node graphs.



Tip: Using Using Graph Scope Variables in Your Workflows is a good place to get started.

Tip: To see these features in practice, also see our Multishot Tutorial.

Using Graph Scope Variables in Your Workflows

Graph Scope Variables (GSVs) are an important tool, enabling a range of workflows, including a native multishot compositing workflow in Nuke.

They allow you to define, store, and access the data required for multiple contexts or scopes in a single Nuke script.

- Once Variables are created, you can more easily manage scopes by using VariableGroup nodes in your node graph.
- You can also switch the scope based on the Variables available with the VariableSwitch node.

These **Variables** provide the flexibility to work at scale and speed up laborious workflows, all whilst ensuring consistency and continuity across the project.



Note: You can think of Variables as containers that hold information, which can then be referenced and called upon in various different locations.



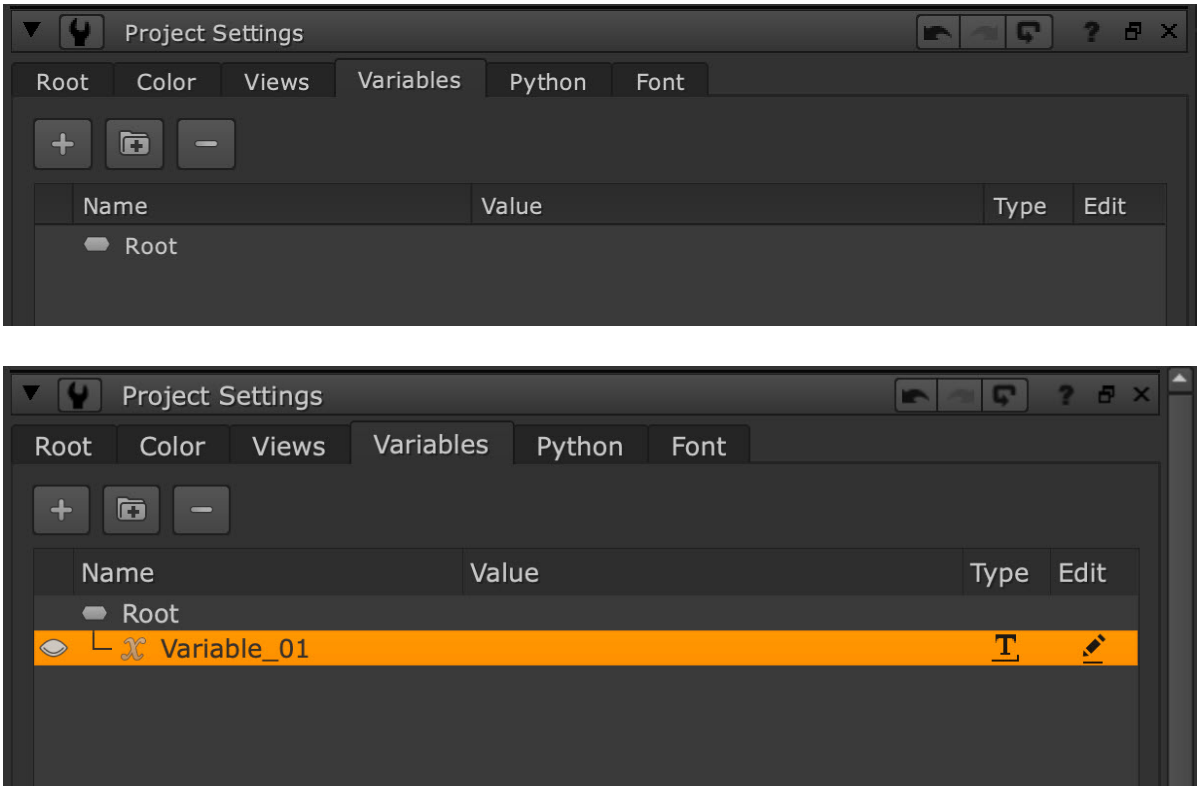
Note: Differences between Graph Scope Variables and the Graph State Variables in Katana: Although both provide similar functionality, VariableGroup nodes in Nuke automatically inherit GSVs from their parent(s) and allow these to be overridden to define a different or modified 'Scope' for the group.

Managing Variables

There are two locations where you can view and manage Variables. One at the root level within the **Project Settings** panel, and another one inside every VariableGroup node or LiveGroup node.

Project Settings

Open **Edit** > **Project Settings**, and navigate to the **Variables** tab:

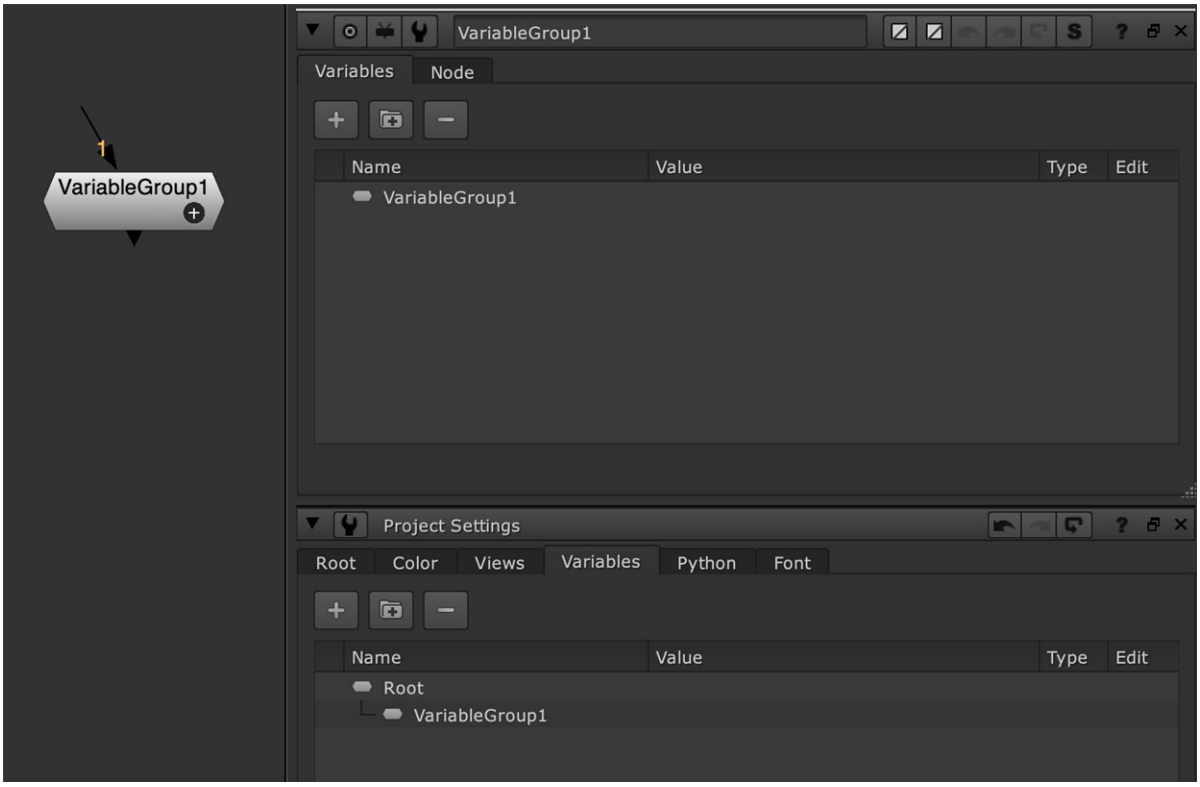


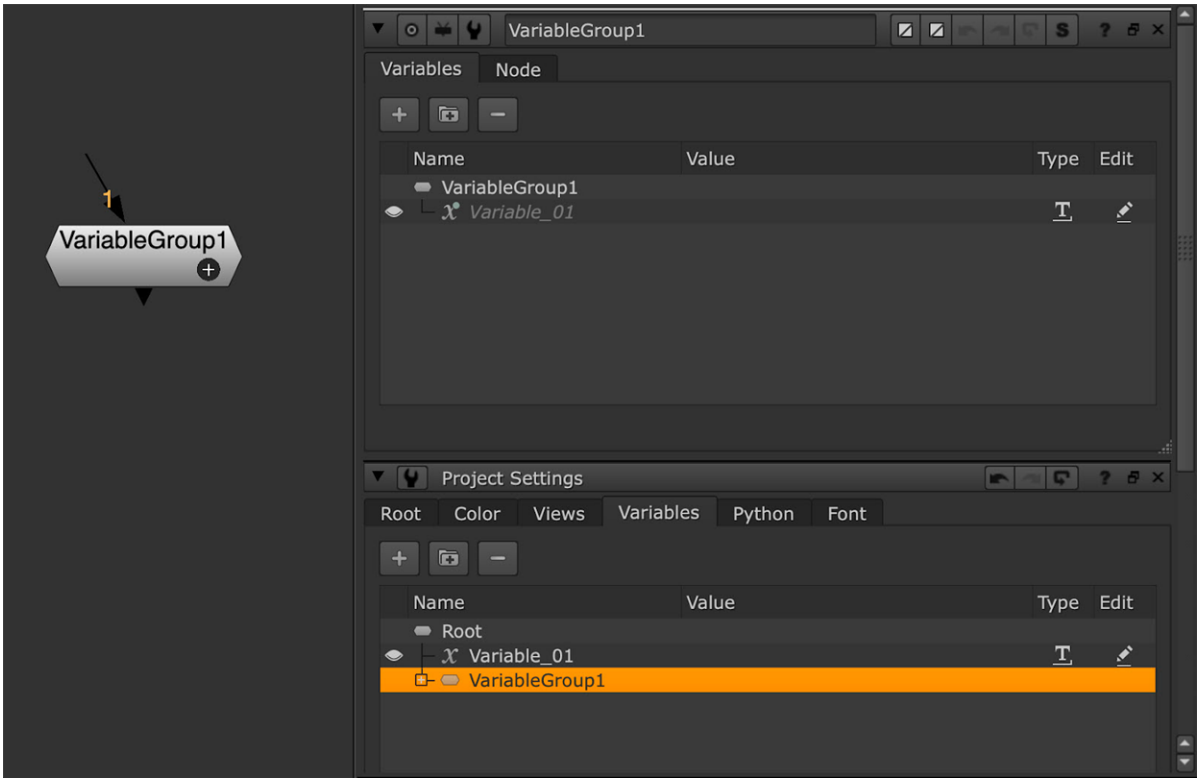
This displays all the VariableGroups, Variable Sets, Variables and Variable Values within a project (.nk script).

OR

Inside the Node

Create a VariableGroup node or LiveGroup node, and navigate to the **Variables** tab:


1
VariableGroup1
VariableGroup1
Variables
Node
Name
Value
Type
Edit
VariableGroup1
Project Settings
Root
Color
Views
Variables
Python
Font
Name
Value
Type
Edit
Root
VariableGroup1


1
VariableGroup1
VariableGroup1
Variables
Node
Name
Value
Type
Edit
VariableGroup1
Variable_01
Project Settings
Root
Color
Views
Variables
Python
Font
Name
Value
Type
Edit
Root
Variable_01
VariableGroup1

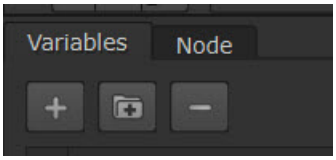
Note: The Variables tab in the VariableGroup or LiveGroup node shouldn't be considered the whole knob value but a convenience to allow Variables to be edited more easily, and making it possible to change an inner group GsvKnob value from its parent VariableGroup node. To view all VariableGroup nodes, navigate to the **Project Settings** Variables tab.

Note: The Variables (GsvKnob) can also be obtained via Python. See Using Python with Graph Scope Variables.

Tip: A Variables tab also shows on the LiveGroup node. This allows you to work with Variables on shared scripts.

Adding and Removing Variables

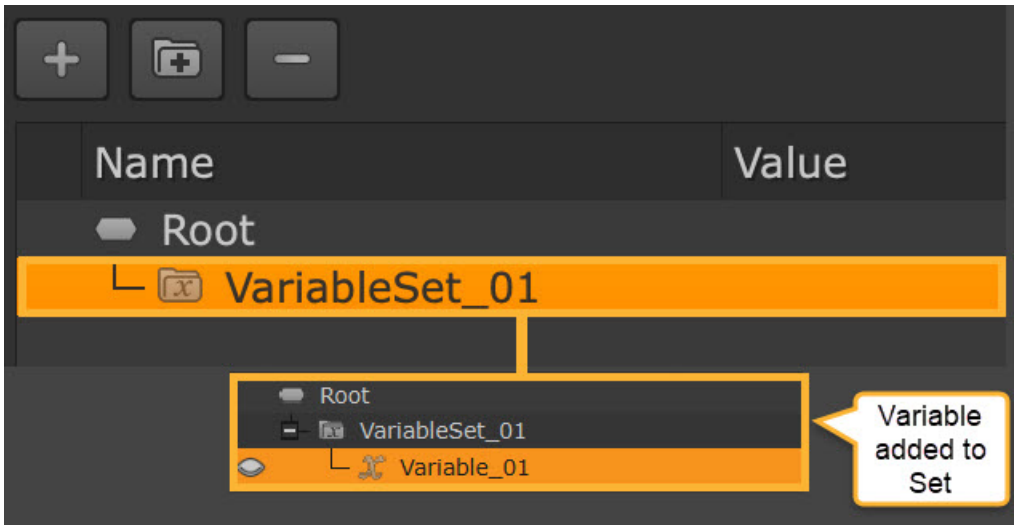
The **Variables** tab allows you to create, edit and remove Variables and Variable Sets.



- To **add** a **Variable** on its own, select the button. This will add a single Variable to whichever group or set you currently have selected.



- To **add** a **Variable Set**, select the folder button. This acts as a group of Variables, and you can add further Variables within it.



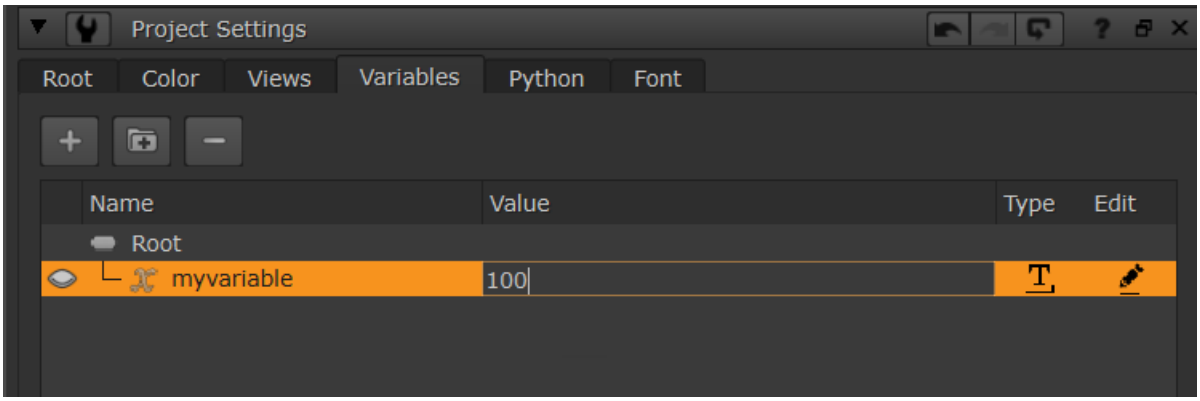
• To **remove** a Variable or Variable Set, select the button.

Setting Up Variables

Name your Variable something unique and give it a **Value**.

The **Name** is what will be used elsewhere in your Nuke script to reference this Variable.
The **Value** is the data that the Variable holds, to be substituted in.

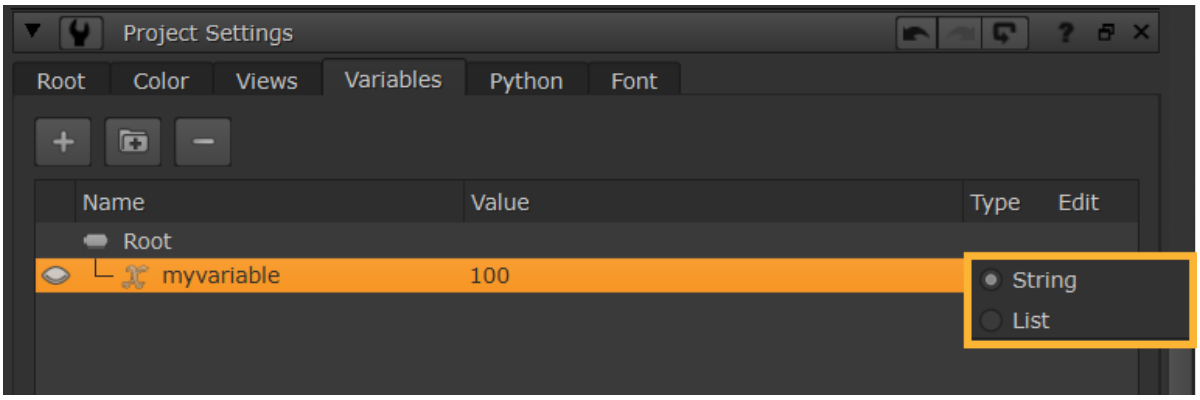
To edit a **Name** or **Value** simply double click in the relevant field.



There are two types of **Value**:

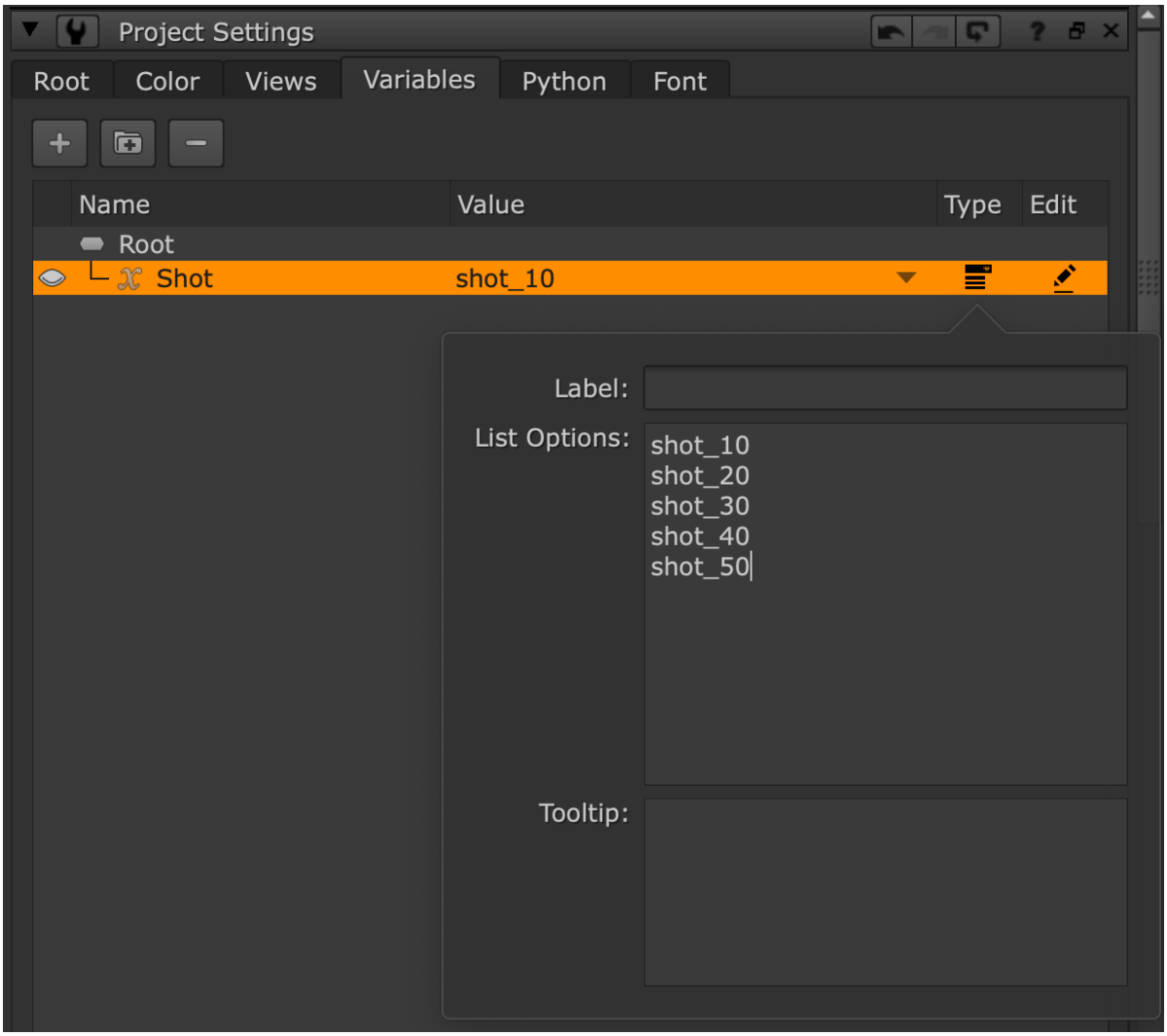
String is a single text value. This is the default option.

List allows you to create multiple string options for what are the possible acceptable values for the Variable.



Note: Any updates to the Variable Names will update the Variables tab in both the VariableGroup node and the Project Settings panel. Valid Names are the ones that begin with an alpha character [a-zA-Z] or '_' (underscore), followed by any number of alphanumeric characters or underscores. Examples of valid names are: '_shot', 'Shot', 'shot01'. Non compliant names will be ignored by the GsvKnob.

The List option can be used to add a list of shots, for example:



Tip: You can change the visibility of each Variable in the Variables Panel. See Configuring Visibility with the Variables Panel.

Tip: The List option is useful when using the VariableSwitch node, allowing you to switch inputs based on the current scope being viewed or rendered. Because you have a fixed set of possible Values, you can use the values to choose between specific shots. See Using the VariableSwitch Node.

Referencing Variables

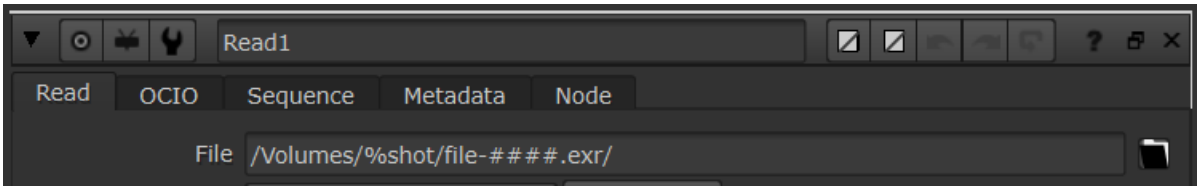
- Once created, you can reference your Variables in nodes and expressions by referencing **%{Variable}**, for example: `%Variable_01`
- Additionally, Variables within Variable Sets can be referenced if you prepend the Variable Set name **%{VariableSet.Variable}**, for example: `%VariableSet_01.Variable_01`
- **Recursive evaluation** of Variables is also supported, giving the option of creating Variables which depend on one another. Special delimiters '{' and '}' can be used to indicate where a valid Graph Scope Variable expression starts and finishes.

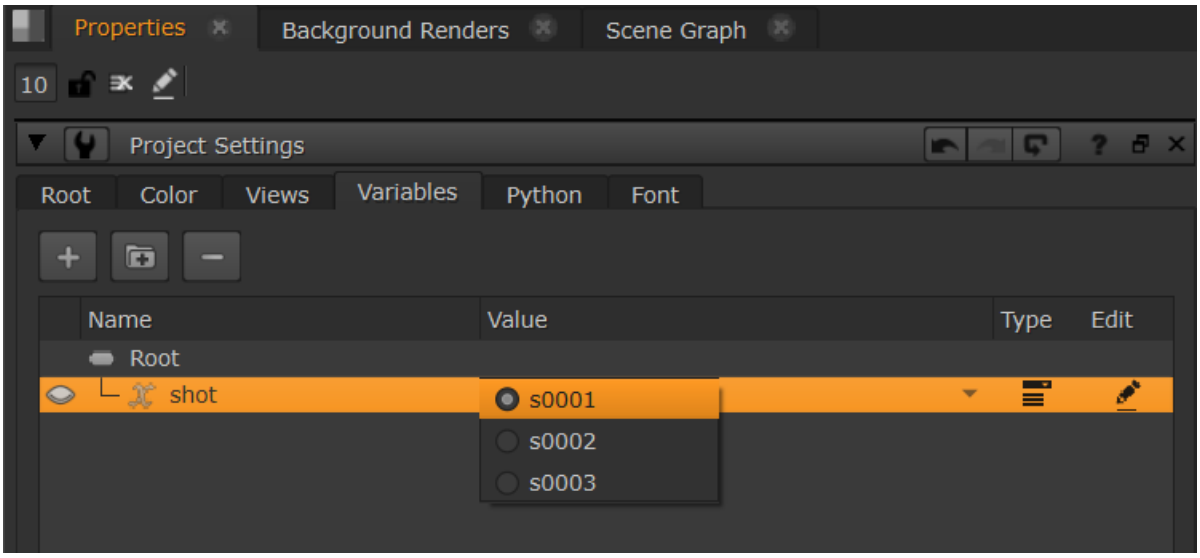
When referencing Variables in expressions, the order of evaluation of expressions in Nuke after the introduction of Graph Scope Variables is:

- Graph Scope Variables
- TCL
- OpenAssetIO entity references
- Nuke's view and frame ranges

To learn more about expressions in Nuke, see Expressions.

Variables Example 1: File Path

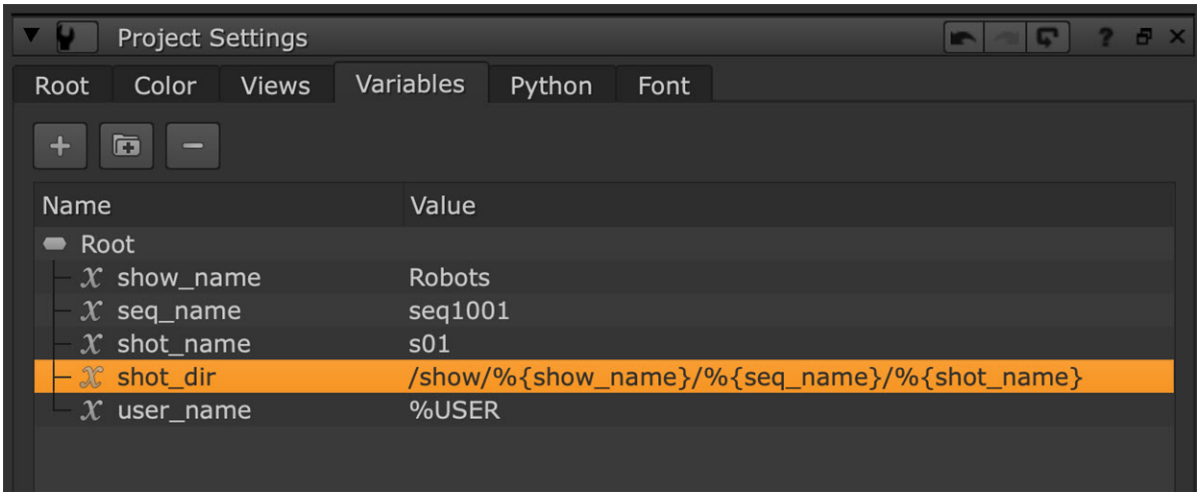




With a file path set to `/Volumes/%shot/file-####.exr`, and a shot Variable defined with value `s0001`, the referred path would be evaluated to: **`/Volumes/s0001/file-####.exr`**.

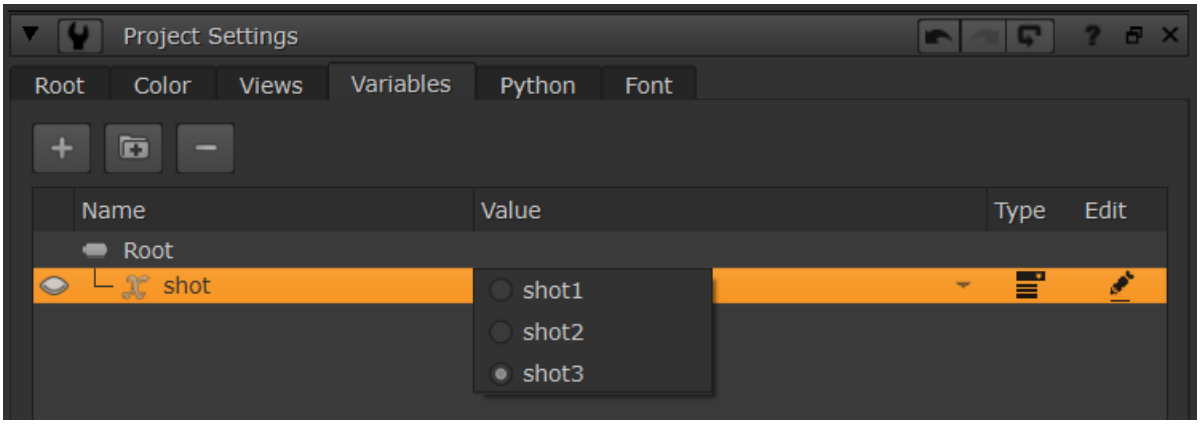
Variables Example 2: Recursive Path Evaluation

Recursive evaluation allows you to create Variables which depend on one another, by using special delimiters '{' and '}' to indicate where a valid Graph Scope Variable expression starts and finishes. This means you can include multiple Variables in one path:

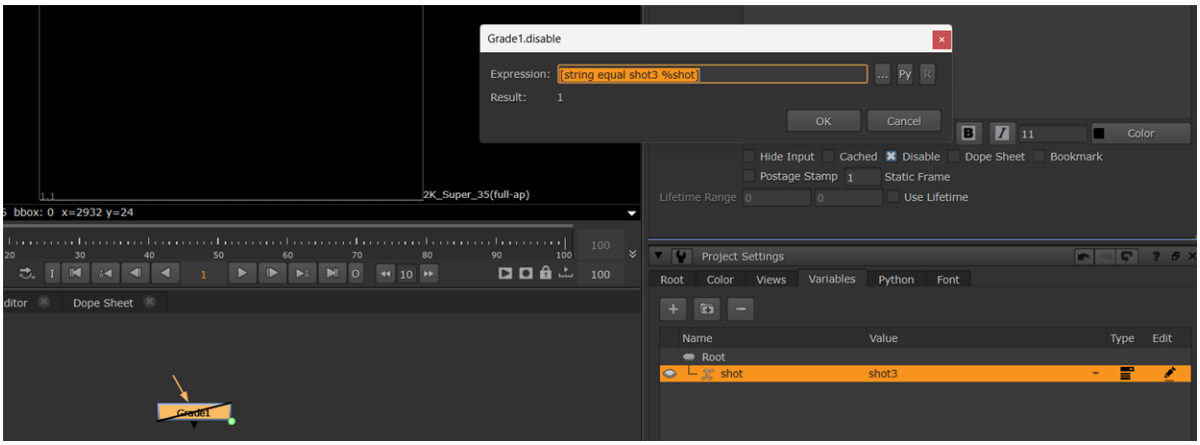


Note: No validation is used against Variable values.

Variables Example 3: Disable Knob



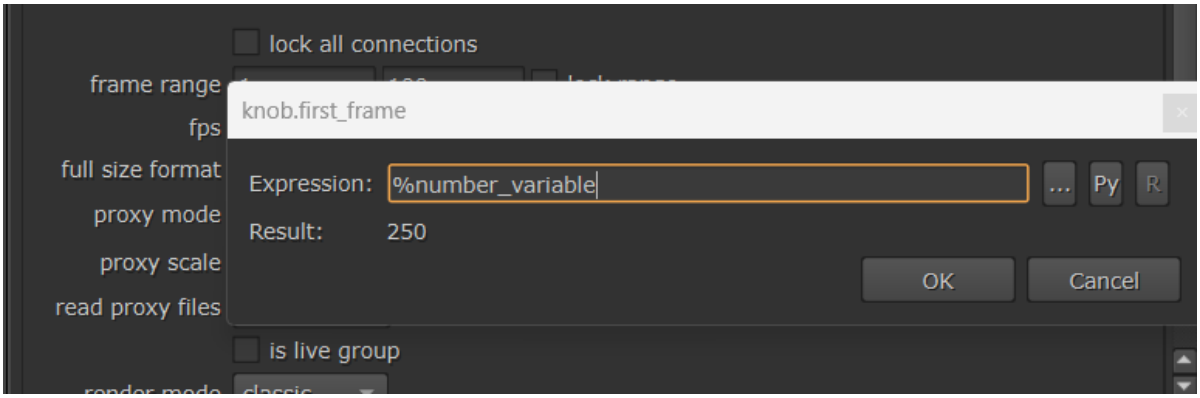
You could use a **List** Variable in an expression on the disable knob `[string equal shot3 %shot]` — this would disable the node when the shot Variable is equal to `shot3`.



Note: You can also use the command line to specify Variables for command line render. See Specifying Variables for Command Line Render.

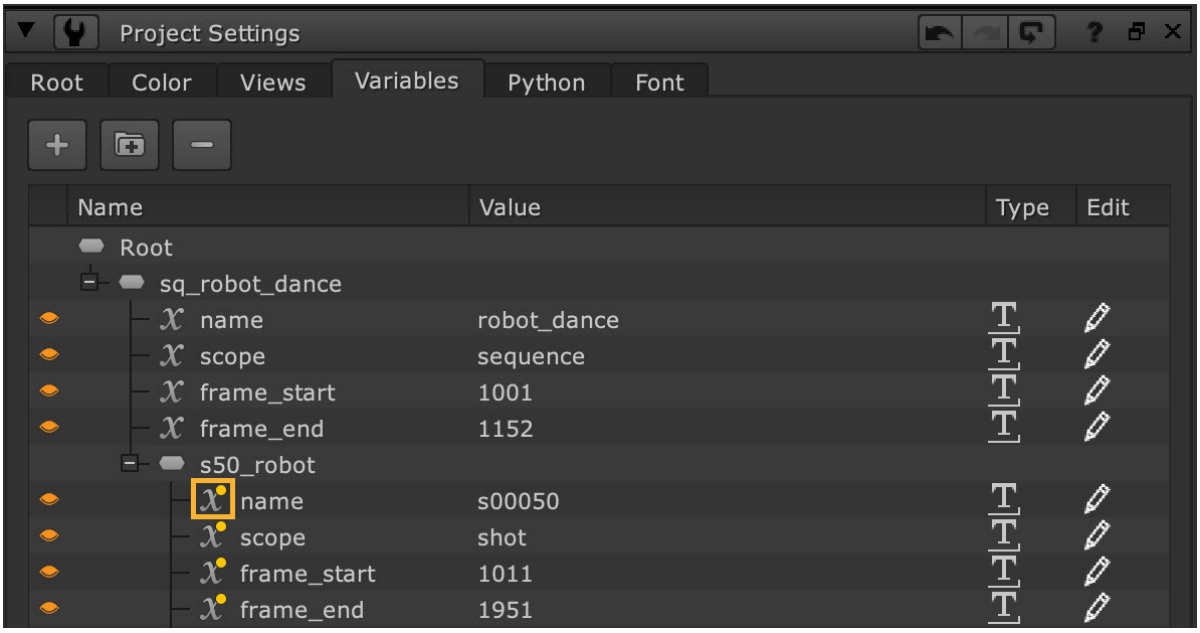
Variables Example 4: Root Node

You can also reference Variables in expressions in several knobs on the Root node - **frame range** (**first_frame** and **last_frame**) and **fps** knobs. See the full steps on Using Variables with the Root Node.



Overriding Variables

New Variables can override a Variable in a parent group, this is represented in the UI with a **yellow icon** next to the Variable. Removing the override Variable will revert it back to inheriting from the parent.



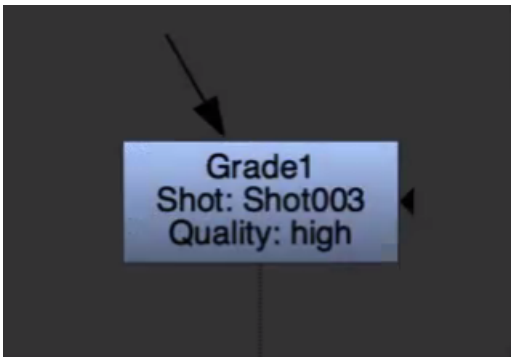
Tip: Setting up Variable groups is made easier with the use of VariableGroup nodes, explained here — Using the VariableGroup Node.

Tip: See our Multishot Tutorial to learn more about how Graph Scope Variables, VariableGroup, and VariableSwitch nodes work together in practice.

Variables in Your Project

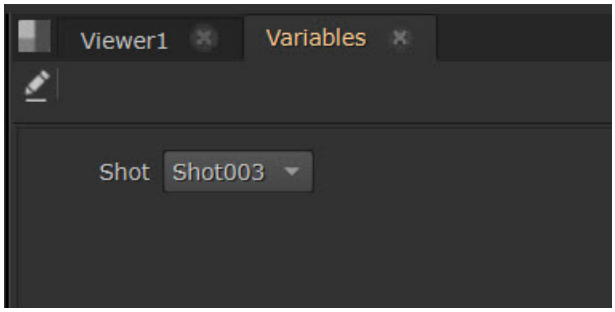
In the Node Graph

Once Variables are set, you can also see them in the Node Graph, where they show resolved on the node label.



In the Variables Panel

The Variables Panel exposes a tailored list of Variables in a separate panel. See Configuring Visibility with the Variables Panel.

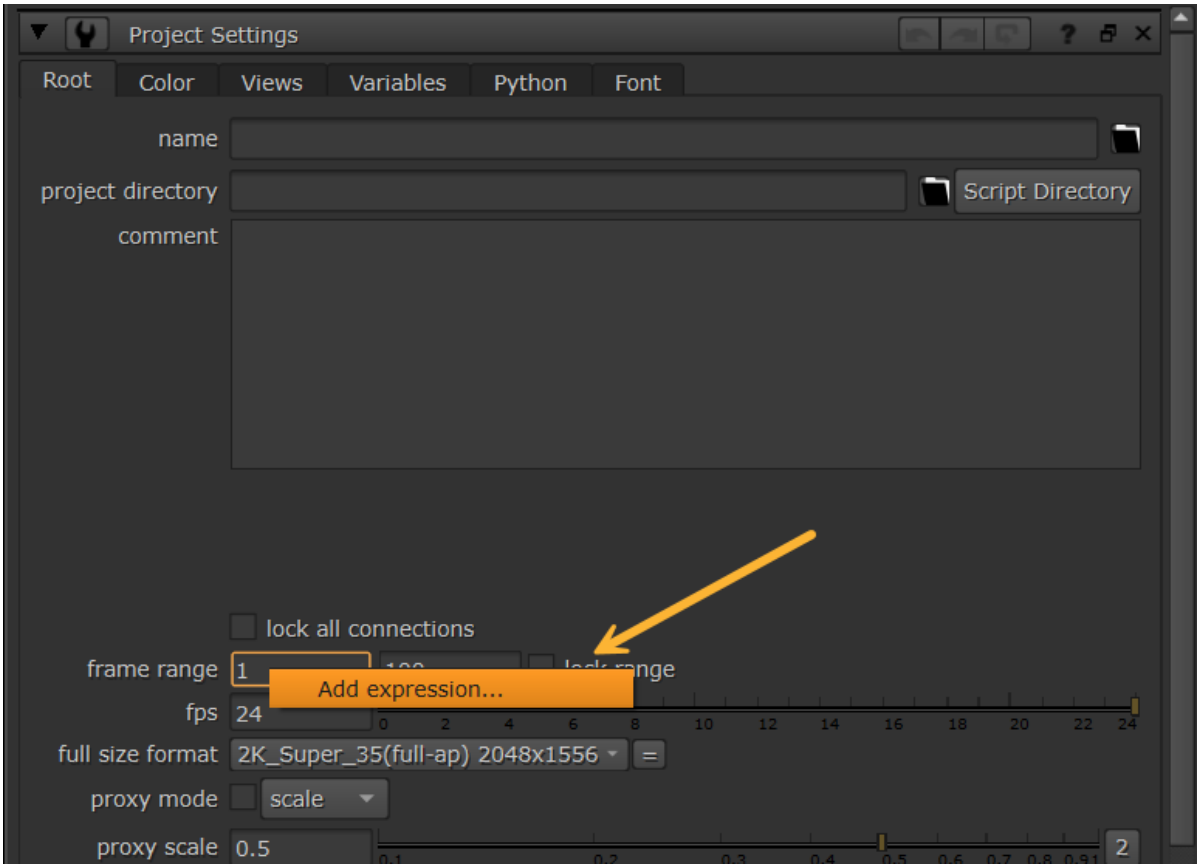


Using Variables with the Root Node

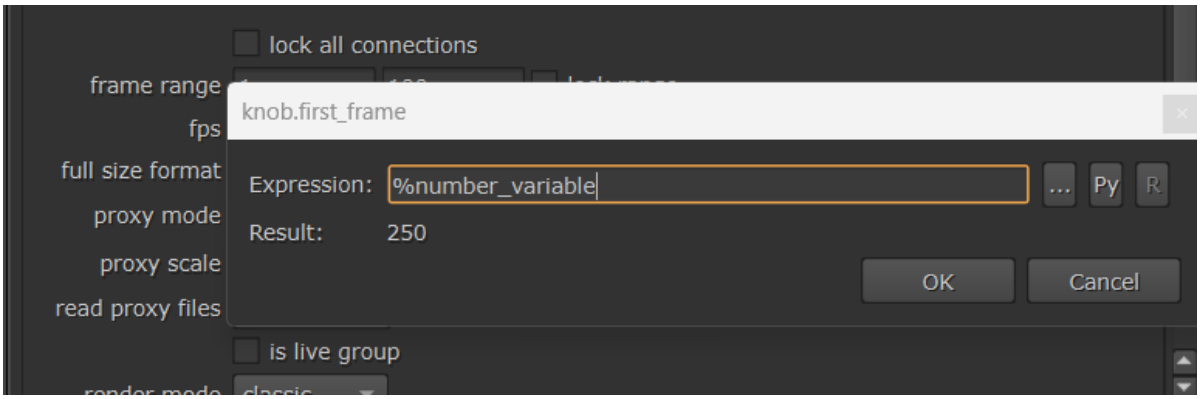
The Root node allows you to control parameters that affect the entire Nuke script, rather than just a single node. You can access the Root node parameters by going to **Project Settings** and viewing the **Root** tab.

The **frame range** (**first_frame** and **last_frame**) and **fps** knobs on the Root node allow expressions, allowing you to reference Variables here.

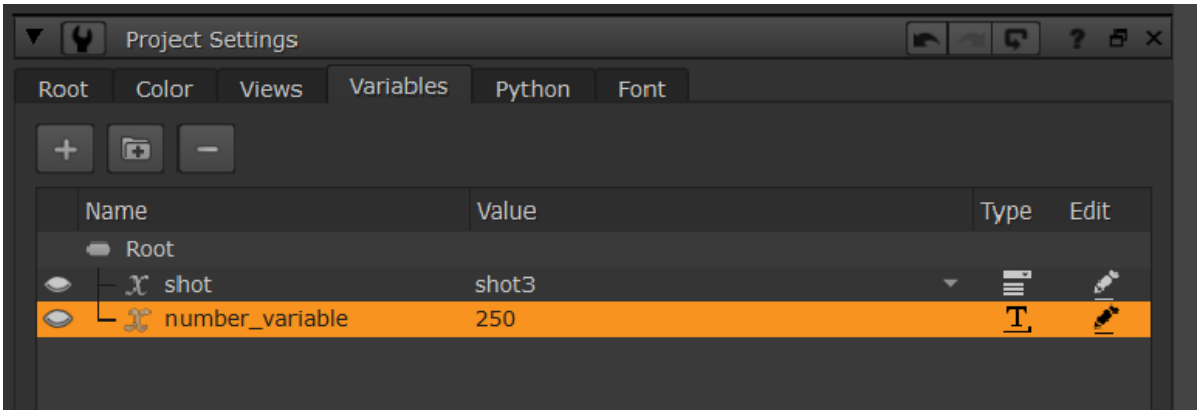
To do this, right click on the knob and select **Add expression...**



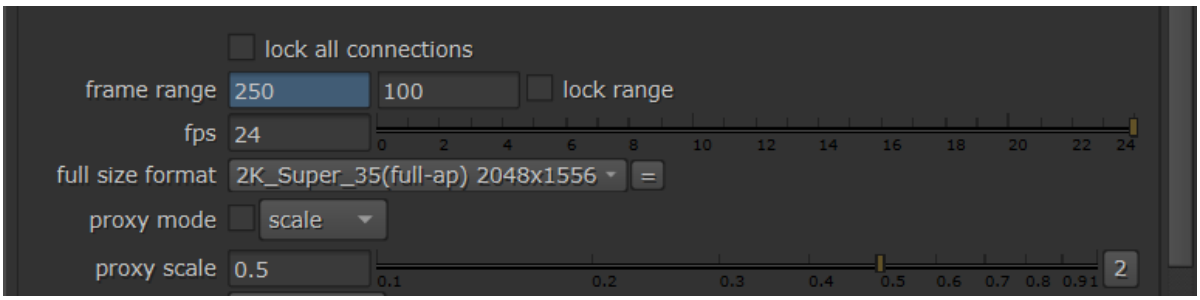
A window will appear. In the **Expression** field, type to reference your Variable using **%** and your variable name. You can check if the result matches your expectations below before clicking OK.



This is the Variable we are referencing in this example:



Once you've clicked OK, evaluated Variables will appear in your Root node in blue:



Now changing the value of your Variable will automatically update the Root node's first_frame knob.

Note: These expressions cannot include animations (e.g. use of the frame, t and x variables) - these expressions will be flagged as invalid and will evaluate to 0 if used.

Note: To see more examples of variables usage see Using Graph Scope Variables in Your Workflows.

Using Python with Graph Scope Variables

Editing Variable Values

You can edit Variable values using Python rather than the Variables tab.

In this case, the value of the GsvKnob can be set and retrieved using the regular ‘setValue’ and ‘value’ functions, using a regular Python dictionary type:

For example:

```
rootGsvKnob = nuke.root()['gsv']
myValue = {'Custom': {'shot': 'sh001'}, '__default__': {'show': 'hot-chocolate'}}
rootGsvKnob.setValue(myValue)
print(rootGsvKnob.value())
# Result: {'Custom': {'shot': 'sh001'}, '__default__': {'show': 'hot-chocolate'}}
```

Python Callbacks

Python code can now be triggered when Variables are added, removed, renamed, and changed in Value.

This enables you to create custom behavior based on these events, such as creating templates of Variables, populating Variables into specific folder structures, and adding password protection for Variable value changes.

The system provides granular control by offering ***before*** and ***after*** callbacks for each operation (Add, Remove, Rename or Change in Value). This distinction is key for implementing pipeline rules:

- **Before Operations (`addBeforeUser*`):** Callbacks that execute before an operation (such as adding or removing a GSV) can **return `False` to abort or prevent the operation**. This is essential for enforcing naming conventions or preventing the removal of crucial variables.
- **After Operations (`addAfterUser*`):** Callbacks that execute after an operation are used for notifications, logging changes, or initiating synchronization with external systems, but they **cannot prevent the operation**.

Available GSV Callback Functions

Nuke provides the following functions, categorized by the action and the scope (individual GSV or GSV set):

Action	Individual GSVs	GSV Sets	Description & Notes
Add	`addBeforeUserAddGsv()`	`addBeforeUserAddGsvSet()`	Called before adding a GSV or GSV set. Can abort the operation by returning `False`.
	`addAfterUserAddGsv()`	`addAfterUserAddGsvSet()`	Called after a GSV or GSV set is added.
Remove	`addBeforeUserRemoveGsv()`	`addBeforeUserRemoveGsvSet()`	Called before removing a GSV or GSV set. Can prevent removal by returning `False`.
	`addAfterUserRemoveGsv()`	`addAfterUserRemoveGsvSet()`	Called after a GSV or GSV set is removed.
Rename	`addBeforeUserRenameGsv()`	`addBeforeUserRenameGsvSet()`	Called before renaming a GSV or GSV set. Can abort the operation by returning `False`.
	`addAfterUserRenameGsv()`	`addAfterUserRenameGsvSet()`	Called after a GSV or GSV set is renamed.
Value	`addBeforeUserSetGsvValue`		Called before a

Action	Individual GSVs	GSV Sets	Description & Notes
Change			variable's value is changed. Can abort the operation by returning `False`.
	`addAfterUserSetGsvValue`		Called after a variable's value is changed.

Note: For unregistering Python Callbacks, there are `nuke.remove...` functions for each of the registration functions above.

Note: See more in the Nuke Developer Guide.

Note: Variable Value Change Callbacks are missing from the List of callbacks in the Nuke Developer Guide in the Beta 4, however they are available in the build.

Specifying Variables for Command Line Render

The command line interface enables you to specify Graph Scope Variables so that scripts can be rendered using the correct shot context.

Here is an example of how you may use this for a single Variable.

(You would input your own Variable names, values, framerange and path)

```
./Nuke16.0 --var <variable1>:<value1> -F <framerange> -X Write1 <path_to_your_nuke_script>
```

If you wish to specify multiple Variables, you can do this using a "," separated list as below:

```
./Nuke16.0 --var <variable1>:<value1>,<variable2>:<value2> -F <framerange> -X Write1 <path_to_your_nuke_script>
```

Applying this to multishot workflows: example

To use the command line to specify Variables across multiple shots, you could first set up Variables in your Write nodes, such as:

```
/project/%{seq}/%{shot}_key.####.exr
```

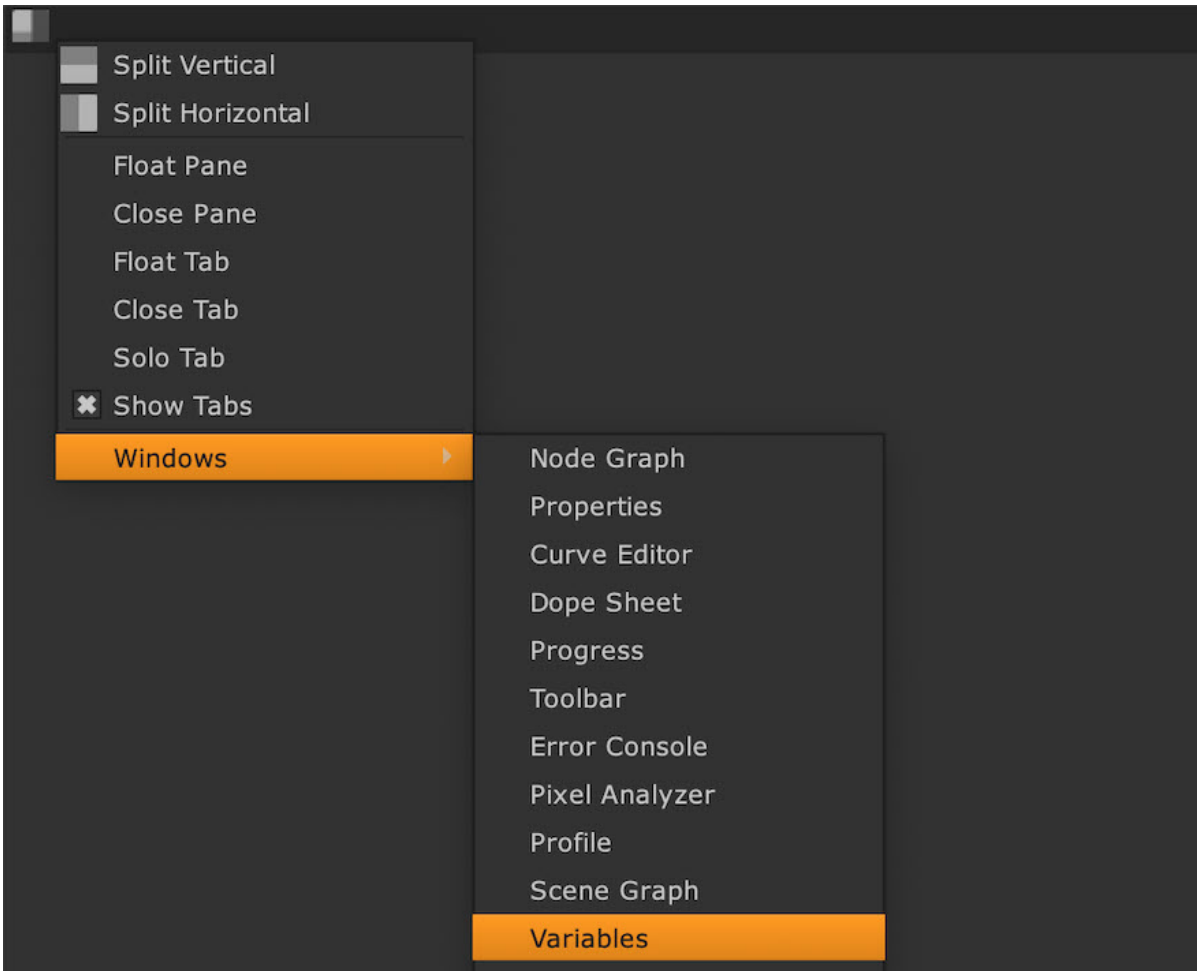
And render using the next command line like this:

```
./Nuke16.0 --var seq:seq030,shot:Sh100 -F 1101-1105 -X Write1 <path_to_your_nuke_script>
```

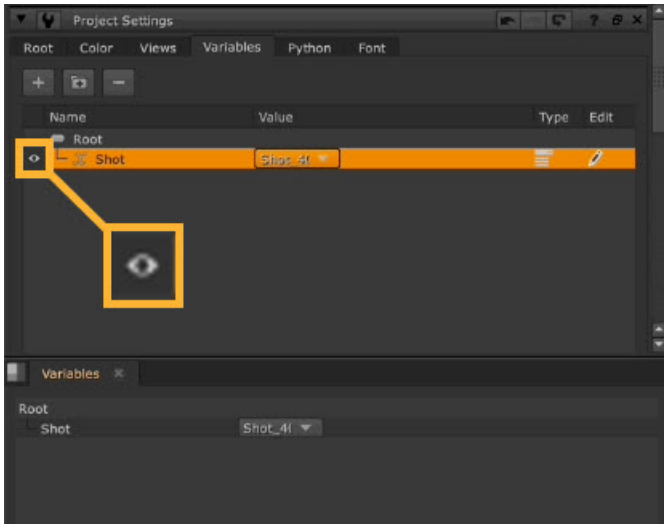
Configuring Visibility with the Variables Panel

The **Variables** panel is a convenient way to expose a tailored list of variables.

You can find this by selecting the Tabs icon  > **Windows** > **Variables**.



1. To set the variables that you wish to include, first navigate to the **Variables** tab in **Project Settings**.
2. In **Project Settings**, toggle on or off the variable visibility on using the eye icon. The eye icon signifies that the variable will appear in the **Variables** panel, as in the example below.



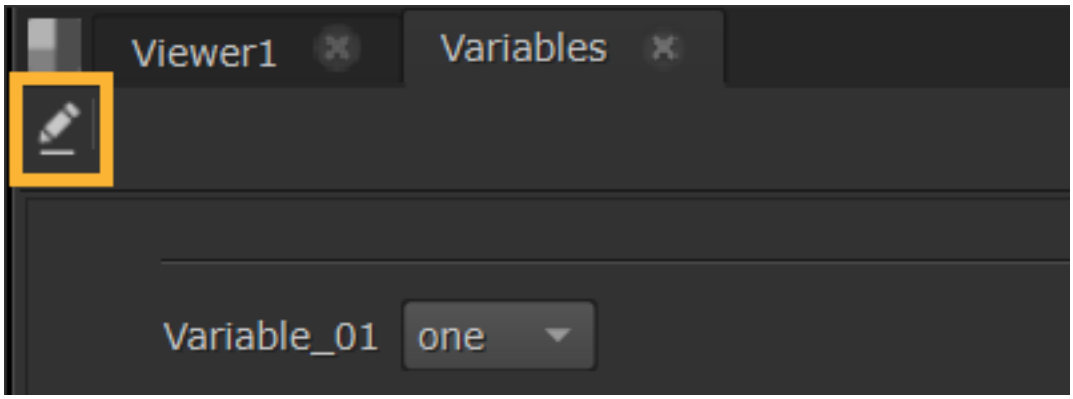
You can see here that with the visibility eye icon present, the 'Shot' variable appears in the Variables panel.

This provides an easy way to work with different shots, or to interact with any other variable previously set up.

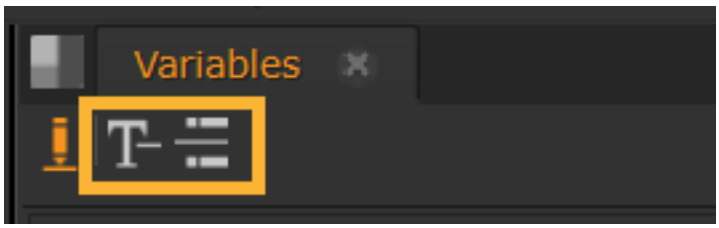
Tip: Whilst having an overview of all possible variables is a necessary and powerful tool for setting up a Multishot sequence, this level of detail may not be necessary when working on a specific subset of shots in that sequence. The Variables panel lets you toggle variable visibility to help you work more efficiently.

Organizing the Variables panel

You can click and drag the variables to change the order that they appear by enabling the edit tool:



You can add divider lines and text using these buttons:

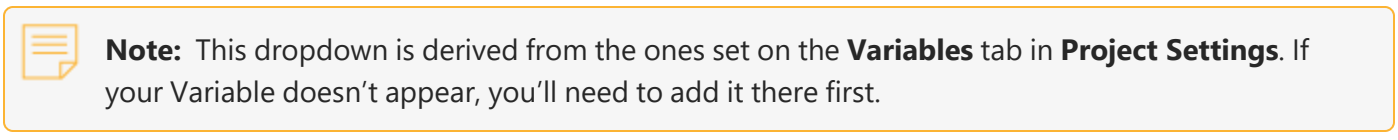


Using the VariableSwitch Node

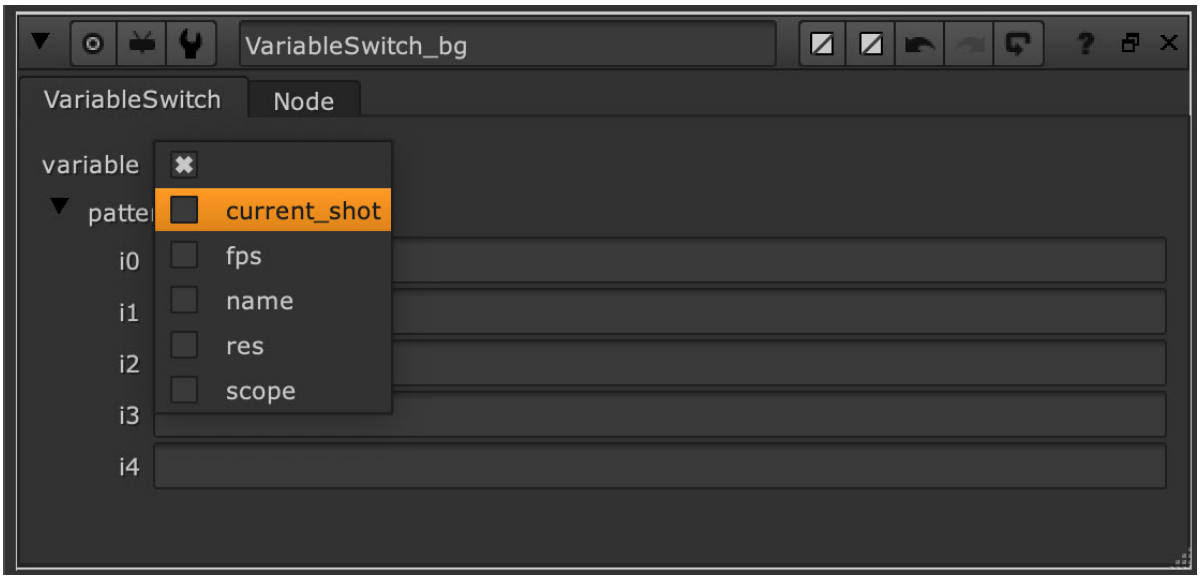
The VariableSwitch node allows you to switch between different inputs, shots, or scopes using Variables.

How to set up switching

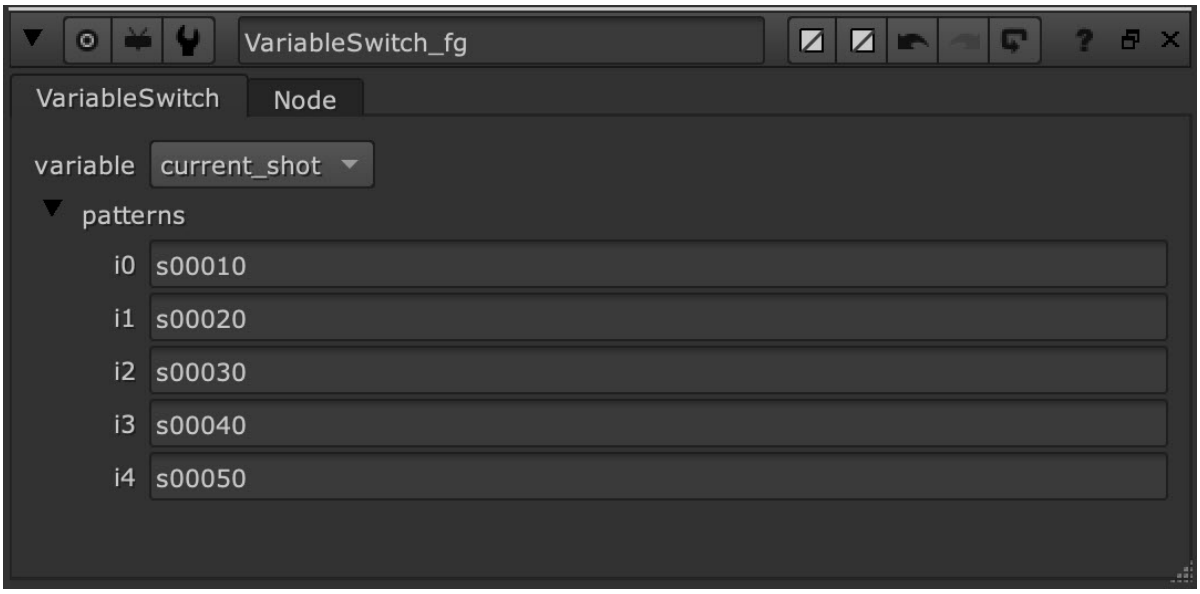
1. After adding a VariableSwitch node to your node graph, choose which Variable is going to be used for switching the node. The Variables to choose from are displayed in the dropdown.



Note: This dropdown is derived from the ones set on the **Variables** tab in **Project Settings**. If your Variable doesn't appear, you'll need to add it there first.



2. In the **patterns** section, you'll see a pattern knob appearing for each input that the node has: if the switch node has 5 inputs, there will be 5 pattern knobs.
   Type Variable names into these **patterns** to define the inputs that are active on the node. The Variables should match the ones you have defined in the Variables tab in Project Settings — see Using Graph Scope Variables in Your Workflows for more information.



3. You can then switch the Variable in use in Project Settings or in the Variable Panel.

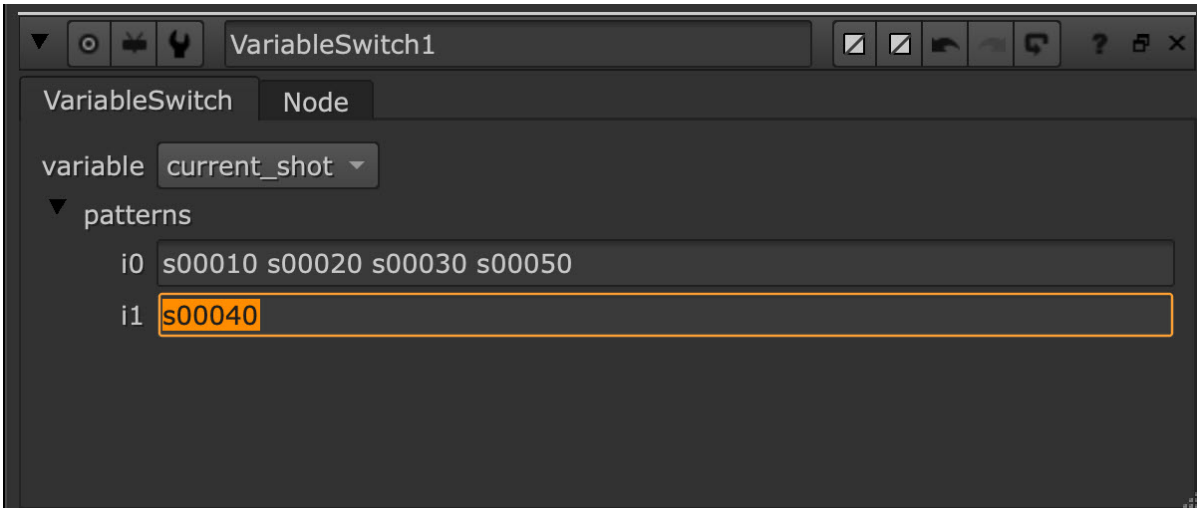
Changing the 'current shot' Variable in the Project Settings to a different value ('s00010', 's00020', 's00030', 's00040', 's00050') will change the inputs in the VariableSwitch.

If there are multiple VariableSwitch nodes with the same patterns, all of them will be affected. If a variable doesn't match the input in the pattern then input 0 will be used.

Adding multiple patterns per input

You can also add multiple Variables per pattern input - this enables a split and join style workflow where multiple scopes can have the same treatment. In the case of a multishot workflow, you can have treatments that apply to a group of shots in one input and another group of shots in the other.

In the example below you can see how shots 's00010', 's00020', 's00030' and 's00050' have one treatment while 's00040' needed a different one.

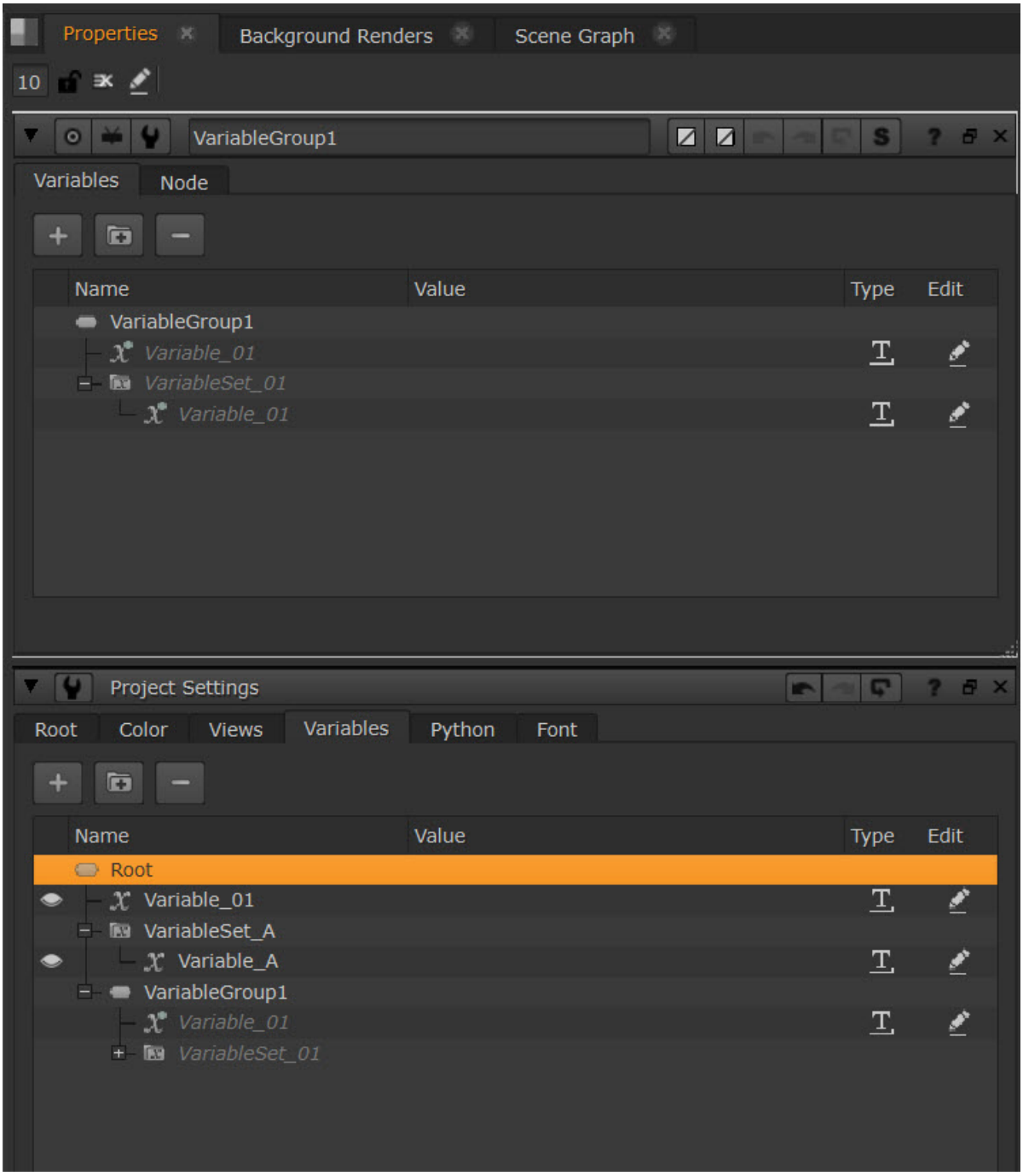


Tip: See our Multishot Tutorial to learn more about how Graph Scope Variables, VariableGroup, and VariableSwitch nodes work together in practice.

Using the VariableGroup Node

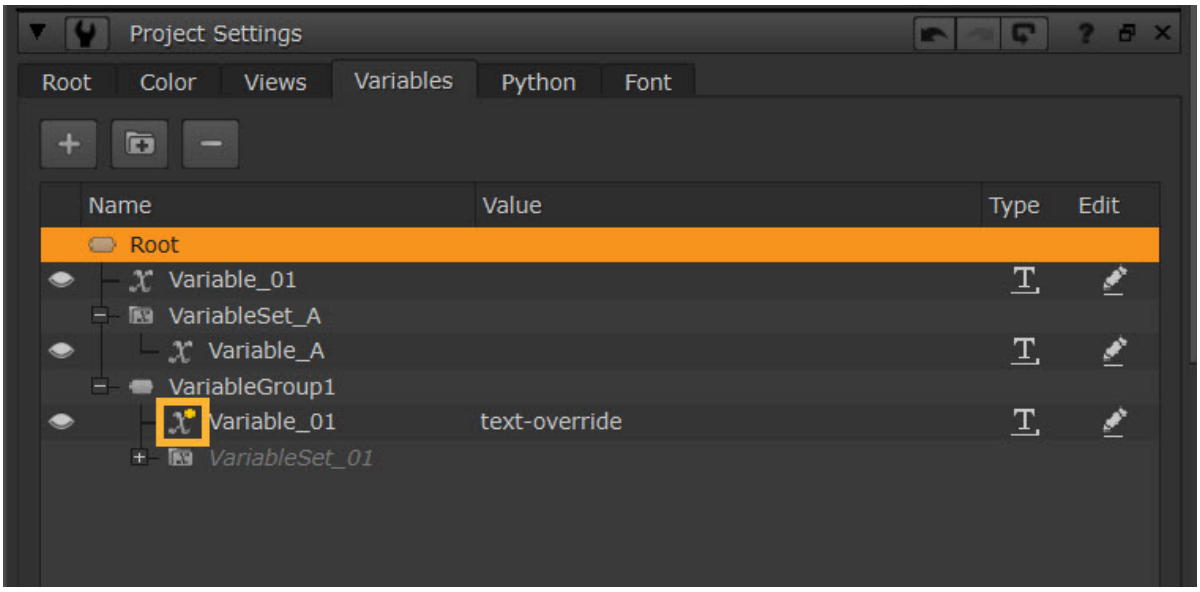
VariableGroup nodes allow you to define groups of Variables or scopes. This works similarly to the regular Group node, but for every new VariableGroup created, a corresponding VariableGroup item is created in the Variables tab in **Project Settings**.

When a VariableGroup node is first added to the script you'll notice that Variables from its parent(s) are automatically inherited and displayed in the tree view. These are denoted by an inherited Variable icon (green) and italicised/dimmed Name and Value.

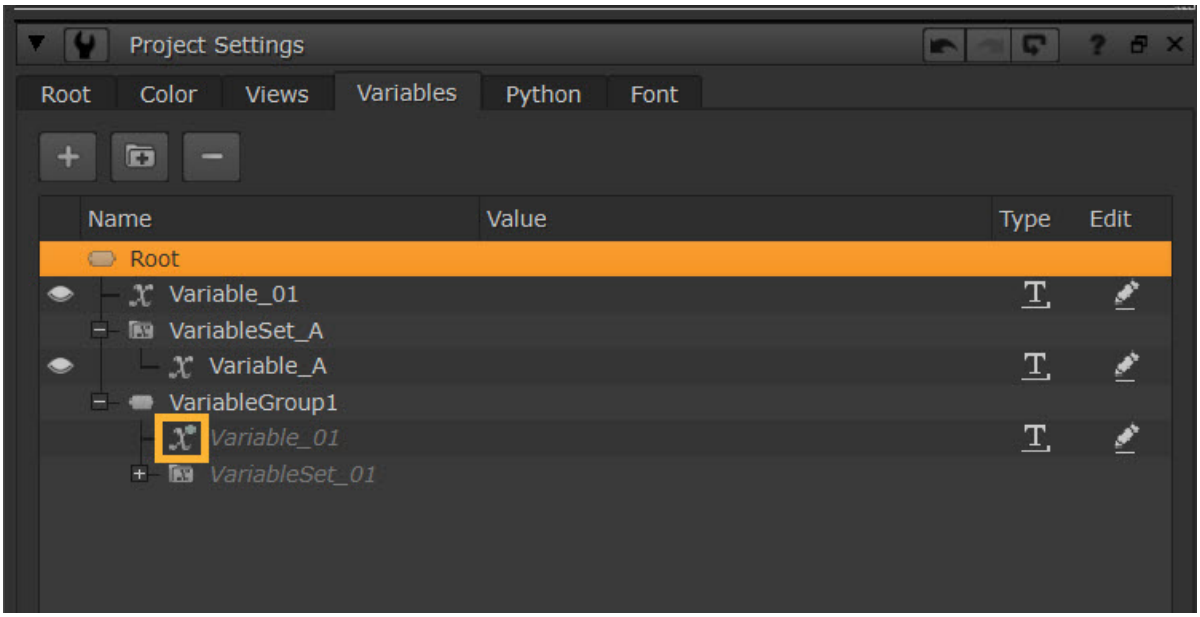


These Variables can be directly referenced inside the VariableGroup without the need to specify which parent they belong to and optionally overridden by changing the value.

• Overridden Variables are denoted by an override icon (**yellow**).



- Removing the override Variable will revert it back to inheriting from the parent, denoted by inherited icon (**green**).



Note: Currently the New 3D System does not work with VariableGroup with multiple differing variable values in the same Node Graph. As a workaround, you can have your ScanlineRender2 and 3D scene completely inside each VariableGroup, or you could avoid using VariableGroup when working with 3D workflows.

How does this help in multishot contexts?

This combination of Graph Scope Variable and VariableGroup nodes provides the ability to define typical production hierarchies and create, store and override Variables at each level as required.

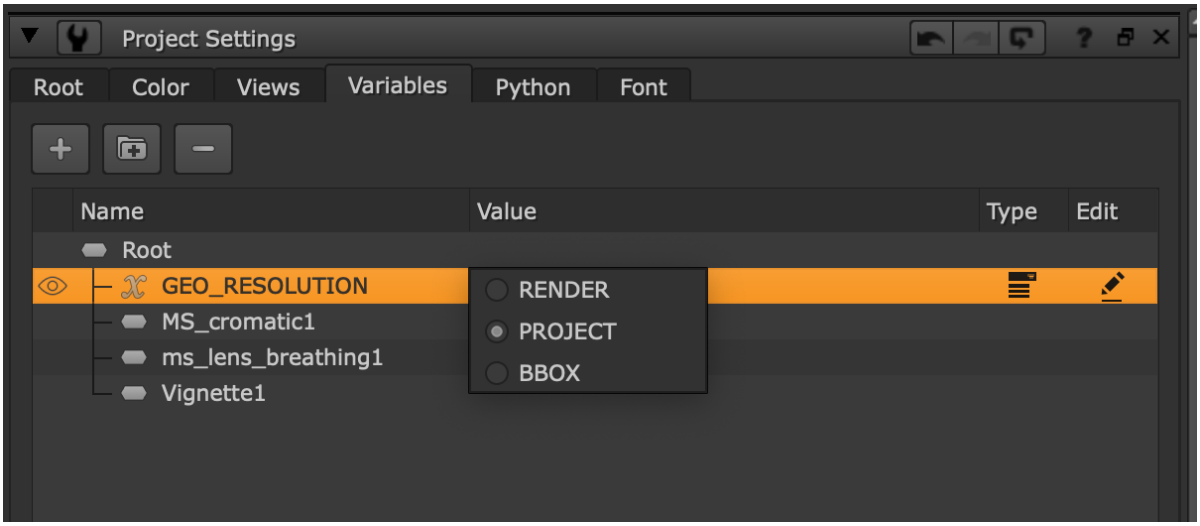
You can also take this even further by using the Using the VariableSwitch Node to create "split and join" workflows based on the Variables set for your multishot workflow.

Using Graph Scope Variables in the New 3D System

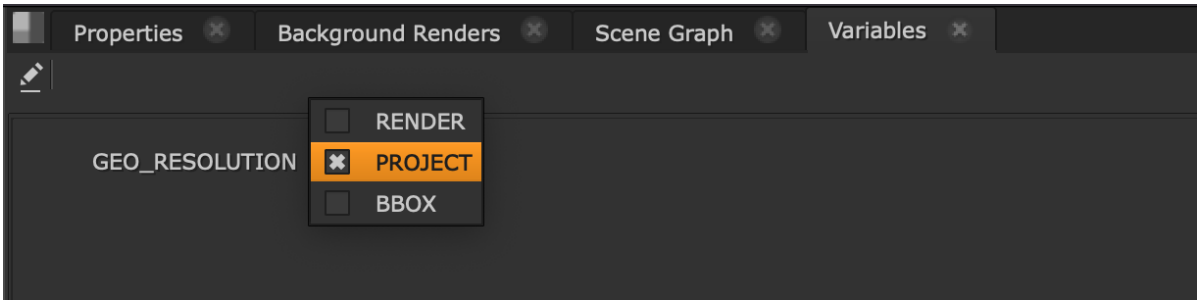
With Graph Scope Variables now able to support 3D nodes, you can set up Variables in your 3D scene to swap out geometry, textures, cameras, really anything you can think of. This means you could set up a 3D scene once and use Variables to swap it between a day and night setup, or swap geometry complexity out based on your needs, loading in a more performant version, while navigating the 3D Viewer and a higher resolution one for rendering.

Simple Workflow Example of Graph Scope Variables in 3D

To work with Graph Scope Variables, simply set up a Variable in your Project Settings. In the example below, we are setting a Variable to change the resolution of the geometry of our 3D scene and have given three different values.



This will show up in the Variables Panel as shown here.

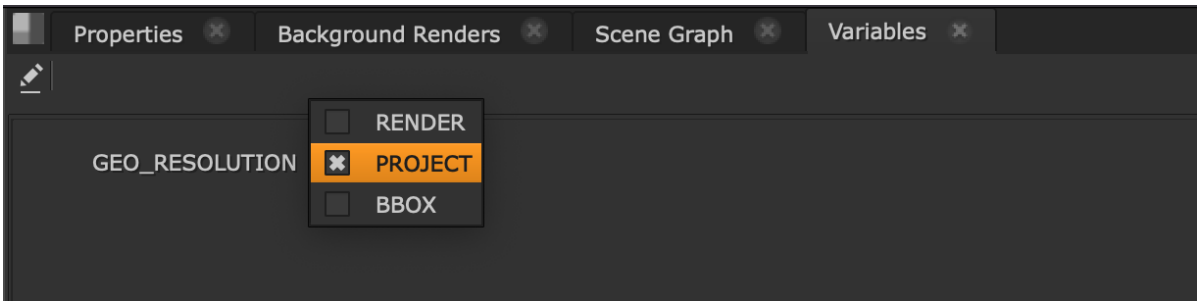


Note: You can learn more about the Variables Panel at Configuring Visibility with the Variables Panel.

We then import the different usd files containing the different geometry details and use a VariableSwitch node to switch the inputs between each import when the Variable is changed in the Variables Panel.

Note: You can learn more about VariableSwitch workflows at Using the VariableSwitch Node.

Note: See more about importing geometry at Importing with GeoImport.



This will result in different geometry being displayed in our 3D Viewer depending on our needs and can be connected to our ScanlineRender output.

There are many ways to use Variables as you could also use Variables in the GeoImport file path, so that you can have just one import node and swap between each usd file there.

Note: Currently the New 3D System does not work with VariableGroup with multiple differing Variable values in the same Node Graph. As a workaround, you can have your ScanlineRender2 and 3D scene completely inside each VariableGroup, or you could avoid using VariableGroup when working with 3D workflows.

Note: Learn more about the New 3D System at 3D Compositing with the New 3D System.

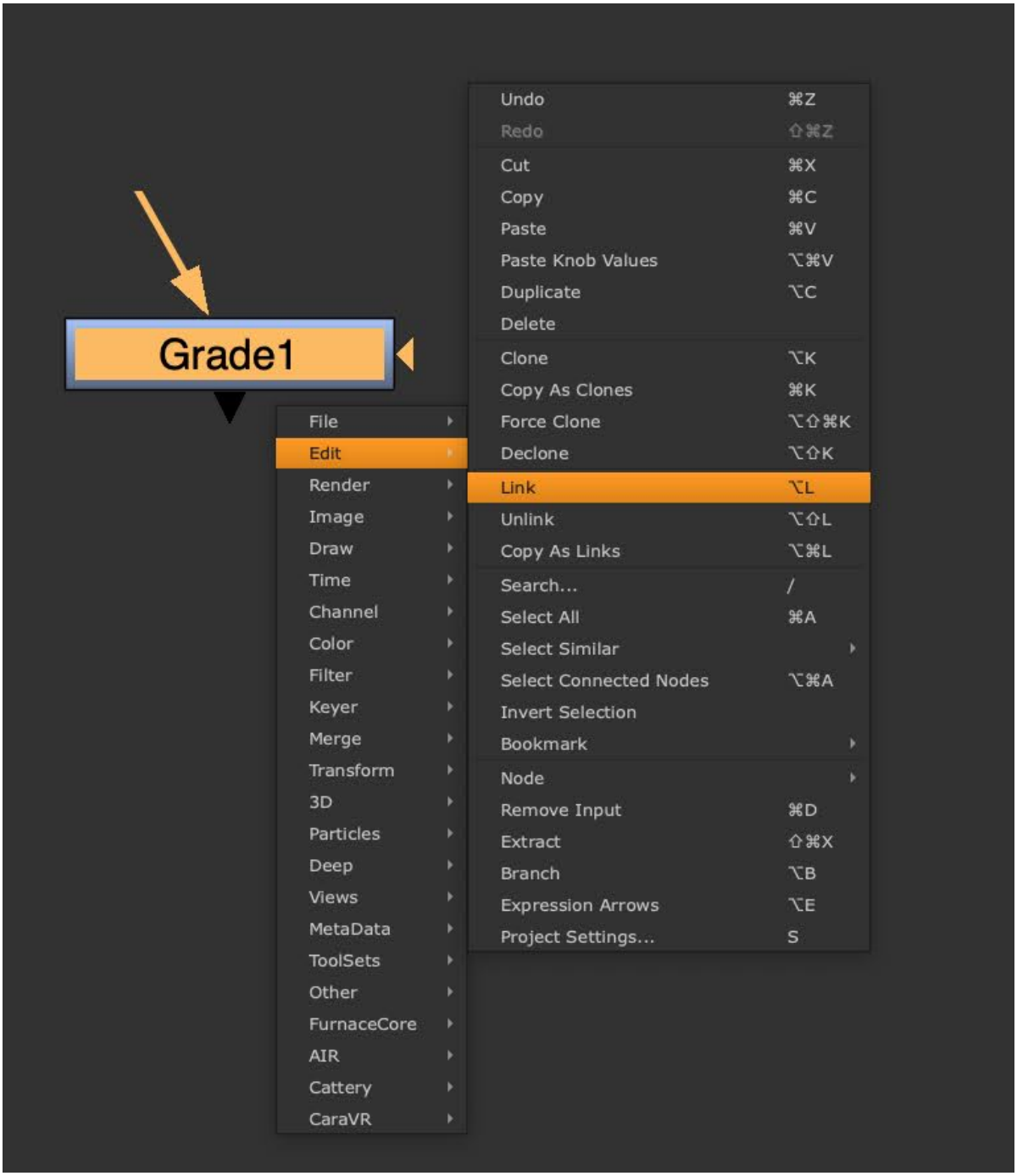
Using Link Nodes

Link Nodes allow you to make a linked copy of a node, and have changes updated in both nodes, whilst optionally being able to override any of the knobs.

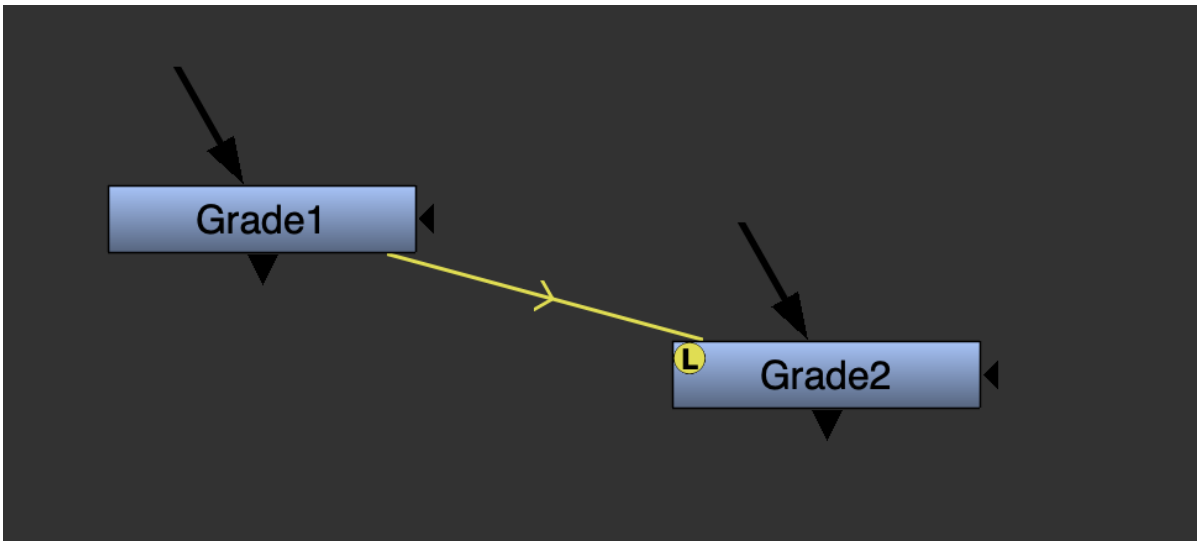
Link Nodes are useful in many workflows in Nuke - similar to a clone but more versatile, and are designed with multishot workflows in mind.

Create a Link Node

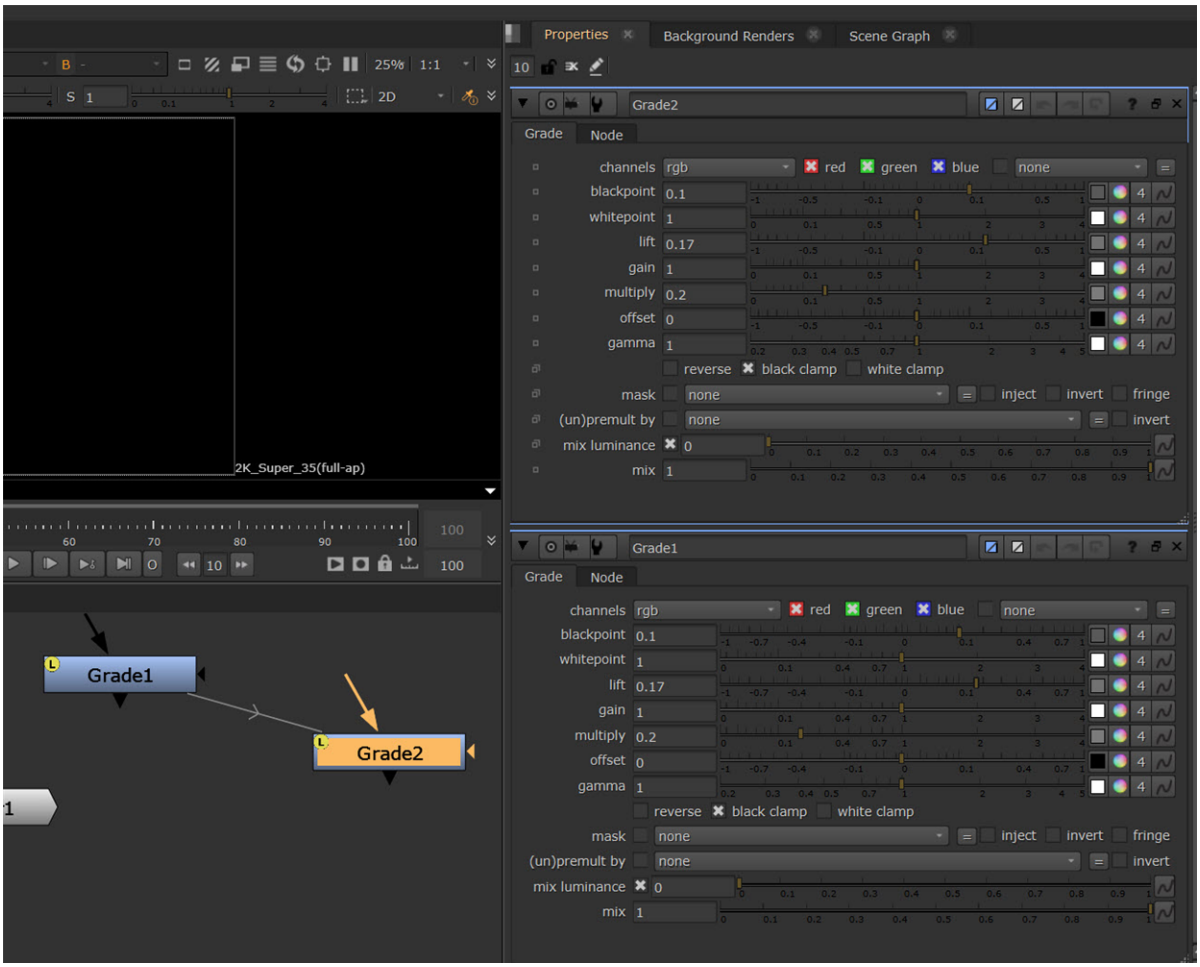
To create a Link Node, select the node you want to make a link node from and press **Alt+L**, or go to the menu bar and select **Edit** > **Link**.



A Link Node will be created nearby, which will appear as a duplicate node. Multiple selection is supported with this function.



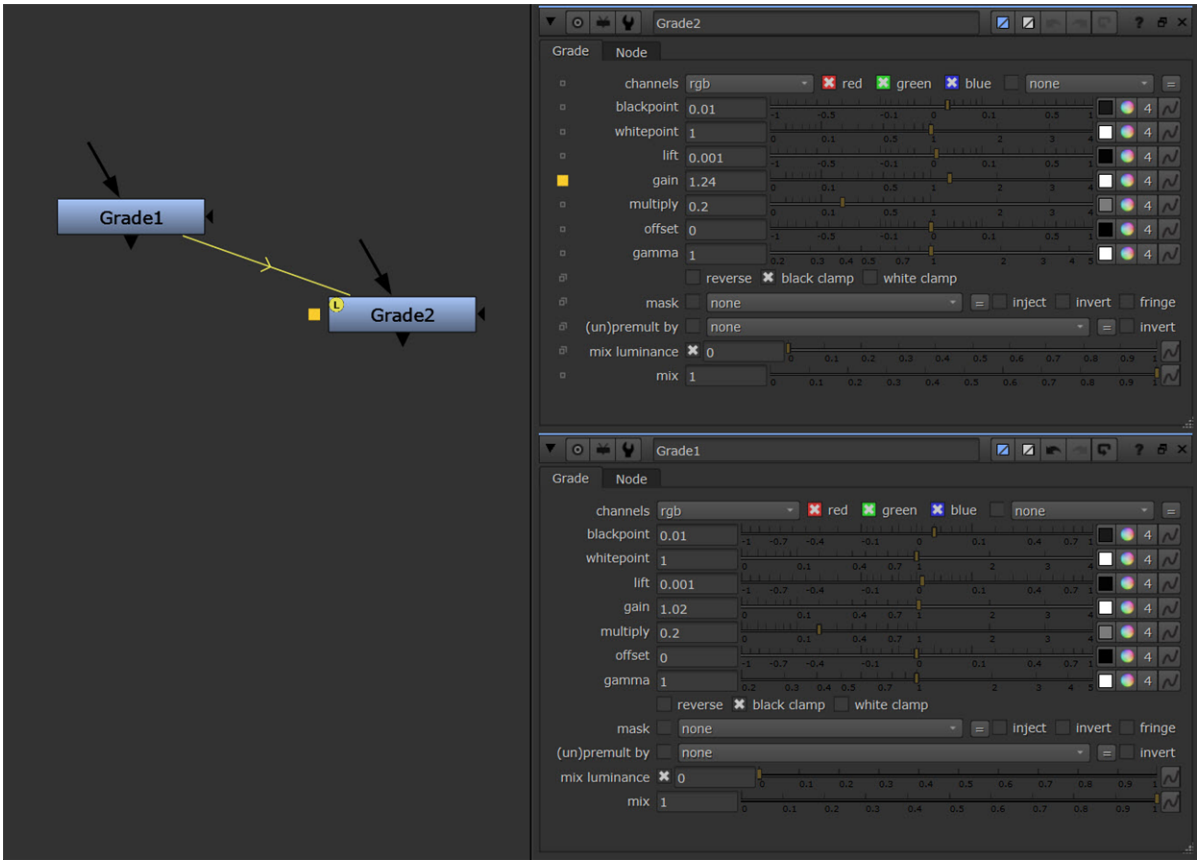
The knobs on the node will be matched both ways, similar to a clone node.



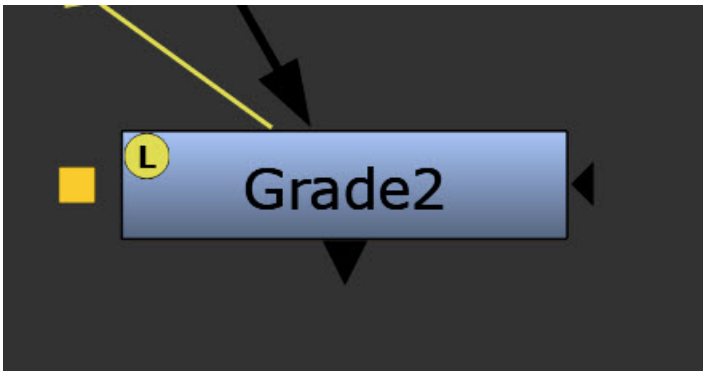
Tip: In large node graphs, creating Link Nodes for nodes like ColorCorrect and HueCorrect may help to ensure visual consistency, allowing you to easily adjust color and see changes applied across all your images.

A new column appears on the Link Node, showing options for overriding each knob.

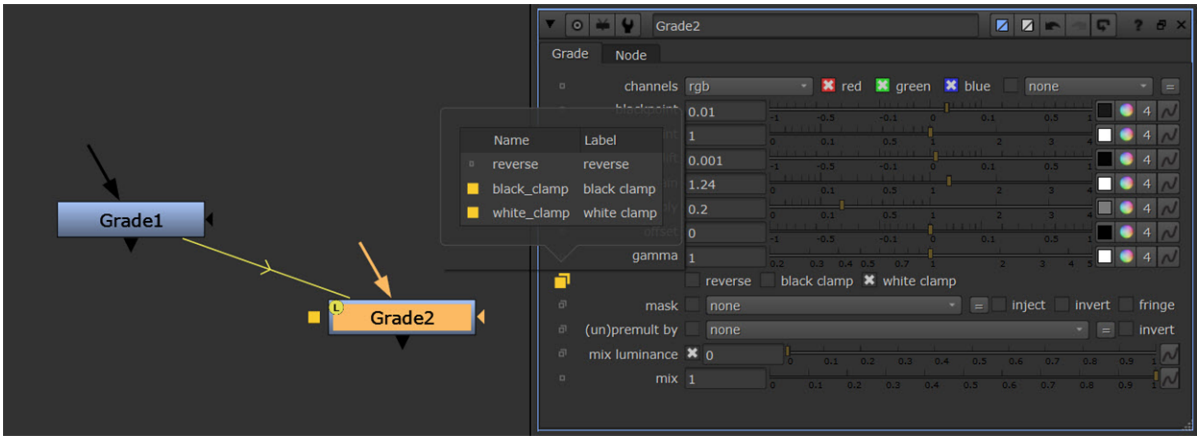
To override a knob, click the **square icon** in the left hand column, next to the knob you wish to override. A yellow filled-in square indicates that the knob has been overridden, and it will be decoupled from the corresponding knob on the other node.



When any knob is overridden on the link node, an override icon will also appear to the left of the node.



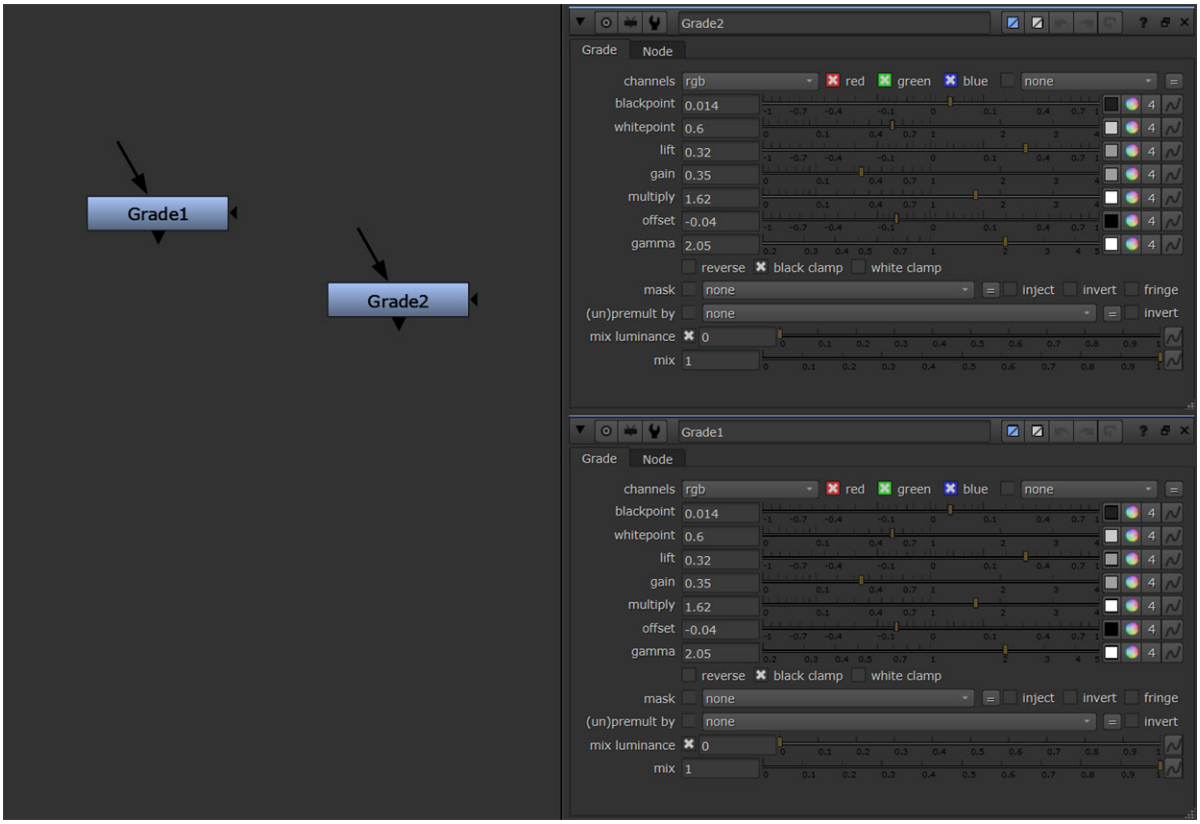
Nodes often have multiple knobs in a row, and these can be individually overridden using the pop-up table. Rows with multiple knobs can be identified by the multiple-override icon.



Unlink a Link Node

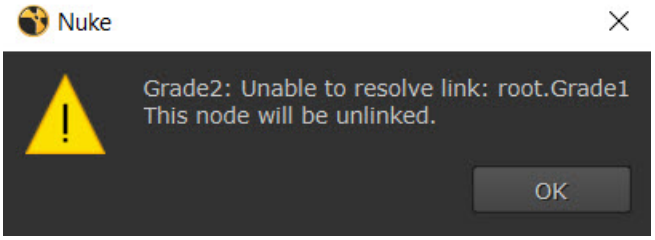
To Unlink a node, select the node you want to unlink from and go to the menu bar and select **Edit** > **Unlink**. Multiple selection is also supported with this function.

A link node that has been unlinked becomes an ordinary node, retaining the link node knob values.



Be aware that certain procedures will also result in the unlinking of a link node. These include:

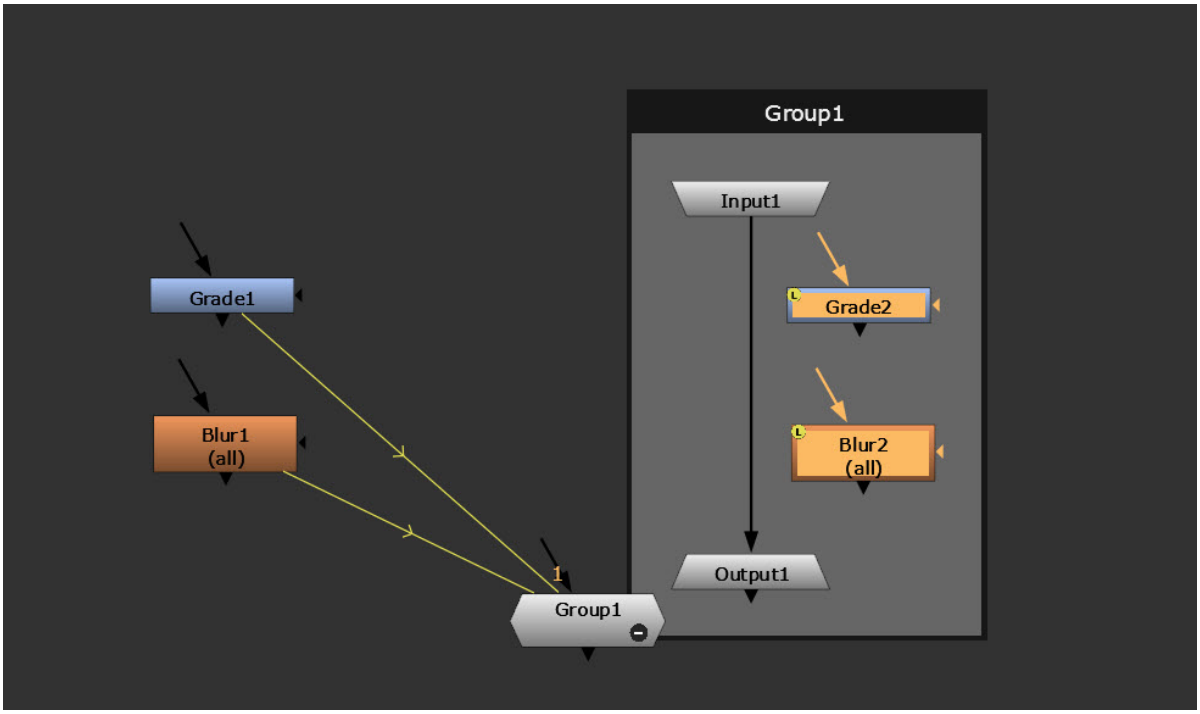
- Deleting the node a linked node is linked to. A pop-up will be shown before this occurs.
- Cross boundary links are not permitted for LiveGroups, Gizmos and Precomps. A cross boundary link is where one part of a link exists in a different group. The following situations will result in an unlink prompt:
    - Copying a node or it's link into a LiveGroup, causing a cross boundary link.
    - Exporting to gizmo a group that contains a cross boundary link.
    - Creating a precomp that would result in a cross boundary link.
- If a Link Node for any reason cannot find a node to link to, it is considered unresolved and will be unlinked.



Copy As Links

Copy As Links allows you to copy a set of nodes and paste link nodes, respectively linked against the copied set of nodes, under the cursor.

Select the nodes you wish to copy, select **Edit** > **Copy As Links** and **Edit** > **Paste** at the intended location. This differs from regular linking which creates the link node immediately and automatically places it. This can be used for example, to place link nodes into a group view by first clicking into the group view before pasting.



Arrow customization

You can customize the color of the link node arrows in the **Edit** > **Preferences** > **Node Graph**.

Nodes not supported for linking

Linking is not supported between certain node types, this includes but is not limited to:

- Clone
- Read
- DeepRead

- DeepWrite
- DeepExpression
- Write
- GeoExport
- ReadGeo
- WriteGeo
- BlinkScript
- Camera
- Axis
- Light
- ScanlineRender
- GeoSetVariant
- Inference
- ModelBuilder
- UnrealReader
- GenerateLUT
- Vectorfield
- Viewer
- VariableSwitch



Note: To be fixed in future releases: Grouping a node with a link connection will break the link.

Overrides via Python

Overrides can also be set via Python.

The code below is an example of how to use Link Nodes in Python:17.0

Python

```
# create a grade
grade = nuke.createNode("Grade")

# make a link node from this grade
link_grade = nuke.link(grade)

# check it is linked:
link_grade.isLinked()
```

```
# Result: True

# override a knob
blackpoint_knob = link_grade.knob("blackpoint")
link_grade.overrideKnob(blackpoint_knob)
blackpoint_knob.setValue(0.5)

# remove the override
link_grade.removeOverrideKnob(blackpoint_knob)

# find the node this is linked to
original_grade = link_grade.getLink()
original_grade.name()
# Result: Grade1

# find the dependent link nodes of this grade
dependent_links = original_grade.dependencies(nuke.LINKNODE_INPUTS)
# Result: [ Grade2 ]

# unlink
link_grade.unlink()
link_grade.isLinked()
# Result: False
```

For more information on the methods used, see the Python Reference Guide.

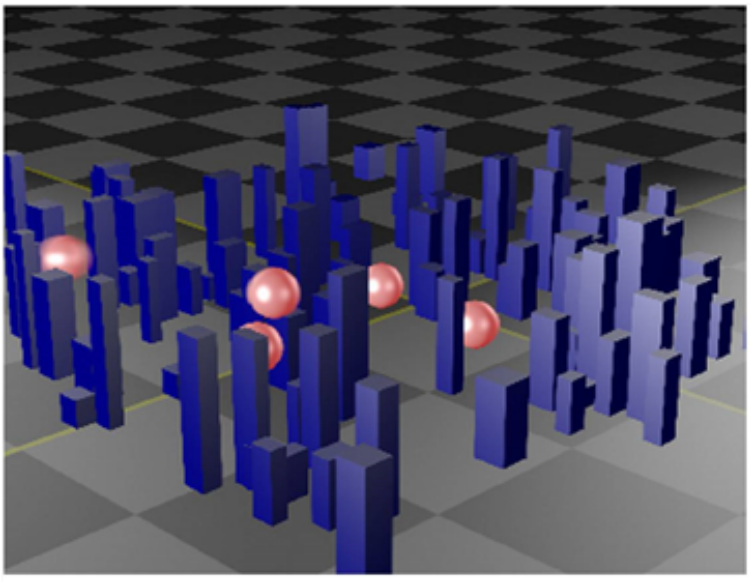
Deep Compositing

Deep compositing is a way of compositing digital images using data in a different format to standard "flat" compositing. As the name suggests, deep compositing uses additional depth data. This reduces need for re-rendering, produces high image quality, and helps you solve problems with artifacts around the edges of objects.

About Deep Compositing

A standard 2D image contains a single value for each channel of each pixel. In contrast, deep images contain multiple samples per pixel at varying depths and each sample contains per-pixel information such as color, opacity, and camera-relative depth.

For example, creating holdouts of objects that have moved in the scene has previously required re-rendering the background, and problems frequently occurred with transparent pixels and anti-aliasing. With Nuke's Deep compositing node set, you can render the background once, and later move your objects to different places and depths, without having to re-render the background. Any transparent pixels, with motion blur for example, are also represented without flaw, so working with deep compositing is not only faster, but you also get higher image quality.



Deep composite with ball objects among blue buildings.

Quick Start

With Nuke's deep compositing node set, you can:

- Read in your deep image with the DeepRead node. See Reading in Deep Footage.
- Merge deep data with the DeepMerge, see Merging Deep Images.
- Generate holdout mattes from a pair of deep images using the DeepHoldout node. See Creating Holdouts with the DeepMerge Node.
- Flatten deep images to regular 2D or create point clouds out of them. See Creating Deep Data.
- Sample information at a given pixel using the DeepSample node. See Sampling Deep Images.
- Crop, reformat, and transform deep images much in the same way as you would a regular image, using the DeepCrop, DeepReformat and DeepTransform nodes. See Cropping, Reformatting, and Transforming Deep Images.
- Create deep images using the DeepFromFrames, DeepFromImage, DeepRecolor, and ScanlineRender nodes. See Creating Deep Data.

Reading in Deep Footage

You read in deep images to Nuke with a DeepRead node, which is rather like reading in any other images with the Read node. Nuke allows you to import deep images in two formats:

- DTEX (generated from Pixar's PhotoRealistic RenderMan® Pro Server).
- Scanline OpenEXR 3.3.6, or above (tiled OpenEXR files are not supported).

Importing DTEX Files

Before importing DTEX files, you need to set up Pixar's RenderMan Pro Server 20, or earlier, on your computer.

Do the following:

1. Install RenderMan Pro Server on your computer, and set up the necessary environment variables that enable Nuke to work with it. For details, have a look at Setting Up RenderMan Pro Server and PrmanRender. Note that you don't need a RenderMan Pro Server license to work with deep images in Nuke, just installing the software is enough.

2. Create a DeepRead by clicking **Deep** > **DeepRead**.
3. Navigate to your **.dtex** image, and click **Open**. For more information about the Read node controls, see the Managing Scripts chapter.
4. By default, Nuke tries to automatically detect the **.dtex** file type by looking at the subimage name. If the name is either **Deep Shadow** or ends with (or is) **.deepopacity**, Nuke treats the file as a deep opacity file. However, if you have manually changed the subimage name when rendering the file, Nuke may not be able to detect the file type correctly. If this is the case, set the **type** dropdown menu to one of the following:
   - **deepopacity** - This forces Nuke to treat the file as an accumulated deep opacity file, corresponding to a RenderMan Display Driver configuration of:

     Display "Filename.dtex" "deepshad" "deepopacity"
   - **alpha** - This forces Nuke to treat the file as the newer point-sampled alpha or color, corresponding to a RenderMan Display Driver configuration of either:

     Display "Filename.dtex" "deepshad" "a" or

     Display "Filename.dtex" "deepshad" "rgba"

Importing Scanline OpenEXR Files

To import scanline OpenEXR files:

1. Create a DeepRead by clicking **Deep** > **DeepRead**.
2. Navigate to your deep **.exr** image, and click **Open**.For more information about the Read node controls, see the Managing Scripts chapter.
3. By default, the **exr** prefix is attached to metadata keys to make them distinct from other metadata in the tree. If you'd rather read metadata in "as is" without attaching a prefix, enable **do not attach prefix**.

Creating Deep Data

You can create deep data in Nuke by:

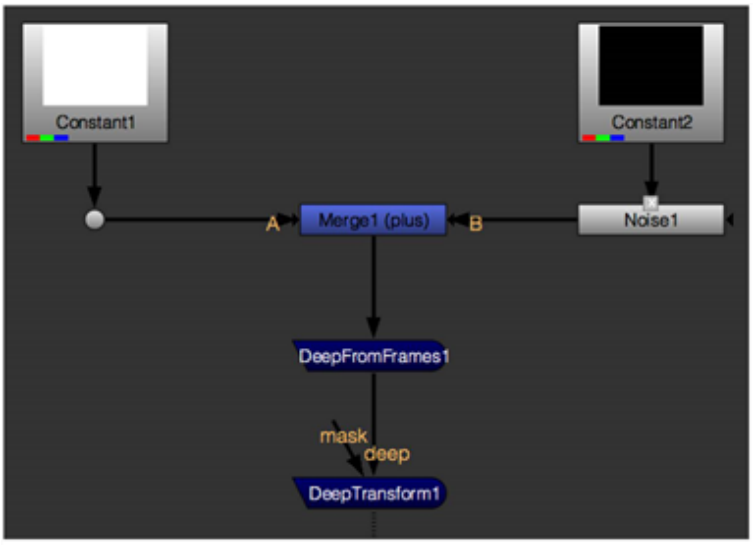
- sampling a regular 2D image sequence at multiple frames to create several samples for each pixel in a single deep frame. See Converting a 2D Image Sequence to a Deep Frame Using Input Frames.
- converting a regular 2D image to a deep image with a single sample for each pixel at the depth defined by the depth.Z channel. See Converting a 2D Image to a Deep Image.
- recoloring depth samples using a regular 2D color image. See Recoloring Depth Data.

• adding a ScanlineRender node to a 3D scene and connecting a Deep node downstream. See Using ScanlineRender to Generate Deep Data.

Converting a 2D Image Sequence to a Deep Frame Using Input Frames

You can use the DeepFromFrames node to create depth samples from input frames.

1. Connect the DeepFromFrames node to your footage. The deep image is created by placing each frame at increasing depths.
2. To adjust the results, use the controls in the properties panel:
   - **samples** - the number of samples to create per pixel in the output deep image.
   - **frame range** - the range of frames to use for one deep image. For example, with the default samples value (5) and frame range value (1-9) DeepFromFrames sample at times 1, 3, 5, 7 and 9.
   - **premult** - check to premultiply the samples.
   - **split alpha mode** - select **additive** to perform a straight division by the number of samples or **multiplicative** to split the alpha so that it can be returned to its original value if flattened later on (using the DeepToImage node, for example). If you select **additive**, the alpha can't be returned to its original value.
   - **zmin** - the depth to assign to the first sample of each deep pixel output, corresponding to the first frame in the range.
   - **zmax** - the depth to assign to the last sample of each deep pixel output, corresponding to the last frame in the range.



A simple setup for creating a deep fog element.

Converting a 2D Image to a Deep Image

Using the DeepFromImage node, you can convert a regular 2D image to a deep image with a single sample for each pixel at the depth defined by the depth.Z channel.

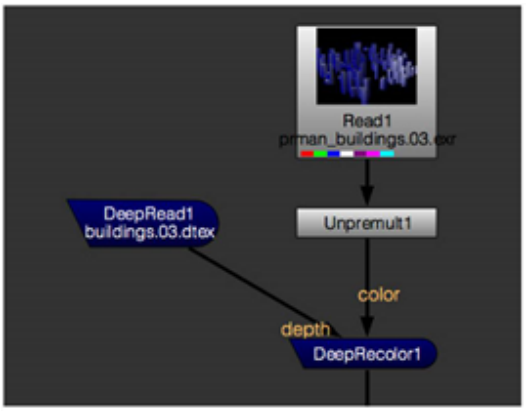
1. Connect DeepFromImage to the footage you want to convert to a deep image.
2. Use the **premult input** box in the properties panel to select whether you want the input channels to be premultiplied or not.
3. Uncheck the **keep zero alpha** box if you want to drop any samples with a zero alpha value from the resulting deep image. By default, the box is checked and the resulting deep image contains the zero alpha samples.
4. You can also specify the depth using the **z** control in the properties panel. In that case, check the **specify z** box to indicate you don't want to use a depth channel from the input.

Recoloring Depth Data

Use the DeepRecolor node to merge deep buffer files that only contain opacity for each sample with a standard 2D color image. DeepRecolor spreads the color at each pixel of the input 2D image across all the samples of the corresponding pixel in the deep input.

1. Connect your deep source to the **depth** input of the DeepRecolor node, and your 2D image to the **color** input. You might want to add an unpremultiply node between your color input and the DeepRecolor if your 2D image is premultiplied.
2. In the properties panel, you can select which channels you want to use from the **color** input image.

   In the example below, DeepRecolor takes an unpremultiplied **.exr** image and uses it to color the **.dtex** file's deep samples.



3. If at this point the alpha from the final high-quality flat render doesn't match the alpha represented by the deep samples (for example, as a result of the compression that usually happens to deep files on disk

or some change to the shader), you can check **target input alpha**. This means the **color** input's alpha is distributed amongst the deep samples, so that the final resulting alpha after flattening of the deep data matches the **color** input's alpha.

If you leave **target input alpha** unchecked, Nuke distributes the color to each sample by unpremultiplying by the **color** image's alpha and then remultiplying by the alpha of each sample. In this case, the alpha from DeepRecolor may not match the alpha from its **color** input.

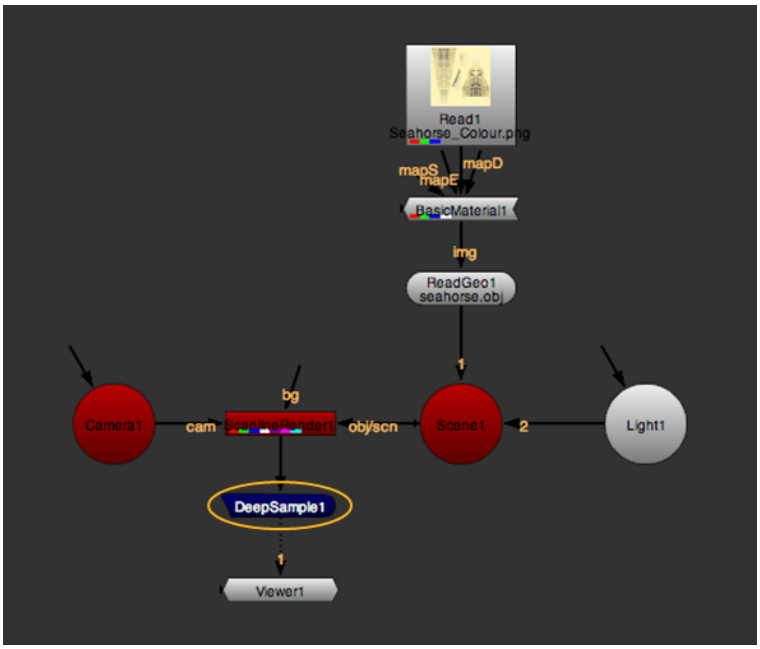
Using ScanlineRender to Generate Deep Data

The ScanlineRender node outputs deep data if there is a Deep node downstream.

1. Create a 3D scene and attach a ScanlineRender node to it to render the scene as a 2D image.

Note: Deep compositing only supports the **over** blend mode. As a result, if there is a BlendMat node in the 3D scene, its **operation** always appears to be set to **over** when converted to Deep.

2. Add a node from the **Deep** menu downstream from ScanlineRender.



3. If you don't want deep samples with an alpha value of 0 to contribute to the output, open the ScanlineRender properties and make sure that **drop zero alpha samples** is enabled.
4. Adjust the rest of the ScanlineRender properties as usual. For example:
   - If you see any aliasing artifacts in the render, go to the **MultiSample** tab and increase **samples**. This increases the number of deep samples per pixel.

     Alternatively, you can set **antialiasing** to **low**, **medium**, or **high** on the **ScanlineRender** tab.



Samples set to a low value.



Samples set to a high value.

- If you want to add motion blur to your 3D scene, increase the **samples** value to sample the image multiple times over the shutter time.

  In the **shutter** field, enter the number of frames the shutter stays open when motion blurring. If rendering becomes very slow, you can approximate multi-sample rendering and reduce render times by increasing **stochastic samples**.



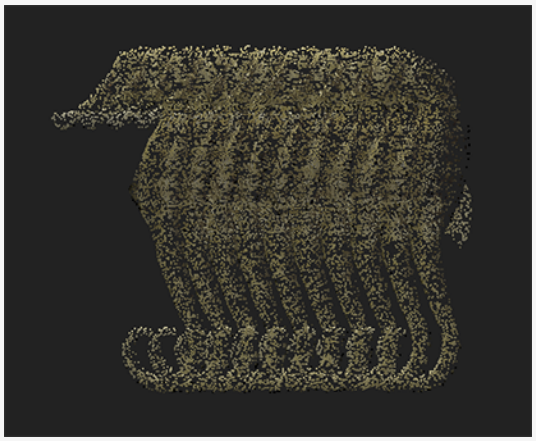
Without motion blur.



With motion blur.

For more information on the ScanlineRender properties, see Rendering a 3D Scene.

Tip: You can use a DeepToPoints node after ScanlineRender to create a 3D point cloud that represents the motion in the scene. For more information on DeepToPoints, see Creating 2D and 3D Elements from Deep Images.



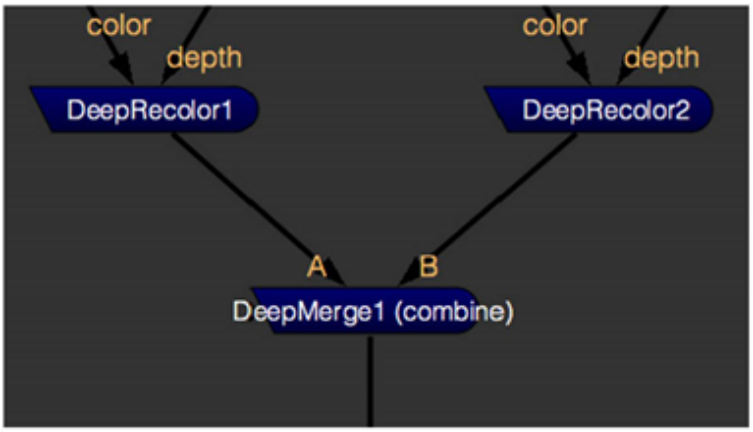
Merging Deep Images

Use the DeepMerge node to **combine** the samples from multiple deep images, so that each output pixel contains all the samples from the same pixel of each input. You can also **plus** overlapping samples from the **A** and **B** inputs.

1. Connect the data you want to merge to the DeepMerge node's numbered inputs.
2. In the DeepMerge properties, make sure **operation** is set to **combine**.

Note: Selecting **plus** adds **A** and **B** overlapping samples, which can be useful when recombining data after a holdout.

3. You can check the **drop hidden samples** box in the properties panel to not include samples that are completely occluded by nearer samples whose alpha value is one.



Merging two DeepRecolor results.

4. You can filter out samples using the **drop zero threshold** control. Increasing the value removes more samples with very small alpha values, such as those caused by floating point inaccuracy.
5. The **metadata from** control allows you to control which input's metadata is passed down the node tree.

Note: When **metadata from** is set to **All** and there are keys with the same name in both inputs, keys in B override keys in A.

See Creating Holdouts with the DeepMerge Node for more information on DeepMerge holdouts.

Creating Holdouts with the DeepMerge Node

The **holdout** operation of the DeepMerge node removes or fades out samples in input **B** that are occluded by samples in input **A**.

To create a holdout using the DeepMerge node:

1. Connect the deep image you want to remove or fade parts from to input **B**.
2. Connect the deep image with the occluding parts to input **A**.
3. In the DeepMerge properties, set **operation** to **holdout**.
4. You can filter out samples using the **drop zero threshold** control. Increasing the value removes more samples with very small alpha values, such as those caused by floating point inaccuracy.
5. The **metadata from** control allows you to control which input's metadata is passed down the node tree.

Note: When **metadata from** is set to **All** and there are keys with the same name in both inputs, keys in B override keys in A.

6. Enable **volumetric holdout** if you want Nuke to calculate occlusion using the values of the **holdout** samples in front of samples from **main**. This is a more accurate representation of occlusion at depth, but can take longer to process.

   For example:

Depth 0						Depth INF
M0	H0	H1	H2	M2	H3	M4

   M = main sample

   H = holdout sample

   M0 remains unchanged since there are no holdout samples before it. **M2** is affected by the combined **H0**, **H1**, and **H2** holdout samples and **M4** is affected by all holdout samples.

   You can now view the result, which is a holdout with red, green, blue, and alpha channels. Note that the output image is still a deep image.

Note: When **volumetric holdout** is disabled, deep samples that coincide with a holdout are removed and everything 'deeper' is also removed.

Creating 2D and 3D Elements from Deep Images

You can create a 2D image or a 3D point cloud from a deep image.

Creating a 2D Image from a Deep Image

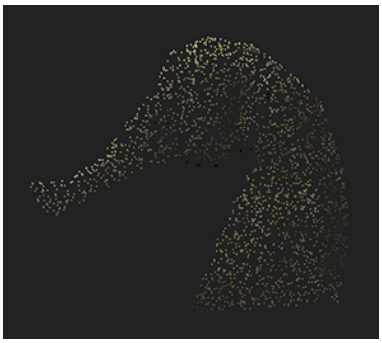
You can use the DeepToImage node to flatten an image, in other words merge all the samples in a deep image into a regular 2D image.

1. Connect the node to a deep image (or a DeepMerge with merged deep data) you want to flatten.
2. In the properties panel, the **volumetric composition** box is checked by default, but if you uncheck it, Nuke only calculates the front depth of each sample and assumes the samples do not overlap. If you uncheck this, the calculation takes less time, but if you have overlapping samples in your deep image, the resulting image might not represent every pixel as expected.

Creating a Point Cloud from a Deep Image

You can use the DeepToPoints node to transform the deep pixel samples into points in 3D space that you can see in Nuke's 3D view, much like a point cloud. This node is useful for position reference.

1. Connect the DeepToPoints node's **deep** input to the deep image you want to view in 3D. If you have a camera that you want to look at the point cloud through, connect it to the **camera** input.
2. Change to 3D view (by pressing **Tab**) to view the results.
3. In the properties panel, you can use the **Point detail** slider to adjust the density of the cloud.

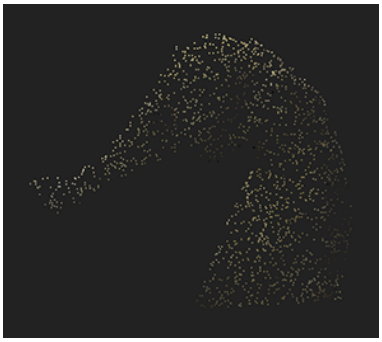


Point detail set to 0.005.



Point detail set to 0.05.

4. Adjust **Point size** to change the size of the points. You can also use 3D object selection on the point cloud, and for example snap on your DeepToPoints results. (For more information on 3D selection, see 3D Selection Tools.)



Point size set to 2.



Point size set to 6.

Modifying Deep Data

Nuke allows you to color correct deep images as well as modify them using expressions.

Color Correcting Deep Images

The DeepColorCorrect node applies the color correction to each sample at each pixel.

There are control sets for adjusting shadows, midtones and highlights, as well as a **master** set for adjusting all these at once. You can use the lookup curves on the **Ranges** tab to control the range of the image that is considered to be in the shadows, midtones, and highlights. For more information about the basics of color correcting, see Working with Color.

Tip: Make mattes by setting the **gain** value for your alpha channel to 0 and setting the **offset** value for the alpha channel to 1 in the range you want a matte for.

Adjusting the Effect of Deep Color Correction

On the **Masking** tab you can set the point among the deep samples where the effect of your color correction starts and finishes.

1. Check the **limit_z** box to activate the **zmap** tool.
2. Adjust the trapezoid so that the **A** delimiter marks the depth where you want the color correction to start, **B** and **C** mark the length of the full effect and delimiter **D** indicates where the effect stops. **Zmap** tool's y axis, therefore, indicates the amount of the effect, and the x axis is the range of your depth samples.
3. Use the **mix** control to adjust the overall mixing between the color corrected result and the original image. Zero value is the original image, and value 1 is the full color correction result.

Modifying Your Deep Images with Expressions

You can use the DeepExpression node to run Nuke expressions on deep data. Use the controls in the properties panel:

1. There are four fields for temporary expressions, just like in the normal Expression node. These can be useful if you need to use a long expression in several fields and want to assign that expression temporarily to a variable. Enter the variable name to the left of the equals (**=**) sign, and the expression to the right. You can then use the variable name to represent the entire expression in the expression fields for the channels.
2. In the **chans0** - **chans3** dropdowns you can then specify which channels you want to create expressions for. This adds or removes expression fields below.
3. You can then enter your expressions for the different channels in the channel expression fields.

For more information about expressions, have a look at Expressions.

Cropping, Reformatting, and Transforming Deep Images

You can crop, reformat and transform deep images much in the same way as you would a regular image, using the corresponding deep nodes.



Note: Remember that since the samples at each pixel can be located at arbitrary depths, resampling during transforming may produce unexpected results since there may not be samples at the same depth in adjacent pixels.

Cropping Deep Images

You can use the DeepCrop node to clip your deep image, much like the normal Crop node:

1. Connect the DeepCrop node to the deep image you want to crop.
2. Adjust the crop box in the Viewer in X and Y directions to define your crop area. Alternatively, define your crop area using the **bbox** fields in the properties panel. If you want to keep the depth samples outside the crop box, you can check the **keep outside bbox** box.
3. Use the **znear** and **zfar** controls in the properties panel to crop samples in depth. If you don't want to use either of these controls, you can disable them by unchecking the **use** box next to them. If you want to keep your depth samples outside of the z range defined by these controls, you should check the **keep outside zrange** box.

Reformatting Deep Images

DeepReformat is the Reformat node for deep data. You can use it to set your deep image's dimensions, scale, and so on. To reformat your deep image:

1. Connect the DeepReformat node to the deep image you want to resize.
2. In the **type** dropdown, select:
   - **to format** - sets the output width and height to the selected format. Select the format in the **output format** dropdown. If the format does not yet exist, you can select **new** to create a new format from scratch. The default setting, **root.format**, resizes the image to the format indicated on the **Project Settings** dialog.
   - **to box** - sets the output width and height to dimensions you define in pixels. Enter values in the **width**, **height** and **pixel aspect** fields to specify the dimensions.
   - **scale** - sets the output width and height to a multiple of the input size. Use the **scale** slider to define the factor. The scale factor is rounded slightly, so that the output image is an integer number of pixels in the direction selected under **resize type**.
3. You can specify what kind of resize you want in the **resize type** dropdown. Select:
   - **none** - to not resize the original.
   - **width** - to scale the original until its width matches the output width. Height is then scaled in such a manner as to preserve the original aspect ratio.
   - **height** - to scale the original so that it fills the output height. Width is then scaled in such a manner as to preserve the original aspect ratio.
   - **fit** - to scale the original so that its smallest side fills the output width or height. The longest side is then scaled in such a manner as to preserve the original aspect ratio.
   - **fill** - to scale the original so that its longest side fills the output width or height. The smallest side is then scaled in such a manner as to preserve the original aspect ratio.
   - **distort** - to scale the original so that both sides fill the output dimensions. This option does not preserve the original aspect ratio, so distortions may occur.
4. Check the **center** box to define whether the input pixels should be resampled to the new size or centered in the output. If you don't center, the lower left corners of the input and output are aligned.
5. To further adjust your image's layout, you can check the respective boxes for:
   - **flip** - to swap the top and bottom of the image.
   - **flop** - to swap the left and right of the image.
   - **turn** - to turn the image 90 degrees.
   - **black outside** - to set pixels outside the format black.
   - **preserve bounding box** - to preserve pixels outside the output format rather than clipping them off.

Transforming Deep Samples

You can use the DeepTransform node to reposition the deep samples.

1. Connect the node to the deep footage you want to transform.
2. Use the **translate x**, **y**, and **z** controls to translate your samples.
3. Scale the samples' z depth using the **zscale** control. Values above 1 increase the depth, whereas values below 1 decrease it.
4. If you connect a mask to the node's **mask** input, you can use it to regulate how much of an effect the depth transformation has in different parts of the frame.

Sampling Deep Images

You can use the DeepSample node to sample any given pixel in a deep image. The Deep Sample node gives you the depth data as figures.

1. Connect the DeepSample node to another Deep node.
2. Position the **pos** indicator over the pixels you want to sample in the Viewer.
3. View the deep sample information in the sample table on the DeepSample properties panel.
4. You can also toggle the **accumulate** box to select whether you want to see the individual sample values of the sample pixel (unchecked), or the final composited value (checked).

Writing Deep Data

You can write out deep images in the scanline OpenEXR 3.3.6, or above, format using the DeepWrite node, which shares a lot of controls with the standard Write node. Do the following:

1. Select **Deep** > **DeepWrite** to insert a DeepWrite node into your script.
2. In the properties panel, click the **file** or **proxy** field's folder icon and browse to the directory where you want to store the deep image.
3. After the path, type a name for the deep image, including the .exr extension, and then click **OK**. If you're rendering an image sequence, include the frame number variable (for example, **####**) in the name.
4. Use the **datatype** dropdown menu to select the bit depth for the rendered file: **16 bit half** or **32 bit float**.
5. Set **compression** to the compression type to apply to the rendered file.

6. From the **metadata** dropdown menu, select what metadata is included with the rendered file:
   - **no metadata** - No custom attributes are created, and only metadata that fills required header fields is written out.
   - **default metadata** - The optional time code, edge code, frame rate, and exposure header fields are also filled using metadata values.
   - **default metadata and exr/***
   - **all metadata except input/***
   - **all metadata**
7. By default, unknown metadata keys have the prefix **nuke** attached to them when they are written into the file. If you'd rather have them written into the file "as is", without the prefix, check **do not attach prefix**.
8. Adjust the rest of the controls as necessary. For more information on them, see Output (Write) Nodes.
9. Click the **Render** button.

   Nuke prompts for a frame range, defaulting to the range in the **frame range** fields.
10. If necessary, change the start and end frames, and then click **OK**.

    Nuke writes the deep data to a scanline OpenEXR 3.3.6, or above, file (tiled OpenEXR files are not supported).

Working with File Metadata

The Read node's **Metadata** tab and the nodes in the **Metadata** menu of the Toolbar let you work with information embedded in your images. This section gives instructions on their usage, teaching you to view, compare, edit, and render metadata.

Metadata

Metadata is a set of information about an image embedded in the image file. This information may include the image's original bit depth, width, and height, for example. It can be attached to the file by the camera used to shoot the images, and/or edited later.

When Nuke loads an image, it reads in the metadata embedded in the image. The metadata is then passed down the node tree so you can view and use it at any point in your script. For example, you can reference metadata via expressions. You can also edit or delete existing metadata, add new metadata to a file, and write the resulting metadata out to files.

Note: Metadata for QuickTime files does not show gamma or bit depth.

Note: The Read and Write node's **timecode** and **edge code** fields have been removed from the Properties panel, but the metadata is still available using the **input/timecode** and **input/edgecode** keys. See Viewing Metadata for more information.

Tip: When using a Merge node, you can choose which input's metadata to pass down the tree. In the Merge controls, set **metadata from** to either **A** or **B**.

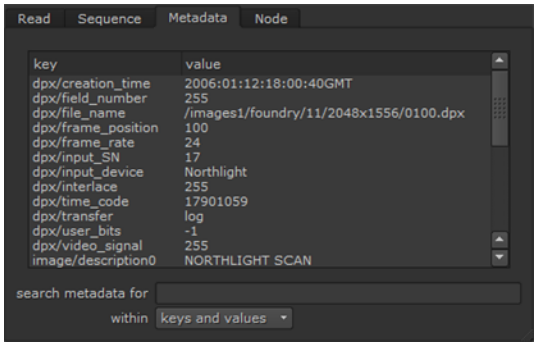
As well as the **Metadata** tab in the Read node, the **MetaData** menu of Nuke's Toolbar features five nodes that help you work with file metadata:

- ViewMetaData lets you inspect the metadata passed down by the input node. See Viewing Metadata.
- CompareMetaData lets you compare the metadata between two inputs and view the differences. See Comparing Metadata Between Inputs.

- ModifyMetaData lets you edit existing metadata in the input stream, add metadata into the stream, and delete metadata from the stream. See Modifying Metadata.
- CopyMetaData lets you copy metadata from one input to another and filter metadata to exclude some of it. See Copying and Filtering Metadata Between Inputs.
- AddTimeCode lets you add a timecode to the metadata passed down by the input node. See Adding a Time Code to Metadata.

Viewing Metadata

The simplest way to view file metadata is by clicking the **Metadata** tab in the **Properties** panel of a standard Read node. All the available metadata is displayed, along with a simple search function.



To filter the lists of metadata, use the **search metadata for** field. For example, if you enter **f** in the **search metadata for** field, only the keys and values that include the letter f are displayed. By default, the search is done within both keys and values. If you want to limit the search to either keys or values only, set **within** to **keys only** or **values only**. For example, you can view metadata specific to the input's file format by entering the file format (for instance, **dpx/**) in the **search metadata for** field and setting **within** to **keys only**.

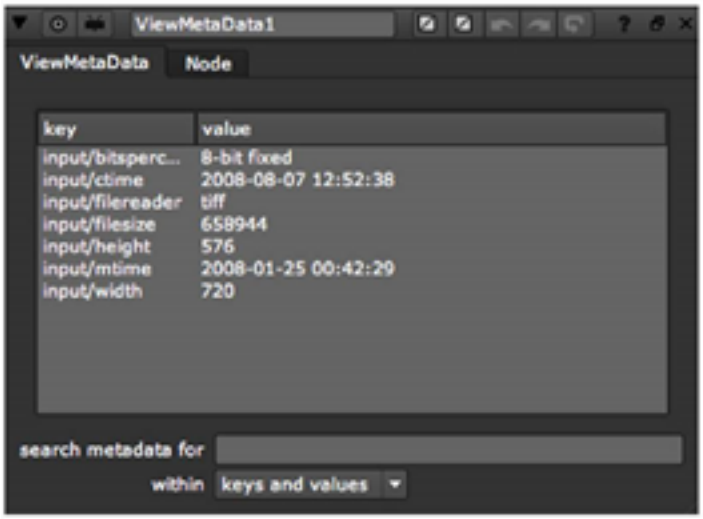
Note: When observing the creation time (input/ctime) of an image, Windows generally differs from Linux and Mac. This is due to the different way in which the operating systems deal with file creation times.

Once you know which keys exist on the input, you can reference them in expressions. See Accessing Metadata Using Tcl Expressions.

You can also view metadata using the ViewMetaData node:

1. Select **MetaData** > **ViewMetaData** to insert a ViewMetaData node after the node whose metadata you want to inspect.

2. In the ViewMetaData properties, you can see a list of the metadata embedded in the image. This is divided into keys and their values.



Comparing Metadata Between Inputs

To compare metadata between two inputs:

1. From the Toolbar, select **MetaData** > **CompareMetaData** to add a CompareMetaData node after the two nodes whose metadata you want to compare.
2. Connect the nodes you want to compare to the **A** and **B** inputs of the CompareMetaData node.

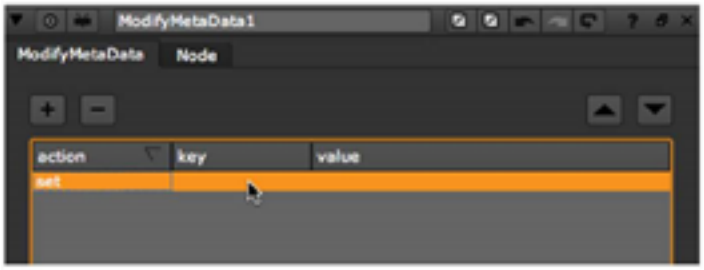
   A list of keys where there is a difference between the two inputs is shown in the CompareMetaData properties.

Modifying Metadata

There are several ways to modify metadata in Nuke.

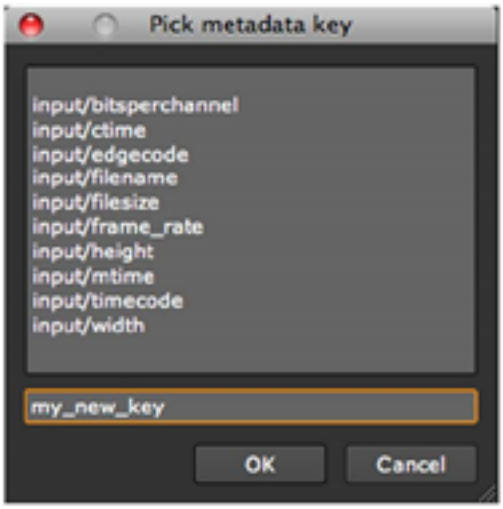
To Add Metadata

1. Select **MetaData** > **ModifyMetaData** to insert a ModifyMetaData node after the node whose metadata you want to add a new key to.
2. In the ModifyMetaData controls, click on the plus (**+**) button. A placeholder appears in the metadata box.
3. Double-click on the placeholder under **key**.



The **Pick metadata key** dialog opens.

4. In the field at the bottom of the dialog, enter a name for the new key you want to add to the metadata. Click **OK**.



5. Double-click on the placeholder under **data** and enter a value for the new key.

   The new key and its value are added to the metadata being passed through.

To Edit Metadata

1. Select **MetaData** > **ModifyMetaData** to insert a ModifyMetaData node after the node whose metadata you want to edit.
2. In the ModifyMetaData controls, click on the plus (**+**) button.

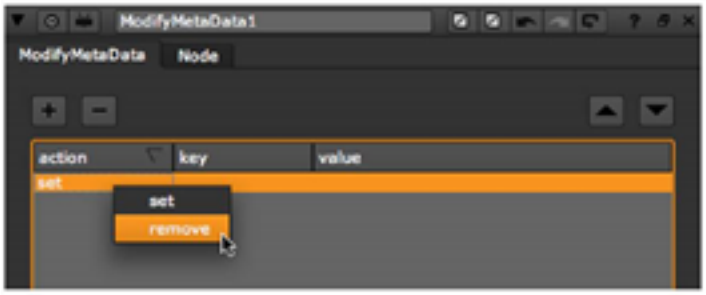
   A placeholder appears in the metadata box.
3. Double-click on the placeholder under **key**.

   The **Pick metadata key** dialog opens.
4. Pick the key whose name or value you want to edit and click **OK**.

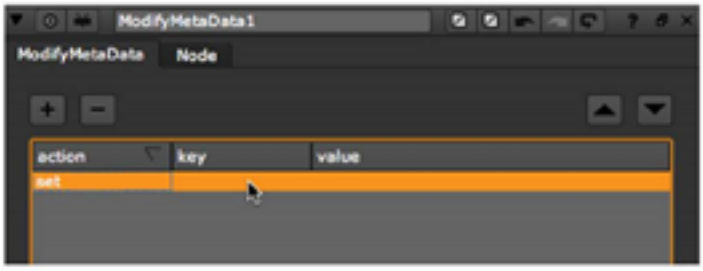
   The key is added to the ModifyMetaData properties.
5. In the ModifyMetaData properties, double-click on the key or its value and edit the information as required.

To Remove Metadata

1. Select **MetaData** > **ModifyMetaData** to insert a ModifyMetaData node after the node whose metadata you want to edit.
2. In the ModifyMetaData properties, click on the plus (**+**) button. A placeholder is added to the metadata list.
3. Double-click on the placeholder under **action** and select **remove** from the menu that opens.

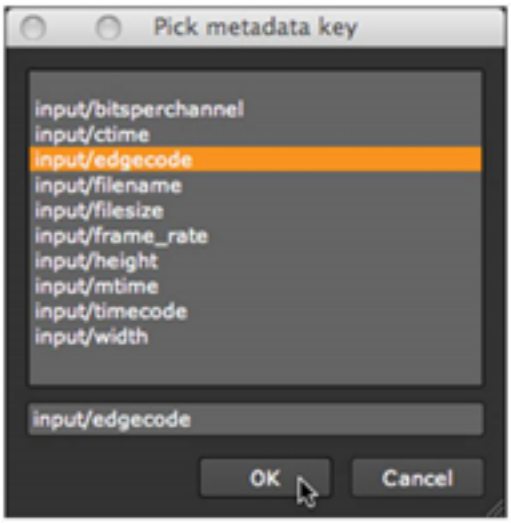


4. Double-click on the placeholder under **key**.



The **Pick metadata key** dialog opens.

5. From the list of existing keys, select the key you want to remove and click **OK**.



The node now removes the selected key from the metadata, as you can see if you view the metadata in the output.

To Edit the List of Actions in the ModifyMetaData Properties

- To perform a new action, click on the plus (**+**) button.
- To cancel an existing action, select it from the list and click on the minus (**-**) button. Note that this only affects the ModifyMetaData actions, and does NOT delete keys from the metadata embedded in the input image.
- To move an item up in the list, select it and click on the up arrow button.
- To move an item down in the list, select it and click on the down arrow button.

Copying and Filtering Metadata Between Inputs

To copy metadata from one input to another and/or filter metadata:

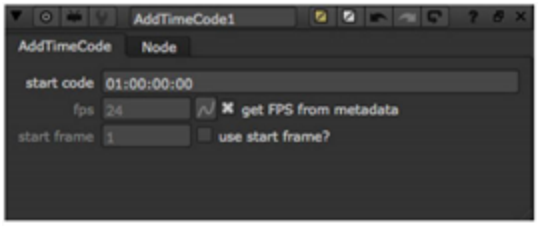
1. Select **MetaData** > **CopyMetaData** to insert a CopyMetaData node into your script.
2. Connect:
   - the **Image** input to the node whose image you want to pass down the tree.
   - the **Meta** input to the node whose metadata you want to copy to the output.
3. Set **metadata from** to one of the following:
   - **Image+Meta** - to add the metadata from the **Meta** input to the metadata from the **Image** input. If the inputs share any common metadata keys, the values taken from the Meta input override those taken from the Image input.
   - **Meta only** - to only use the metadata from the Meta input.
   - **Meta+Image** - to add the metadata from the **Image** input to the metadata from the **Meta** input. If the inputs share any common metadata keys, the values taken from the Image input override those taken from the Meta input.
   - **Image only** - to only use the metadata from the Image input. This produces the same result as not using a CopyMetaData at all: both the image and metadata are taken from the Image input. However, this option can be useful if you want to filter the metadata to exclude some of it (see the next step for how to do this).

4. To filter the metadata taken from the inputs, use the **copy only** fields under **Meta filtering** and/or **Image filtering**. For example, if you enter **f** in the **copy only** field under **Meta filtering**, only the keys and values that include the letter f are copied from the Meta input. By default, the search is done within both keys and values. If you want to limit the search to either keys or values only, set **within** to **keys only** or **values only**. For example, you can copy metadata specific to the input's file format by entering the file format (for instance, **dpx/**) in the **copy only** field and setting **within** to **keys only**.

Adding a Time Code to Metadata

1. Select **MetaData** > **AddTimeCode** to insert an AddTimeCode node into your node tree.

   A time code is added to the metadata being passed through. By default, the time code is 01:00:00:00 on the first frame. It is updated throughout the frame range according to the input clip's playback speed, which in turn is controlled by the **fps** (frames per second) parameter in the Project Settings. If you change the **fps** value in the Project Settings, the time code in the metadata is updated to reflect the change.



   Nuke can also deal with drop frames, such as when a clip's frame rate is 29.97 or 59.94. Instead of the default HH:MM:SS:FF time code format, use the format HH;MM;SS;FF, delimited by **;** (semicolon).



Note: Using semicolon delimiters with non-drop frame time codes displays an error in the Viewer.

2. If you don't want the time code on the start frame to be 01:00:00:00, enter a new time code in the **startcode** field.
3. If you want to specify the playback speed manually rather than get it from the metadata and project settings, uncheck **get FPS from metadata** and enter a new value in the **fps** field.
4. If you want to specify a different start frame than the first frame, check **use start frame?** and enter a new value in the **start frame** field.

If you want to display the time code on the image, insert a Text node after the AddTimeCode node and enter **[timecode]** in the **message** field. For more information on referencing metadata via expressions, see Accessing Metadata Using Tcl Expressions.

Rendering Metadata

When rendering with the Write node, Nuke lets you write out metadata into the following file formats: **.exr**, **.cin**, **.dpx**, and **.jpg**. You cannot write out metadata into any other formats.

When rendering metadata into an **.exr** file, you can use the **metadata** dropdown menu in the Write node controls to specify what to write out:

- **no metadata** - Do not write out any metadata, except for metadata that fills required header fields (for example, file name and bbox).
- **default metadata** - Write out the time code, edge code, frame rate, and exposure.
- **default metadata and exr/*** - Write out the time code, edge code, frame rate, exposure, and anything in **exr/**.
- **all metadata except input/*** - Write out all metadata, except anything in **input/**.
- **all metadata** - Write out all metadata.

When rendering any other file format than **.exr**, Nuke writes whatever metadata the file format header is known to support. Therefore, what is written out varies according to the file format.

Accessing Metadata Using Tcl Expressions

You can access metadata via Tcl expressions in the following ways:

- To get a list of all keys in the incoming metadata, use the expression **[metadata]**. For example, if you add a Text node after an image and enter **[metadata]** in the **message** field, a list of all the keys in the incoming metadata appears on the image. The values of the keys are not displayed.

  To get a list of all keys and values, use the expression **[metadata values]**.
- To get all the all values from a particular node, us the **-n** argument. For example, the expression **[metadata -n Read3 values]** returns all metadata values from the node Read3.
- To get the value of a particular key in the incoming metadata, use the expression **[metadata *key*]**. Replace ***key*** with the name of the key whose value you want to use. For example, to display the name and location of the image file on the image, add a Text node after the image and enter **[metadata input/filename]** in the **message** field.

- To get a filtered list of keys in the incoming metadata, use the expression **[metadata keys *filter*]**. Replace ***filter*** with whatever you want to use to filter the list. You can use asterisks (*****) as wildcards in your filter to substitute zero or more characters in the key names. For example, to get a list of all the keys with the letter **f** in them, use the expression **[metadata keys *f*]**. To get a list of all the keys starting with **input/**, use the expression **[metadata keys input/*]**.

  By default, the keys are listed on separate lines. To change this, you can use **-s "*separator*"** to have the keys separated by a separator of your choice. Replace ***separator*** with whatever you want to appear between the different keys. For example, to get a list of all the keys starting with **input/** and separated by spaces, you can use **[metadata -s " " keys input/*]**. To get the same list separated by commas, use **[metadata -s ", " keys input/*].**

  You can also use a custom separator between the key and value pairs, rather than the default **key=value**. For example, the expression **[metadata -v "--" values]** returns all metadata values using "--" instead of "=", like: **exr/pixelAspectRatio--1**

By default, if you attempt to access metadata that does not exist in the stream, Nuke returns an empty string. To make this error instead, use the **-e** flag before other parameters.

For more information on using expressions, see the Expressions chapter.

Accessing Metadata Using Python

You can also access metadata using the Python programming language. For more information, see the Nuke Python documentation (**Help** > **Documentation**).

Audio in Nuke

In many compositing projects it's vital to be able to key visual changes to cues on the audio track that goes with the picture. You can use Nuke's AudioRead node to read in an audio file, view it in the Curve Editor and Dope Sheet in order to line up keyframes of your composition with the waveform of the sound. You can then flipbook the audio with your footage to preview your comp with sound.

Quick Start

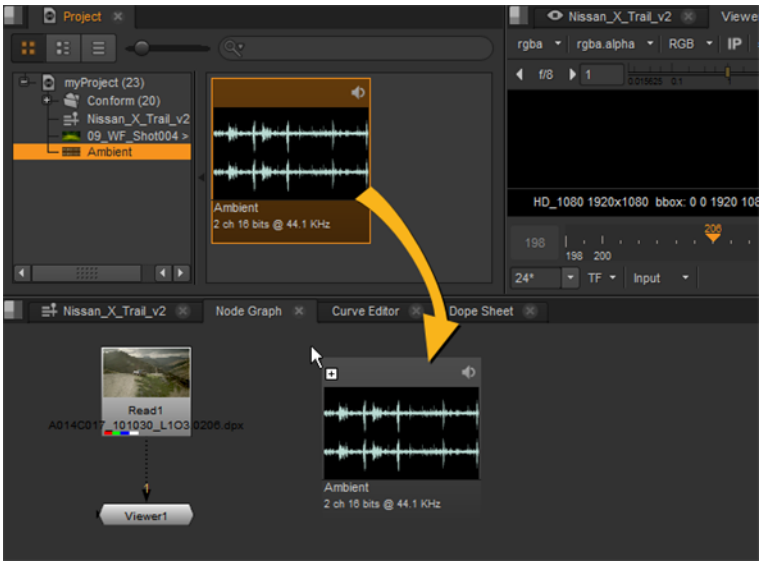
You can load audio files into Nuke using the AudioRead node, in much the same way as you read in images with the Read node. You can read in uncompressed **.wav** and **.aiff** files and flipbook them with your footage for playback.

Here's a quick overview of the workflow:

1. Read in an audio file. See Reading Audio Files into the Node Graph.
2. Display an audio waveform for your audio clip and access its animation curve in the Curve Editor or the Dope Sheet. See Creating and Editing Audio Curves.
3. When you're done, you can flipbook your script to view and listen to the results. See Flipbooking the Audio Track.

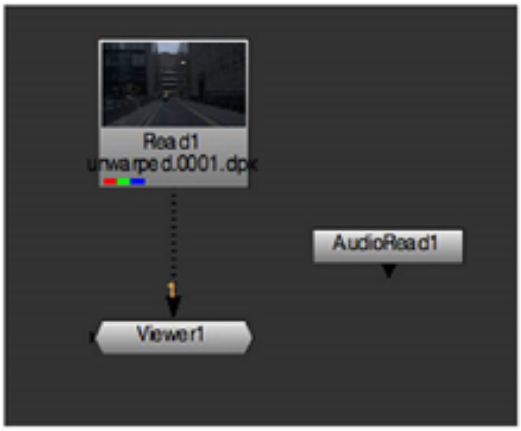
Reading Audio Files into the Node Graph

You can drag and drop audio clips from the **Project** tab to the Node Graph, if the clip is already in Nuke.



Otherwise, use the AudioRead node to read in an audio file:

1. To create an AudioRead node, click **Other** > **AudioRead** in the Nuke Toolbar.
   The AudioRead node doesn't have to be connected to other nodes.



Simple AudioRead node setup.

Tip: You can also load an audio file by creating a normal Read node and navigating to a supported audio file.

2. In the AudioRead properties, use the **file** control to navigate to the audio file you want to read in. You can read in uncompressed **.wav** and **.aiff** files.
3. Use the **time range** fields to enter the start and end times in seconds for the audio in Nuke.
4. In the **file time range** fields, enter the start and end times in seconds of the audio file read in. These are automatically set to the values in the file, but you can change them to trim the data used in Nuke.
5. If you want to discard your changes and reload the audio file, click **reload.**

6. Use the **ratesource** menu to select the source for the sample rate:
   - **file** - reads the rate from the audio file.
   - **custom** - lets you specify a custom sample rate in the **rate** field.

Creating and Editing Audio Curves

Once you have read in an audio file (see Reading Audio Files into the Node Graph), you can display an audio waveform for your audio clip and access its animation curve in the Curve Editor or the Dope Sheet.

Creating a Keyframe Curve

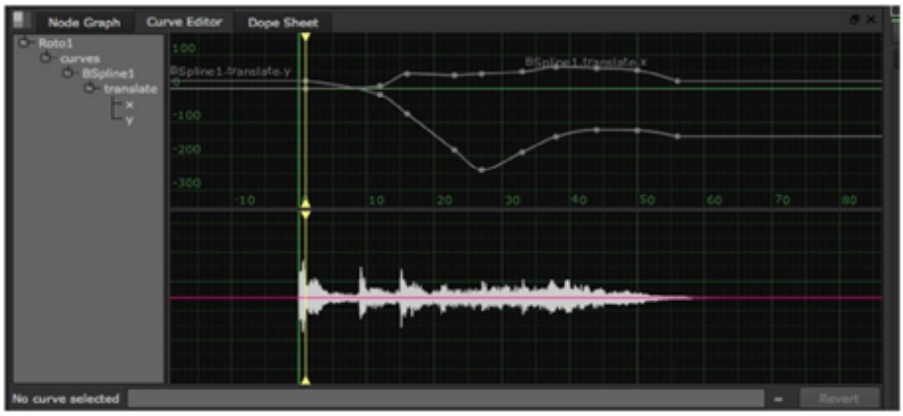
In the **curves** section of the AudioRead properties panel, you can generate curves out of the audio data:

1. Set the keyframe interval you want to use when creating the curves in the **key interval** field. For example, if you enter 3, keyframes are created to every third frame of the input footage.
2. Click **generate** to generate the audio data as a curve that you can use in the Curve Editor and Dope Sheet.
3. View the left and right stereo levels on the current frame in the **left** and **right** fields and adjust if necessary. Any changes are reflected on the curve automatically.

Modifying the Audio Curve in the Curve Editor and Dope Sheet

When you're working with your audio curve in the Curve Editor or Dope Sheet, there are a few right-click options that you can use to adjust how the clip's waveform displays:

1. Right-click in the Curve Editor or Dope Sheet and select **View** > **Audio**.
2. Then select **Source** and check the box for either **ProjectDefault** or an **AudioRead** node depending on which one you want to view. If you've only got one AudioRead, it is the project default.
3. If you're working with a stereo clip with more than one audio channel, you can select your audio channel by ticking the appropriate box under **Channel**.
4. Select a style in which you want your waveform to be drawn by selecting one of the **DrawStyle** options:
   - **Off** - to draw no audio waveform.
   - **Behind** - to draw a waveform behind the animation curves.
   - **Below** - to draw a waveform below the animation curves.



Audio waveform in the Curve Editor.

Flipbooking the Audio Track

When you're done, you can proceed to flipbooking your results:

1. Click the **flipbook this Viewer** button in the Comp Viewer.
2. In the **Flipbook** dialog, select the AudioRead file you want to use in the **Audio** dropdown.
3. Click **OK**. View and listen to your clip in Flipbook.

Previews and Rendering

Nuke supports a fast, high-quality internal renderer, with superior color resolution and dynamic range without a slowdown in the workflow.

About Rendering in Nuke

These are some of the key features of Nuke's rendering engine:

- Multi-threaded rendering to take advantage of multiple processors in its calculations.
- Scanline (as opposed to buffer-based) rendering allows you to immediately see portions of render output.
- Calculations performed with 32-bit precision, using linear light levels.

This chapter teaches you how to use the renderer's various features to preview a script's output and generate its final elements. You'll also learn how to preview using Flipbooking and Capture, and check output on an external broadcast video monitor.

Quick Start

Here's a quick overview of the workflow:

1. With the Viewer, you can preview your footage and use the **ROI** button  to focus on a particular part of it. For more information, see Previewing Output.
2. You can then flipbook your clip. A quick way of doing this is to click the **Flipbook** button  in the Viewer, set the frame range and other settings in the **Flipbook** dialog, and click **OK** to flipbook and automatically launch the **Flipbook** Viewer. See Flipbooking Sequences.
3. Save out low resolution **.jpg** sequences using **Capture** to share your work, such as for peer review purposes. Click the **Capture** button  in the Viewer, set the frame range and other settings in the **Capture** dialog, and click **OK**.
4. If you've read in an audio clip with the AudioRead node, you can flipbook that with your footage just by selecting the right AudioRead node in the **Audio** dropdown. For more information, see Audio in Nuke.
5. To check the final result in correct video colorspace and pixel aspect ratio, preview your footage on an external broadcast video monitor. See Monitor Output in Nuke Studio (NDI - SDI/HDMI).

6. When you're happy with your preview results, you can render out your clip. To do this, you need to connect at least one Write node to your clip, and then set the render properties in the properties panel. You can specify your render format in the **filename** field, and use the **frame** control to offset your frame numbers if necessary. For more information, see Output (Write) Nodes.
7. If you have more than one Write node connected to your node tree, you can render out all of them, or select the ones you want to render. You can then click **Render** > **Render all** or **Render selected** in the menu bar to start the renders. For more information, see Output (Write) Nodes.

Previewing Output

This section explains how to preview individual frames in a Nuke Viewer window (see Previewing in a Nuke Viewer), how to render a flipbook for a sequence of frames (see Flipbooking Sequences), and how to preview output on an external broadcast video monitor (see Monitor Output in Nuke Studio (NDI - SDI/HDMI)).

Previewing in a Nuke Viewer

When you connect a Viewer to a given node's output (by selecting the node and pressing a number key), Nuke immediately starts rendering the output in the Viewer using all available local processors.

Keep in mind the following tips in order to speed up this type of preview rendering:

- First, if you don't need to evaluate the whole image, zoom into the area of interest. Nuke then renders only the portion of scan lines visible within the Viewer.
- Alternatively, you can use the Viewer's region of interest (ROI) feature to render only a portion of the image, while seeing that result in the context of the whole image.

To Enable the ROI Render Feature

1. Press **Alt**+**W** over the Viewer. The Viewer's **ROI** button turns red , indicating that the feature is enabled.
2. Drag on the Viewer to draw the region of interest. The Viewer now renders only the pixels within the region.

To Edit the Position or Size of Current ROI

1. Click the **ROI** button so that it turns red . The overlay for the current ROI appears in the Viewer.
2. To reposition the ROI:
   Using the crosshair in the middle of the ROI, drag the ROI to the desired location.
3. To resize the ROI:
   Drag any corner or side of the ROI until you achieve the desired size.

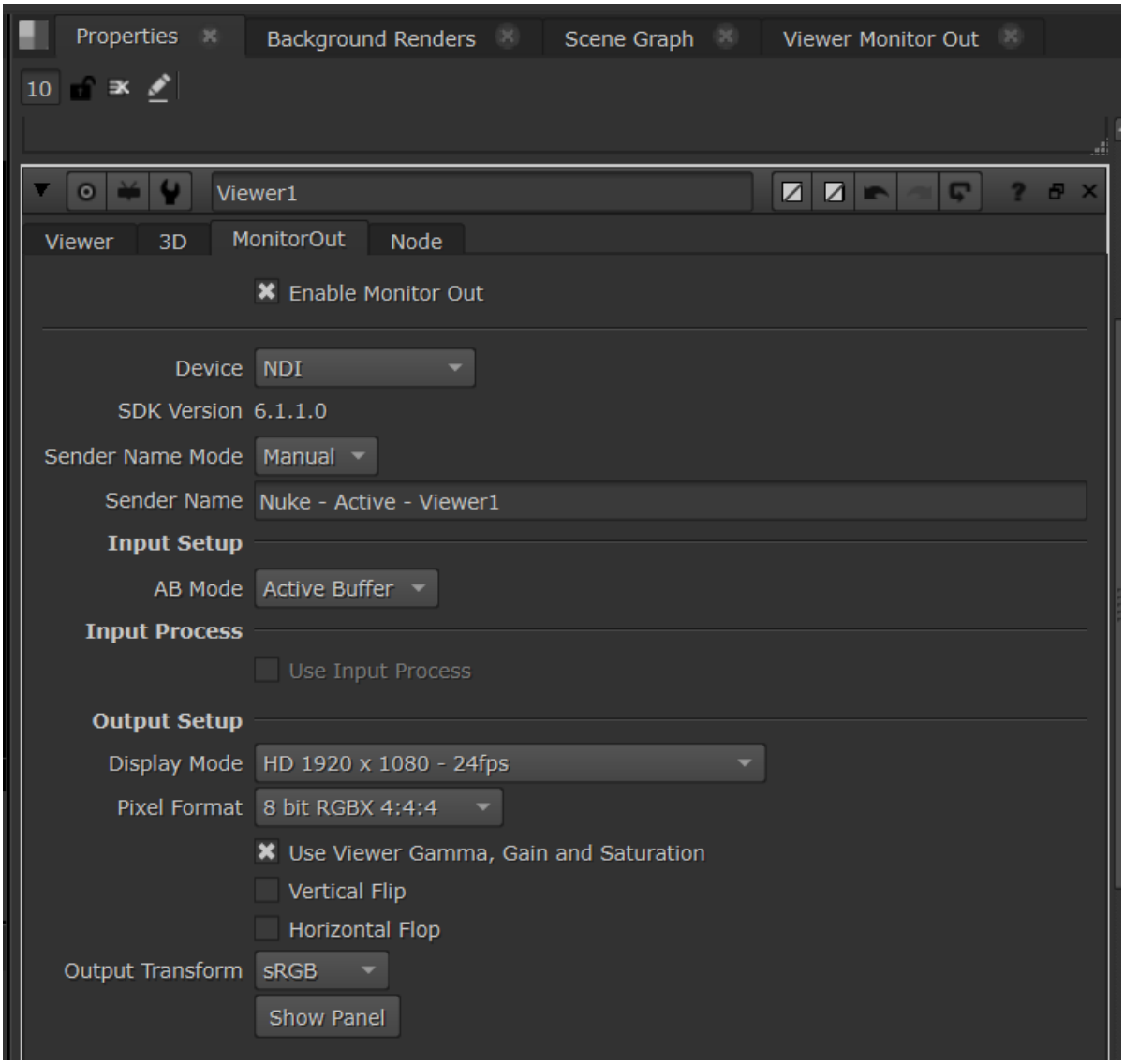
To Disable the ROI Render Feature

Click the Viewer's **ROI** button. It turns gray , signaling that it is off. The Viewer now renders all of the visible image.

Reviewing Your Work with Monitor Out in Nuke (NDI - SDI/HDMI)

Monitor Out allows you to preview Viewer images in a floating window, NDI session, or on an external projector/video monitor to check the final result, including the correct output transform and aspect ratio.

A dedicated **Viewer Monitor Out** workspace provides easy access to the monitor out strip and its controls, but you can also access the controls by double-clicking a Viewer node to display the **Properties** panel.



Nuke supports internal monitor out using a **Floating Window**, NDI sessions, and dedicated AJA or Blackmagic cards, selected from the **Device** dropdown in the Viewer **Properties**:

- **Floating Window** - Nuke's native monitor out uses a floating monitor and does not require additional hardware. Mac users can also take advantage of additional HDR technology including an extended sRGBf LUT. See Working with HDR Images for more information.
- **NDI** - A network protocol that enables video and metadata signals to be sent over standard IP networks in real-time. This option allows you to easily share a stream of the Viewer output to anyone else on the network.

Note: To view the NDI stream, you need the NDI Studio Monitor or NDI Video Monitor application installed. See Streaming Viewer Output Over the Internet with NDI® for more information.

- **Dedicated AJA or Blackmagic Design cards** - Nuke supports various AJA and Blackmagic devices that can be selected from the **Device** dropdown after installation. See AJA and Blackmagic Design Monitor Out Devices for more information on supported hardware and drivers.

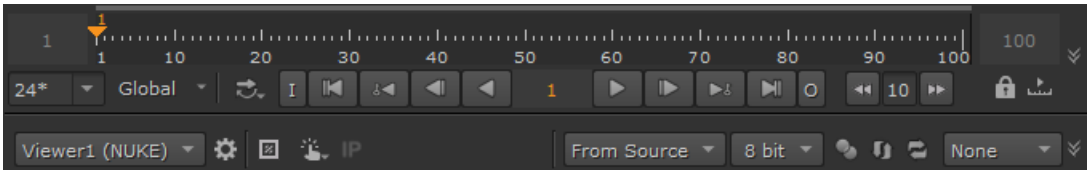
Note: Audio scrubbing is not currently available through monitor out cards. Audio scrubbing is only supported through internal audio output devices.

Nuke also supports monitor out through VR headsets. See Reviewing on VR Headsets (HMD) for information on supported headsets.

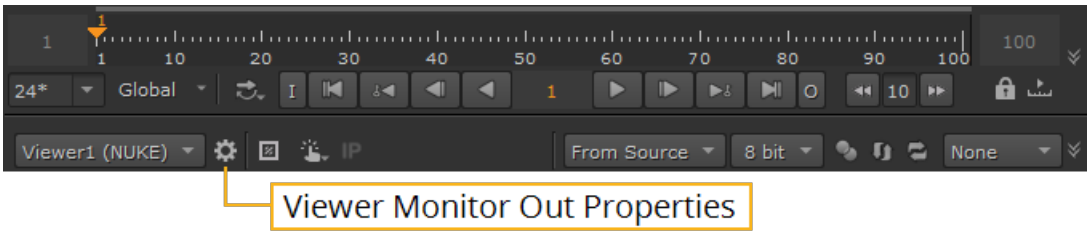
Enabling Nuke Monitor Out

Nuke's monitor out uses a floating monitor, NDI session, or additional hardware from AJA or Blackmagic to display monitor out. The floating monitor allows Mac users to take advantage of additional HDR technology including an extended sRGBf LUT. See Working with HDR Images for more information.

1. Navigate to **Workspace** > **Viewer Monitor Out**.
   The **Viewer Monitor Out** workspace is displayed with the monitor out toolbar under the Viewer by default.



   If you're working with multi-view footage, use **Interactive** buffer mode to show the output from the selected Viewer. See Stereoscopic Scripts or Stereoscopic and Multi-View Projects for more information.
2. Click the cog icon in the bottom-left of the **Viewer Monitor Out** panel to display the **Properties** panel or double-click the **Viewer** node.



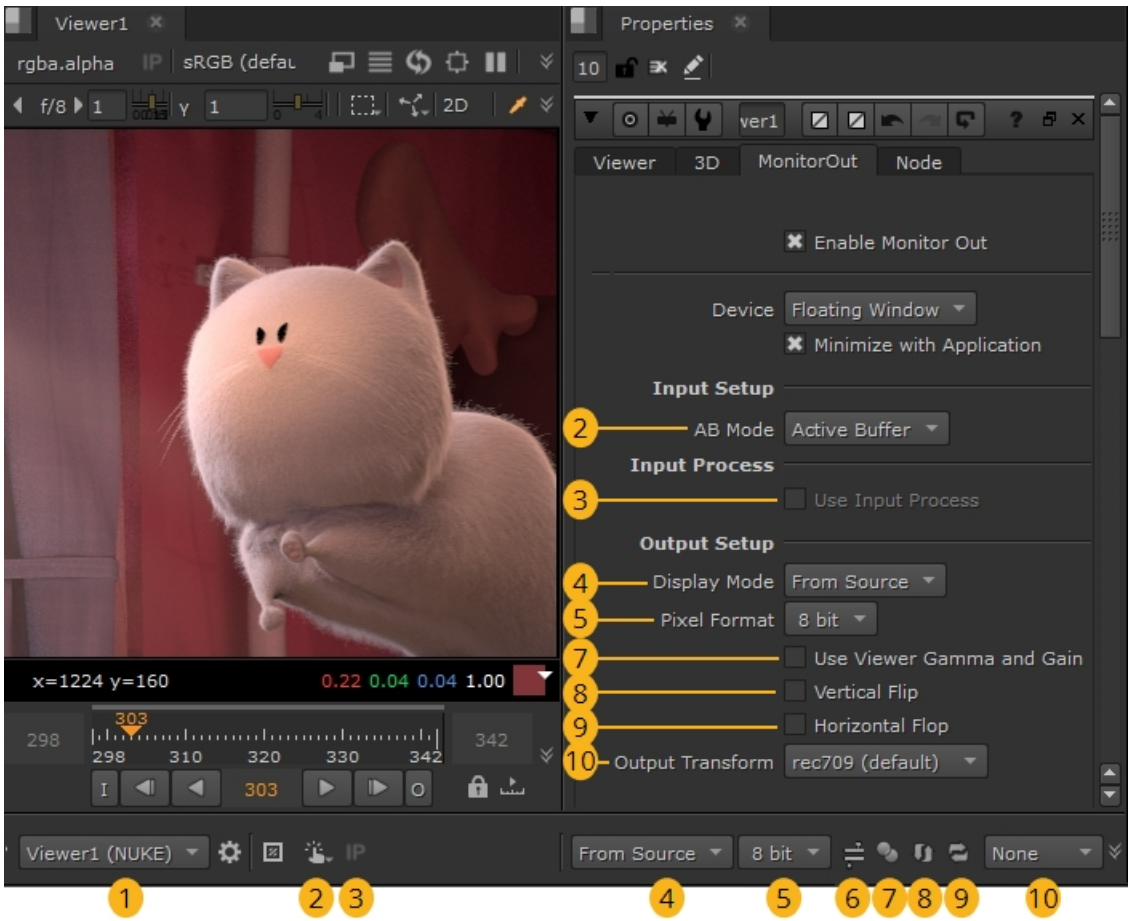
3. Select the device you want to use from the output **Device** dropdown. All available devices are automatically detected and listed in this menu, along with the default options:
   - **Floating Window** - opens a floating monitor window, without the need for a monitor device and card. This is intended for full-screen use without displaying any of the interface.
   - **NDI** - A network protocol that enables video and metadata signals to be sent over standard IP networks in real-time. This option allows you to easily share a stream of the Viewer output to anyone else on the network.



Note: To view the NDI stream, you need the NDI Studio Monitor or NDI Video Monitor application installed. See Streaming Viewer Output Over the Internet with NDI® for more information.

   - **Installed Cards** - transfers the Viewer contents to the selected card and displays SDK and driver information associated with the card.

The **Monitor Out** panel and Viewer **Properties** panel duplicate some controls for convenience.



1	Viewer	Select the **Viewer** to feed to the output monitor using the source viewer dropdown. Selecting **Active Viewer** always displays the Viewer that is currently selected in the workspace.
2		Select the view mode using the **AB Mode** dropdown: • **Active Buffer** - the monitor always shows the output from the buffer currently in use by the selected Viewer. • **Buffer A** - the monitor always shows the output from the selected Viewer buffer A. • **Buffer B** - the monitor always shows the output from the selected Viewer buffer B. • **Interactive** - the monitor shows the output from the selected Viewer including the A/B player mode currently in use, such as **wipe**.
3		Click the button or enable **Use Input Process** to apply an input process node before displaying the image on the external monitor or floating window. This allows you to display different color information in Nuke's Viewer and monitor or floating window. To define the node to use, enter its name in the Viewer node's **input process** field or, from the Node Graph right-click menu, select **Edit** > **Node** > **Use as Input Process**.
4	Display Mode	Select the resolution output from the monitor card currently in use. The resolutions available depend on the card manufacturer.  **Note:** Floating monitor out always uses the resolution of the Viewer in use.
5	Pixel Format	Select the pixel format output from the monitor card currently in use. The formats available depend on the card manufacturer.
6		Select the output range between 0-255, when the button is enabled and SMPTE-range 16-236, when the button is disabled.  **Note:** This control is not included in the **Viewer Monitor Out** properties and is only displayed when a dedicated monitor out card is selected from the **Device** dropdown.

7		Click the button or enable **Use Viewer Gamma, Gain and Saturation** to apply the active Viewer's filtering, gamma, gain and saturation to the monitor out.
8		Click the button or enable **Vertical Flip** to flip the output vertically. This control is handy if you're using a ceiling mounted projector.
9		Click the button or enable **Horizontal Flip** to flop the output horizontally.
10	Output Transform	Select the color transform to apply to the image. If you've specified an OCIO configuration file in the preferences, these custom LUTs are also applicable.

Comparing and Combining Multi-Viewer Output

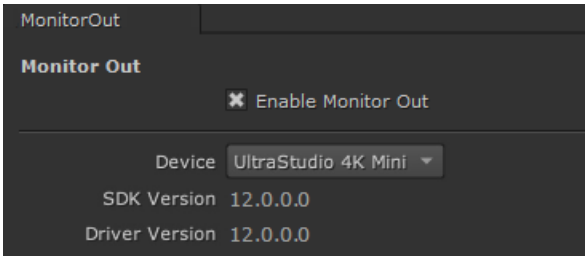
Nuke monitor out functionality is controlled by Viewer nodes, so you can compare or combine multiple streams from as many Viewers as you like to ensure shot consistency. Each Viewer can be enabled separately to output your shots to multiple output devices, whether they're dedicated monitor out devices, NDI streams, or Nuke's native Floating Window.

1. Double-click a source **Viewer** to open its **Properties** panel.
2. Go to the **Monitor Out** tab and then select the device you want to use from the output **Device** dropdown.
   All available devices are automatically detected and listed in this menu, along with the default options:
   - **Floating Window** - opens a floating monitor window, without the need for a monitor device and card.
   - **NDI** - a network protocol that enables video and metadata signals to be sent over standard IP networks in real-time.

   Note: To view the NDI stream, you need the NDI Studio Monitor or NDI Video Monitor application installed. See Streaming Viewer Output Over the Internet with NDI® for more information.

   - **Installed Cards** - transfers the Viewer contents to the selected card and displays SDK and driver information associated with the card.

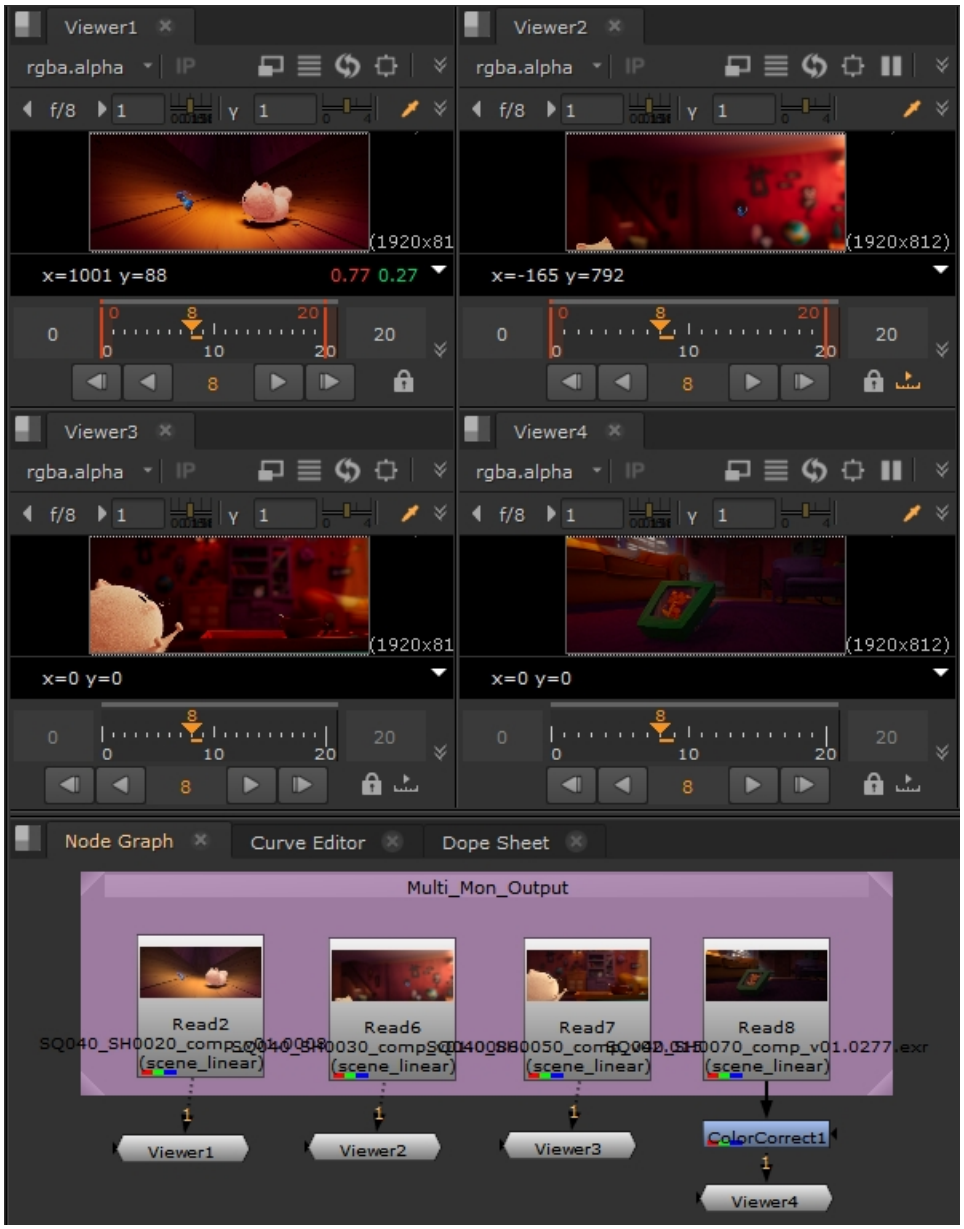
3. 



Note: SDI cards are grayed-out when they are in use by another Viewer and cannot be selected.

4. Repeat the process with as many Viewer nodes as you like. You can combine device types as you see fit, allowing you to compare dedicated monitor out, floating windows, and NDI streams as required.

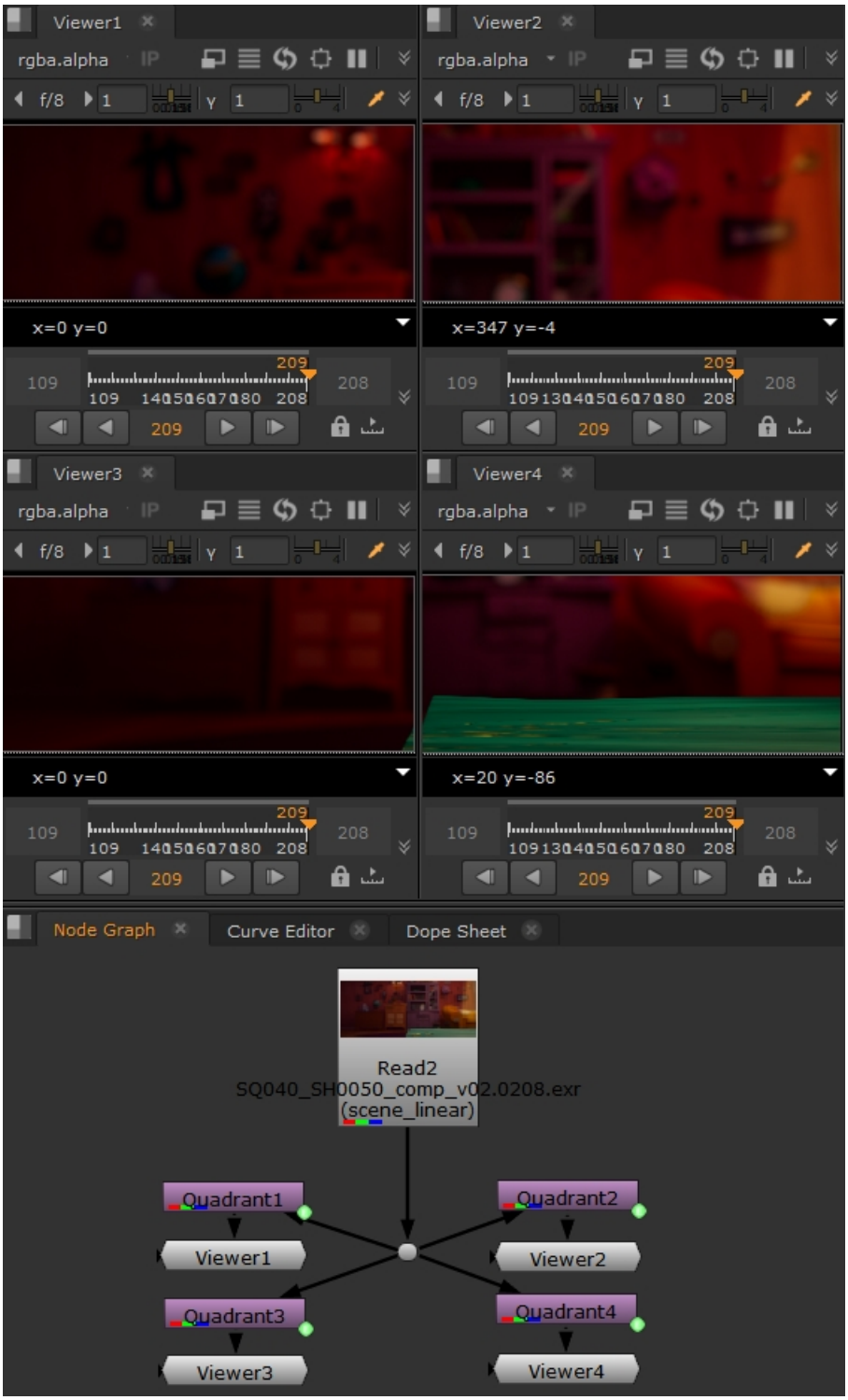
   For example, using four Viewers and four Nuke **Floating Windows**, you could compare the format of multiple shots.



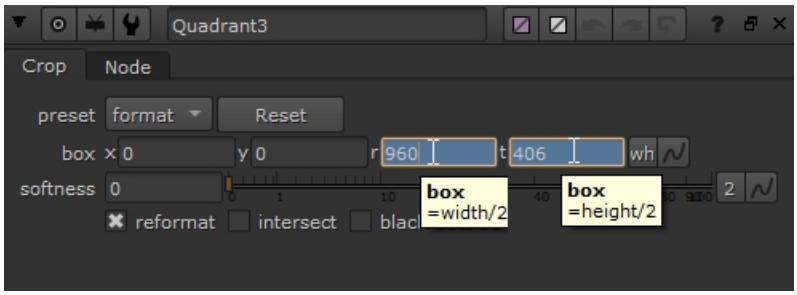
You can switch the **Output Transform** on each Viewer independently to check color consistency between the shots.

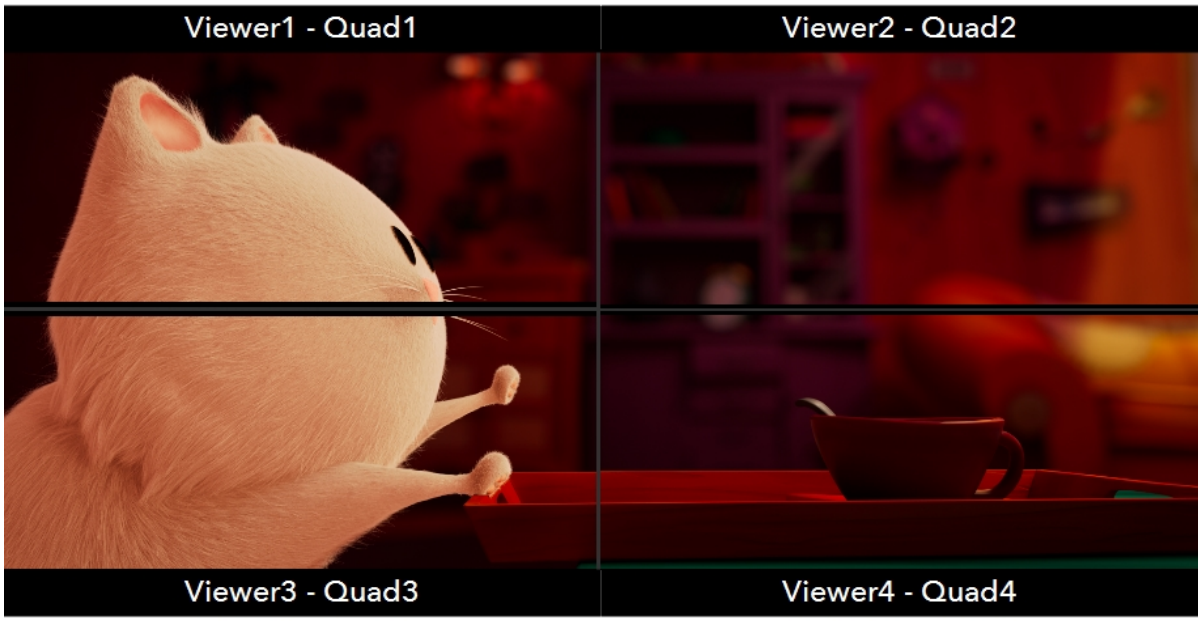


Using a similar four-Viewer setup, you could send four quadrants over NDI to four projectors to produce a larger image somewhere else on the same network. See Streaming Viewer Output Over the Internet with NDI® for more information.



Each quadrant is a Crop node that defines which part of the overall image is handled by each Viewer. For example, Quadrant3 is set up as shown here, with simple expressions to handle different formats.

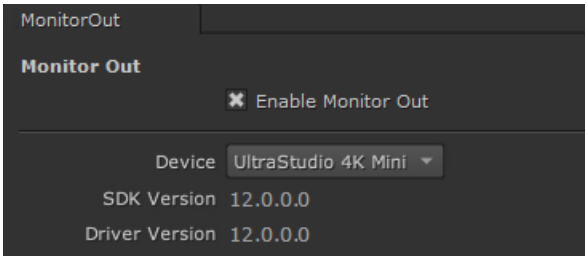




AJA and Blackmagic Design Monitor Out Devices

Our monitor out architecture interfaces directly with the AJA and BlackMagic Design device drivers, which are unified across their respective hardware lines, meaning all current supported cards for the versions detailed in Third-Party Libraries and Fonts should work.

You can select installed cards from the MonitorOut's **Properties** panel using the **Device** control. The SDK and driver versions for the selected card are displayed for convenience.



We've tested the following AJA and Blackmagic Design hardware:

AJA Card:	KONA 4	KONA 5	KONA iO4k+	T-Tap Pro
See https://www.aja.com/family/desktop-io and https://www.aja.com/products/t-tap-pro#techspecs for more details on supported formats.				
Connection				
BNC	●	●	●	●
HDMI	●	●	●	●
Stereoscopic Support				
	Yes	Yes	No	No
Platforms				
	Win, Mac, Linux	Win, Mac, Linux	Win, Mac, Linux	Win, Mac
Drivers				
	Driver 17.5 or later, available here: • **Windows**: https://www.aja.com/support/item-download/10199/AJA-Software-Installer_Windows_v17.6_Release.zip • **macOS**: https://www.aja.com/support/item-download/10183/AJA-Software-Installer_macOS_v17.6_Release.zip • **Linux**: https://www.aja.com/support/item-download/10191/AJA-Software-Installer_Linux-Ubuntu_v17.6_Release.zip			

Blackmagic Card:	UltraStudio 4K Mini	4K Extreme	4K Extreme 12G	8K Pro
See https://www.blackmagicdesign.com/uk/products/ for more details on supported formats per card.				
Connection				
BNC	●	●	●	

Blackmagic Card:	UltraStudio 4K Mini	4K Extreme	4K Extreme 12G	8K Pro
HDMI			●	●
Stereoscopic Support				
	No	No	Yes	No
Platforms				
	Win, Mac, Linux	Win, Mac, Linux	Win, Mac, Linux	Win, Mac, Linux
Drivers				
	Driver 14.4 or later, available here: • **Windows**: https://www.blackmagicdesign.com/support/download/fd605bc94cd4424da637483c41165995/Windows • **macOS**: https://www.blackmagicdesign.com/support/download/fd605bc94cd4424da637483c41165995/Mac%20OS%20X • **Linux**: https://www.blackmagicdesign.com/support/download/fd605bc94cd4424da637483c41165995/Linux			

Blackmagic Card:	DeckLink Studio 4K	DeckLink 4K Extreme	DeckLink 4K Extreme 12G
See https://www.blackmagicdesign.com/uk/products/ for more details on supported formats per card.			
Interface			
BNC	●	●	●
HDMI	●	●	●
Stereoscopic Support			
	Yes (Both views through one output, so the **Full**	Yes	Yes

Blackmagic Card:	DeckLink Studio 4K	DeckLink 4K Extreme	DeckLink 4K Extreme 12G
	Resolution option is not available.)		
Platforms			
	Win, Mac, Linux	Win, Mac, Linux	Win, Mac, Linux
Drivers			
	Driver 14.4 or later, available here: • **Windows**: https://www.blackmagicdesign.com/support/download/fd605bc94cd4424da637483c41165995/Windows • **macOS**: https://www.blackmagicdesign.com/support/download/fd605bc94cd4424da637483c41165995/Mac%20OS%20X • **Linux**: https://www.blackmagicdesign.com/support/download/fd605bc94cd4424da637483c41165995/Linux		

Some monitor out cards allow you to extend or mirror your desktop so that the user interface is visible on the monitor. Please refer to your card's documentation for more information.

Working with HDR Images

Nuke handles HDR (high dynamic range) images in the same way as any other image, and so manipulating HDR footage in Nuke does not compromise the range of data available. Nuke can read in and out HDR with associated metadata and color primaries.

In the Read and Write nodes **MOV Options**, for the **YCbCr Matrix**, you can choose between **Rec 601**, **Rec 709**, **Rec 2020**, or **default** (which will be selected based on the metadata). **Rec 2020** can be used for HDR images, for more accurate color conversion.

Quicktime Color Metadata Control

If **default** is selected, movReader reads the nclc values, including the color matrix, from the container metadata (the colr atom), and uses this information to suggest a default value for the **YCbCr Matrix** knob.

When using the Write node, under **Advanced**, there are **Metadata Export** knobs for **Color Primaries** and **Transfer Function** , allowing for additional options.

- **Color Primaries** specifies the color primaries metadata to be written to the MOV file. This defines the XYZ coordinates of the RGB primaries and white point. Auto - Uses legacy colorspace assignment based on content resolution.
- **Transfer Function** specifies the transfer function (gamma) metadata to be written to the MOV file. This describes how RGB values are converted to and from linear light. Auto - Uses legacy colorspace assignment based on content resolution.

Nuke, Nuke Studio, and Hiero support viewing HDR output through approved monitors and Monitor Output devices such as AJA, Blackmagic Design, and Bluefish SDI/HDMI cards. See AJA and Blackmagic Design Monitor Out Devices for more details on supported cards. The following table contains details of the output devices we've tested and approved.

Device	Operating System	Connection
Dell UP3221Q	Linux and Windows	HDMI
	macOS (Catalina 10.15.x, or later)	Thunderbolt
LG EP950 and BP95E	Linux and Windows	HDMI
	macOS (Catalina 10.15.x, or later)	USB-C DisplayPort Alt Mode
Apple XDR	macOS (Catalina 10.15.x, or later)	Thunderbolt
EIZO CG3146, CG319X, CG279X, CG2700S, CG2700X, CS2740	Linux and Windows	HDMI, Display Port
	macOS (Catalina 10.15.x, or later)	USB-C Alt Mode

EDR Displays - Mac Only (Beta)

If you're running Nuke on macOS Catalina or later and have a compatible screen capable of displaying values above 1, you can view your projects with HDR luminance ranges using **sRGBf** in OCIO Color Management.

You can also display images in the P3 gamut, giving more accurate color on wide gamut displays.



Note: This feature requires a 2019 Mac Pro running macOS 10.15, or later, and a suitable display or a current generation iMac Pro, iMac or MacBook Pro with an HDR screen.

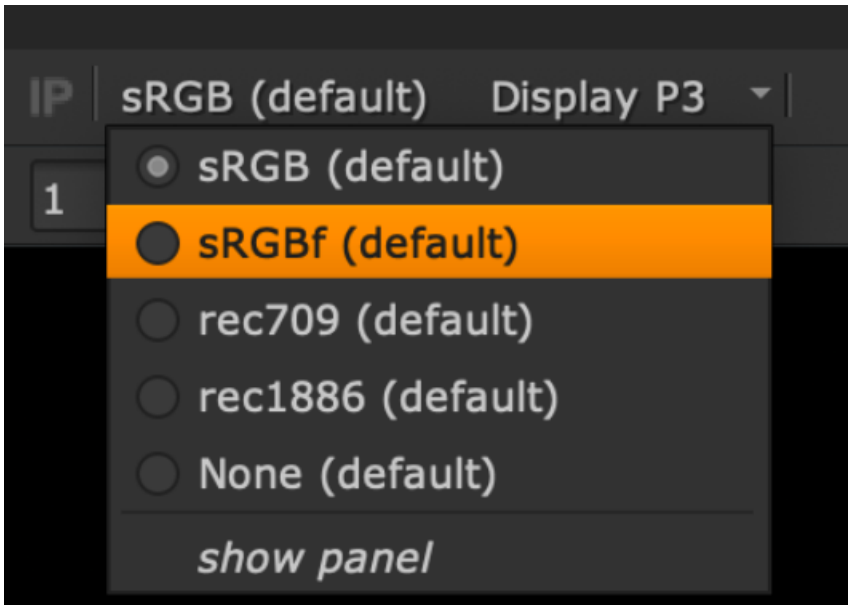
1. On an Apple XDR, set the profile of your display to one of the HDR options in the OS System Display Preferences:
   - Pro Display XDR (P3-1600 nits)
   - Apple Display (P3-500 nits)
   - HDR Video (P3-ST2084) - 1000 nits

   On other monitors, enable the **High Dynamic Range** checkbox.
2. In the **Preferences**, open the **Color Management** > **HDR** section and check **Enable macOS HDR Color Profile (Display P3) (Beta)**.

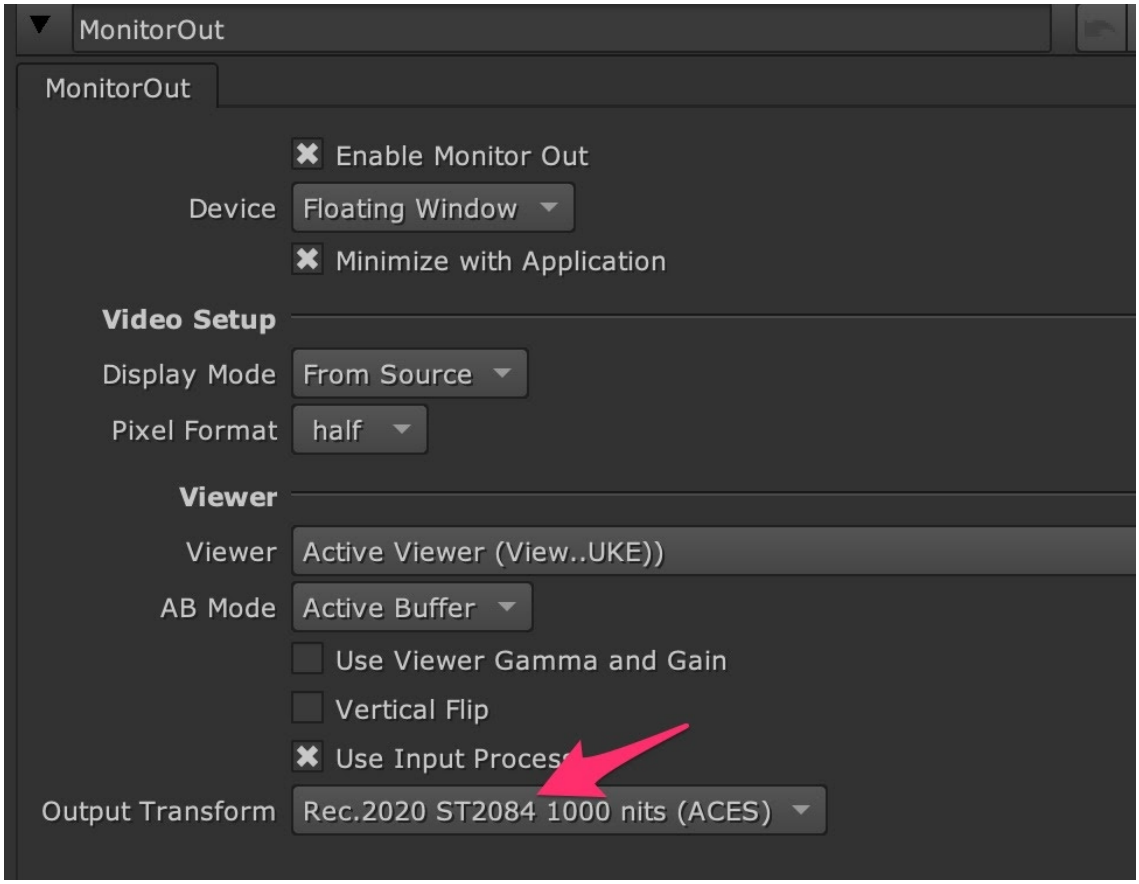
   

   Note: Enabling this control requires you to restart the application.

   This control sets the Viewer colorspace to **Display P3**, so that the screen can display values above 1. This setting also applies the P3 colorspace to the following;
   - All Viewers
   - Node Graph
   - Dope Sheet
   - Curve editors
   - Scopes
3. In the Viewer **Properties**, set the **gl buffer depth** to **half-float** or **float** to allow HDR values to display.
4. Set the **Viewer Process** to **sRGBf** using OCIO Color Management or provide your own extended color space. This ensures that the frame buffer is not clamped at 1.



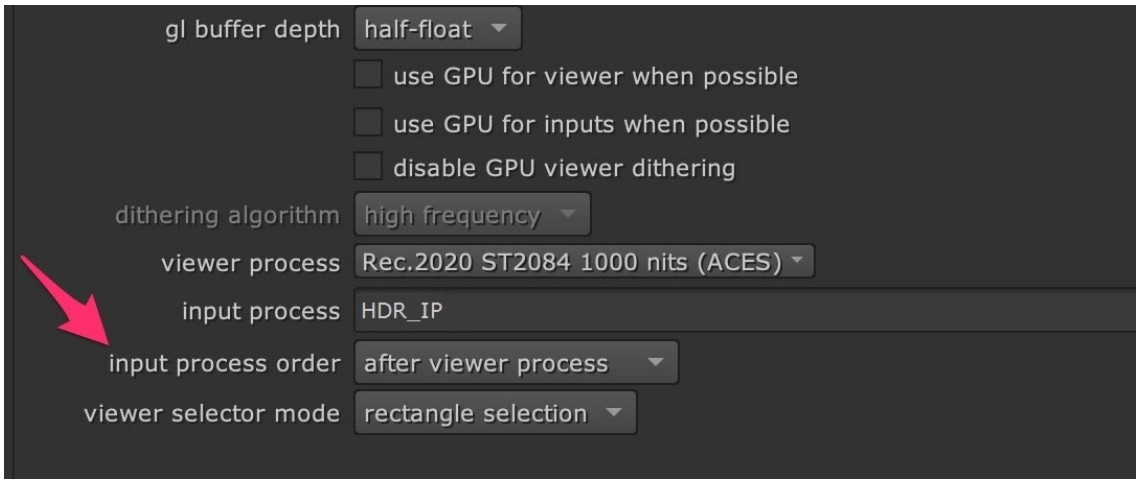
5. To use the **HDR Video (P3-ST2084)** setting on the monitor, select an appropriate ST2084 **Output Transform** in the MonitorOut **Properties** or in the Viewer.



A further color space conversion is required to allow HDR images to display correctly. An example of how this can be achieved in Nuke can be seen using a gizmo created by a customer, Nick Shaw - the gizmo is available for download here: PQ-EDR_v101.gizmo

For more information on gizmos and how to use them, see Accessing Gizmos in Nuke.

Using this gizmo as an **Input Process** and setting the **input process order** to **after viewer process** renders the correct image in the Viewer.



The gizmo also contains a slider called **Normalization (nits)**. This allows you to set the PQ nit value which is normalized to 1.0. This is useful, for example, to prevent clipping of a 1000 nit PQ signal on the 16" MacBook Pro's 500 nit display.

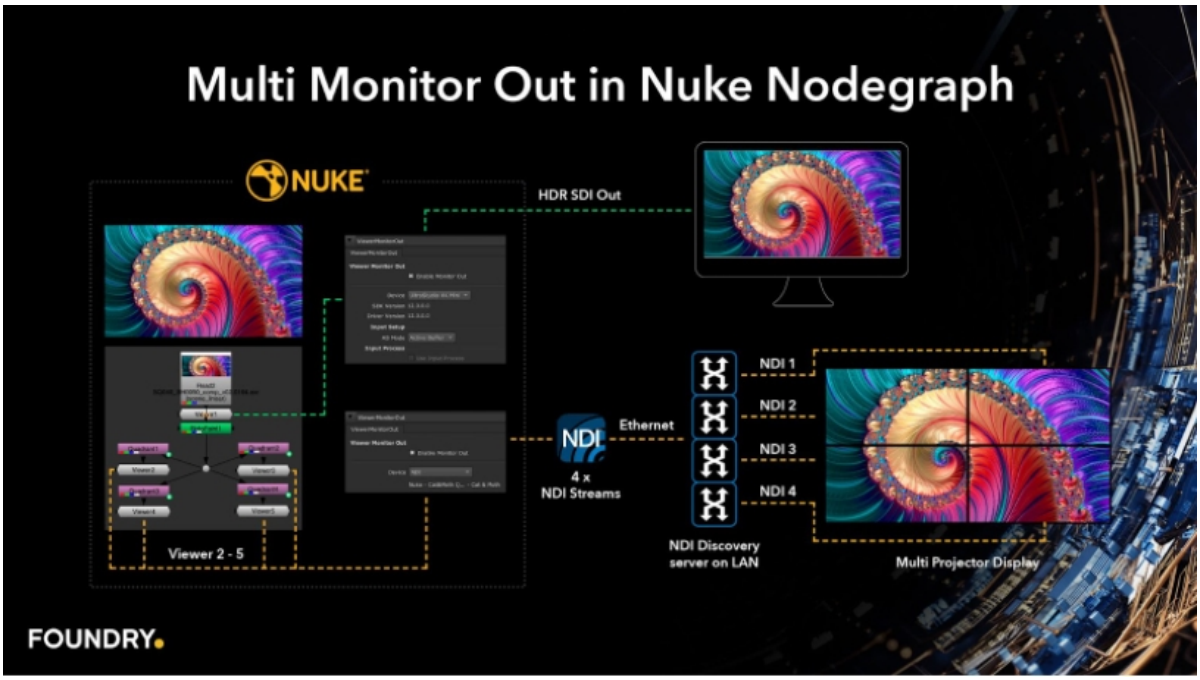
Streaming Viewer Output Over the Internet with NDI®

Network Device Interface, or NDI, is a network protocol from NewTek that enables you to send video, audio, and metadata signals over standard IP networks in real-time. NDI allows you to easily share a Nuke Viewer's output with anyone else on the same network.

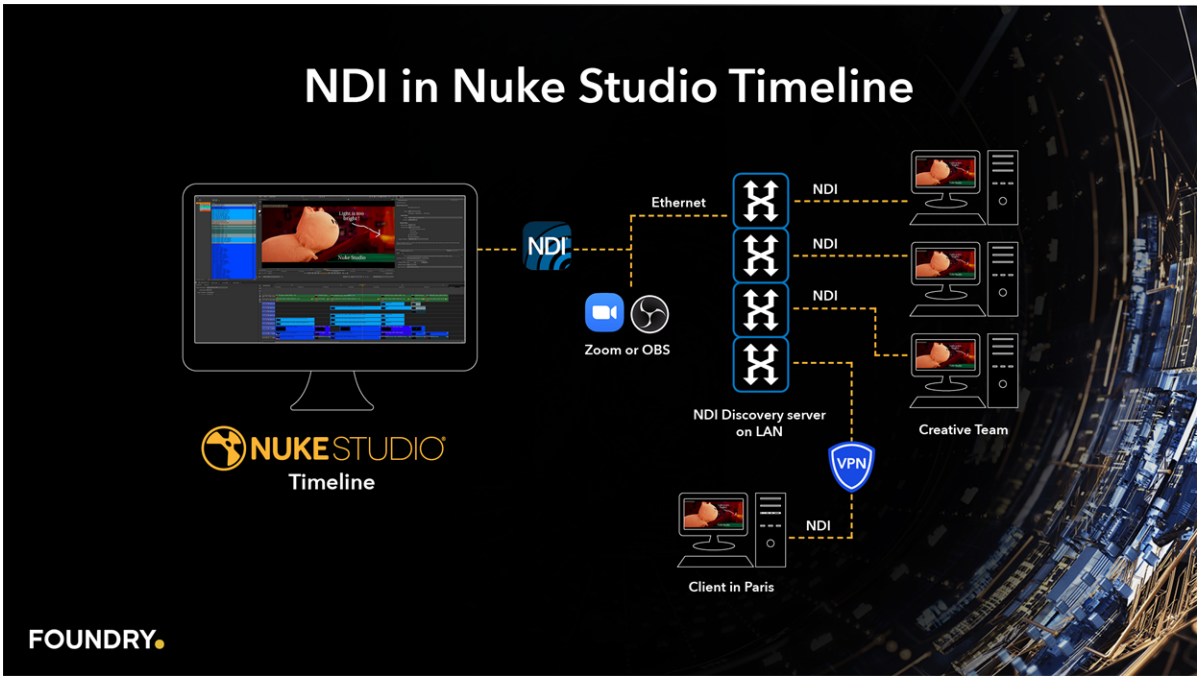
Nuke's monitor out functionality is controlled by Viewer nodes, so you can compare or combine multiple streams from as many Viewers as you like. Each Viewer can be enabled separately to output your shots to multiple output devices, whether they're dedicated monitor out devices, NDI streams, or Nuke's native Floating Window. The example shows five Nuke Viewers in total. The first Viewer at the top of the image is

feeding an HDR SDI monitor using a dedicated AJA or Blackmagic Design card and the other four are combined over NDI connections to a remote multi-projector display.

See Comparing and Combining Multi-Viewer Output with NDI for more details.



Nuke Studio's monitor out only allows you to stream one timeline Viewer to a dedicated monitor out device, NDI stream, or Nuke's native Floating Window. Sharing your Viewer over an NDI connection is a great way to collaborate in a review session with your creative team on the same local network or over VPN to remote customers.



Installing the NDI Tools

To view an NDI stream, you need the NDI Studio Monitor (Windows) or NDI Video Monitor (macOS) application installed on your machine. The monitor searches your network for NDI signals from Nuke or Nuke Studio and allows you to select the stream you'd like to view.

NDI Studio Monitor and NDI Video Monitor are available for download as part of the NDI Tools on Windows and macOS here:

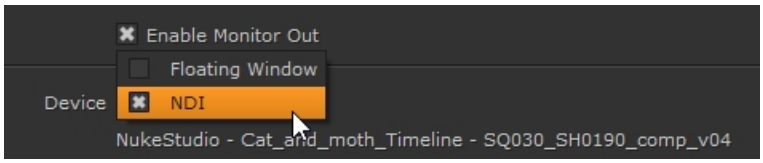
- NDI Tools

Note: NDI streams require you to bypass the operating system's firewall software on the remote machine to work correctly. Installing the NDI Tools on the remote machine should set exceptions in the firewall automatically, but you can add exceptions manually as described under Setting Up an NDI Stream from Nuke and Nuke Studio and Adding Firewall Exceptions on macOS.

Setting Up an NDI Stream from Nuke and Nuke Studio

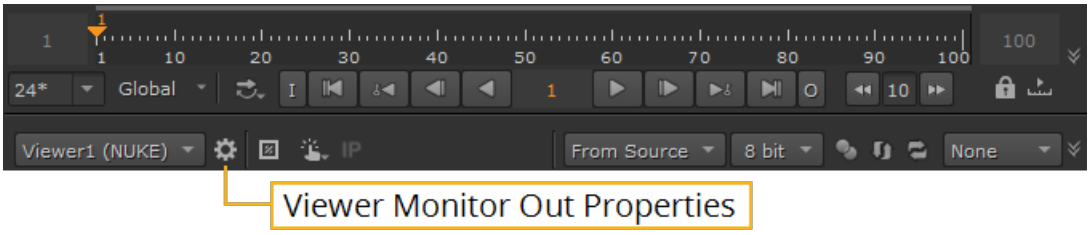
1. On the local machine in Nuke or Nuke Studio, go to **Edit** > **Preferences**, or press **Shift**+**S**, and navigate to **Panels** > **Viewer Monitor Out**.
2. Ensure that **Enable NDI** is checked.

3. In the monitor out **Properties**, set the monitor out **Device** to **NDI** to start broadcasting the stream over the network.

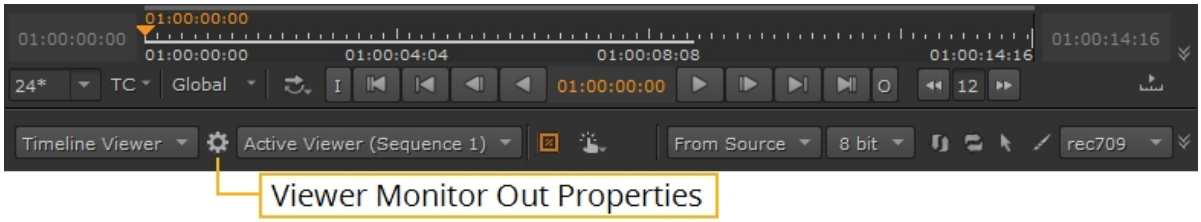


To display the **Properties** panel:

- **Nuke** - click the cog icon in the bottom-left of the **Viewer Monitor Out** panel or double-click the Viewer node.



- **Nuke Studio** - click the cog icon in the bottom-left of the **Viewer Monitor Out** panel.



4. Set the **Display Mode** for the receiver device of the NDI stream. If you don't change this, Nuke broadcasts the **source resolution** of the project over NDI. However, this format may not be supported on the receiving device, so you can set the NDI stream to scale to a specific resolution and frame rate here.

> **Note:** If the **Display Mode** is set from a timeline sequence, the NDI resolution is auto-scaled. Whereas the node graph viewer outputs an absolute pixel-for-pixel view, center cropping or padding, based on the resolution of the NDI signal. For example, a 3K (2880x1620) resolution comp set to output from the node graph viewer to UHD (3840x2160) would appear padded on a UHD device because the image is smaller than the frame size being broadcast, whereas the same 3K (2880x1620) comp set to output to HD (1920x1080) would appear cropped on the receiver side.

5. On the remote machine, download and install the NDI Tools for your operating system from the links under Installing the NDI Tools.

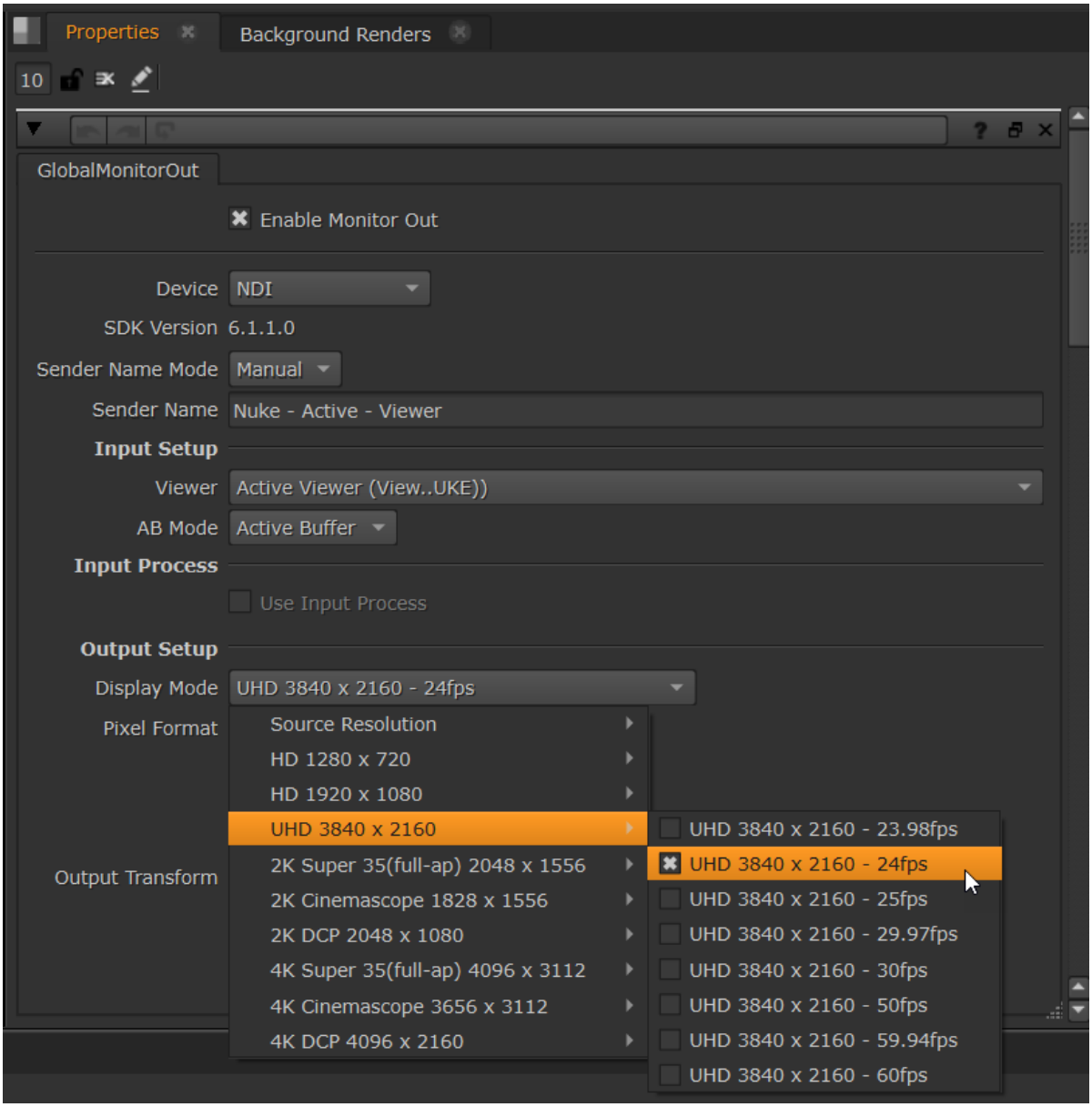
6. Launch the NDI Studio Monitor (Windows) or NDI Video Monitor (macOS), depending on your operating system:

   - **Windows** - In the NDI Studio Monitor, click the options  icon at the top-left and then select the required stream.
   - **macOS** - In the NDI Video Monitor, click the **NDI** menu at the top of the screen and then select the required stream.

   The stream name is shown in the Viewer **Properties** and on the NDI stream for identification and is constructed as follows: [Application] - [Project or Script] - [Viewer]

   For example: **Nuke - cat_and_moth - Viewer2**

   The Viewer is output through the NDI Studio Monitor or NDI Video Monitor.



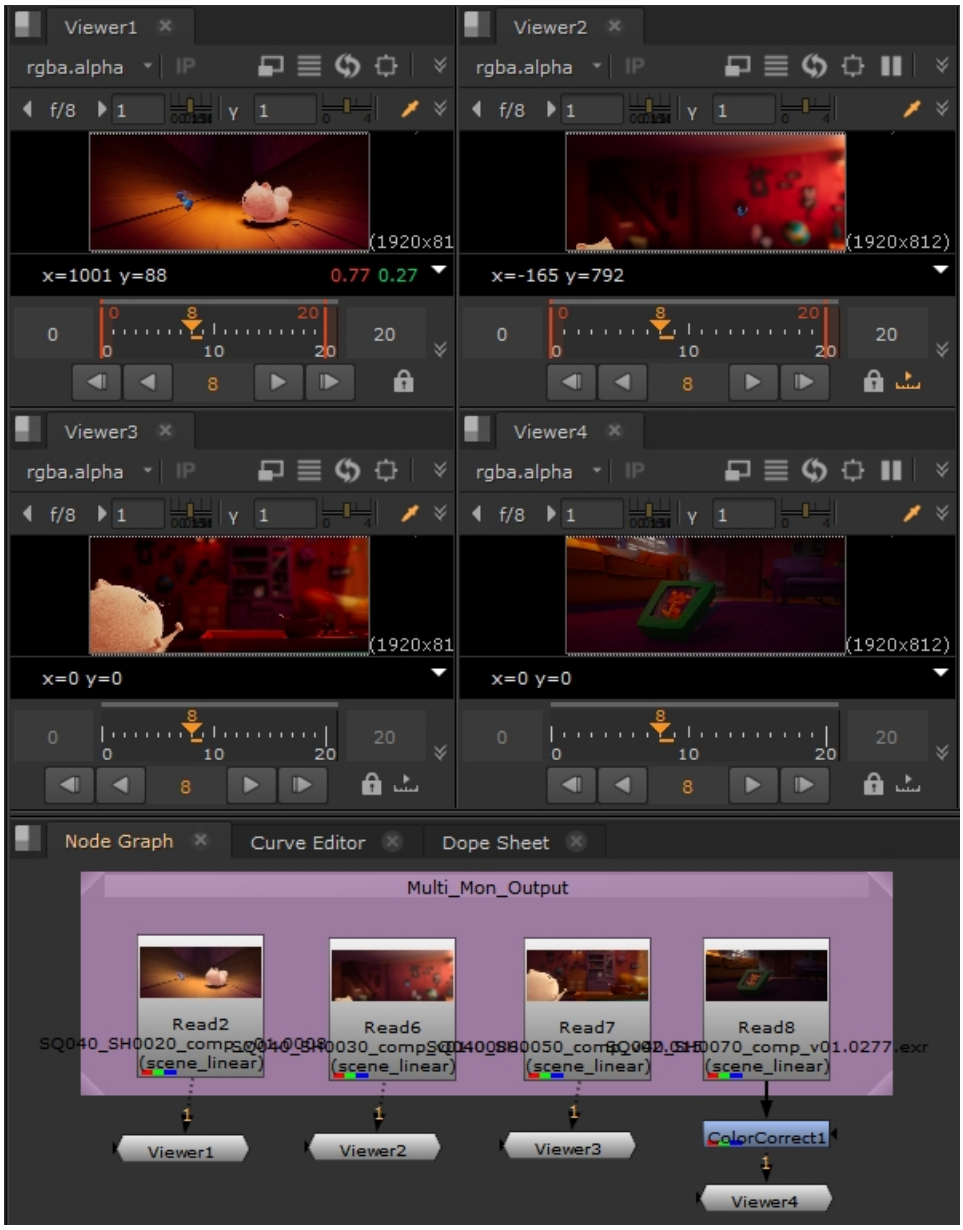
NDI output showing a custom named stream set in **Sender Name** and the **Display Mode** resolution set to UHD 3840 x 2160 24fps.

Note: When NDI is enabled in Monitor Out, audio will only play through the NDI stream.

Comparing and Combining Multi-Viewer Output with NDI

Nuke's monitor out functionality is controlled by Viewer nodes, so you can compare or combine multiple streams from as many Viewers as you like. Each Viewer can be enabled separately to output your shots to

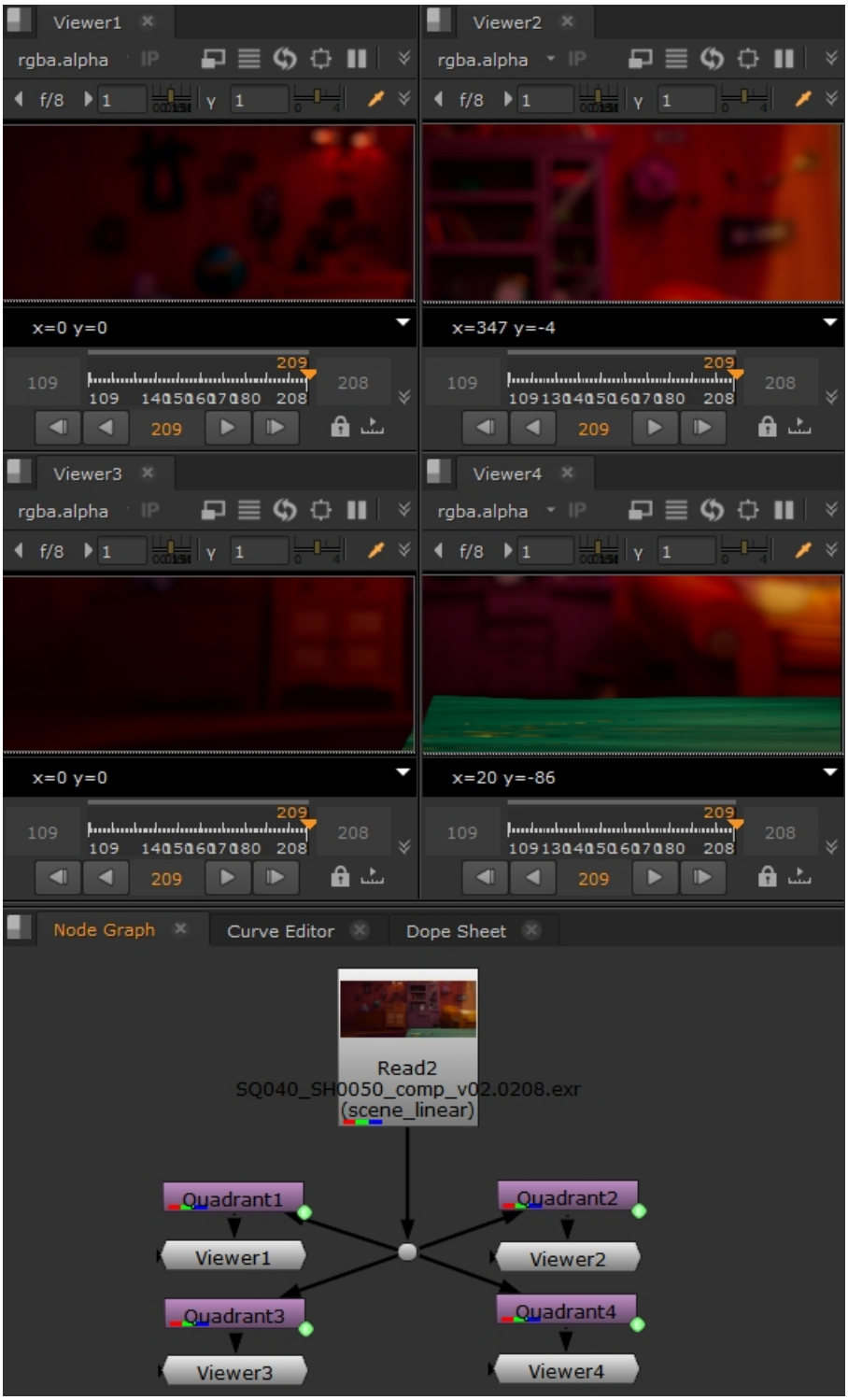
multiple NDI streams. For example, using four Viewers and four NDI streams, you could compare the format of multiple shots.



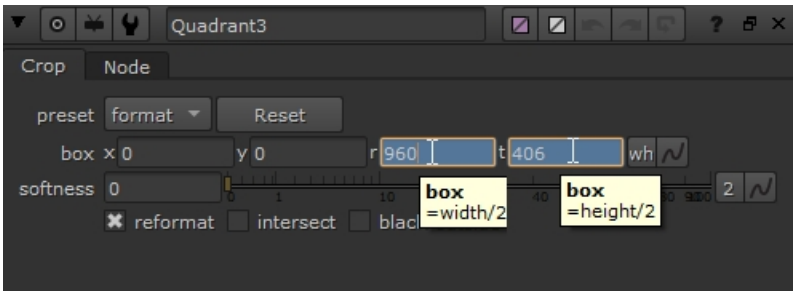
You can switch the **Output Transform** on each Viewer independently to check color consistency between the shots.

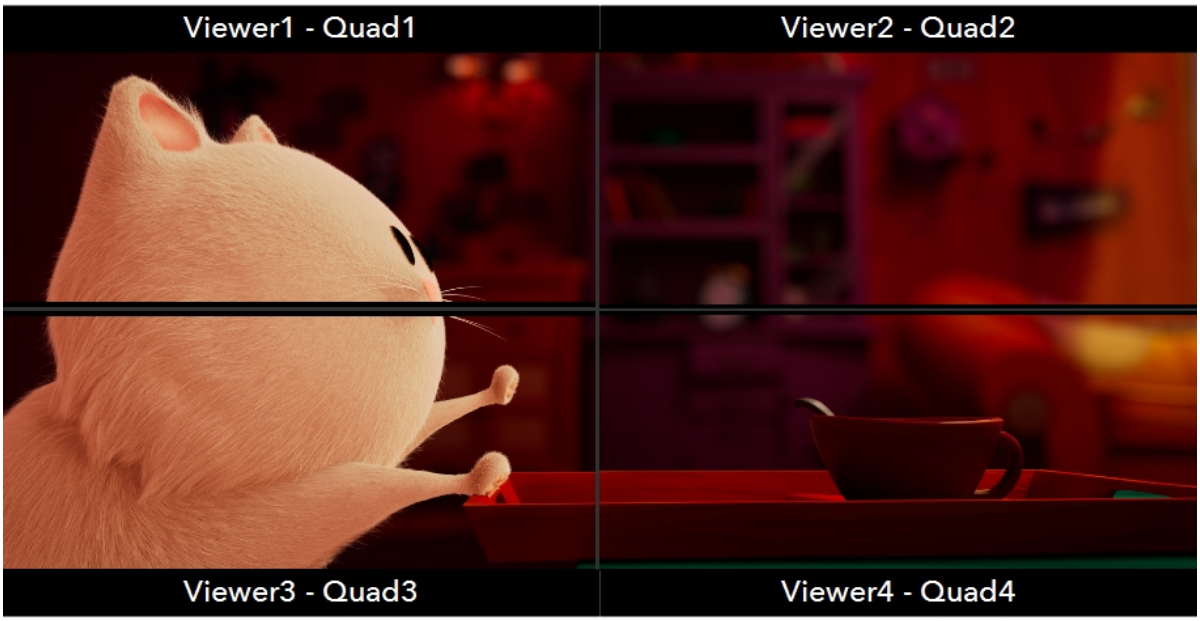


Using a similar four-Viewer setup, you could send four quadrants over NDI to four projectors to produce a larger image somewhere else on the same network.



Each quadrant is a Crop node that defines which part of the overall image is handled by each Viewer. For example, Quadrant3 is set up as shown here, with simple expressions to handle different formats.

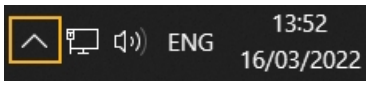




Adding Firewall Exceptions on Windows

Installing the NDI Tools on the remote machine should set exceptions in the firewall automatically, but you can add exceptions manually as follows:

1. Click the show hidden icons button on the right-hand side of the taskbar and then click **Windows Security**,

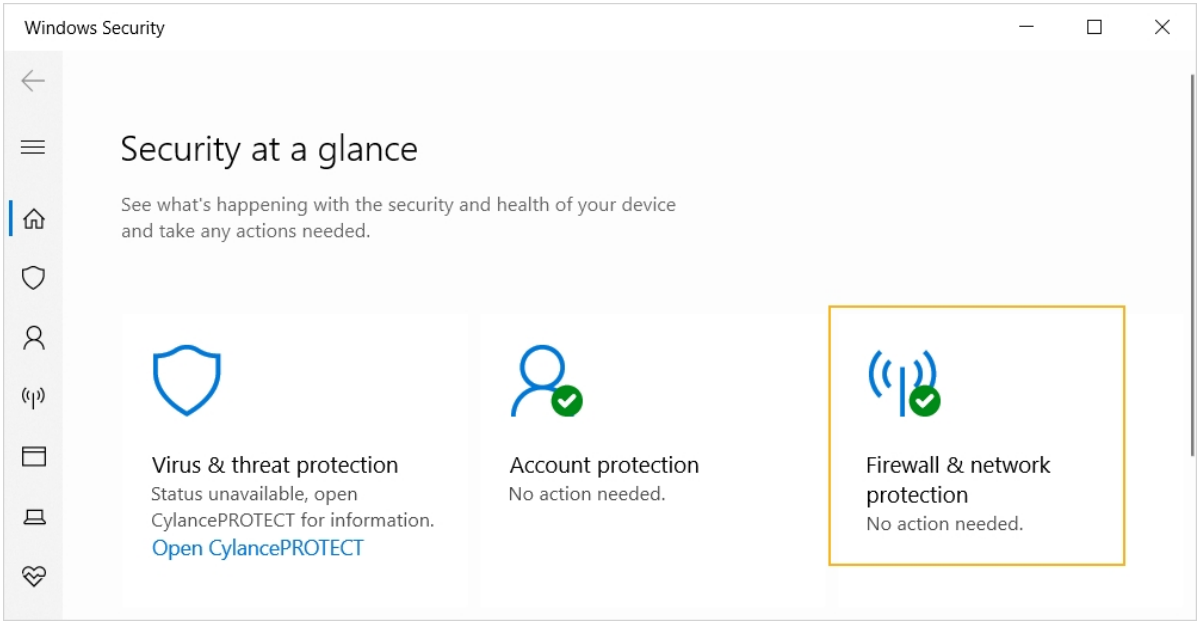
   

   OR

   Click the Windows Start button and type **Windows Security** and then click **Windows Security**.

   The **Windows Security** dialog is displayed.

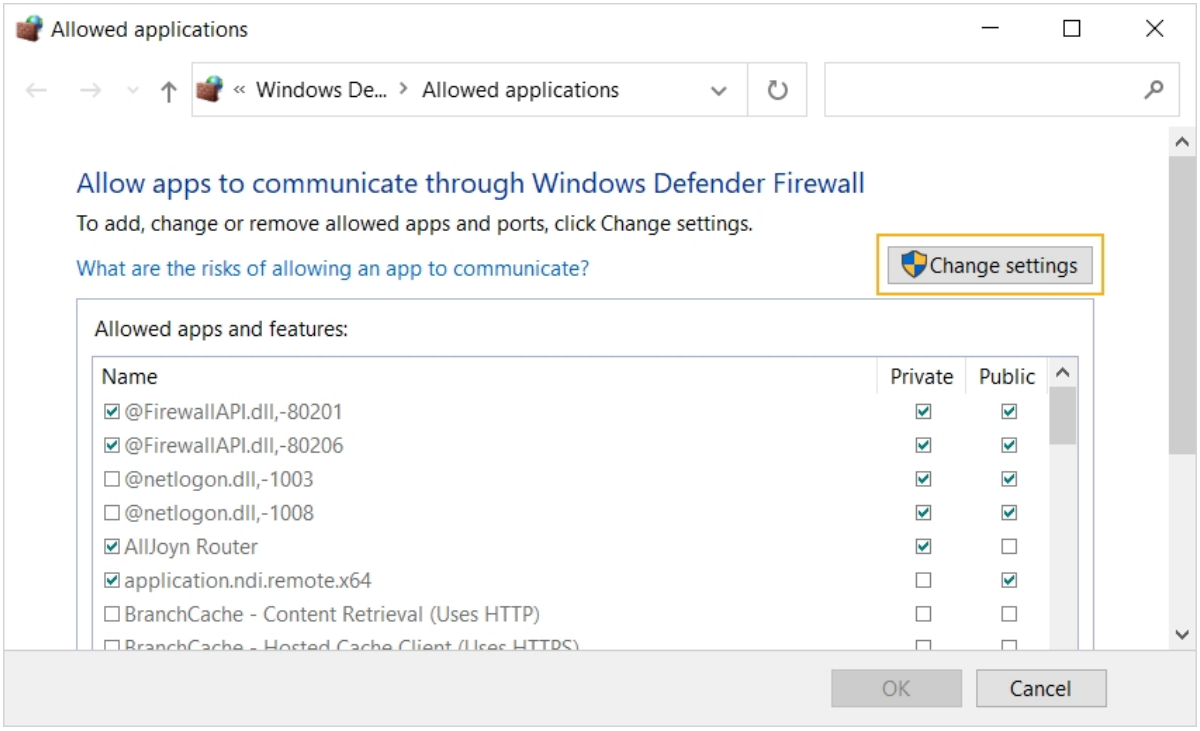
2. Click **Firewall & network protection**.

3. 

4. Click **Allow an app through the firewall**.

   The **Allowed applications** dialog is displayed.

5. Click **Change settings** to enable editing.

6. 

7. Scroll to the bottom of the window and click **Allow another app**.

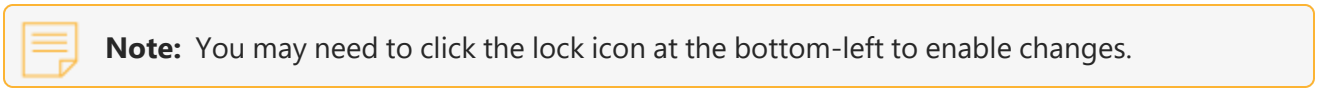
8. Enter the file path to the NDI Studio Monitor executable or click **Browse** to locate the **.exe** using the explorer. NDI Studio Monitor is installed here by default:
   ```
   C:\Program Files\NDI\NDI 5 Tools\Studio Monitor\Application.Network.StudioMonitor.x64.exe
   ```
9. Click **Add** and then click **OK** to complete the firewall exception.

Refer to your operating system's documentation or speak to your network administrator for more details. For more information on the NDI Tools and network configuration best practice, see: https://ndi.video/tech/

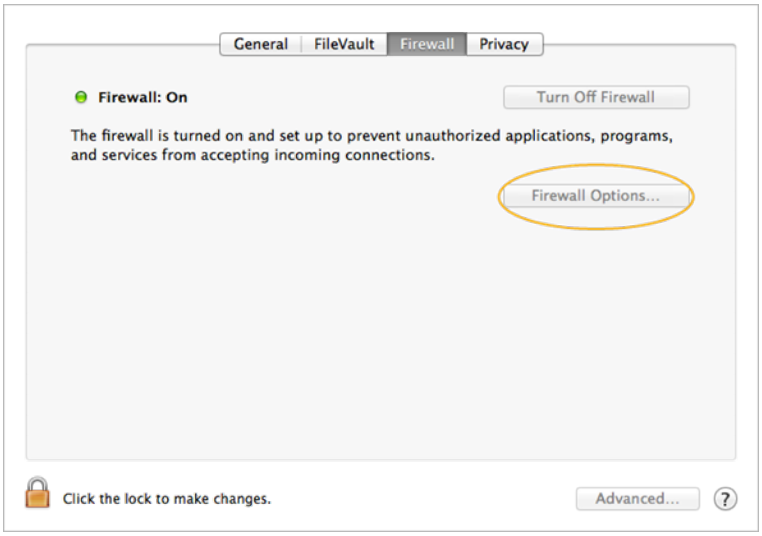
Adding Firewall Exceptions on macOS

Installing the NDI Tools on the remote machine should set exceptions in the firewall automatically, but you can add exceptions manually as follows:

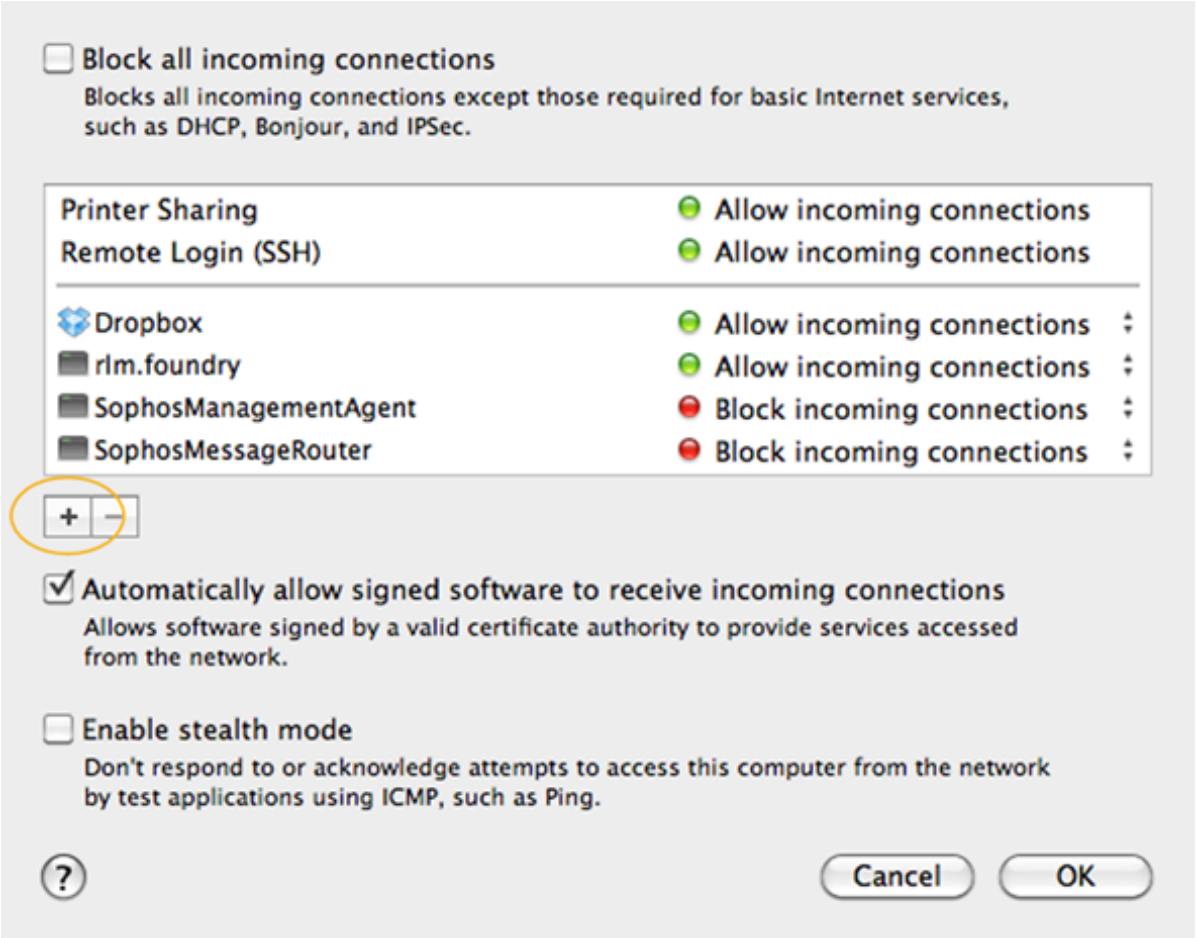
1. Launch **System Preferences**.
2. Click **Security & Privacy** and go to the **Firewall** tab.
3. Click the **Firewall Options** button.



Note: You may need to click the lock icon at the bottom-left to enable changes.



4. Under the list of incoming connections, click on the **+** button.



5. Enter the file path to the NDI Video Monitor application or click **Browse** to locate the **.app** using the Finder. NDI Video Monitor is installed here by default:
   `/Applications/NDI Video Monitor.app`

6. Select **NDI Video Monitor.app** and click **Add** and then **OK** to complete the firewall exception.

Refer to your operating system's documentation or speak to your network administrator for more details. For more information on the NDI Tools and network configuration best practice, see:
https://ndi.video/tech/

Flipbooking Sequences

Flipbooking a sequence refers to rendering a range of images (typically at proxy resolution), then playing them back in order to accurately access the motion characteristics of added effects.

You have a few options for flipbooking within Nuke:

- You can enable automatic disk caching of rendered frames, then play these frames back using Nuke's native Viewer. This option does not let you define a specific playback rate.
- You can render out temporary image sequences using the default flipbooking tool, a RAM-buffering playback utility, which is displayed in its own Viewer and plays back sequences at the defined frame rate. If you have a license for HieroPlayer or Nuke Studio, you can flipbook using HieroPlayer rather than the default flipbook that ships with Nuke. See Anaglyph - displays the views simultaneously using a red hue for left and green hue for right, and adds selection controls above the Viewer. for more information.
- You can also set up an external flipbooking application in Nuke using Python. For more information, see the Nuke Python documentation (**Help** > **Documentation**).

The Nuke Viewer automatically saves to disk a version of every frame it displays. When you play through sequences in the Viewer, it reads, where possible, from this cache of pre-rendered images, making real-time play back possible (depending, of course, on image resolution and your hardware configuration). You can define the location and size of the Viewer cache in the **Preferences**.

Depending on what **viewer buffer bit depth** has been set to in the Viewer settings, the cache can contain 8-bit (**byte**), 16-bit (**half-float**), or 32-bit (**float**) image data. This offers a trade-off between speed and quality. Half-float and float modes provide higher precision than byte, but are also slower to process.

Setting the Viewer Cache Location and Size

1. Click **Edit** > **Preferences** to display the **Preferences** dialog.
2. In the **Performance** > **Caching** section, there is a **Disk Caching** – **temp directory** field. Use this to enter the path name of the directory in which you want to store the flipbook images (for example, c:/temp).
3. Using the **comp disk cache size** control, specify the number of gigabytes you want to allow the image cache to consume.
4. Click **OK** in the bottom-right corner of the **Preferences** dialog to update preferences and then restart Nuke.

The Viewer now caches each frame it displays in the directory specified. When you click the playback buttons on the Viewer, or drag on the scrub bar, Nuke reads in images from this cache.

Note that the cached images have unique names reflecting their point of output location in the script. This means that you can cache images from multiple nodes in the script without overwriting previously cached images. For more information on caching, see Image Caching.

Flipbooking a Sequence

To flipbook an image sequence, do the following:

1. Select the node that you want to flipbook the output of.

Note: If you select a Write node in the step above, you must first click its **Render** button in order to manually render its output to the destination defined in the **file** field. This step is necessary only in the case of Write nodes.

2. Select **Render** > **Flipbook selected** (or press **Alt**+**F**).
   Alternatively, you can click the **Flipbook this Viewer** button  at the bottom-right of the Viewer. This flipbooks the nodes that are connected to the Viewer.
   A **Flipbook** dialog opens.
3. Check that your settings are correct in the dialog. The default values are copied from the Viewer you currently have active. You can change them if necessary:
   - **Flipbook** - set the flipbooking application you want to use.
   - **Use settings from -** set which Viewer should be used to draw default values from.
   - **Enable ROI -** Check to define your region of interest.
   - **Channels** - select which layer to display in the flipbook result.
   - **Frame range -** set the frame range you want to flipbook.
   - **Delete existing temporary files -** Check to delete any existing temporary files with the same file name before flipbooking.
   - **LUT** - select the LUT appropriate for viewing. By default, the flipbook renders your files with a linear colorspace and attempt to pass a LUT file to the flipbook.
   - **Burn in the LUT** - If you check this box the flipbook files are rendered with the LUT applied. If you uncheck it, the flipbook is displayed using it's equivalent LUT (based on the LUT's name). If you have an equivalent LUT available in the flipbook program, then it's better not to check the **Burn in the LUT** box. This way, when you measure pixel values in the flipbook application they match what you get in the Nuke Viewer.
   - **Audio** - if you want to flipbook an audio file with your clip, select the AudioRead node you need in this dropdown. For more information on audio files in Nuke, see Audio in Nuke.
   - **Buffer** - set which buffer you want include.
   - **Views** - check the view(s) to output, if you're working in a multi-view comp.

Note: The **Views** control is only available in multi-view comps. See Stereoscopic Scripts for more information.

   - **Use proxy -** check to use proxy mode.
   - **Render using frame server** - check to render in the background using Nuke's frame server.
   - **Continue on error** - check to keep rendering even if an error occurs during the process.
4. Click **OK**.

Nuke renders as a temporary sequence the output of the selected node using the frame range and resolution defined in the script's settings. This may take a few moments.

After the render is complete, Nuke launches Flipbook Viewer and loads in the temporary sequence. You can play it back and view it using Flipbook Viewer controls.

If you flipbooked a stereo comp, you can right-click in the Flipbook and choose **Stereo Modes** to view your flipbook in various configurations:

- **Side by Side** - displays the views side by side at the correct aspect ratio, and adds selection controls above the Viewer.
- **Squeezed Side by Side** - displays the views side by side and squeezed to fit the format horizontally, and adds selection controls above the Viewer.
- **Squeezed Above by Below** - displays the views above and below each other and squeezed to fit the format vertically, and adds selection controls above the Viewer.
- **Interlace H** - displays the views interlaced horizontally, and adds selection controls above the Viewer.
- **Interlace V** - displays the views interlaced vertically, and adds selection controls above the Viewer.
- **Checkerboard** - displays the views using an alternating checkerboard pattern (one pixel from left and one pixel from right), and adds selection controls above the Viewer.
- **Anaglyph** - displays the views simultaneously using a red hue for left and green hue for right, and adds selection controls above the Viewer.
- **Flicker** - displays both views alternately, and adds selection controls above the Viewer.
- **OpenGL Stereo** - displays both views at once on a 3D monitor for review purposes. See Enabling OpenGL Stereo Output for more information.

Using HieroPlayer as Nuke's Flipbook

If you have a HieroPlayer license (get one free if you have Nuke or NukeX), you can flipbook using HieroPlayer rather than the default flipbook that ships with Nuke:

1. Select the node that you want to output to the HieroPlayer flipbook.



Note: If you select a Write node, you must first click its **Render** button in order to manually render its output to the destination defined in the **file** field. This step is necessary only in the case of Write nodes.

2. Select **Render** > **Flipbook selected** (or press **Alt**+**F**).

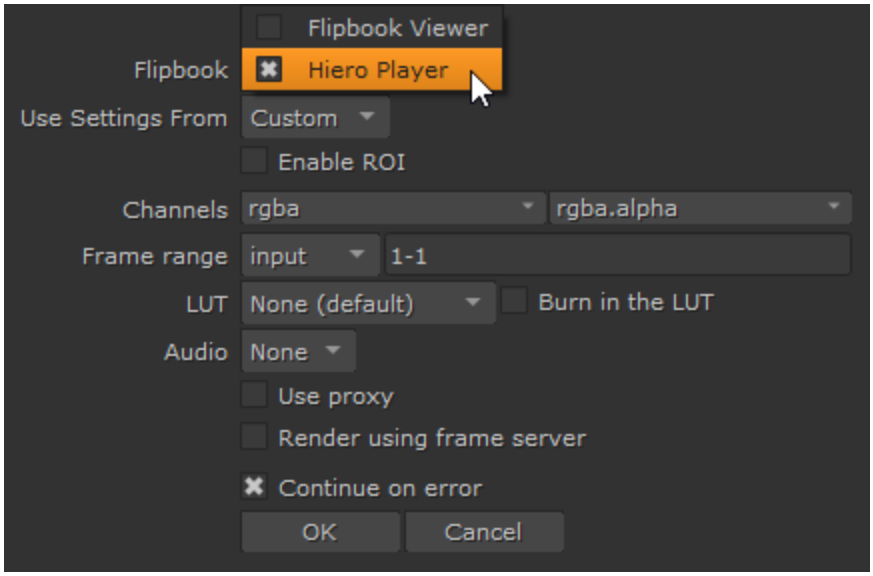
   Alternatively, you can click the **Flipbook this viewer** button  at the bottom-right of the Viewer. This flipbooks the nodes that are connected to the Viewer.

Note: If the Viewer node is not connected to any nodes, a warning displays: **Can't launch flipbook, there is nothing connected to the viewed input.**

A **Flipbook** dialog opens.

3. Select **Flipbook** > **HieroPlayer** and adjust the other controls as described in Flipbooking a Sequence.

Note: If you enable **Burn in the LUT**, the flipbook files are rendered with the Viewer LUT applied. If you uncheck it, the flipbook is displayed using its equivalent LUT in HieroPlayer (based on the LUT's name).



4. Click **OK**.

Nuke renders the output and then opens HieroPlayer with a clip in the **Project** bin and displays it in the Viewer. The clip is named automatically as **[project]_[node]_[date]_[time]** by default. For example: `myProject_Blur1_2020-2-21_12-53-58`

Note: Image clips in HieroPlayer flipbooks are always rendered as **.exr** files.

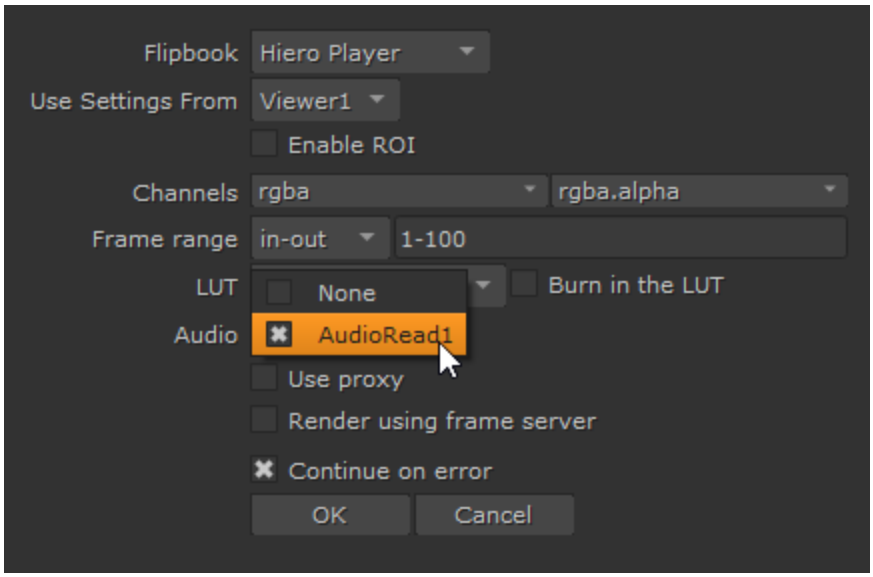
If you change the **OCIO config** in the **Project Settings** and flipbook again, HieroPlayer creates a new project to contain flipbooks using that configuration. For example, flipbooking using **nuke-default** and then switching to **aces_1.1** creates two distinct projects in HieroPlayer.

Subsequent flipbooks using either configuration are then automatically grouped in the correct HieroPlayer project.

Flipbooking with Audio

If you have an AudioRead node in your script, you can flipbook the audio file along with the image clip in HieroPlayer. Adding audio to a flipbook creates a bin containing an **.exr**, a **.wav**, and a sequence. For more information on audio files in Nuke, see Audio in Nuke.

To flipbook with audio, follow the steps in Anaglyph - displays the views simultaneously using a red hue for left and green hue for right, and adds selection controls above the Viewer. and then select **Audio** > **AudioRead1** and click **OK**.



Note: The AudioRead node does not need to be connected to the main node tree when you flipbook to HieroPlayer.

Nuke renders the output and then opens HieroPlayer with a bin in the **Project** containing the imported audio file, an **.exr** clip, and a sequence. The bin items are named automatically as **[project]_[node]_[date]_[time]** by default. For example:

```
myProject_AudioRead1_2020-2-21_12-53-58
myProject_Blur1_2020-2-21_12-53-58
```

Note: Image clips in HieroPlayer flipbooks are always rendered as **.exr** files.

If you change the **OCIO config** in the **Project Settings** and flipbook again, HieroPlayer creates a new project to contain flipbooks using that configuration. For example, flipbooking using **nuke-default** and then switching to **aces_1.1** creates two distinct projects in HieroPlayer.

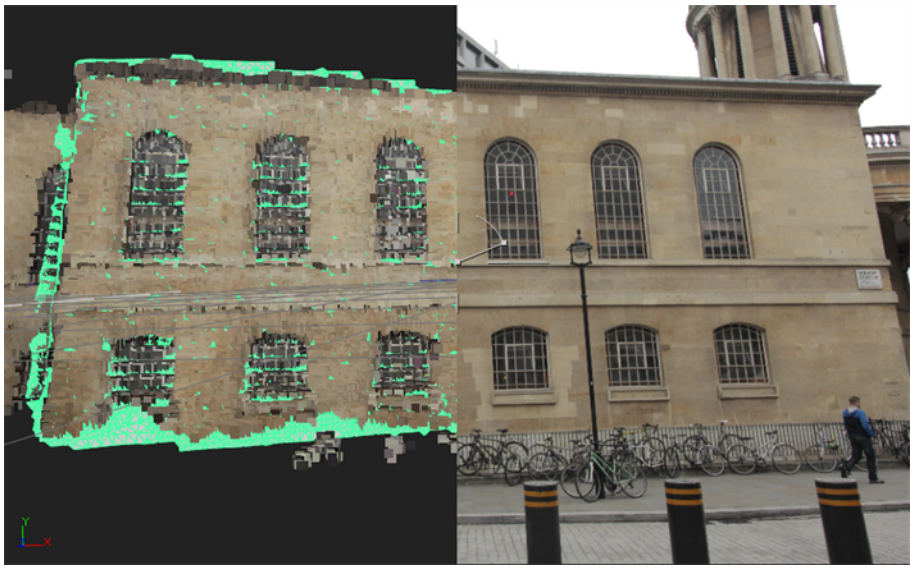
Subsequent flipbooks using either configuration are then automatically grouped in the correct HieroPlayer project.

Capturing the Viewer

You can capture the contents of the Viewer for a quick real-time flipbook, also known as a playblast, and save the content out to **.jpg** for review.

Tip: Capturing the contents of the Viewer is very useful for quickly checking the animation in your 3D scene in real-time without having to do a full scanline render.

Playblast flipbooks the Viewer 'as is', including 2D and 3D scenes, handle and transform overlays (such as roto shape outlines), wipes, and so on.

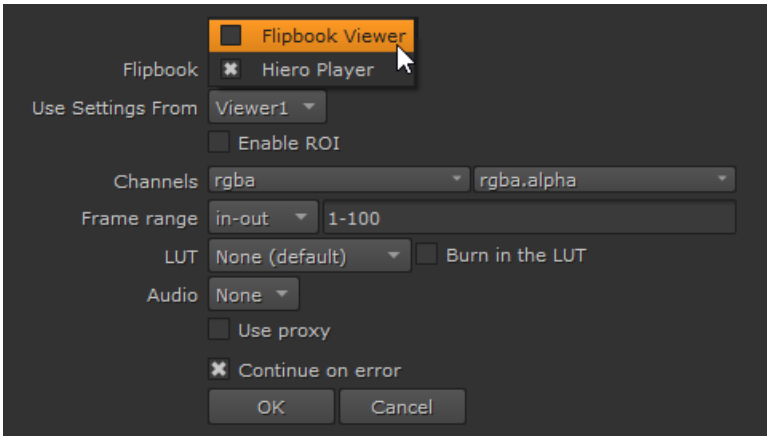


The contents of a Viewer containing 3D and 2D information, including the wipe handle, captured in **.jpg** format.

To capture the Viewer contents:

1. Click the **Capture** button under the Viewer, to the right of the flipbook button.

   The **Capture** dialog displays.



2. Select the required **Flipbook** tool using the dropdown.
3. Check that your settings are correct in the dialog. The default values are copied from the Viewer you currently have active. You can change them if necessary:
   - **Frame range** - set which Viewer you want to draw default values from, and set the frame range that you want to flipbook.
   - **Buffer** - select the buffer you want to capture.
   - **Use proxy -** check to use proxy mode.
   - **Continue on error** - check to keep rendering even if an error occurs during the process.
   - **Customise write path** - Check this to manually set the location (in the **Write path** field below) where the **.jpg** files are stored.



Note: Nuke only captures sequences using the **.jpg** format. Don't forget to include printf or hash frame padding, such as **%4d** or **####**.

4. Click **OK** to capture the contents of the Viewer.

   The frame range is cached as **.jpg**s to either the default location set in the **Preferences** dialog, or – if enabled – the directory specified in the **Write path** control.

Rendering Output

Nuke can render images locally on your workstation (see Output (Write) Nodes) or it can be setup to render images on a network render farm (see Using the Frame Server on Render Machines). Before rendering, make sure that you are using the appropriate file name syntax (see File Name Conventions for Rendered Images) and verify that your project settings have the correct output format and proxy format selected (see Render Resolution and Format).

By default, if you choose to render frame 15, the resulting file is numbered accordingly, for example image.0015.rgb. However, you can change this behavior via expressions, specified start frames, and constant offsets (see Changing the Numbering of Rendered Frames).

Sometimes, you may want to render an image just to read the rendered image back in (see Using a Write Node to Read in the Rendered Image). Because reading the output from a file is faster than calculating its output by processing the node tree upstream, this can speed up large projects.

Note: Scripts that require Nuke to load a large number of files concurrently (for example, by having hundreds of Read nodes followed by TimeBlurs) may exceed the number of available file handles per process, causing problems when rendering scripts.

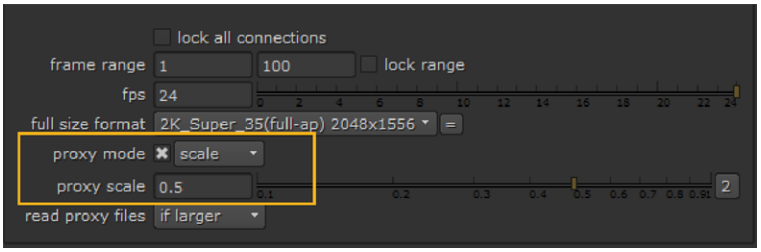
Nuke itself supports up to 2048 file handles on all systems; however, you may need to increase the file handle limit on your system.

On Mac, you can increase the default limit of 256 (depending on your version) by entering the following command from the Terminal and then running Nuke from the same Terminal session: **ulimit -Sn 2048**

Render Resolution and Format

Before rendering a script, it's important to check what is the currently active mode: the full-size or the proxy mode. Nuke executes all renders at the currently active scale. Thus, when a script is rendered in proxy mode, processing is done at the proxy scale and image output goes to the file name in the Write node's **proxy** field. If you do not specify a proxy file name, the render fails with an error. It never resizes the proxy image, and it does not write the proxy image over the full-size one.

To view and change the proxy resolution for the current script file, select **Edit > Project Settings** from the menu bar, or press **S** with the mouse pointer over the Node Graph or the Properties Bin.

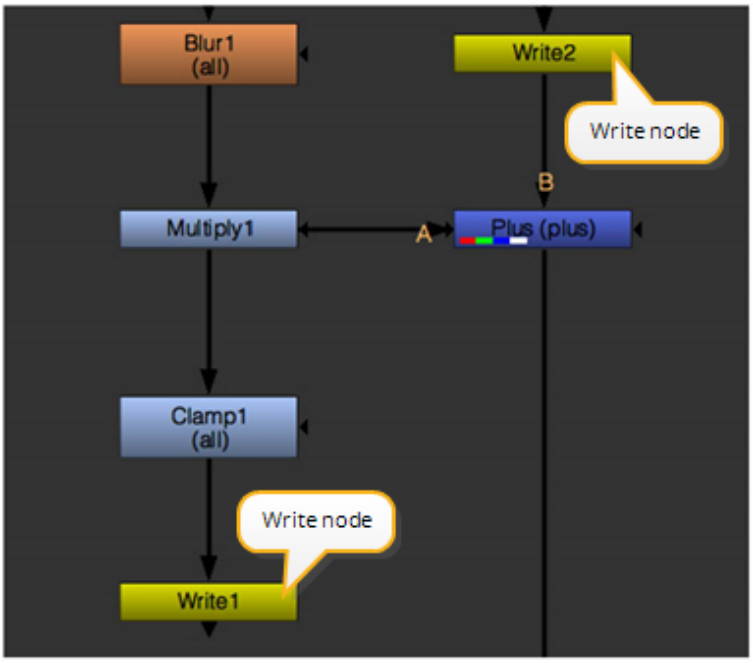


Changing the output resolution under project settings.

From the **Project Settings** properties panel, you can select a new render format from the dropdown menu of predefined resolutions, and toggle proxy rendering. You can also select the **new** option under either **full size format** or **proxy format** or use the **proxy scale** fields to define custom render resolutions for the composite. When rendering in proxy mode, use the dropdown menu on the right to select whether to use the resolution defined under **proxy format** or **proxy scale**. Also check that you have set **read proxy file** to what you want - this setting controls how Read nodes select the file to read (full res file or proxy) in the proxy mode. For more information on these settings, refer to Setting Up Your Script .

Output (Write) Nodes

With the correct resolution and format selected, you then insert Write nodes to indicate where you want to render images from the script.



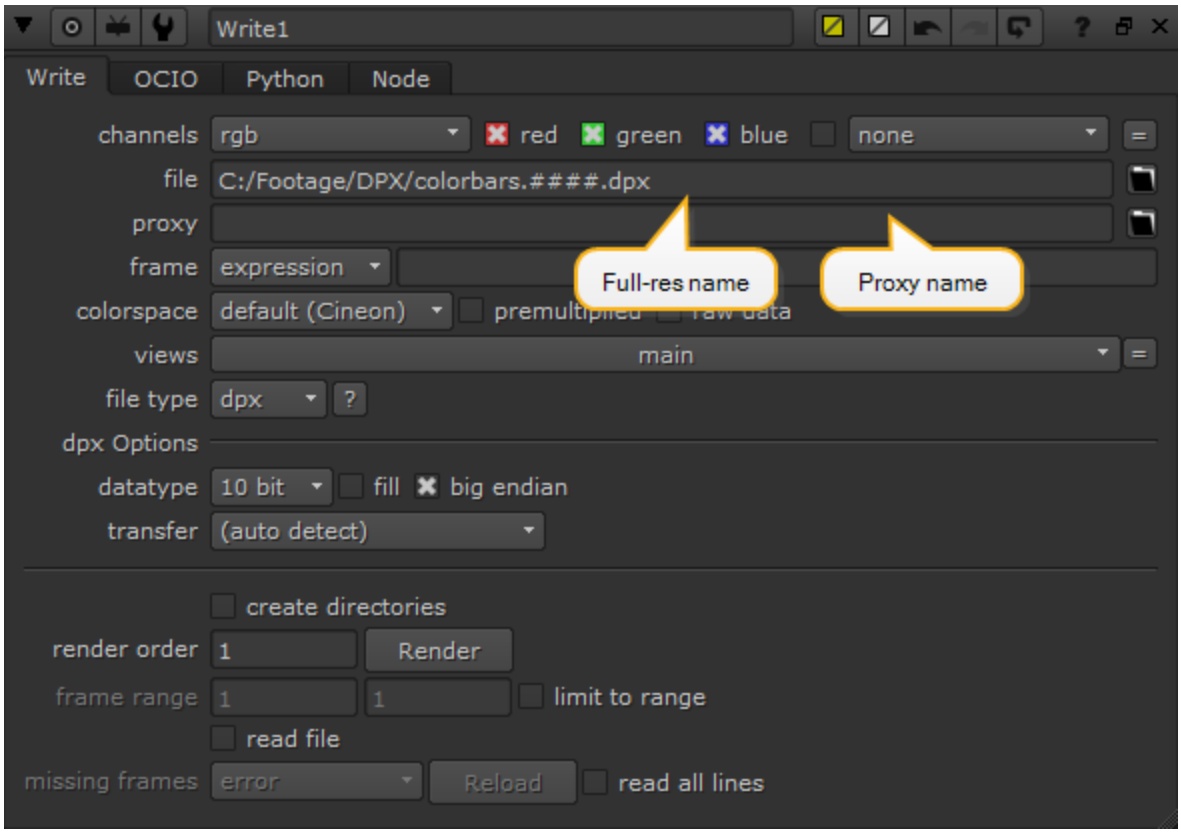
Inserting Write nodes for rendering.

One Write node is usually placed at the bottom of the compositing tree to render the final output. However, Write nodes have both input and output connectors, so they may be embedded anywhere in the compositing tree.

You can execute renders for a single Write node or all Write nodes in your compositing script.

To Render a Single Write Node

1. Select the node in the script from which you want to render an image.
2. Select **Image > Write** (or press **W** over the Node Graph). Nuke attaches a Write node and opens its properties panel.
3. Connect a Viewer to the Write node you want to render and verify that the correct resolution is displayed for output. If necessary, press **Ctrl/Cmd+P** to toggle between full-res and proxy resolution. The displayed output resolution is used for rendering.
4. In the properties panel, click the **file** or **proxy** field's folder icon (depending on whether you want to render high res or low res images) and browse to the directory where you want to store the rendered sequence. For instructions, see Using the File Browser.



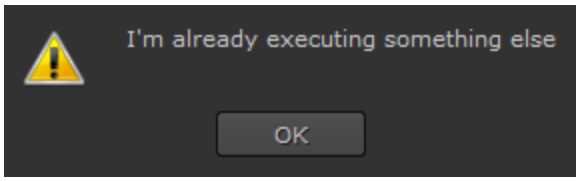
5. After the path, type a name for the rendered image and then click **OK**. If you're rendering an image sequence, include the frame number variable (for example, **####**) in the name.

   See To Render Selected or All Write Nodes in the Script below for examples of valid file names with the frame number variable.
6. If necessary, adjust the following controls:

   • Using the **channels** dropdown menu and checkboxes, select the channels you want to render.

- Using the **frame** dropdown menu and input field, set the relation between the currently processed frame and the numbering of the frame written out. For more information, see Changing the Numbering of Rendered Frames.
- check **read file** if you want the output of the Write node to be produced by reading the rendered file back in rather than by processing the upstream node tree. For more information on this and the **missing frames** control, see Using a Write Node to Read in the Rendered Image.
- From the **output transform** dropdown menu, select which lookup table to use when converting between the images' color space and Nuke's internal color space.
- From the **file type** dropdown menu, select the file format for the rendered images. If you don't specify a file format, Nuke uses the extension in the file name to figure out the format.
- Check the **limit to range** box if you want to disable the node when executed outside of a specified frame range. In the **frame range** fields, enter the range of frames you want to make executable with this Write node.

7. In the Write node properties, click the **Render** button.
8. In the Render dialog, adjust the render settings if necessary:
   - **Frame range** - set the frame range you want to render.
   - **Use proxy** - check to use proxy mode.
   - **Render using frame server** - enable this checkbox if you want to use Nuke's frame server to render the frames specified in the current render task. When the frame server is disabled, Nuke can only perform one render task at a time. Executing another render displays a warning and the current render is paused until you acknowledge the message:



   See Rendering Using the Frame Server for more information.
   - **Continue on error** - check to keep rendering even if an error occurs during the process.
   - **Views** - set which views to include in the render. See Selecting Which Views to Render for more information.
9. Click **OK**.

Tip: When specifying the frame range to render, you can enter complex frame ranges into the frame range prompt dialog. For example, if you enter **1-5 8 10 15 22-25**, it only renders those frames. You can also increment rendered frames using something like **10-50x10**, which resolves to only frames 10, 20, 30, 40, and 50 from the range.

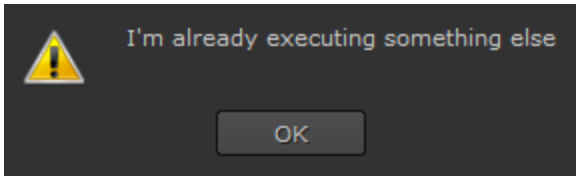
Likewise, you can specify multiple ranges on the command line, for example:

```
nuke -F 1-5 -F 8 -F 10 -F 15 -F 20-40x5 -x myscript.nk
```

Tip: When rendering with the Write node, you can force a certain data type by adding the data type and a colon before the file path. For example, you can enter ftiff:C:\Temp\test.tif as the file path to render a file whose data type is ftiff and extension **.tif**.

To Render Selected or All Write Nodes in the Script

1. Connect a Viewer to a Write node you want to render and verify that the correct resolution is displayed for output.
2. If necessary, press **Ctrl/Cmd+P** to toggle between full-res and proxy resolution. The displayed output resolution is used for rendering.
3. If you want, you can change the order in which your Write nodes are rendered by giving them custom render order numbers in the **render order** field.
4. Do one of the following:
   - With the desired Write node selected, select **Render** > **Render selected** (or press **F7**).
   - Select **Render** > **Render all** (or press **F5**).
5. In the Render dialog, adjust the render settings if necessary. The default values are drawn from the Viewer you have active.
   - **Frame range** - set the frame range you want to render.
   - **Use proxy** - check to use proxy mode.
   - **Render using frame server** - disable this checkbox if you don't want to use Nuke's frame server to render the frames specified in the current render task. If you disable the frame server, Nuke can only perform one render task at a time. Executing another render displays a warning and the current render is paused until you acknowledge the message:



See Rendering Using the Frame Server for more information.

- **Continue on error** - check to keep rendering even if an error occurs during the process.
- **Views** - set which views to include in the render. See Selecting Which Views to Render for more information.

6. Click **OK**.

Tip: When specifying the frame range to render, you can enter complex frame ranges into the frame range prompt dialog. For example, if you enter "1-5 8 10 15 22-25", it only renders those frames. Likewise, you can specify multiple ranges on the command line, for example:
nuke -F 1-5 -F 8 -F 10 -F 15 -F 22-25 -x myscript.nk

You can see the progress of your render in the status window that appears. When the render is complete, the rendered images are added to the directory you specified.

Tip: When rendering with the Write node, you can force a certain data type by adding the data type and a colon before the file path. For example, you can enter ftiff:C:\Temp\test.tif as the file path to render a file whose data type is ftiff and extension **.tif**.

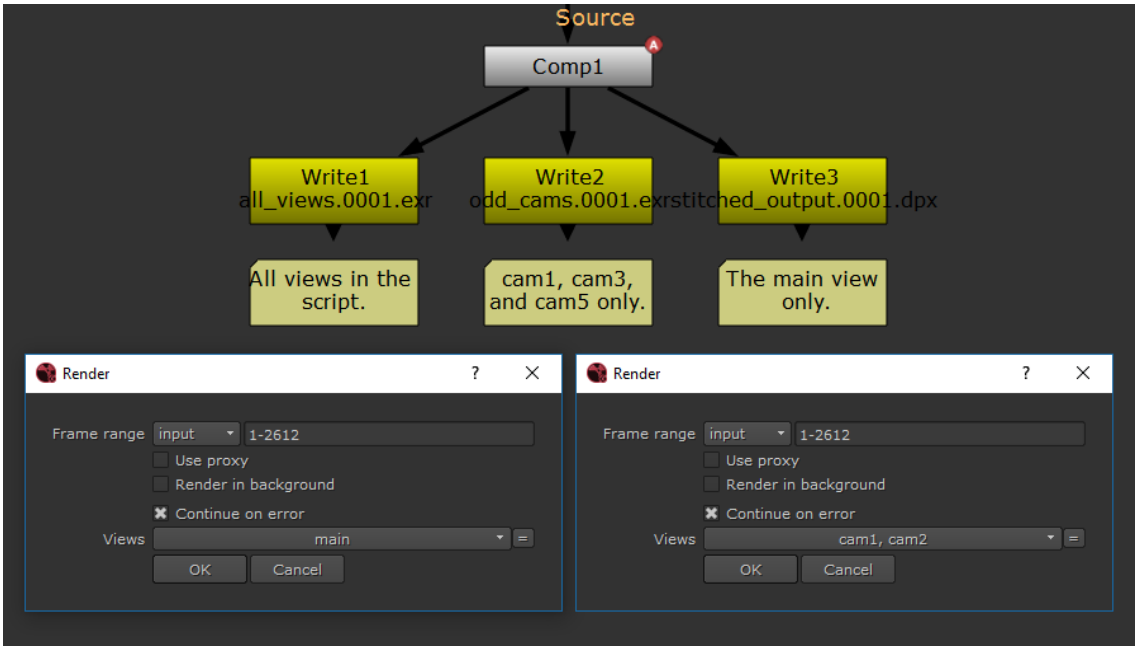
Selecting Which Views to Render

The Write nodes in a script determine which views are available at render time, but the **Render** dialog can be used as a filter to limit the views rendered from those available. Write nodes and the **Render** dialog default to all available views.

For example, if your script contained two Write nodes calling for the **main** view and **left** and **right** views, but the **Render** dialog is set to render **main**, then only the **main** view is rendered.

Conversely, if the Write nodes are calling for the **left** and **right** views, but the **Render** dialog is set to render **main**, then no views are rendered.

A slightly more complex example is shown in the image.



In the example above, the left-hand dialog is set to render only **main** and the right-hand dialog is set to render only **cam1** and **cam2**.

So, in the case of the first render, Write1 renders **main**, Write2 renders nothing, and Write3 renders **main**.

In the case of the second render, Write1 renders **cam1** and **cam2**, Write2 renders **cam1**, and Write3 renders nothing.

Notes on Rendering MOV Files

If you are rendering **.mov** files, you can:

- choose the codec from the **Codec** dropdown menu.
- **FPS** - set the playback frames per second for the output file.
- Set the **Codec Profile** or **Pixel Format** used to write the file, depending on the **Codec** chosen.
- **Audio File** - allows you to specify a separate audio file to include in the output. Either enter the file path manually or click the browse button to locate the audio file.
- **Audio Offset** - sets the start time of any audio file specified in the **Audio File** control. The unit of measure is specified using the **Units** control. Negative values cause the audio to begin before the video and positive values cause the audio to begin after the video.

Advanced MOV Options

You can adjust advanced codec options by opening the **Advanced** dropdown. The options available vary, depending on the Codec you're using to render the output.

For a full description of the **.mov** options available, see the Write node topic in the Nuke Reference Guide.

Control	Description
Animation Codec	
Data Range	Sets the minimum and maximum color range values for encoding DNx files: • **Full Range** - encodes with the full data range (0-255 for 8-bit or 0-1023 for 10-bit). • **Video Range** - encodes with a reduced, or video/legal, data range (16-240 for 8-bit or 64-960 for 10-bit).
Fast Start	When enabled, MOVs are playable while still down loading during streaming.
Write Timecode	When enabled, a time code track is added to the rendered **.mov** file. You can also use the quicktime/reel metadata, if present, to give the track its reel name. You can add this key using the ModifyMetaData node if it doesn't exist.  **Note:** The **input/timecode** key must be present in the sequence metadata in order to write a time code.
GOP Size	Sets how many frames can be placed together to form a compression GOP (group of pictures).  **Note:** Use caution with this control as large alterations can stop other applications reading the rendered file.
Apple ProRes Codec	
Data Range	Sets the minimum and maximum color range values for encoding DNx files: • **Full Range** - encodes with the full data range (0-255 for 8-bit or 0-1023 for 10-bit). • **Video Range** - encodes with a reduced, or video/legal, data range (16-240 for 8-

Control	Description
	bit or 64-960 for 10-bit).
Fast Start	When enabled, MOVs are playable while still down loading during streaming.
Write Timecode	When enabled, a time code track is added to the rendered **.mov** file. You can also use the quicktime/reel metadata, if present, to give the track its reel name. You can add this key using the ModifyMetaData node if it doesn't exist.  **Note:** The **input/timecode** key must be present in the sequence metadata in order to write a time code.
Avid DNxHD Codec	
Data Range	Sets the minimum and maximum color range values for encoding DNx files: • **Full Range** - encodes with the full data range (0-255 for 8-bit or 0-1023 for 10-bit). • **Video Range** - encodes with a reduced, or video/legal, data range (16-240 for 8-bit or 64-960 for 10-bit).
Fast Start	When enabled, MOVs are playable while still down loading during streaming.
Write Timecode	When enabled, a time code track is added to the rendered **.mov** file. You can also use the quicktime/reel metadata, if present, to give the track its reel name. You can add this key using the ModifyMetaData node if it doesn't exist.  **Note:** The **input/timecode** key must be present in the sequence metadata in order to write a time code.
Avid DNxHR Codec	
Data Range	Sets the minimum and maximum color range values for encoding DNx files: • **Full Range** - encodes with the full data range (0-255 for 8-bit or 0-1023 for 10-bit). • **Video Range** - encodes with a reduced, or video/legal, data range (16-240 for 8-bit or 64-960 for 10-bit).

Control	Description
Fast Start	When enabled, MOVs are playable while still down loading during streaming.
Write Timecode	When enabled, a time code track is added to the rendered **.mov** file. You can also use the quicktime/reel metadata, if present, to give the track its reel name. You can add this key using the ModifyMetaData node if it doesn't exist.  **Note:** The **input/timecode** key must be present in the sequence metadata in order to write a time code.
H.264 Codec	
Quality	Selects a preset level that automatically adjusts the **Advanced** controls. Higher values produce larger files with better quality. Select **Custom** to adjust the **Advanced** controls manually.
YCbCr Matrix	Sets the way RGB is converted to Y'CbCr. **Rec 601** and **Rec 709** follow the ITU.BC specifications. **Auto** uses a set a heuristics to determine the best color matrix to use, including **Format-based**, which sets the color matrix to **Rec 601** for formats with a width below 840 pixels and **Rec 709** for formats with a width of 840 pixels or above.
Data Range	Sets the minimum and maximum color range values for encoding DNx files: • **Full Range** - encodes with the full data range (0-255 for 8-bit or 0-1023 for 10-bit). • **Video Range** - encodes with a reduced, or video/legal, data range (16-240 for 8-bit or 64-960 for 10-bit).
Fast Start	When enabled, MOVs are playable while still down loading during streaming.
Write Timecode	When enabled, a time code track is added to the rendered **.mov** file. You can also use the quicktime/reel metadata, if present, to give the track its reel name. You can add this key using the ModifyMetaData node if it doesn't exist.  **Note:** The **input/timecode** key must be present in the sequence metadata in order to write a time code.

Control	Description
GOP Size	Sets how many frames can be placed together to form a compression GOP (group of pictures). **Note:** Use caution with this control as large alterations can stop other applications reading the rendered file.
B Frames	Sets the maximum number of B frames that can be consecutive in the rendered file. The default, 0, does not impose any maximum number of B frames in the output.
Bitrate	Sets the target bitrate that the codec attempts to reach, within the limits set by the **Bitrate Tolerance** and **QuantizerMin/Max** controls.
Bitrate Tolerance	Sets the amount that the bitrate can vary from the **Bitrate** setting. Setting this tolerance too low can result in renders failing.
Quantizer Min	Sets the quality range within which the codec can vary the image to achieve the specified bitrate. Higher ranges can introduce image degradation.
Quantizer Max	
Motion JPEG A Codec	
Quality	Selects a preset level that automatically adjusts the **Advanced** controls. Higher values produce larger files with better quality. Select **Custom** to adjust the **Advanced** controls manually.
YCbCr Matrix	Sets the way RGB is converted to Y'CbCr. **Rec 601** and **Rec 709** follow the ITU.BC specifications. **Auto** uses a set a heuristics to determine the best color matrix to use, including **Format-based**, which sets the color matrix to **Rec 601** for formats with a width below 840 pixels and **Rec 709** for formats with a width of 840 pixels or above.
Data Range	Sets the minimum and maximum color range values for encoding DNx files: • **Full Range** - encodes with the full data range (0-255 for 8-bit or 0-1023 for 10-bit). • **Video Range** - encodes with a reduced, or video/legal, data range (16-240 for 8-bit or 64-960 for 10-bit).
Fast Start	When enabled, MOVs are playable while still down loading during streaming.

Control	Description
Write Timecode	When enabled, a time code track is added to the rendered **.mov** file. You can also use the quicktime/reel metadata, if present, to give the track its reel name. You can add this key using the ModifyMetaData node if it doesn't exist. **Note:** The **input/timecode** key must be present in the sequence metadata in order to write a time code.
Bitrate	Sets the target bitrate that the codec attempts to reach, within the limits set by the **Bitrate Tolerance** and **QuantizerMin**/**Max** controls.
Bitrate Tolerance	Sets the amount that the bitrate can vary from the **Bitrate** setting. Setting this tolerance too low can result in renders failing.
Quantizer Min	Sets the quality range within which the codec can vary the image to achieve the specified bitrate. Higher ranges can introduce image degradation.
Quantizer Max	
Motion JPEG B Codec	
Quality	Selects a preset level that automatically adjusts the **Advanced** controls. Higher values produce larger files with better quality. Select **Custom** to adjust the **Advanced** controls manually.
YCbCr Matrix	Sets the way RGB is converted to Y'CbCr. **Rec 601** and **Rec 709** follow the ITU.BC specifications. **Auto** uses a set a heuristics to determine the best color matrix to use, including **Format-based**, which sets the color matrix to **Rec 601** for formats with a width below 840 pixels and **Rec 709** for formats with a width of 840 pixels or above.
Data Range	Sets the minimum and maximum color range values for encoding DNx files: • **Full Range** - encodes with the full data range (0-255 for 8-bit or 0-1023 for 10-bit). • **Video Range** - encodes with a reduced, or video/legal, data range (16-240 for 8-bit or 64-960 for 10-bit).
Fast Start	When enabled, MOVs are playable while still down loading during streaming.
Write Timecode	When enabled, a time code track is added to the rendered **.mov** file.

Control	Description
	You can also use the quicktime/reel metadata, if present, to give the track its reel name. You can add this key using the ModifyMetaData node if it doesn't exist. **Note:** The **input/timecode** key must be present in the sequence metadata in order to write a time code.
Bitrate	Sets the target bitrate that the codec attempts to reach, within the limits set by the **Bitrate Tolerance** and **QuantizerMin**/**Max** controls.
Bitrate Tolerance	Sets the amount that the bitrate can vary from the **Bitrate** setting. Setting this tolerance too low can result in renders failing.
Quantizer Min Quantizer Max	Sets the quality range within which the codec can vary the image to achieve the specified bitrate. Higher ranges can introduce image degradation.
MPEG-4 Codec	
Quality	Selects a preset level that automatically adjusts the **Advanced** controls. Higher values produce larger files with better quality. Select **Custom** to adjust the **Advanced** controls manually.
YCbCr Matrix	Sets the way RGB is converted to Y'CbCr. **Rec 601** and **Rec 709** follow the ITU.BC specifications. **Auto** uses a set a heuristics to determine the best color matrix to use, including **Format-based**, which sets the color matrix to **Rec 601** for formats with a width below 840 pixels and **Rec 709** for formats with a width of 840 pixels or above.
Data Range	Sets the minimum and maximum color range values for encoding DNx files: • **Full Range** - encodes with the full data range (0-255 for 8-bit or 0-1023 for 10-bit). • **Video Range** - encodes with a reduced, or video/legal, data range (16-240 for 8-bit or 64-960 for 10-bit).
Fast Start	When enabled, MOVs are playable while still down loading during streaming.
Write Timecode	When enabled, a time code track is added to the rendered **.mov** file. You can also use the quicktime/reel metadata, if present, to give the track its reel

Control	Description
	name. You can add this key using the ModifyMetaData node if it doesn't exist. **Note:** The **input/timecode** key must be present in the sequence metadata in order to write a time code.
GOP Size	Sets how many frames can be placed together to form a compression GOP (group of pictures). **Note:** Use caution with this control as large alterations can stop other applications reading the rendered file.
B Frames	Sets the maximum number of B frames that can be consecutive in the rendered file. The default, 0, does not impose any maximum number of B frames in the output.
Limit Bitrate	When enabled, the maximum **Bitrate** is capped and the **Bitrate Tolerance** is set to 0. This control is designed to reduce output file size, but the render quality is generally lower.
Bitrate	Sets the target bitrate that the codec attempts to reach, within the limits set by the **Bitrate Tolerance** and **QuantizerMin**/**Max** controls.
Bitrate Tolerance	Sets the amount that the bitrate can vary from the **Bitrate** setting. Setting this tolerance too low can result in renders failing.
Quantizer Min Quantizer Max	Sets the quality range within which the codec can vary the image to achieve the specified bitrate. Higher ranges can introduce image degradation.
PHOTO JPEG Codec	
Quality	Selects a preset level that automatically adjusts the **Advanced** controls. Higher values produce larger files with better quality. Select **Custom** to adjust the **Advanced** controls manually.
YCbCr Matrix	Sets the way RGB is converted to Y'CbCr. **Rec 601** and **Rec 709** follow the ITU.BC specifications.

Control	Description
	Auto uses a set a heuristics to determine the best color matrix to use, including **Format-based**, which sets the color matrix to **Rec 601** for formats with a width below 840 pixels and **Rec 709** for formats with a width of 840 pixels or above.
Data Range	Sets the minimum and maximum color range values for encoding DNx files: • **Full Range** - encodes with the full data range (0-255 for 8-bit or 0-1023 for 10-bit). • **Video Range** - encodes with a reduced, or video/legal, data range (16-240 for 8-bit or 64-960 for 10-bit).
Fast Start	When enabled, MOVs are playable while still down loading during streaming.
Write Timecode	When enabled, a time code track is added to the rendered **.mov** file. You can also use the quicktime/reel metadata, if present, to give the track its reel name. You can add this key using the ModifyMetaData node if it doesn't exist. **Note:** The **input/timecode** key must be present in the sequence metadata in order to write a time code.
Bitrate	Sets the target bitrate that the codec attempts to reach, within the limits set by the **Bitrate Tolerance** and **QuantizerMin/Max** controls.
Bitrate Tolerance	Sets the amount that the bitrate can vary from the **Bitrate** setting. Setting this tolerance too low can result in renders failing.
Quantizer Min	Sets the quality range within which the codec can vary the image to achieve the specified bitrate. Higher ranges can introduce image degradation.
Quantizer Max	Quantizer Max
PNG Codec	
Data Range	Sets the minimum and maximum color range values for encoding DNx files: • **Full Range** - encodes with the full data range (0-255 for 8-bit or 0-1023 for 10-bit). • **Video Range** - encodes with a reduced, or video/legal, data range (16-240 for 8-bit or 64-960 for 10-bit).
Fast Start	When enabled, MOVs are playable while still down loading during streaming.

Control	Description
Write Timecode	When enabled, a time code track is added to the rendered **.mov** file. You can also use the quicktime/reel metadata, if present, to give the track its reel name. You can add this key using the ModifyMetaData node if it doesn't exist. **Note:** The **input/timecode** key must be present in the sequence metadata in order to write a time code.
Uncompressed Codec	
YCbCr Matrix	Sets the way RGB is converted to Y'CbCr. **Rec 601** and **Rec 709** follow the ITU.BC specifications. **Auto** uses a set a heuristics to determine the best color matrix to use, including **Format-based**, which sets the color matrix to **Rec 601** for formats with a width below 840 pixels and **Rec 709** for formats with a width of 840 pixels or above.
Data Range	Sets the minimum and maximum color range values for encoding DNx files: • **Full Range** - encodes with the full data range (0-255 for 8-bit or 0-1023 for 10-bit). • **Video Range** - encodes with a reduced, or video/legal, data range (16-240 for 8-bit or 64-960 for 10-bit).
Fast Start	When enabled, MOVs are playable while still down loading during streaming.
Write Timecode	When enabled, a time code track is added to the rendered **.mov** file. You can also use the quicktime/reel metadata, if present, to give the track its reel name. You can add this key using the ModifyMetaData node if it doesn't exist. **Note:** The **input/timecode** key must be present in the sequence metadata in order to write a time code.

Nuke writes the selected colorspace and other information into the metadata of the file. If you then read the rendered file in, Nuke reads that metadata and is able to pick the correct defaults for the file. To see the file metadata yourself, use a ViewMetaData node.

When writing **.mov** files, some users have reported receiving the following error at the end of rendering: *Failed to flatten movie data: the movie is open in another application.*
This is because the output file has been opened by some other process. This can be the Finder on Mac trying

to show a preview, an OS file system search trying to index the file, or a virus checker, for example. The workaround is to turn off **Fast Start** in the Write node controls to skip the flattening process that is affected.

Notes on Rendering OpenEXR Files

Nuke supports multi-part OpenEXR 3.3.6 files, which allow you to store your channels, layers, and views in separate parts of the file. Storing the data this way can make loading **.exr** files faster, as Nuke only has to access the part of the file that is requested rather than all parts. However, for backwards compatibility, you also have the option to render your **.exr** files as single-part images.

To set how the data is stored in your rendered **.exr** file, open the Write properties and set **interleave** to:

- **channels, layers and views** - Write channels, layers, and views into the same part of the rendered **.exr** file. This creates a single-part file to ensure backwards compatibility with earlier versions of Nuke and other applications using an older OpenEXR library.
- **channels and layers** - Write channels and layers into the same part of the rendered **.exr** file, but separate views into their own part. This creates a multi-part file and can speed up Read performance, as Nuke only has to access the part of the file that is requested rather than all parts.
- **channels** - Separate channels, layers, and views into their own parts of the rendered **.exr** file. This creates a multi-part file and can speed up Read performance if you work with only a few layers at a time.

Rendering Using the Frame Server

The Frame Server reduces render times by sharing work over the number of render processes specified in the **Preferences**. The Frame Server logs renders in the **Background Renders** panel, including information such as script name, Write node, frame range, progress, and whether or not the render produced errors. The Frame Server does not display a progress bar for render tasks, but you can disable **Render using frame server** in the **Render** dialog to use the process from older versions of Nuke.



Note: The Frame Server can only render sequential formats, like **.dpx** and **.exr** files. Container formats, such as **.mov** and **.mxf** are not supported and the **Render using frame server** option is not displayed in the **Render** dialog.

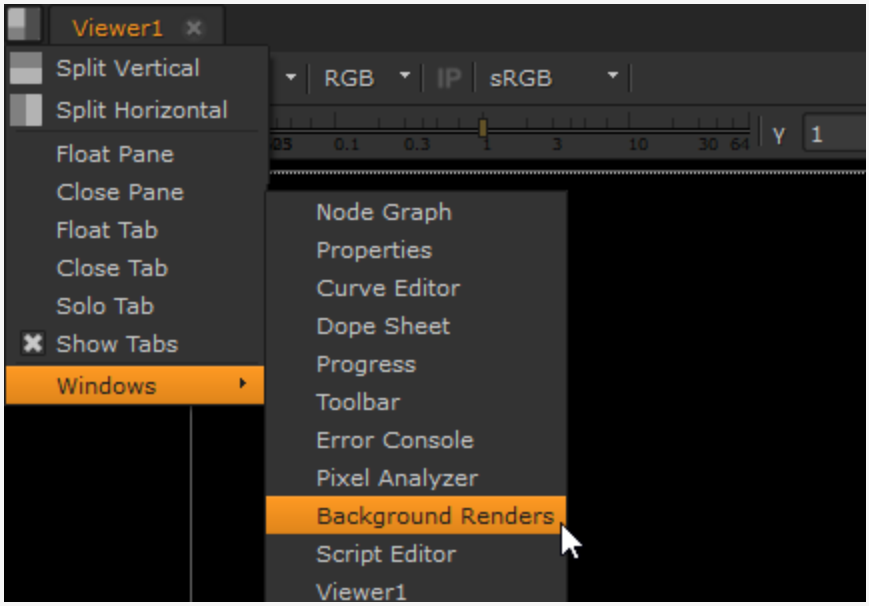
The Frame Server is disabled by default, but you can enable it in the **Preferences** > **Performance** > **Threads/Processes** or on a per render basis in the Write node's **Render** dialog.

Note: Local Frame Server processes use ports 5558-5662.

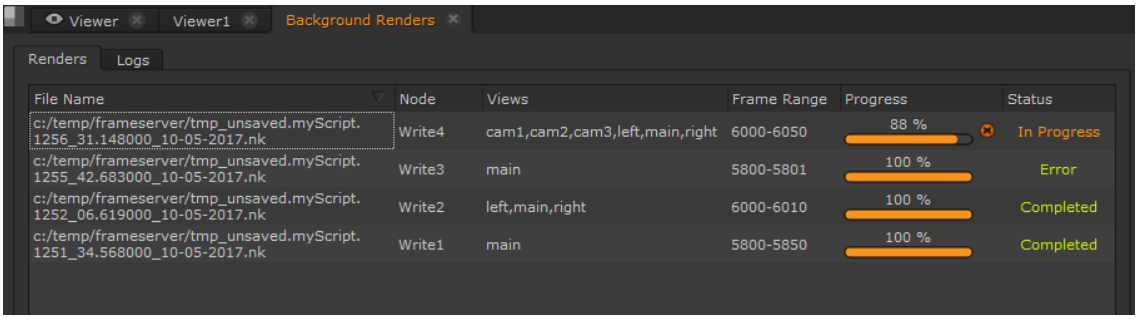
Rendering Write nodes, either from a node's **Properties** or from the **Render** menu, displays the **Render** dialog. See Output (Write) Nodes for more information on initiating renders. If the Frame Server is enabled, clicking **OK** in the **Render** dialog starts the selected renders in the background, allowing you to keep working.

The **Background Renders** tab is not displayed as part of the workspaces that ship with Nuke, but you can enable **Performance** > **Threads/Processes** > **focus background renders** in the **Preferences** to open the panel automatically when the Frame Server is in use.

Tip: You can also add the panel manually by right-clicking the button in the top-left of any pane, and then selecting **Windows** > **Background Renders**.



The **Background Renders** panel includes information such as script name, Write node, frame range, progress, and whether or not the render produced errors.



Frame Server Preferences

You can control how the Frame Server behaves using the **Performance** > **Threads/Processes** tab in the **Preferences**:

- **render using frame server (Nuke)** - when enabled, the Frame Server is always used for rendering.
- **focus background renders** - when enabled, rendering using the Frame Server automatically opens the **Background Renders** panel, or if it is already open, shifts focus to the panel.
- **frame server processes to run** - sets the number of render processes available to the Frame Server.



Note: You must restart Nuke if you change the number of processes available.

- **export renders** - allows you to set limitations on the resources available for render processes:
  - **limit renderer (more responsive ui)** – select this option to make the user interface more responsive during rendering. It tells Nuke to use 2 threads to transcode and to use 25% of RAM to cache. Using this option is likely to result in slower renders.
  - **no renderer limits (fastest transcoding)** – select this option to ensure that renders happen as quickly as possible. This option may result in a less responsive user interface during rendering.
  - **customize render limits** – select this option to manually configure the **number of threads** used and **cache memory** available during renders.

Using the Frame Server on External Machines

Although Nuke is capable of rendering frames internally, running the Frame Server on an external machine can accelerate the process considerably by sharing work across a network of machines.



Note: The Frame Server requires a Nuke license (nuke_i) on the main workstation, but only a Nuke render license (nuke_r) on the render machine.

If you want to use an interactive license (nuke_i) on the render machines, add the **--useInteractiveLicense** argument to the **runframeserver.py** command described below.

Configuring the Frame Server on External Machines

Nuke's Frame Server can be set up on an external machine (or a number of machines) to render from your session. To do this, you need to run the **runframeserver.py** script on the render machines, found inside the Python site-packages, with specific command line arguments.



Warning: In order for everything to work smoothly, you need to ensure that both your external render machines and main Nuke session can read and write files to a shared location, such as an NFS share.

Depending on platform this can be done by manipulating your default **umask** setting, but be aware that this alters the permissions of the created files.

Additionally, Macs and certain Linux distributions, such as RHEL, can not function as the main workstation if the firewall is blocking the communication port 5560. You can configure the firewall to allow certain ports through the firewall using the **iptables** command, but use caution when doing so. For example:

```
sudo iptables -I INPUT -p tcp --dport 5560 --syn -j ACCEPT
```

Please refer to the documentation on firewalls for your particular platform for more information.

The Frame Server uses a number of worker processes on the render machine, each of which requires allocated resources, such as threads, memory, and so on. There are a number of arguments that you must pass to **runframeserver.py** for the server to work correctly:

- **--numworkers** - this is the number of concurrent Nuke processes that are launched when you run this server render node.
- **--nukeworkerthreads** - the number of threads that each worker is allocated. This is similar to setting the **-m** argument when running Nuke from the command line.
- **--nukeworkermemory** - the amount of memory, in MB, allocated to each frame server worker.
- **--workerconnecturl** - the TCP port address of the main workstation you want to serve. For example:

  ```
  tcp://bob:5560
  ```

  where **bob** is the resolved hostname of a machine you wish to serve. You can also use an IP address.

Tip: To ensure that you're entering a valid URL, try using the **ping** command to see if you get a response.

• **--nukepath** - the path to the Nuke application on the render machine.

Tip: On Windows, if there are spaces in the file path, remember to place the path in quotes. For example, **--nukepath="C:\Program Files\Nuke17.0v3\Nuke17.0.exe"**

On a Linux render machine, an example command prompt entry running from the install directory might look like this:

```
./python ./pythonextensions/site-
packages/foundry/frameserver/nuke/runframeserver.py --numworkers=2 --
nukeworkerthreads=4 --nukeworkermemory=8096 --workerconnecturl=tcp://bob:5560 -
-nukepath=./Nuke17.0
```

On a Windows render machine, an example command prompt entry running from the install directory might look like this:

```
python.exe pythonextensions\site-
packages\foundry\frameserver\nuke\runframeserver.py --numworkers=2 --
nukeworkerthreads=4 --nukeworkermemory=8096 --workerconnecturl=tcp://bob:5560 -
-nukepath=Nuke17.0.exe
```

On a macOS render machine, an example command prompt entry running from the application directory might look like this:

```
./python ../Resources/pythonextensions/site-
packages/foundry/frameserver/nuke/runframeserver.py --numworkers=2 --
nukeworkerthreads=4 --nukeworkermemory=8096 --workerconnecturl=tcp://bob:5560 --
nukepath=/Applications/Nuke12.1v1/Nuke12.1v1.app/Contents/MacOS/Nuke12.1
```

In the examples, we specify that the render machine uses two Nuke workers, with four threads and 8 GB RAM each, and are on the same network as the main Nuke workstation running on **bob**.

Tip: If your render machines run a different OS than your main Nuke machine, you can use the **--remap** command line argument to convert file paths between them. The host file path is read first followed by the render file path. Nuke expects all file paths to use **/** (forward slash) between directories. For example:

```
--remap "P:/,/mnt/renders/"
```

converts host paths beginning with **P:/** (Windows style) to remote paths beginning with **/mnt/renders/** (Linux style).

You can check that the Frame Server and workers are connected by running the following lines in the Script Editor on the main workstation:

```
from hiero.ui.nuke_bridge.FnNsFrameServer import frameServer
print([worker.address for worker in frameServer.getStatus(1).workerStatus])
```

Successful connections should report something similar to the following in the output panel:

```
['Worker 0 - henry.local - 192.168.1.11', 'Worker 0 - bob.local -
192.168.1.111', 'Worker 1 - henry.local - 192.168.1.11']
```

Where **henry.local** is the name of the remote render machine, and **bob.local** is the name of the main Nuke session.



Note: If the workers cannot contact the Frame Server, an exception is printed in the Script Editor's output panel.

Frame Server Logs

Broker and Worker logging can to help diagnose Frame Server issues. The logs are written to NUKE_TEMP_DIR/logs by default, and take the form:

```
broker.log
worker-0.log
worker-1.log
worker-2.log
```



Note: Running the Frame Server using Python, as described above, always writes log files to the specific OS temporary directory. For example, on Windows **C:\temp** is used.

Tip: You can use the FRAMESERVER_LOG_DIR environment variable to force Frame Server logs into a different location. See for more information.

Bypassing Nodes During Renders

As a Nuke script gets larger and starts to contain quite a few nodes, the processing of the script may take longer to respond. This is especially the case with nodes that require more processing power for their calculations, such as GPU accelerated node, and can cause lagging when working in the script. Some examples of heavy processing nodes that may introduce slowdowns when using Nuke include: BlinkScript, Convolve, Denoise, Defocus, Kronos, MotionBlur, OFlow, VectorGenerator, and ZDefocus.

For such scripts where working becomes difficult and lagging occurs, there is a handy **$gui** expression operator that can be used to disable user interface (GUI) processing. The **$gui** operator in expressions returns either 1 or 0 as result:

- 1 is returned when the node is calculated through the GUI.
- 0 is returned at render time, when the node is not being processed by the GUI.

The standard way to add this into a node is through an expression. Functionality-wise, it acts similar to a Switch node but is driven by whether the GUI is being used or not, which makes it autonomous.



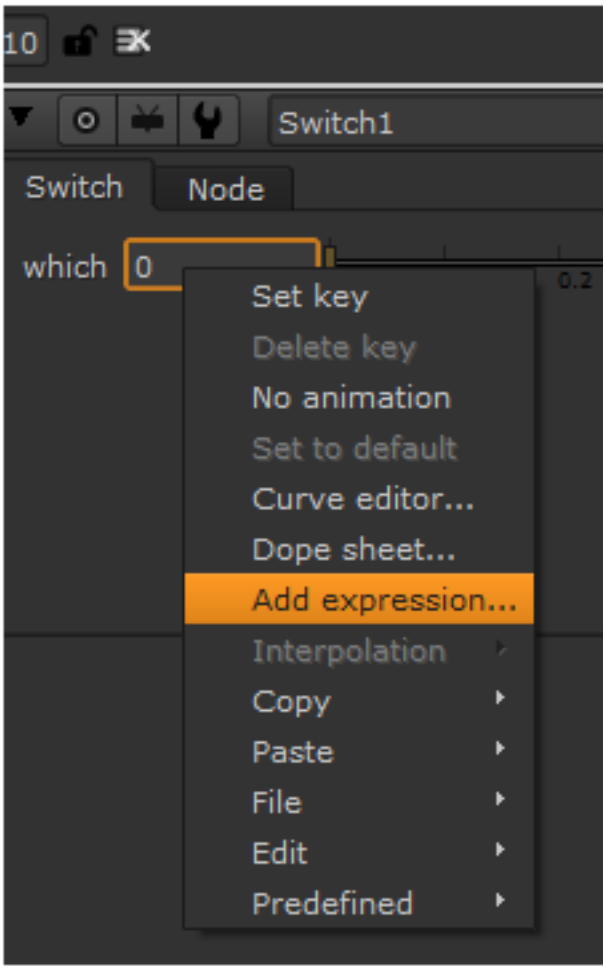
Note: These nodes are GPU accelerated nodes and if you have a fast graphics card, the **$gui** operator in expressions might not be needed. However, if you are experiencing lagging, using **$gui** in expressions in conjunction with your GPU acceleration may help.

The most common uses for the **$gui** operator in Nuke are: The Switch Method, The Disable Method, and The Selective Variation Method.

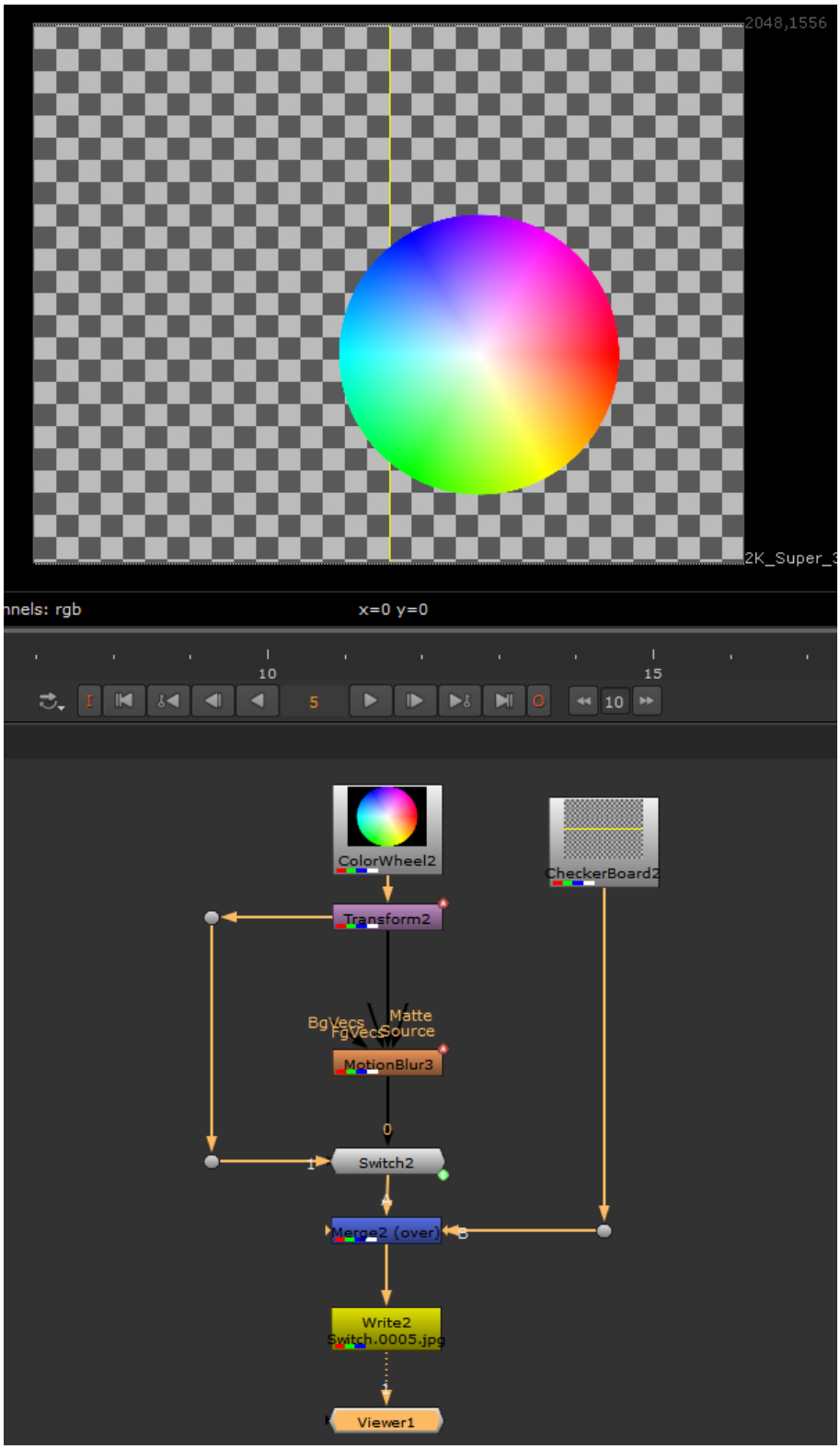
The Switch Method

The Switch method is the most common way of using **$gui** in expressions. Unlike the other methods, it allows you do disable the Switch node and turn off the expression when not needed.

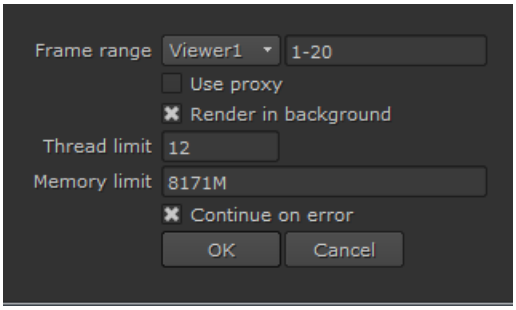
1. Create a Switch node after the processor heavy node.
2. Right-click on the Switch node's **which** control and then select **Add expression...**



3. In the expression box type **$gui**.
   You should see the result displayed as 1 since the node is processed within the Nuke GUI.
4. Connect the **0** input to the processor heavy node (in this example a MotionBlur node with 20 samples).
5. Connect the **1** input to the node tree before the processor heavy node (in this example a Transform node).
6. Playback the Viewer frames through the GUI and you will see the processing done faster since it bypasses the MotionBlur node and not display the blur results applied to the final image.

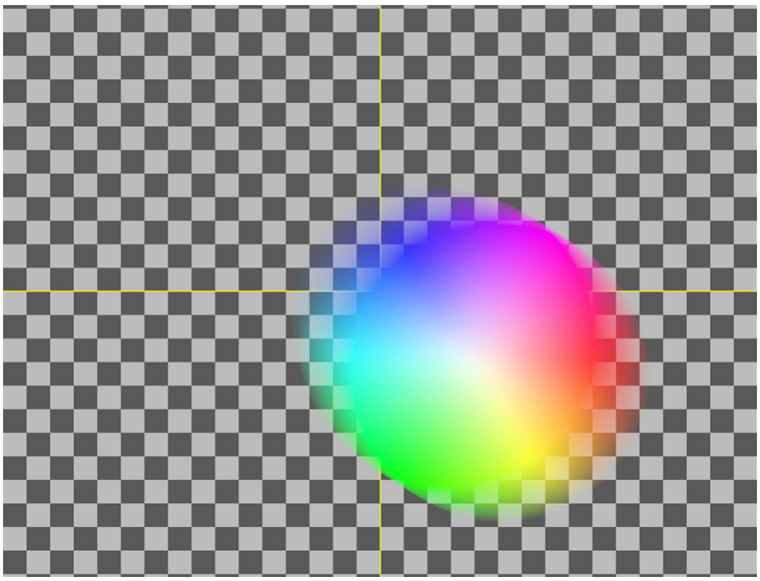


7. To process the MotionBlur and see the final result, render to disk using **Render in background** or the **Frame Server**. This uses an external process outside of the Nuke GUI and should not slowdown the script manipulation in the meantime.



Note: If you don't render in the background or use the Frame Server, the render uses the same process as the Nuke GUI and will not work as expected.

8. Once rendered, the motion blurred result is processed and displayed.

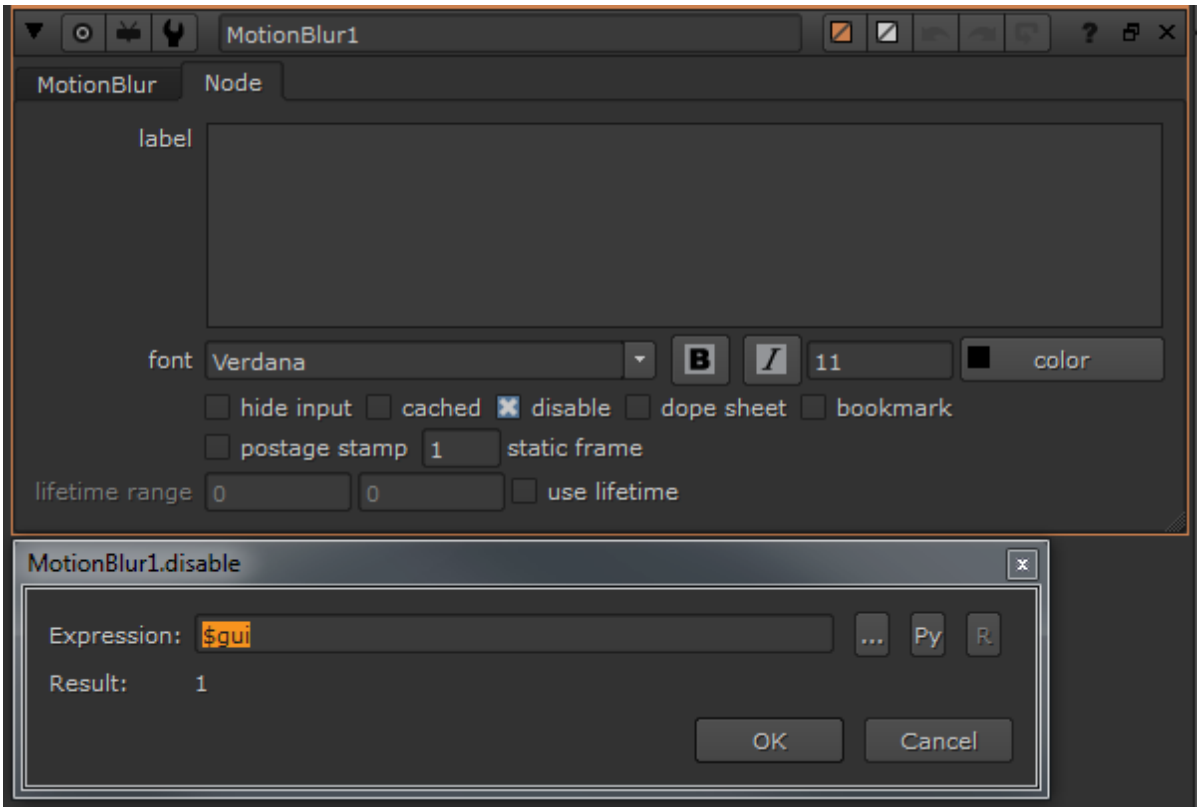


The Disable Method

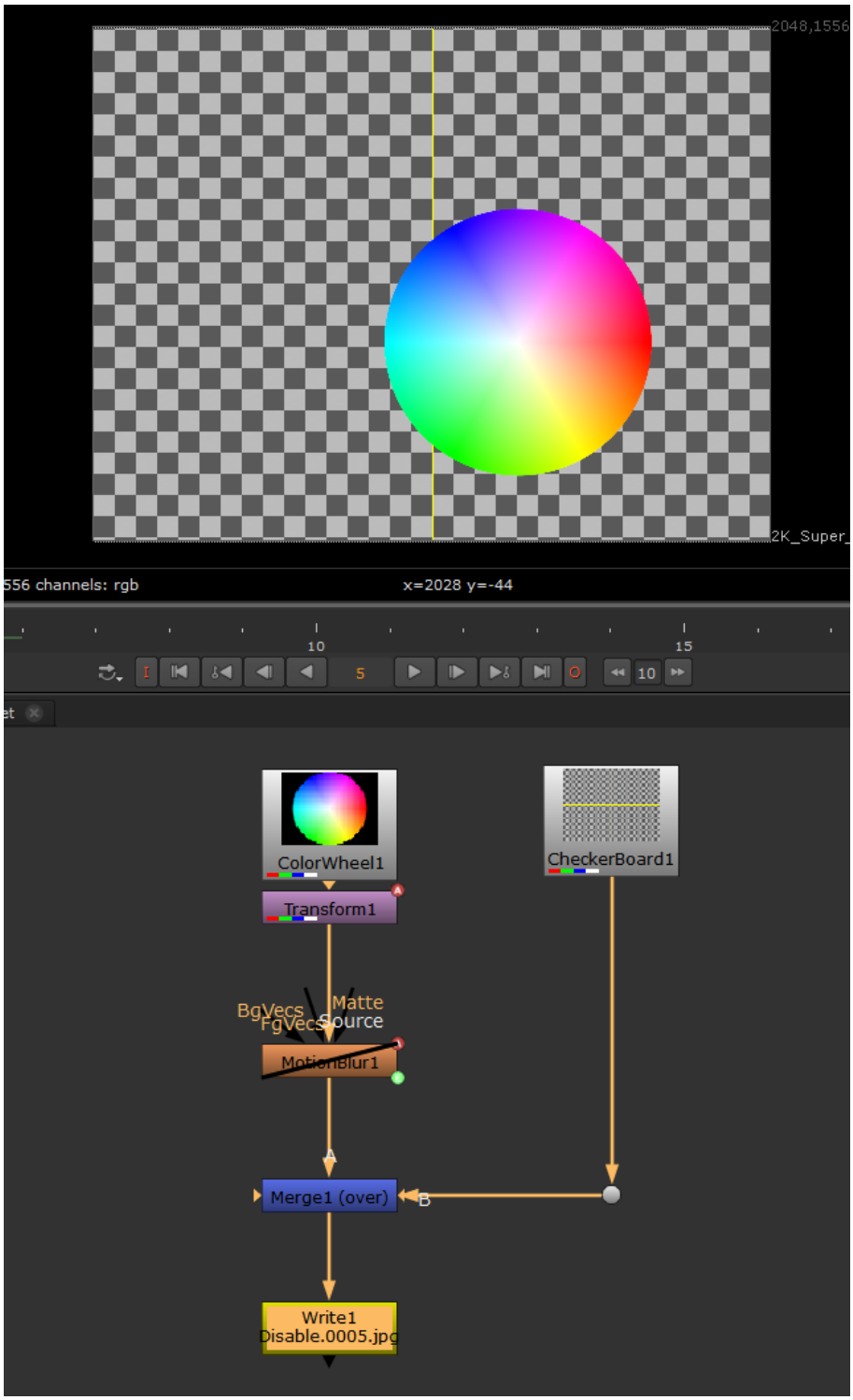
The Disable method uses the **disable** option in the node's settings, rather than using the Switch node. This is a bit cleaner with less nodes, but more difficult to turn off the expression.

Note: Using the **Disable** method means you must delete the expression to view the node processing results in the GUI, it completely bypasses the node.

1. Right-click the MotionBlur's **disable** control select **Add expression...**
2. Type **$gui** in the Expression control and click **OK**.



3. Play back the Viewer to see that the MotionBlur node processing is bypassed and does not display any blur effect.



4. The results of the render are the same as The Switch Method, however, you must remove the expression completely to re-enable the blur effect.

The Selective Variation Method

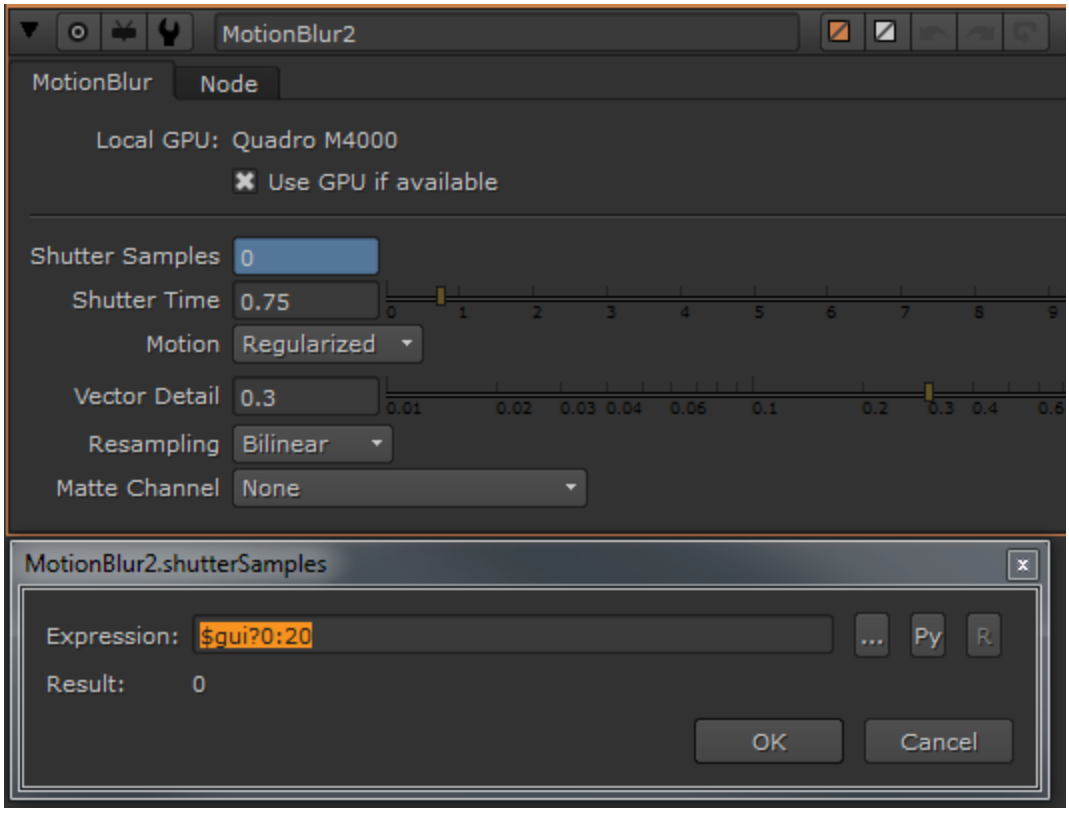
The Selective Variation Method uses the **$gui** operator expression in combination with two other values. This allows setting two independent numeric values that can be assigned to a parameter depending on which mode Nuke is in (either GUI or during rendering in non-GUI), as appose to only an on/off value result. The expression looks like this:

```
$gui?0:20
```

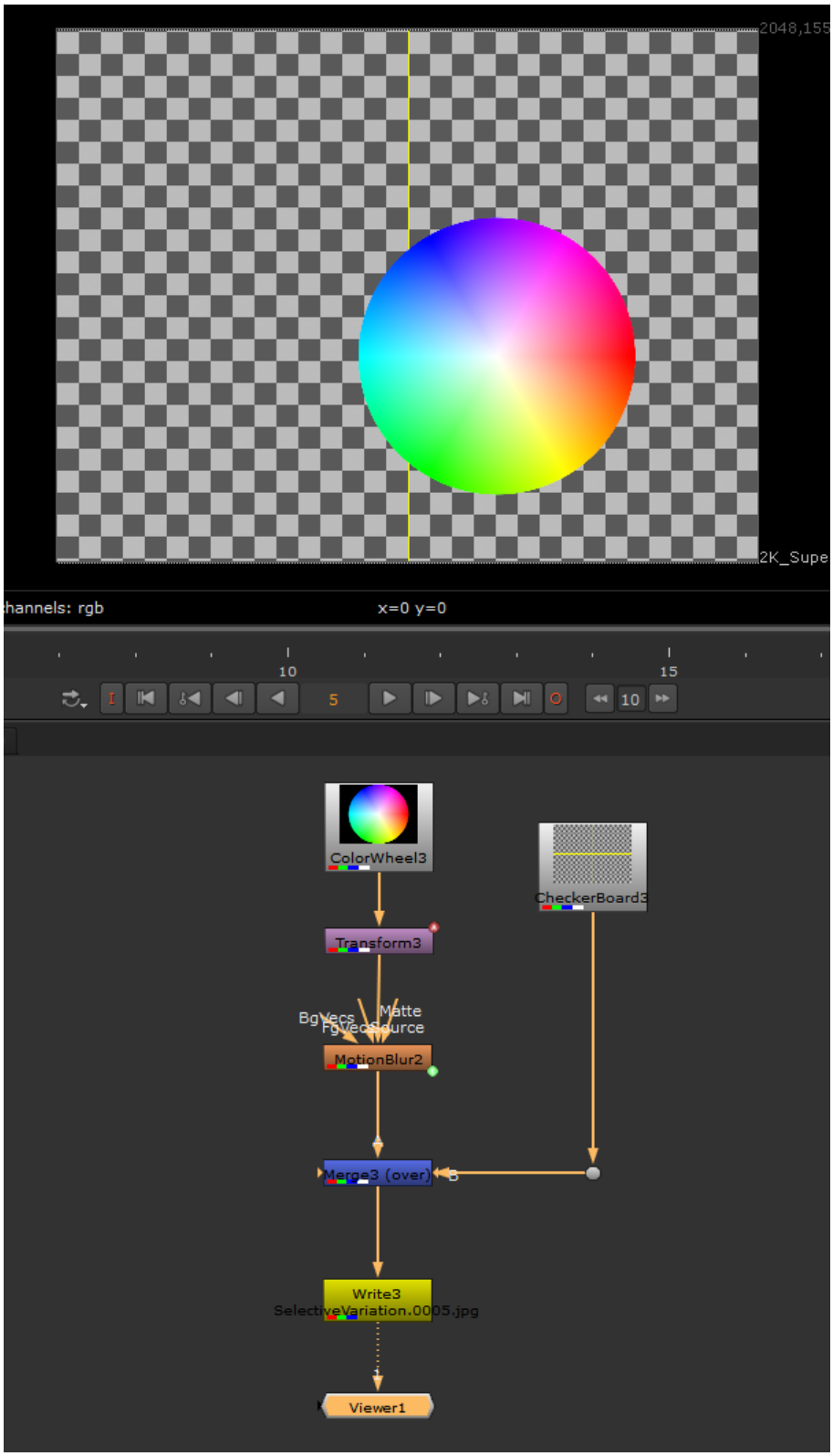
0 is the value to use in Nuke GUI mode and **20** the value to use outside GUI mode, during rendering.

Note: Using the **Selective Variation** method means you must delete the expression to view the node processing results in the GUI, it completely bypasses the node.

1. In the node (in this example, the MotionBlur node), the expression **$gui?0:20** is added and assigned to the **Shutter Samples** control. This renders 0 samples in the GUI and 20 samples during a background render.



2. If you play this through the GUI, you will see that it accesses the MotionBlur node, but the values remain at 0 until rendered using **Render in background**, when it will use 20 samples.



3. The results of the render are the same as The Switch Method.

File Name Conventions for Rendered Images

There is no parameter in the Write node to specify output format. Instead, format is indicated by a prefix or an extension when you type the file name. Here is the appropriate syntax:
`<prefix>:/<path>/<name>.<frame number variable>.<extension>`

The optional <*prefix*>: can be any valid extension. <*path*> is the full path name to the directory where you want to render. The <*frame number variable*> is usually entered as **####**, with #### indicating frame numbers padded to four digits.

You can change the padding by substituting #### with any number of hash marks. For example, two-digit padding would be **##**, three-digit would be **###**, and five-digit would be **#####**.

With these conventions in mind, suppose you want to save an image sequence called "final_comp_v01" to the TIFF16 format. Here are examples of names that work in the Write node:
`tiff16:/<path>/final_comp_v01.####.tiff`
`/<path>/final_comp_v01.####.tiff16`
`tiff16:/<path>/final_comp_v01.####`

All extensions supported by Nuke may be used for the prefix. See Appendix C: Supported File and Camera Formats for a complete list of recognized extensions.

When a prefix is used, it takes precedence over the format represented by an extension. In fact, the prefix makes the extension unnecessary.

You could, for example, enter **exr:/<path>/###** as the file name, and this would create an OpenEXR image sequence with frame numbers only, padded to three digits.

Writing Versions of Rendered Images

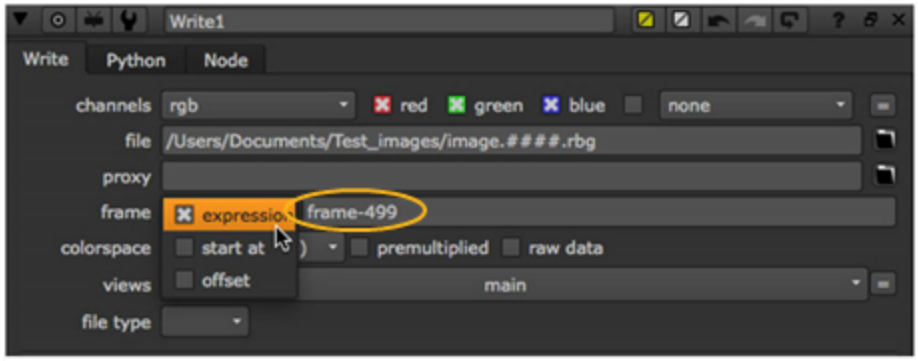
You can write out versions of a file using the **Alt**+**Up**/**Down** arrow keys. Versions must written in the following format in order for Nuke recognize them:
`<path>/final_comp_v01.####.tiff`
`<path>/final_comp_v02.####.tiff`
`<path>/final_comp_v03.####.tiff`

Changing the Numbering of Rendered Frames

By default, when you render an image sequence, Nuke assumes an exact relation between the currently processed frame and the numbering of the frame written out. Therefore, if you choose to render frame 15, the resulting file is numbered accordingly, for example image.0015.rgb. However, the **frame** parameter on the Write node lets you change this behavior. For instance, if you have a sequence that runs from image.0500.rgb to image.1000.rgb, you may want to render it so that the frame numbering in the resulting files runs from image.0001.rgb to image.0501.rgb. You can do so via expressions, specified start frames, and constant offsets. Each method is described below.

Using Expressions

1. Select **Image** > **Write** to insert a Write node into your script.
2. In the Write properties panel, click the **file** folder icon , then navigate to the directory path where you want to save the rendered image sequence. Enter a name for the image sequence.
3. Set **frame** to **expression**. Enter an expression in the field on the right. The expression changes the relation between the currently processed frame and the numbering of the frame written out. The resulting file name for the current frame is displayed on the Write node in the Node Graph.

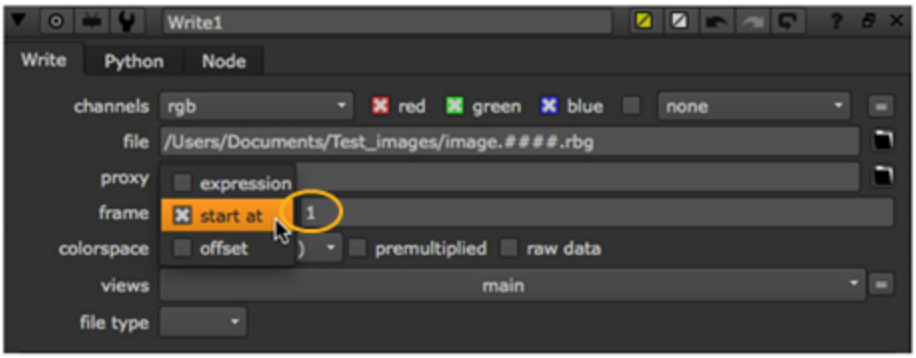


For example, if your clip begins from frame 500 and you want to name that frame image.0001.rgb rather than image.0500.rgb, you can use the expression **frame-499**. This way, 499 frames are subtracted from the current frame to get the number for the frame written out. Frame 500 is written out as image.0001.rgb, frame 501 is written out as image.0002.rgb, and so on.

Another example of an expression is **frame*2**. This expression multiplies the current frame by two to get the number of the frame that's written out. At frame 1, image.0002.rgb is written out; at frame 2, image.0004.rgb is written out; at frame 3, image.0006.rgb is written out; and so on.

Specifying a Frame Number for the First Frame in the Clip

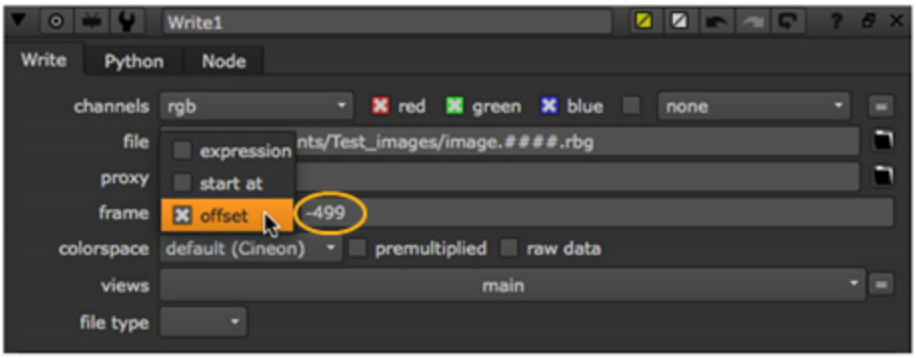
1. Select **Image** > **Write** to insert a Write node into your script.
2. In the Write properties panel, click the **file** folder icon , then navigate to the directory path where you want to save the rendered image sequence. Enter a name for the image sequence.
3. Select **start at** from the **frame** dropdown menu. Enter a start frame number in the field on the right. This specifies the frame number given to the first frame in the sequence. The numbering of the rest of the frames is offset accordingly.



For example, if your sequence begins from frame 500 and you enter **1** in the field, frame 500 is written out as image.0001.rgb, frame 501 as image.0002.rgb, and so on. Similarly, if you enter **100** in the field, frame 500 is written out as image.0100.rgb.

Offsetting All Frame Numbers by a Constant Value

1. Select **Image** > **Write** to insert a Write node into your script.
2. In the Write properties panel, click the **file** folder icon , then navigate to the directory path where you want to save the rendered image sequence. Enter a name for the image sequence.
3. From the **frame** dropdown menu, select **offset**. Enter a constant offset in the field on the right. This constant value is added to the current frame to get the number for the frame that's written out.



For example, if your clip begins from frame 500 and you want to render this first frame as image.0001.rgb rather than image.0500.rgb, you can use **-499** as the constant offset. This way, 499 is subtracted from the current frame to get the number for the frame that's written out. At frame 500, image.0001.rgb is written out; at frame 501, image.0002 is written out, and so on.

Using a Write Node to Read in the Rendered Image

You can use a Write node to both render an image and read the rendered image back in. Because reading the output of a Write node from a file is faster than calculating its output by processing the node tree upstream, this can be particularly useful on large comps. When you have finished working on a branch of the node tree, you can insert a Write node after it, render the output, and use the same Write node to read the rendered image in. If you later need to edit the nodes upstream, simply make your changes and render the Write node again to update the image being read in.

To Use a Write Node to Read in the Rendered Image

1. Render an image as described in Output (Write) Nodes. We recommend rendering the image as an **.exr**. This way, Nuke writes the hash value of the incoming node tree into the rendered file. If the node tree changes and the rendered file gets out of date, the hashes won't match and Nuke notifies you of the problem.
2. In the Write node properties, check **read file**. When this is on, Nuke ignores the upstream node tree and uses the rendered image as the output of the Write node.
3. To check whether the input file is up to date with the input tree connected to the Write node, check **check file matches input**. This only works with **.exr** files written by Nuke and when the proxy mode and down-rez are disabled. If the input file cannot be checked, Nuke displays the word **unchecked** on the Write node in the Node Graph.

4. If there is an error when loading the rendered file, select what to do from the **missing frames** dropdown menu:
   - **error** - display an error message on any missing frames.
   - **black** - replace any missing frames with black.
   - **checkerboard** - replace any missing frames with a checkerboard image.
   - **read input** - display the result of the input tree rather than the rendered file on any missing frames.

Tip: You can also use the Precomp node (**Other** > **Precomp**) to reduce portions of the node tree to pre-rendered image inputs. For more information, see Using the Precomp Node.

What is the Hash Value?

The hash value is a unique number (for example, b1c9c0aff2012a8) calculated from a node and the entire tree of nodes connected to its input. The class of the node and all the current control settings contribute to the hash value.

You can display the hash value at any point in the node tree by selecting a node in the Node Graph and pressing **I**. The hash is different at different points in the tree.

Render Farms

Nuke is supported by virtually all third-party and proprietary render-queuing software. By integrating Nuke with such a system, the render load can be distributed across all the Nuke-licensed machines on your network, whether Windows, Mac, or Linux-based.



Note: Instead of setting up a render farm, you can take advantage of the internal Frame Server, which allows you to setup external render machines to process renders faster. See Using the Frame Server on Render Machines for more information.

Your installation of Nuke may be configured to send jobs to a network render farm, which is usually made available under the **Render** menu (i.e., **Render** > **Render**). However, because this option must be customized for each studio, you should check with your system administrator for instructions on how to send a Nuke script for network rendering.

Tip: If you're attempting to force Nuke to retry an operation rather than failing, such as for license issues during rendering, you may find Nuke's exit codes helpful:
0 = success
1 = render error
100 = license failure

Organizing Scripts

As scripts grow in complexity and are worked on by several people, it becomes increasingly important to organize them in a clear and meaningful way. These pages teach you how to:

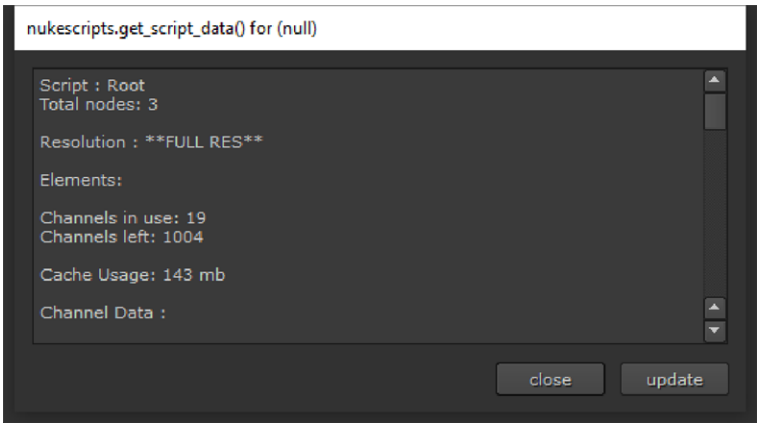
- display information about your Nuke script. See Displaying Script Information.
- find and replace all or part of file names or file paths in any node with **file** or **proxy** controls. See File Name Search and Replace.
- group nodes in the Node Graph using the Backdrop node or the Group node. See Grouping Nodes in the Node Graph.
- add notes to the Node Graph. See Adding Notes to the Node Graph.
- use the Precomp node to save a subset of the node tree as a separate **.nk** script, render the output of this saved script, and read the rendered output back into the main comp as a single image input. See Using the Precomp Node.
- use the LiveGroup node, which combines the power of groups and precomps with the utility of gizmos. See Using the LiveGroup Node.

Displaying Script Information

To display script information, such as the node count, channel count, cache usage, and whether the script is in full-resolution or proxy mode, do the following:

1. Select **File** > **Comp Info** (or press **Alt**+**I**).

   The script information window opens.



2. If you make changes to your script while the window is open, click **update** to update the information.
3. To close the information window, click **close**.

Using Visual Diagnostics

Nuke can display accurate script profile data onscreen or output it to **.csv** or **.xml** file to help you troubleshoot bottlenecks in slow scripts. When visual diagnostics are calculating, timing information is displayed in the Node Graph, and the nodes themselves are colored according to the proportion of the total processing time spent in each one. The data is then displayed in the **Profile** tab as a bar or pie chart, timeline, or as a table.

Enabling Visual Diagnostics

Nuke's visual diagnostics use Profile nodes in the node tree to gather data. You can add as many Profile nodes as required, but one strategy might be to start at the foot of the Node Graph and work your way up until CPU or memory load decreases significantly.

Note: Profile nodes add processing overhead, but the data gathered still provides an accurate picture of where processing time is spent in the script.

1. Navigate to **Image** > **Profile** to add a Profile node to the Node Graph.
2. Connect the Profile node to the point in the node tree where you want to collect profiling data.

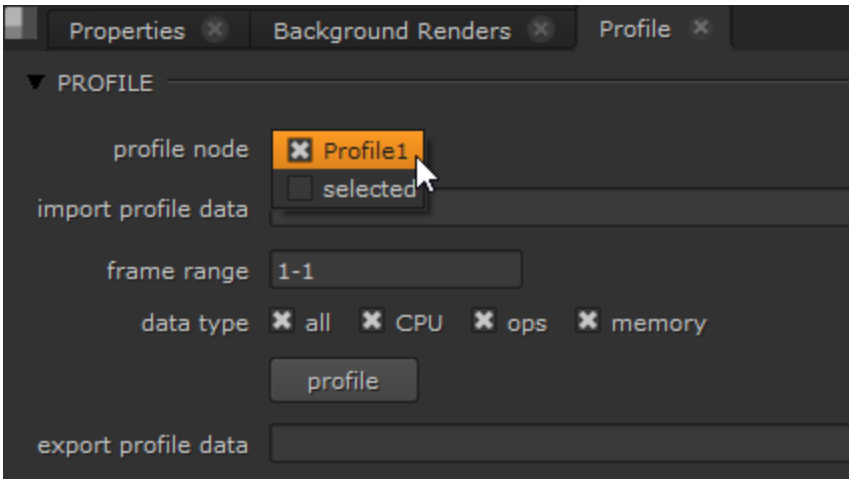
Note: You cannot connect Profile nodes directly to 3D, Particle, or Deep nodes. To profile these nodes, you'll need to render their output through a render node, such as ScanlineRender, RayRender, or a third-party renderer like PRManRender.

3. Click **open profile panel** in the node properties or click the content menu and select **Windows** > **Profile**.

   The **Profile** tab opens.
4. Click the **profile node** dropdown and then select the a Profile node in the Node Graph by name,

   OR

   Select the a Profile node in the Node Graph and then choose **selected**.



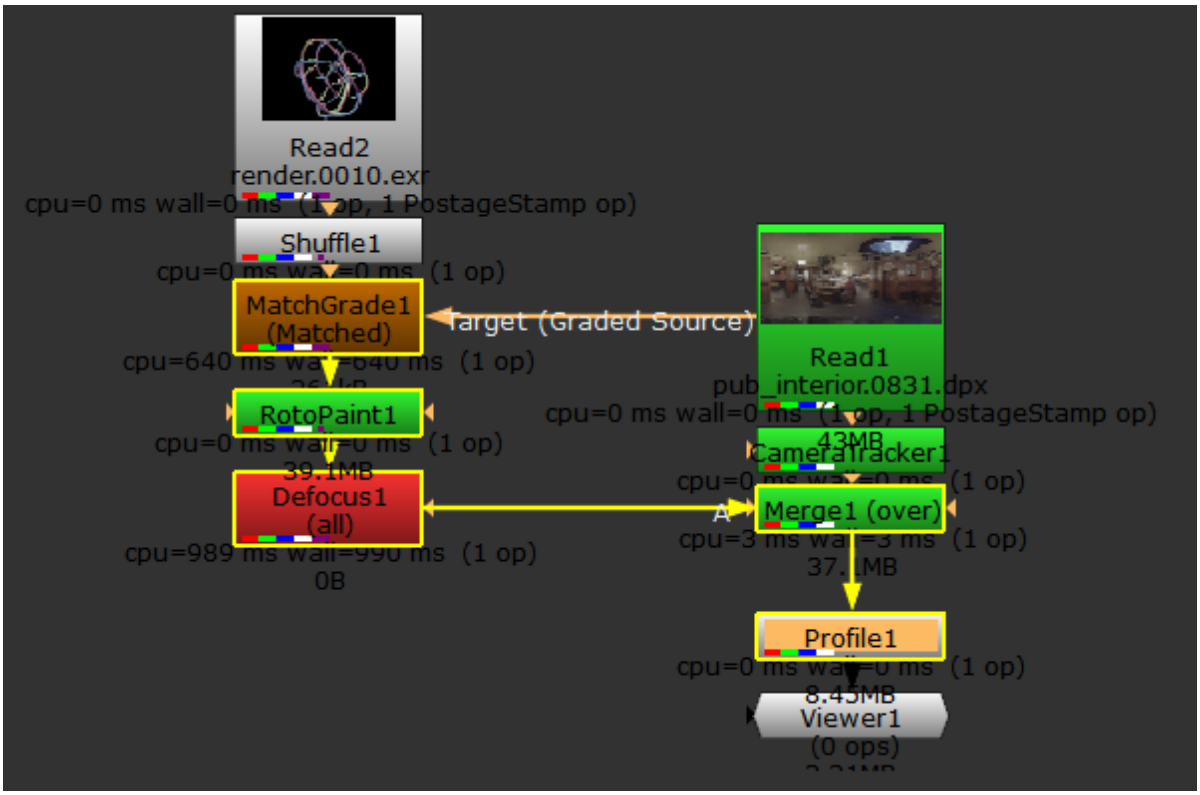
5. Set the **frame range** to collect data from and the required **data type** using the checkboxes.

Tip: The initial frame range is determined by the **Project Settings** > **frame range** control and **all** data types are collected by default. The **frame range** control accepts complex ranges, such as **1-10, 15-25** and **1-200x10**. See Defining Frame Ranges for more information.

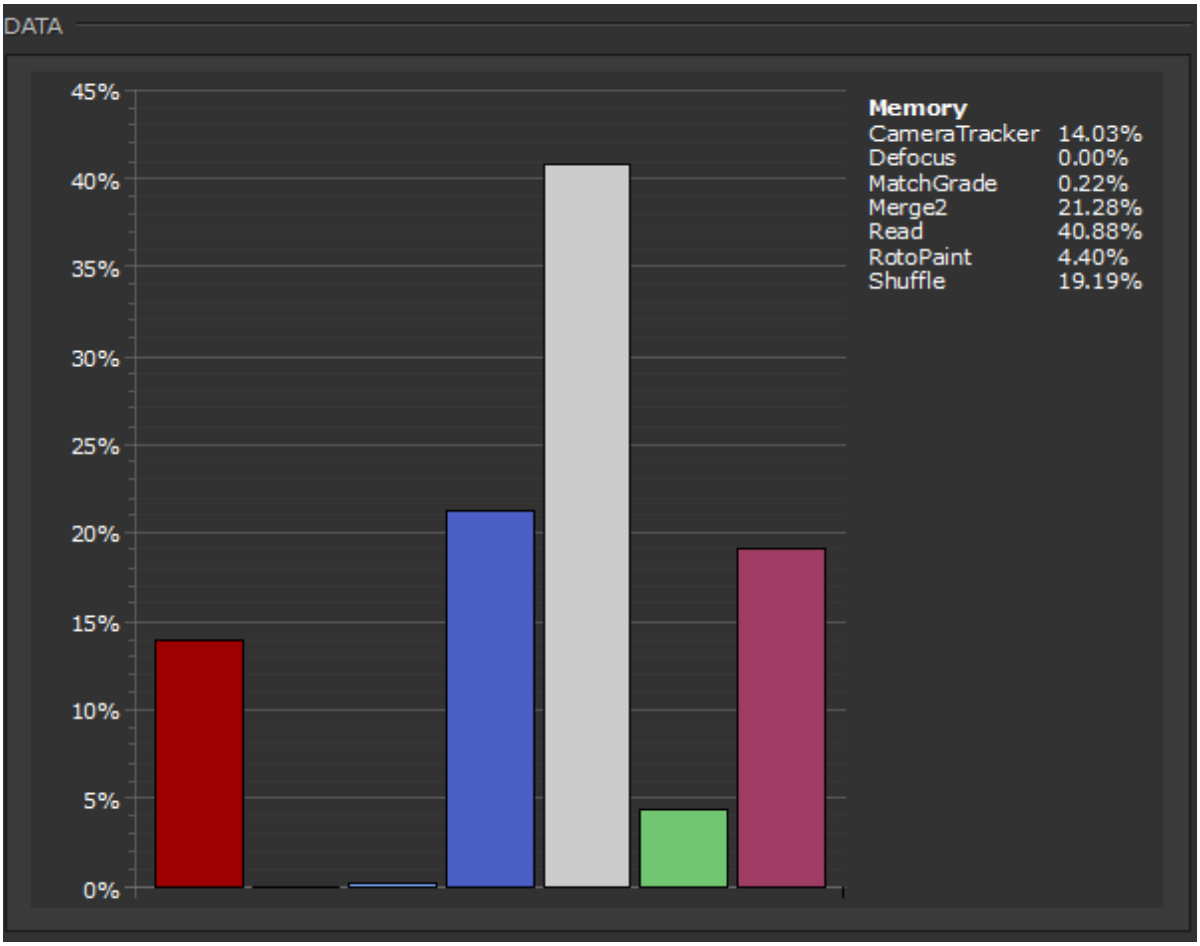
6. Click **profile** to start data collection.

Note: If you've imported existing profiling data from a **.csv** or **.xml** file, you can't profile the current script. See Exporting and Importing Profile Data for more information.

As the specified frame range is calculated, timing information is displayed and the nodes in the Node Graph are colored according to the proportion of the total processing time spent in each one.



The profiling data is stored in the **Profile** tab's **DATA** panel when complete. See Filtering Profile Data for information on how to control what profiling data is displayed.

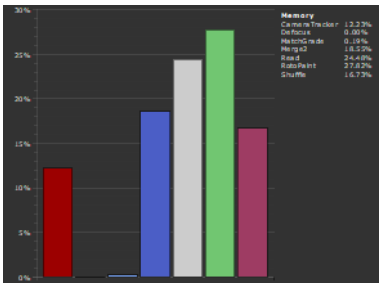


Filtering Profile Data

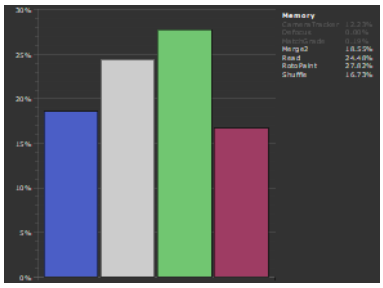
The **Profile** tab's **FILTER** dropdown allows you to output only the profiling data you need, but the full **data type** set selected in the **PROFILE** dropdown remains available if you change your mind.

The following controls determine what output is available on-screen and for export to **.csv** or **.xml** file:

- **minimum threshold** - allows you to hide display data that doesn't reach a certain minimum value. For example, setting the threshold to 15% can hide any nodes that don't use at least 15% of the available memory.



0% minimum threshold.



15% minimum threshold.

Tip: You can zoom in and out of the **DATA** pane using the + − buttons.

- **display data** - determines what data is displayed from **CPU**, **wall**, **ops**, and **memory**. The default, **all**, displays all the available profiling data:
  - **CPU** - the time that the CPU spent executing the processing code, in microseconds, aggregated over all CPU threads. For example, with multi-threaded processing this is typically much larger than the **wall** time. If the average CPU time per thread (**CPU** divided by the number of threads used) is much shorter than the **wall** time, this suggests the CPU threads have spent a lot of time not executing code and perhaps waiting on locks, which could indicate a performance problem.

Note: On Mac and Windows, the CPU time is not currently accurate. On Mac, the **CPU** value is always similar to the **wall** time.

  - **wall** - the time taken as it would be measured by a clock on the wall - the actual time you have to wait for the processing to complete. The **wall** time is also measured in microseconds.
  - **ops** - the number of operators called in the node. Operators are Nuke's building blocks that perform certain tasks. Nodes can contain one or more ops. For example, when a node needs to resize something it would use a Transform op rather than an implementation of its own to do the same thing.

    See https://learn.foundry.com/nuke/developers/17.0/ndkdevguide/intro/terminology.html#op for more information.
  - **memory** - the total amount of system memory used by the node.

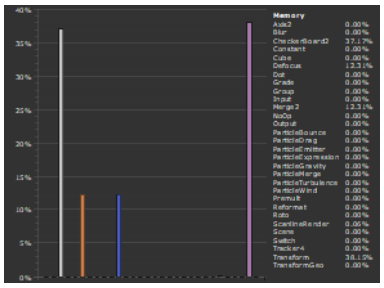
Tip: Bear in mind that some node classes also contain numerical values to differentiate between current and deprecated classes, such as Text and Text2.

- **display nodes** - for large scripts, it is impractical to display all the nodes in the Node Graph in the **Profile** tab. In **bar**, **pie** and **timeline** mode, only the top 15 results by resource usage are listed. In **table** mode, all results are listed.

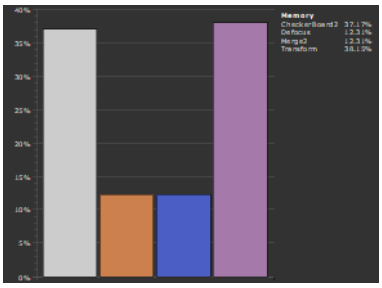
The **display nodes** control allows you to be more selective with your output:

- **all** - displays data for all the nodes in the script.
- **list** - enter the explicit node names to display in the **DATA** panel. You can type the names manually or select nodes in the Node Graph and then click **add selected**.

  You can also find nodes listed in the **FILTER** panel by clicking **select** to highlight them in the node tree.



With **display nodes** set to **all**.



With **display nodes** set to a **list**.

You can display data **by node class**, such as CameraTracker, or explicitly **by node name**, such as Read2.

Tip: You can zoom in and out of the **DATA** pane using the  buttons.

- **display frames** - allows you to view a sub-set of the profiled frames:
  - **average** - the mean average of all the frames specified by the display frames control.
  - **per frame** - allows you to step through all the frames in the **display frames** range using the  buttons.
- **chart type** - sets the form of the output produced from the profiling data:
  - **bar** or **pie** chart, and **timeline** - displays the top 15 results by resource usage.
  - **table** - displays all results in table form, a visual representation of the **.csv** or **.xml** file output.

Exporting and Importing Profile Data

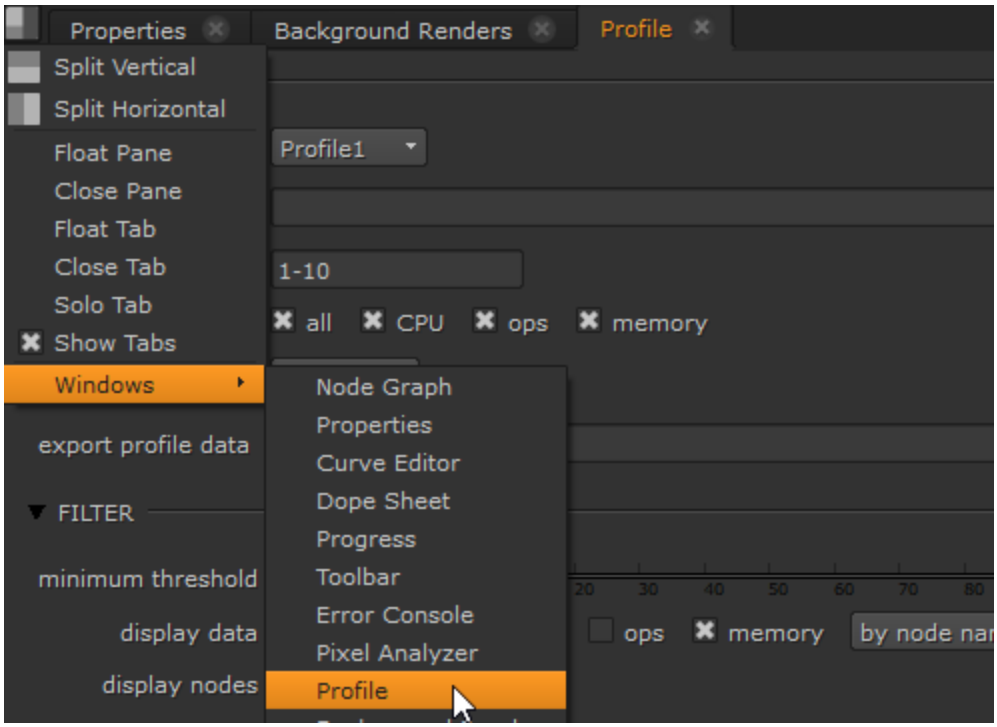
You can write the onscreen profile data to **.csv** or **.xml** file for analysis or comparison using the **export data profile** control on the **Profile** tab. You can also import existing profile data from **.csv** or **.xml** files for

analysis using the **import data profile** control on the **Profile** tab.

Exporting Profile Data

To export profile data to a file:

1. Click **open profile panel** in the Profile node's properties or click the content menu and select **Windows** > **Profile**.



2. Enter a valid file path in the **export profile data** control and then click **export**.

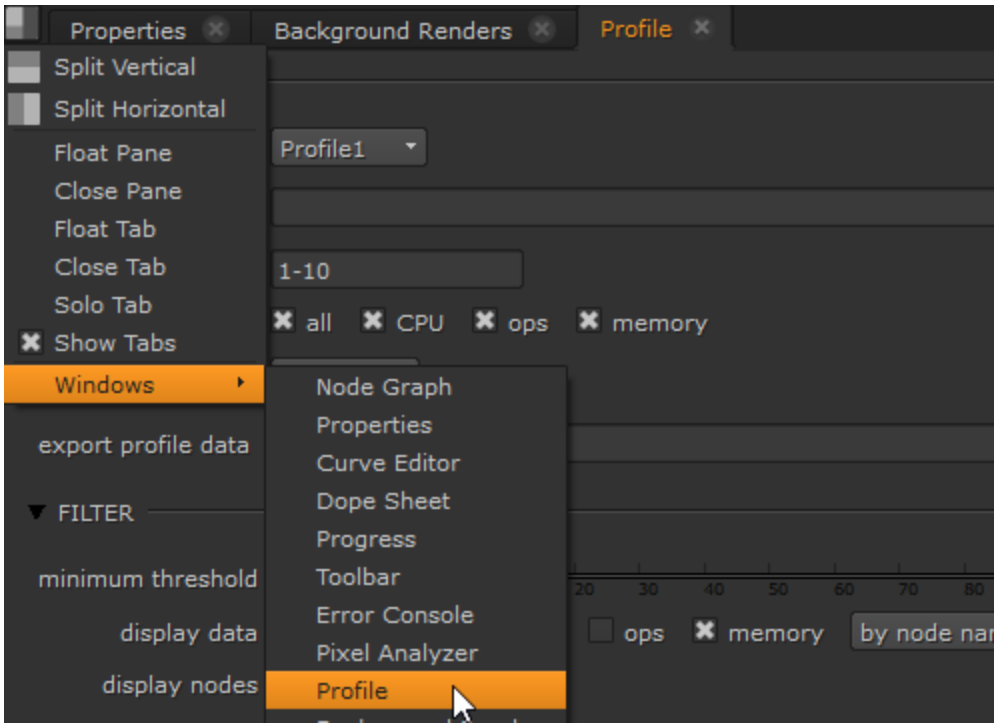
Tip: You can also browse to the export location using the button.

The profile data is saved to the specified location as a **.csv** or **.xml** file.

Importing Profile Data

To import profile data from a file:

1. Click **open profile panel** in the Profile node's properties or click the content menu and select **Windows** > **Profile**.



2. Enter a valid file path in the **import profile data** control and then click **reload**.

Tip: You can also browse to the location of the file using the button.

The profile data is imported from the specified location and displayed in the DATA panel.

Using Performance Timing

Nuke can display accurate performance timing data onscreen or output it to XML file to help you troubleshoot bottlenecks in slow scripts. When performance timing is enabled, timing information is displayed in the Node Graph, and the nodes themselves are colored according to the proportion of the total processing time spent in each one, from green (fast nodes) through to red (slow nodes).

Note: You can also access timing information for individual nodes using Python. See https://learn.foundry.com/nuke/developers/17.0/pythondevguide/performance.html for more information.

Enabling Performance Timings

You can enable profiling from the Script Editor or from the command line using the **-P** argument. You can also output the timing data to an XML file by adding the **-Pf** argument and a file name, though the output is only produced at render time.



Note: Enabling performance timings interactively or from the command line adds an extra processing overhead, but the data still provides an accurate picture of where processing time is spent in the script.

Outputting Performance Timings Onscreen

Enabling performance timings onscreen allows you to view cumulative data, as you work, on a per node basis.

From the Script Editor, enter:

```
nuke.startPerformanceTimers()
```

From the command line, enter:

```
./Nuke<version>/Nuke<version>.exe -P
```



Note: You can reset the timing data using `nuke.resetPerformanceTimers()` from the Script Editor.

When the performance timers are active, the following information is available onscreen:

- **cpu** - the time that the CPU spent executing the processing code, in microseconds, aggregated over all CPU threads. For example, with multi-threaded processing this is typically much larger than the **wall** time. If the average CPU time per thread (**cpu** divided by the number of threads used) is much shorter than the **wall** time, this suggests the CPU threads have spent a lot of time not executing code and perhaps waiting on locks, which could indicate a performance problem.

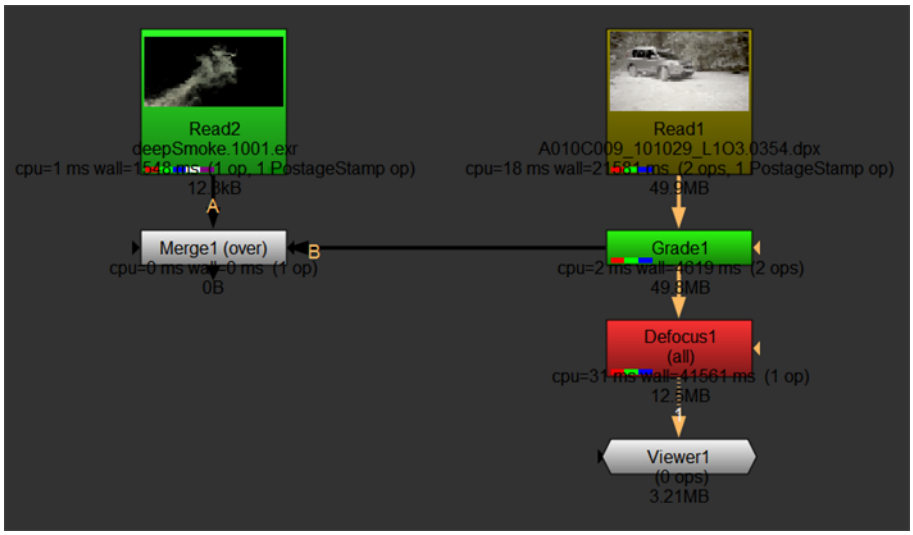


Note: On Mac and Windows, the CPU time is not currently accurate. On Mac, the **cpu** value is always similar to the **wall** time.

- **wall** - the time taken as it would be measured by a clock on the wall - the actual time you have to wait for the processing to complete. The **wall** time is also measured in microseconds.
- **ops** - the number of operators called in the node. Operators are Nuke's building blocks that perform certain tasks. Nodes can contain one or more ops. For example, when a node needs to resize something it would use a Transform op rather than an implementation of its own to do the same thing.

  See https://learn.foundry.com/nuke/developers/17.0/ndkdevguide/intro/terminology.html#op for more information.
- **memory** - the total amount of system memory used by the node.

In addition to the timing information, nodes are color-coded according to their profiling, green through red, where red is a slow node. You can see from the example script that Defocus is a slow node, whereas Merge is doing no work at all.



Note: You can stop displaying timing data using `nuke.stopPerformanceTimers()` in the Script Editor.

Writing Performance Timings to File

You can output the timing data to an XML file using the **-Pf** argument and a file name. The output is only produced at render time, either from a Write node or from a command line render.

On Windows, for example, the following command writes profile data to file when you render from a Write node in the Node Graph:

```
Nuke<version>\Nuke<version>.exe -Pf <file path and name>.xml
```

Rendering from the command line, the following command renders 10 frames from a script called **profileTest.nk** and writes the profile data to an XML file:

```
Nuke<version>\Nuke<version>.exe -x -Pf C:\temp\profileTest.xml
C:\temp\profileTest.nk 1-10
```

File Name Search and Replace

With the Search and Replace function, you can quickly replace all or part of file names or file paths in any node with **file** or **proxy** controls (for example in Read and Write nodes).

To Search for a File Name or File Path and Replace It

1. Select the node(s) where you want to replace all or part of a file name or file path.
2. Select **Edit** > **Node** > **Filename** > **Search and Replace**.

   OR

   Press **Ctrl**+**Shift**+**/** (Mac users press **Cmd**+**Shift**+**/**).
3. In the dialog that opens, enter the string you want to search for and the string you want to replace it with. Click **OK**.

Nuke searches for the string in the selected nodes and replaces it with the new string.



Note: You can also enter expressions into the Search and Replace dialog. Just remember that the search field in the dialog only takes regular expressions. Any characters that have specific meanings in regular expressions, such as **[** and **]**, need to be preceded by the **** character. For example, **[getenv HOME]** would need to be entered as **\[getenv HOME\]**.
You can also pass flags alongside the expression itself to control how the expression behaves. For example, to perform case-insensitive searches, you can enter **(?i)** in the beginning of the expression or after one or more whitespace characters.

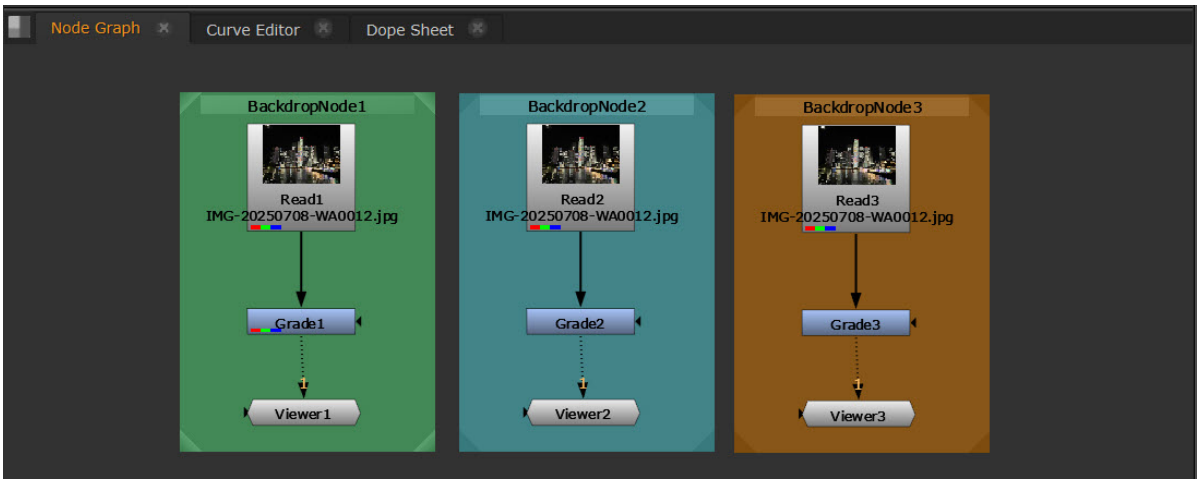
Grouping Nodes in the Node Graph

You can group nodes in the Node Graph using the Backdrop node or the Group node. The Backdrop node adds a background box behind the nodes, separating the nodes visually from the rest of the node tree. A Group node, instead, combines a set of nodes into a single node, acting as a nesting container for those nodes. With Group View you can also view and edit the grouped nodes directly in the main node graph.

Grouping Nodes with the Backdrop Node

You can use the Backdrop node to visually group nodes in the Node Graph. Inserting a Backdrop node creates a box behind the nodes. When you move the box, all the nodes that overlap the box are moved, too. By inserting several backdrop nodes, you can group the nodes in your node tree onto boxes of different colors and titles. This makes it easier to find a particular node in a large node tree, for example.

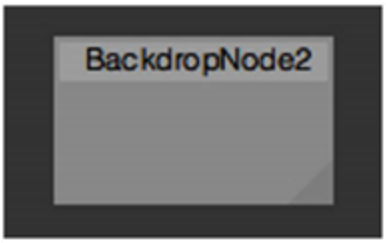
You can also use the **Z Order** control in the Properties panel to layer-up Backdrop nodes. Backdrops with lower **Z Order** values appear underneath those with a higher value.



Nodes grouped with Backdrop nodes.

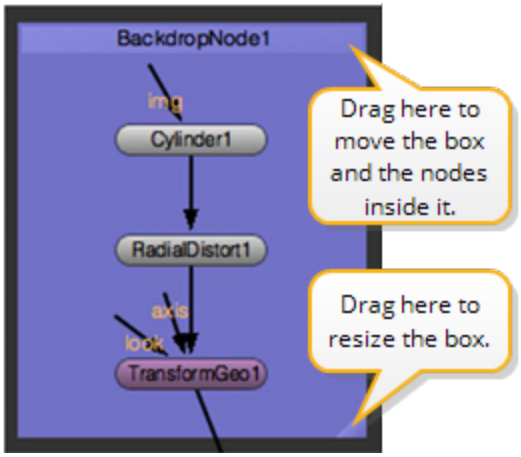
To Group Nodes with a Backdrop Node

1. Select **Other** > **Backdrop**. A Backdrop node box appears in the Node Graph.

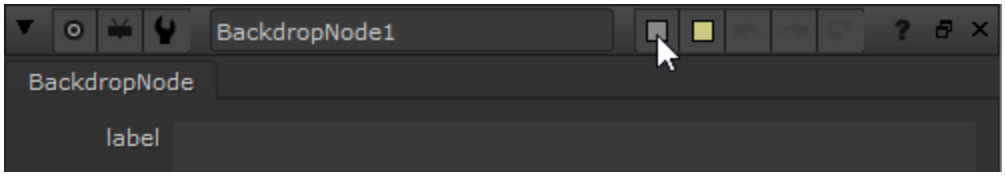


2. Drag the triangle in the lower right corner of the box to resize the box as necessary.

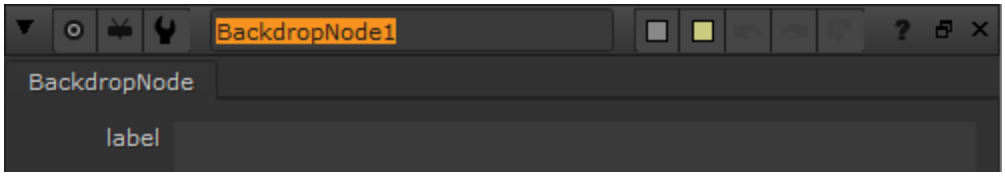
3. Click on the box title bar and drag it to move the box behind the nodes you want to group together. If there are any nodes on the box, they move together with the box.



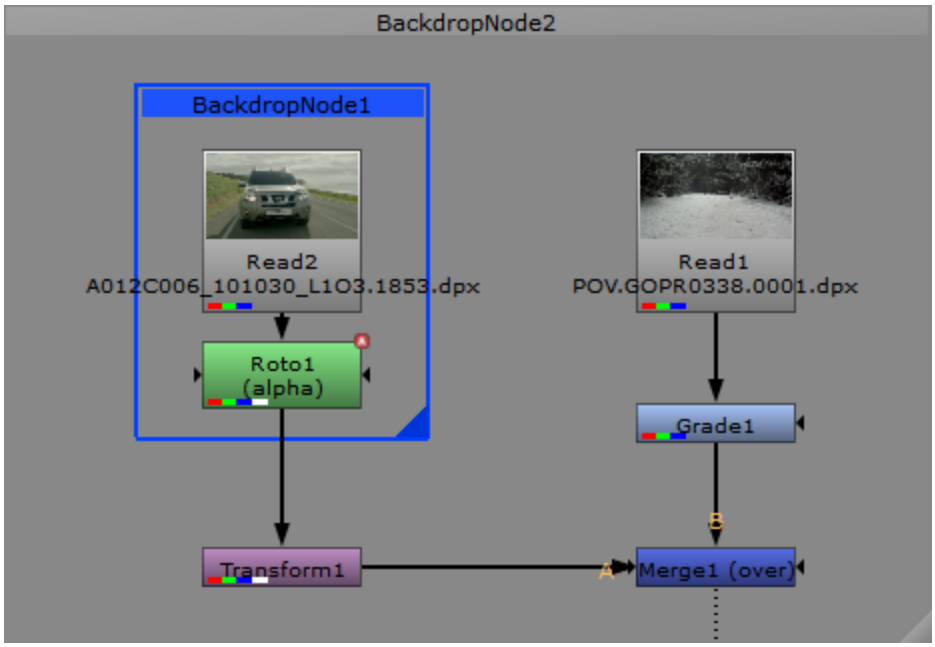
4. To change the color of the box, open the Backdrop node's controls by double-clicking on the title bar, then click the left color button and pick a new color with the color picker that appears.



5. To change the title of the box, enter a new name for the Backdrop node in the node's controls.



6. To layer-up Backdrop nodes, enter a value in the **Z Order** control in the Properties panel. Backdrops with lower **Z Order** values appear underneath those with a higher value.



Tip: In the example, Backdrop1 is set to **appearance** > **Border** in the **Properties** panel. This control removes the fill color and only draws the border of the backdrop, which can help with eye strain in low light environments.

7. If you later want to remove both the box and the nodes inside it, click on the title bar and press **Delete**. **Ctrl/Cmd**+clicking on the title bar and pressing **Delete** only removes the box and leaves the nodes untouched. To remove the nodes and leave the box untouched, click on the triangle in the lower right corner of the box and press **Delete**.

Tip: You may want to use Backdrop nodes as jump-to points throughout a project. For more information, see Navigating Inside the Node Graph.

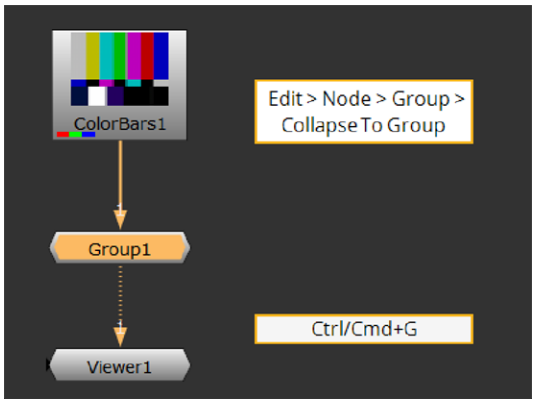
Grouping Nodes with the Group Node

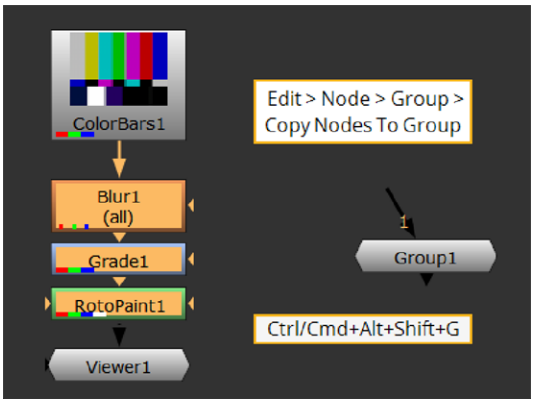
You can use the Group node to nest multiple nodes inside a single node.

To Group Nodes with a Group Node

1. Select all the nodes you want to nest inside the Group node.
2. If you want to replace the original nodes with the Group node, right-click and select **Edit** > **Node** > **Group** > **Collapse To Group** (or press **Ctrl/Cmd**+**G** on the Node Graph).

If you want to keep the original nodes in the layout in addition to the Group node, right-click and select **Edit** > **Node** > **Group** > **Copy Nodes To Group** (or press **Ctrl/Cmd**+**Alt**+**Shift**+**G** on the Node Graph).





The selected nodes are nested into a group.

Tip: As an alternative to **Edit** > **Node** > **Group** > **Collapse to Group**, you can also select **Other** > **Group** from the Toolbar or the Node Graph right-click menu.

To Ungroup Nodes

1. Select the Group node in the Node Graph.
2. Select **Edit** > **Node** > **Group** > **Expand Group** (or press **Ctrl/Cmd**+**Alt**+**G**).
   The Group node is replaced with the nodes that were nested inside it.

Viewing Grouped Nodes with Group View

With **Group View**, you no longer need to open separate tabs to see what's inside each Group. Group View enables you to easily see inside and edit the whole node graph inside a Group in the context of the main graph. This makes it easier to organise your scripts, as well as providing a way of defining variables.

Using Group View, you can for example:

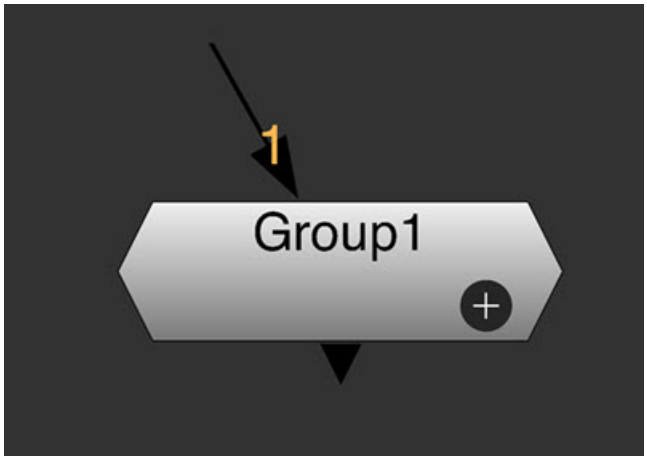
- Add and remove nodes
- Perform any Edit operation
- Attach Viewers to nodes

This applies to Group, VariableGroup and LiveGroup nodes, and Gizmos if the environment variable for editing Gizmos is in use.

Opening and closing Group View

• Select the node

To quickly open and close Group View, you can click on the **+** directly on the node.



After selecting the Group node, the hotkey '**V**' will also toggle open and close the Group View.

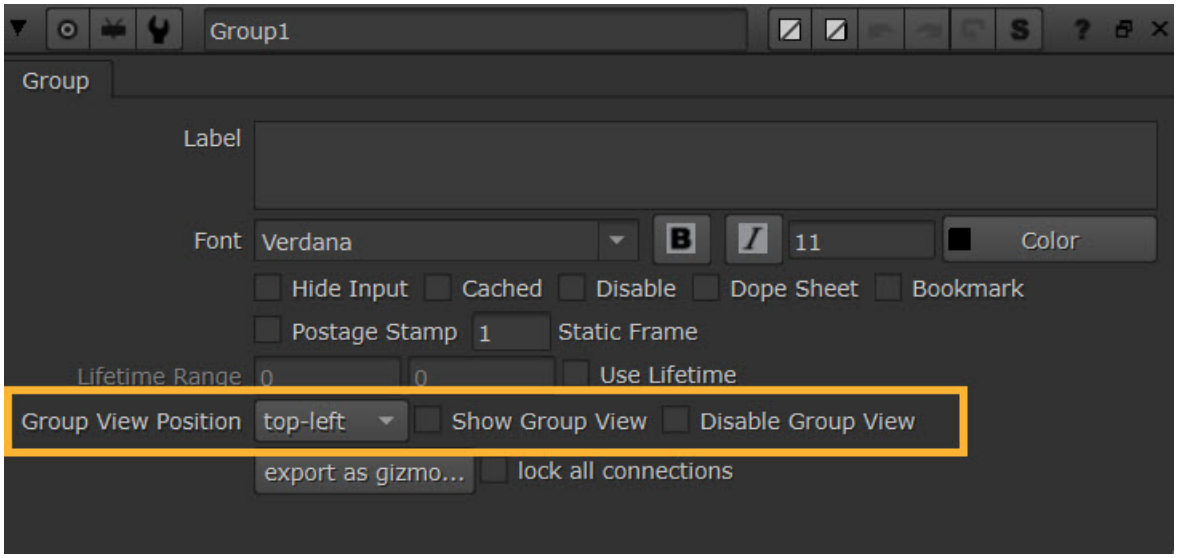
Note: The '**V**' hotkey can still be used for other operations when the Group node is not selected.

• Via Group node Properties

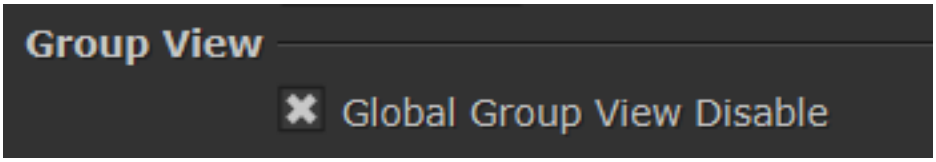
In Group Node Properties, select the checkbox **Show Group View**, which opens and closes the Group View.

Select the option **Group View Position** to change the position of the Group View to either the right or left of the node, depending on how you want to lay out your scripts. The Group View defaults to the left hand side. For customized Group nodes, this will appear in the Node tab.

You can also choose to disable this functionality entirely by selecting **Disable Group View**.

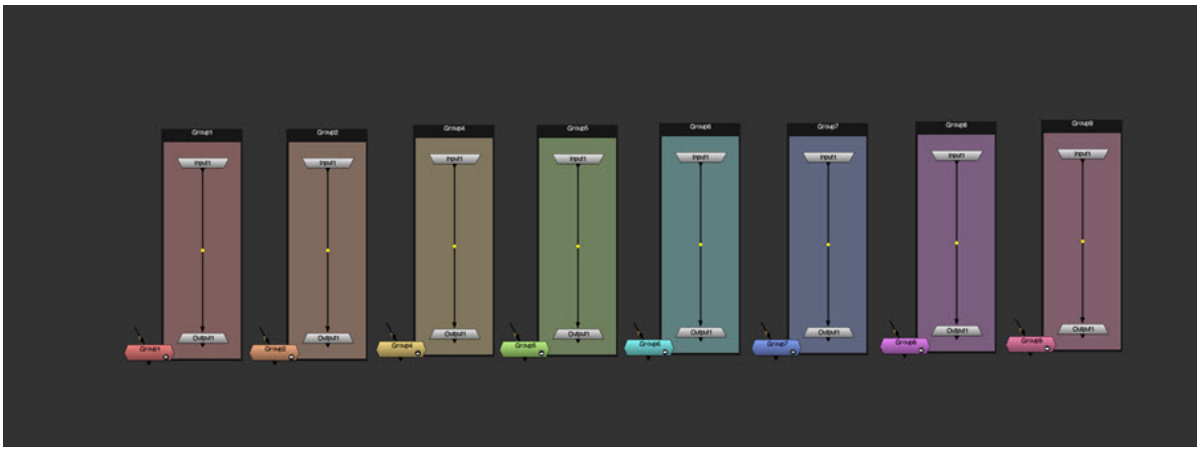


If you wish to disable Group View at a **global** level, there is also an option in the Preferences panel. Go to **Preferences** > **Node Graph** > **Group View** > **Global Group View Disable**. When this is checked, the ability to show group views will be disabled and all currently visible group views will become invisible.



Using Group View in your Node Graph

- You can open the Properties, copy and paste, search, and perform many other operations with nodes inside Groups directly from the Group View, without having to go into the Group in a separate tab.
- The Group View will dynamically resize, to fit the nodes contained in the Group. You can also view Groups within Groups.
- The color of the Group View is derived from the color of the Group Node.



Creating a node in Group View

Creating a node in a group view requires clicking into the Group View, and then using standard node creation procedures such as tab menu, hotkeys or the toolbar menu.

Moving nodes into a Group View

An existing node cannot be moved into a Group View by drag and drop, but can be cut and paste. First select the node(s), **Edit** > **Cut**, click into the group view and **Edit** > **Paste**.

Drag and drop into Group View is supported for files.

Python support for Group View

To find out the group of the last hit Group View, use nuke.lastHitGroup(). This can be used to support Group View when creating custom menu items. For example:

Python
```
with nuke.lastHitGroup():
nuke.createNode("Blur")
```

The above code would create a Blur within the group of the last hit Group View.

Tip: Grouped nodes are especially useful for Multishot workflows, in particular with Nuke's Graph Scope Variables feature. When organizing and managing multiple scopes, being able to see inside your Groups from the main node graph can help immensely.

Tip: With the ability to have multiple shots in the same node graph, setting up a Group per shot is a nice way to visualize and organize your script.



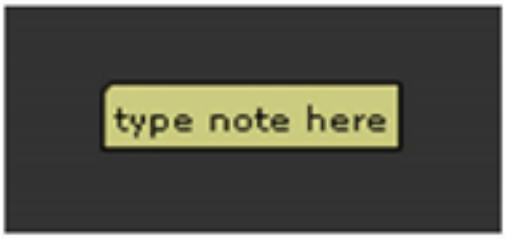
Tip: See also Using the VariableGroup Node.

Adding Notes to the Node Graph

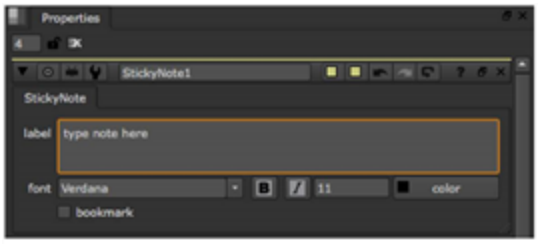
Using the StickyNote node, you can add notes to the Node Graph. The notes can be any text or HTML mark-up. Usually, they are made as annotations to the elements in the node tree.

To Add a Note to the Node Graph

1. Click on the part of the Node Graph where you want to add a note.
2. Select **Other** > **StickyNote**. A note box appears in the Node Graph.



3. In the StickyNote controls, enter your note in the **label** field. If you like, you can use HTML mark-up. For example,
   - to have a note appear in bold, you can use **my note.** This would appear as **my note**.
   - to have a note appear in italics, you can use **my note**. This would appear as *my note*.
   - to add an icon to your note, you can use ****. This adds the Nuke color wheel icon. You can also use your own icons in the same way as long as you save them in your plug-in path directory. Most common image formats work, but we recommend using **.png** files.



Using the Precomp Node

The Precomp node is like a Group node, but its content is stored in an independent **.nk** file. This allows you to save a subset of the node tree as a separate **.nk** script, render the output of this saved script, and read the rendered output back into the main comp as a single image input.

Note: Precomp nodes are unsupported in Nuke Assist. You cannot create a Precomp in Nuke Assist; however, if one already exists in the script, you can view, but not modify its output. To indicate this, Precomp nodes are outlined in red in the Node Graph, and their controls are grayed out. See Nuke Products for more information.

Precomp nodes can be useful in at least two ways. Firstly, they can be used to reduce portions of the node tree to pre-rendered image inputs. This speeds up render time, as Nuke only has to process the single image input instead of all the nodes that were used to create it. Because the original nodes are saved in a separate **.nk** script, you also maintain access to them and can adjust them later if necessary.

Secondly, Precomp nodes enable a collaborative workflow. While one artist works on the main comp, others can work on the sections that have been exported using the Precomp node. These sections can be edited, versioned, and managed independent of the main comp. For example, say you have a comp that involves a complex, multi-layered CG render. A 3D artist can produce this as a separate script that the compositor finishing the shot then reads in using a Precomp node. This way, the 3D artist can modify and re-render the CG element portion of the comp without having to share the main comp with the compositor.

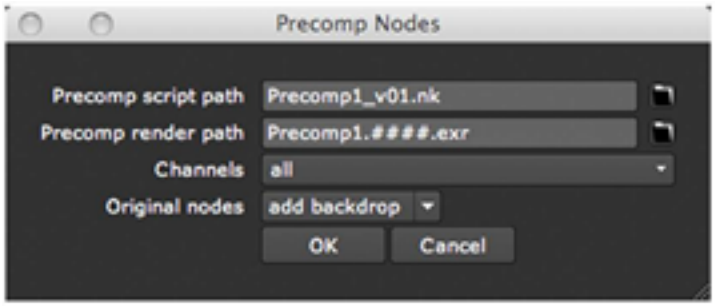
Creating Precomp Nodes

Whether you want to use the Precomp node to speed up rendering or to enable a collaborative workflow, the process of creating the Precomp is the same.

To Create a Precomp Node

1. Select the nodes you want to include in the separate precomp script. If you select a Group node, the nodes nested inside it are copied into the precomp script.
2. Select **Other** > **Precomp** (or press **Ctrl/Cmd**+**Shift**+**P** on the Node Graph).

   The **Precomp Nodes** dialog opens.



Note: If you don't select any nodes when you create a Precomp node, the Precomp Nodes dialog is not displayed and the node is left blank. Using the **file** parameter in the Precomp controls, you can then browse to an existing script to load into the Precomp node.
If you have several Write nodes in the existing script and want to control which of them is used for the output of the Precomp node, you can select **Other** > **Output** to insert an Output node after the Write node you want to use.
If the Precomp node cannot find an Output node in the script, it looks for a Write node and sets **Output node** in the Precomp controls to the name of that node. The **Output node** field can also be used later to override what is set as the Output node in the precomped script. To do so, make sure you check **enable**. This check box allows you to toggle between the output node chosen by default and the output node you specified in the **Output node** field.

3. Click the file browser icon  next to **Precomp script path**, and browse to the directory where you want to save the precomp **.nk** script. After the directory path, enter a name for the precomp script, for example **Precomp1_v01.nk**. By default, the precomp script is saved next to the main script. If the main script has not been saved, the precomp script is saved in the current directory.
4. Click the file browser icon  next to **Precomp render path**, and browse to the directory where you want to save the rendered output of the precomped nodes. After the directory path, enter a name for the rendered image, for example **Precomp1_####.exr**. If you like, you can also use version numbers in the name, for example **Precomp1_v01_####.exr**.

Warning: We recommend rendering the image as an **.exr** because that way Nuke can write the hash value of the incoming node tree into the rendered file. If the precomp script changes so that the hashes won't match, Nuke can then notify you, and you can update the resulting image. If you use a file format other than **.exr**, you do not get this notification and the rendered file is likely to become out of date. For more information on hash values, see About the Hash Value.

5. From the **Channels** dropdown menu, select the channels you want to include in the rendered output of the precomped nodes.
   If you later need to adjust this selection, you can do so by setting the channels on the appropriate Write node (by default, Write1) in the precomp **.nk** script.
6. From the **Original nodes** dropdown menu, select:
   - **add backdrop** to create a backdrop behind the precomped nodes in the Node Graph.
   - **delete** to delete the precomped nodes.
   - **no change** to do nothing to the precomped nodes.
7. Click **OK**.

Nuke saves the selected nodes in the **.nk** script specified. This script also includes input and output nodes, a Write node, and the current project settings.

In the Node Graph, the selected nodes are replaced with the Precomp node. Nuke opens the properties of the Precomp node.

About the Hash Value

The hash value is a unique number (for example, b1c9c0aff2012a8) calculated from a node and the entire tree of nodes connected to its input. The class of the node and all the current control settings contribute to the hash value.

You can display the hash value at any point in the node tree by selecting a node in the Node Graph and pressing **I**. The hash can be different at different points in the tree.

Precomp Nodes and Project Settings

When you create a Precomp node, the precomp **.nk** script gets its project settings from the main script. The main script's project settings are also used whenever the precomp script is loaded into the main script. Therefore, if you open the precomp script in a separate instance of Nuke and change its project settings so that they no longer match the main script's settings, your changes do NOT have an effect when the precomp script is loaded into the main script. If you want to change the project settings, you should always do so in the main script rather than the precomp script.

Using a Precomp Node to Speed-up Rendering

If you want to use a Precomp node to speed up rendering, you need to have the Precomp node read in the output of the precomp script.

To Render a Precomp Node

1. Create a Precomp node. For more information on how to do this, see Creating Precomp Nodes.
2. In the Precomp node controls, click **Render**. Enter a frame range (for example, **1-100**) in the dialog that opens and click **OK**.

Nuke renders the output of the precomp script and saves the resulting image in the **Precomp render path** directory you specified when you created the Precomp node.

If you have activated the proxy mode and have specified a proxy file name in the output node, the rendered image is a proxy image.

3. If the render finishes successfully, **read file for output** is automatically turned on in the Precomp properties. When this is checked, the Precomp node turns green and reads in the rendered precomp image rather than calculating the output of the precomp script. The word **(Read)** and the name of the image read in are also displayed on the Precomp node in the Node Graph to indicate this.
4. In case there is an error opening the rendered output of the precomped nodes, select what to do from the **missing frames** dropdown menu in the Precomp properties:
   - **error** - display an error message on any missing frames.
   - **black** - replace any missing frames with black.
   - **checkerboard** - replace any missing frames with a checkerboard image.
   - **read input** - display the result of the input tree rather than the rendered file on any missing frames.

When several Precomp nodes are used like this to replace sections in a large, complex node tree, the render times may become significantly faster.



Note: When rendering the output of the precomp script, Nuke automatically selects the Write node to use by first looking for an Output node. If it can't find any Output nodes, it tries to find a Write node, and sets **output node** in the Precomp node controls to the name of the Write node. If it can't find any Output or Write nodes, it produces an error.
At any point, the Precomp node properties can be used to override what is set as the Output node in the precomped script. To do so, open the Precomp properties and the **advanced** controls. In the **output node** field, enter the name of the Write node whose output you'd like to use. Make sure you also check **enable**. This check box allows you to toggle between the output node chosen by default and the output node you specified in the **output node** field.

Tip: You can also use Write nodes in a similar manner and have their output read in from the rendered file rather than calculated using the nodes upstream. For more information, see Using a Write Node to Read in the Rendered Image.

Precomp Revisions

The following describes how to open, edit, and reload a precomp script.

To View and Edit a Precomp Script

1. In the Precomp node properties, click **Open**.

   This starts a new Nuke session and loads the precomp script.
2. Edit the precomp script as necessary.
3. If you are using version numbers in the Write node file name, select the Write node in the Node Graph and choose **Edit** > **Node** > **Filename** > **VersionUp** from the menu bar. This changes the version number in the file that's rendered, for example, from Precomp1_v01_####.exr to Precomp1_v02_####.exr.
4. Render the Write node. Note that if the proxy mode is on and you have entered a proxy file name for the output node, Nuke renders a proxy image.
5. Select **File** > **Save New Comp Version** (or **File** > **Save Comp As**) to save the script with a new version number, for example, Precomp1_v02.nk instead of Precomp1_v01.nk.

To Reload a Revised Precomp Script

1. Load the main comp.
2. Make sure the filename in the Precomp node's **file** field matches the current name of the precomp script.

   If the precomp script name has been versioned up, you can simply select the Precomp node and choose **Edit** > **Node** > **Filename** > **VersionUp** from the menu bar. This changes the name of the script that is read in, for example, from Precomp1_v01.nk to Precomp1_v02.nk.

   If the precomp script has been saved with a different name or you want to use a different script as the precomp script, edit the filename in the **file** field.
3. In the Precomp properties, click **Reload**.

Collaborative Workflow Example

In this example, a compositor is responsible for the main comp that is delivered to the client. A lighting TD is responsible for providing a multi-layered CG render for use in the main comp. The following describes how the compositor and lighting TD could use the Precomp node to enable a collaborative workflow.

To Enable a Collaborative Workflow

1. The compositor starts building the main comp.
2. The lighting TD does a render of the CG element and a comp of the layers, saving the script as cg_v01.nk.



Warning: Note that we recommend rendering precomp images as **.exr** files. This way, Nuke writes the hash value of the incoming node tree into the rendered file. If the precomp script changes so that the hashes won't match, Nuke can notify the user who can then update the resulting image. If you use a file format other than **.exr**, you don't get this notification and the rendered file is likely to become out of date.
For more information on hash values, see Creating Precomp Nodes.

3. The compositor creates a Precomp node reading from the file cg_v01.nk, and continues working on the main comp.
4. The lighting TD continues to revise the CG render and the comp, versioning up the Write node and the precomp script .nk name. (See Precomp Revisions.)

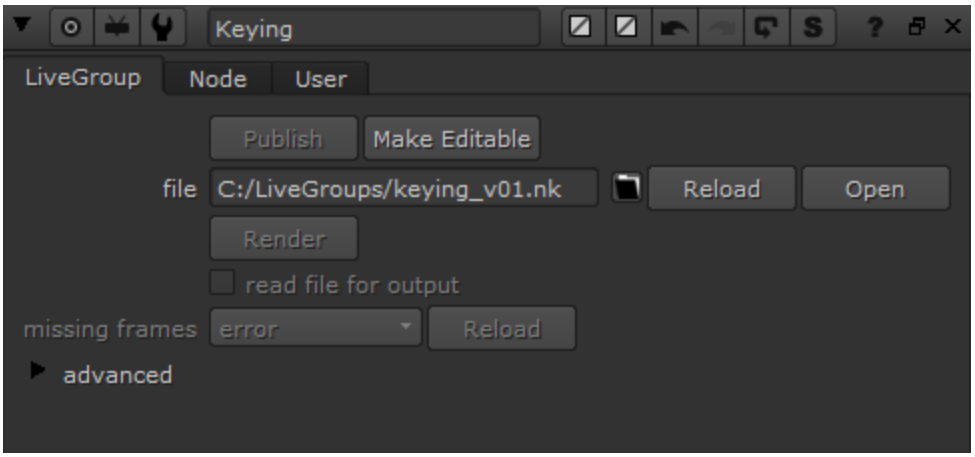
   When better results are achieved, the lighting TD notifies the compositor of the new and improved version of the precomp script.
5. The compositor versions up the Precomp node to the new, improved version and reloads the precomp script. (See Precomp Revisions.)

Using the LiveGroup Node

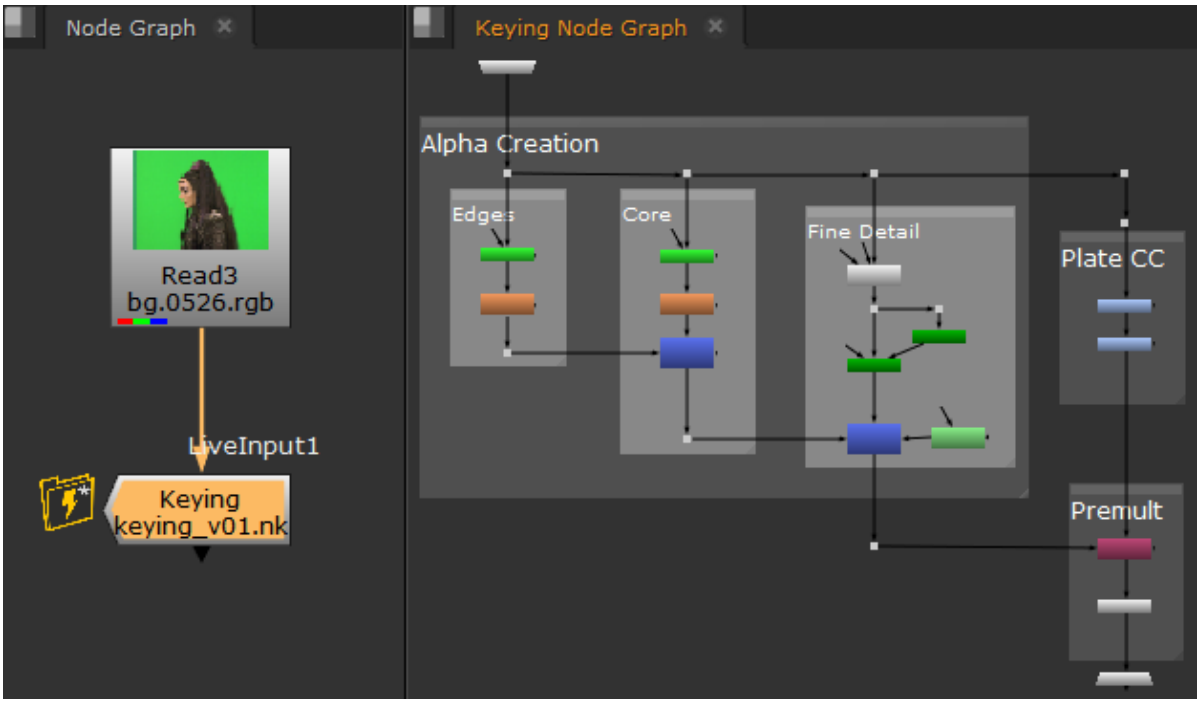
LiveGroups are a type of container node that can be used in conjunction with LiveInput nodes so that multiple artists can work on different parts of the same shot as separate scripts, without the need for rendering. LiveGroups also offer all the functionality of Precomps, Groups, and Gizmos combining all the functionality that they lack individually.

Function	Group/Gizmo	Precomp	LiveGroup
Open source file		●	●
Version up and down without recreating		●	●
Expose controls from within the node	●		●
View internal nodes in the same Nuke session	●		●
Accepts inputs in the same Nuke session	●		●

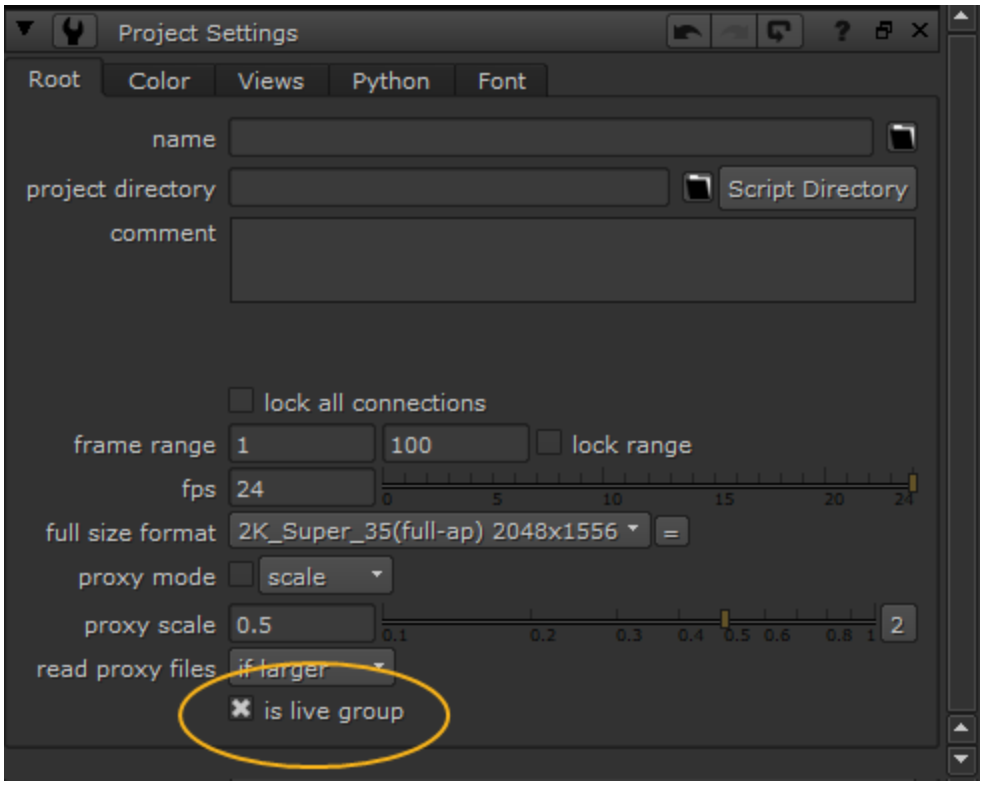
Just like Precomps, LiveGroups can store independent, open source **.nk** files, allowing you to save a subset of the node tree as a separate **.nk** script, render the output of this saved script, and read the rendered output back into the source comp as a single image input.



You can also use LiveGroups like Group nodes to nest multiple nodes inside a single node. The original nodes are replaced with the LiveGroup node. When you create a LiveGroup node, its internal structure is shown in a separate Node Graph tab.

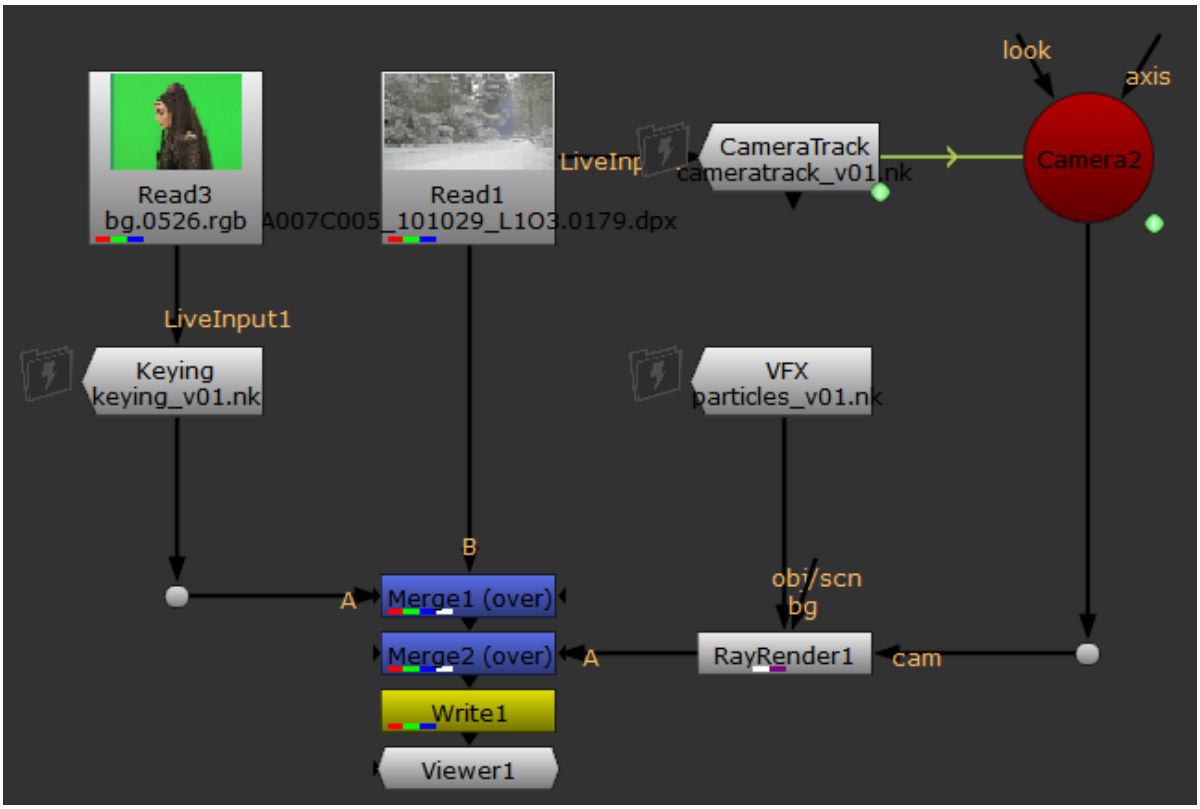


LiveGroups can also be used as shared toolsets, similar to Gizmos, but without having to export them. You can import a LiveGroup script into any other Nuke script, as long as **Project Settings** > **is live group** is enabled and the script contains an Output node.



Collaborative Scripts

The LiveGroup node is different to Precomps and Groups in that it allows multiple artists to work on the same script at the same time, drawing work from disparate scripts into the source script. A simple example is shown below, where the **Keying**, **CameraTrack**, and **VFX** LiveGroups contain the node trees for those operations in separate scripts.



In this example, the artists can continually improve the expression linked Camera and green-screen key while the 3D artist works on the shot. See Collaborative Workflow Examples for more information.

Collaborative Workflow Examples

LiveGroups allow several artists to work on the same source script at the same time, without having to render the separate components like Precomps. You can set up a source script before beginning or start work on individual sessions and then import them into another script, depending on how you like to work.

Creating a Source Script

As a Supervisor, you might want to set up a source script to hold the work of various artists working on a shot. In this example, we'll assume that there are Keying, Tracking, and 3D artists working simultaneously.

To create a source script:

1. Create a Nuke script with the required **Project Settings** and read in the required assets.

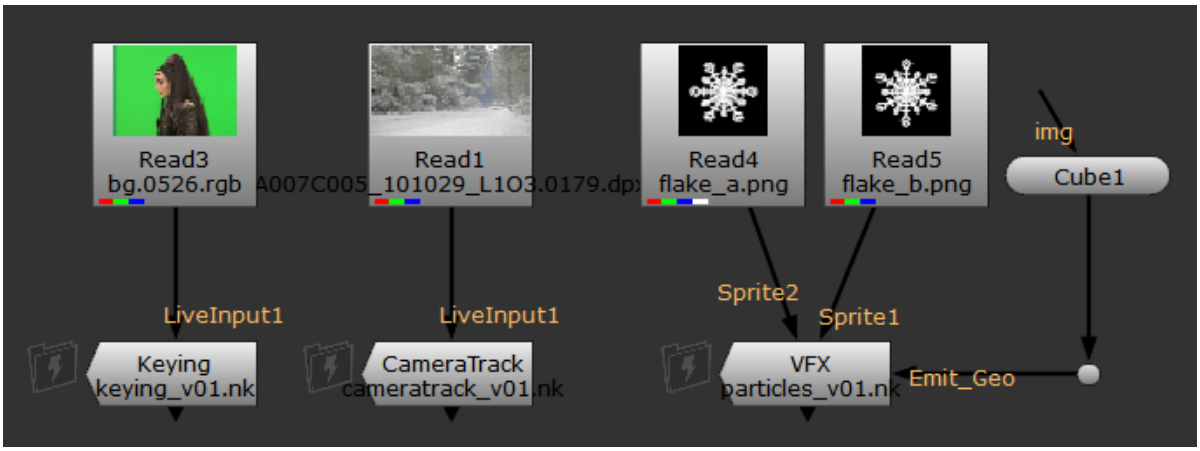
   See Setting Up Your Script for more information.

2. Add the basic nodes required for the Keying artist and place them in a LiveGroup. See Creating LiveGroups for more information on creating LiveGroups.



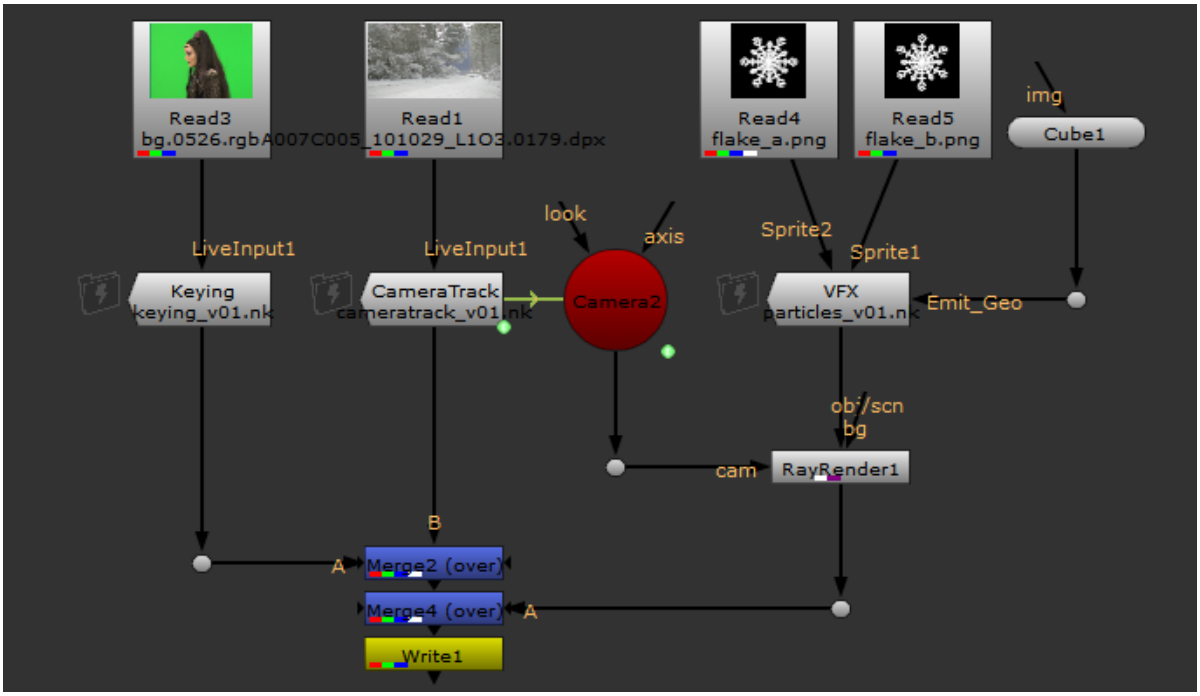
Tip: You can give the LiveGroup node a more descriptive name to make things easier to read.

3. **Publish** the LiveGroup to a network location accessible by the Keying artist. See Editing and Publishing LiveGroups for more information. Published LiveGroups are locked and cannot be edited unless you click **Make Editable** in the script.
4. Repeat steps 2 and 3 for the Tracking and 3D artists. An example is shown in the image.



5. As the sub-script work is completed, you can then add the required nodes to complete the source script.

   For example, you can expression link the Camera data from the CameraTracking LiveGroup to a Camera in the source script. See Managing LiveGroup Controls for more information.



6. As the sub-script artists version up their scripts, you can version up your LiveGroups in the source script to read in the latest work without the need to re-render in the source script. See Versioning LiveGroup Scripts for more information.

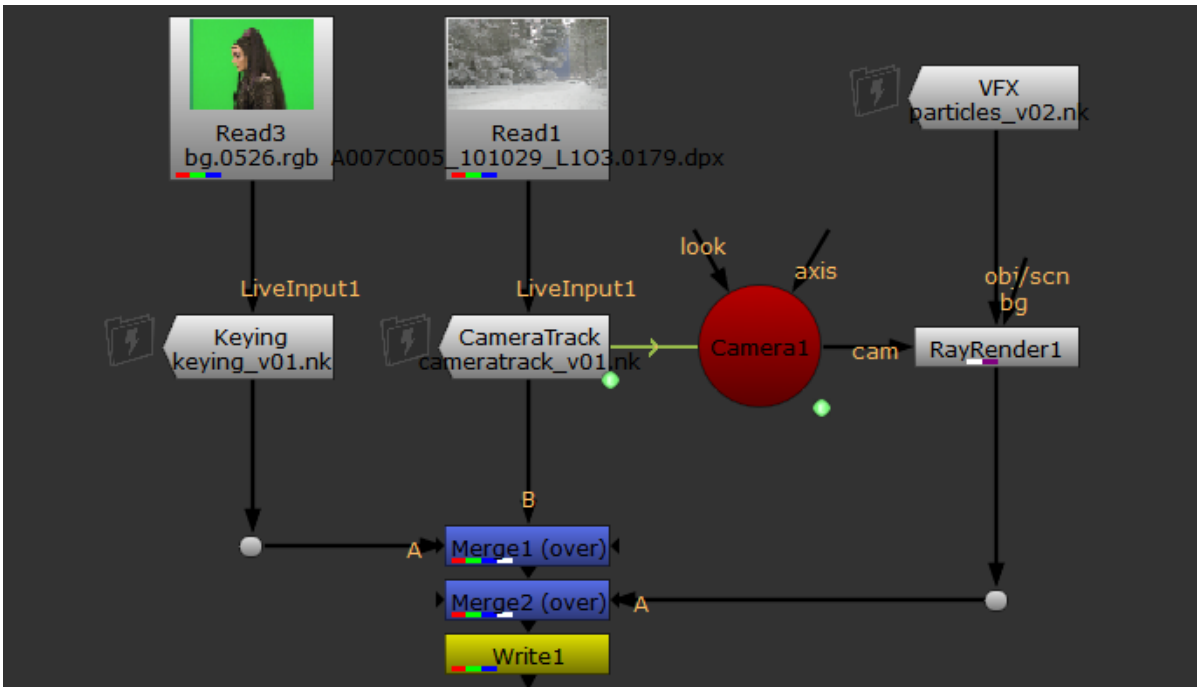
Combining Existing Scripts

As an artist, you might be working on a Nuke script in isolation that needs to be incorporated into a source script at a later date. LiveGroups allow you to point to a sub-script in the same way as a Precomp, but with the added functionality of LiveGroups.

To import a sub-script using a LiveGroup:

1. Open up the source script and add a LiveGroup node. The **VFX** LiveGroup referencing **particles_v02.nk** in the example.
2. Double-click the LiveGroup to open its **Properties** panel.
3. Enter the location of the sub-script in the **file** control.
4. Click **Reload**.

   This example differs from creating a source script first because the VFX LiveGroup is self-contained, meaning that the assets are inside the LiveGroup rather than in the source script.



You can also work the other way, with LiveInputs referencing LiveGroups, but there are a few caveats to working this way. See Referencing a Source Script Using LiveInputs for more information.

Creating LiveGroups

You can create empty LiveGroups to read in existing **.nk** scripts or select a set of nodes in the Node Graph and create a LiveGroup to contain those nodes, with the option to write them out to a new **.nk** script.

Note: If you create and publish a LiveGroup Pythonically, you need to add **xpos** and **ypos** values to the nodes in the Node Graph to position them correctly before you publish the LiveGroup. See Nuke's Python Developer's Guide for more information.

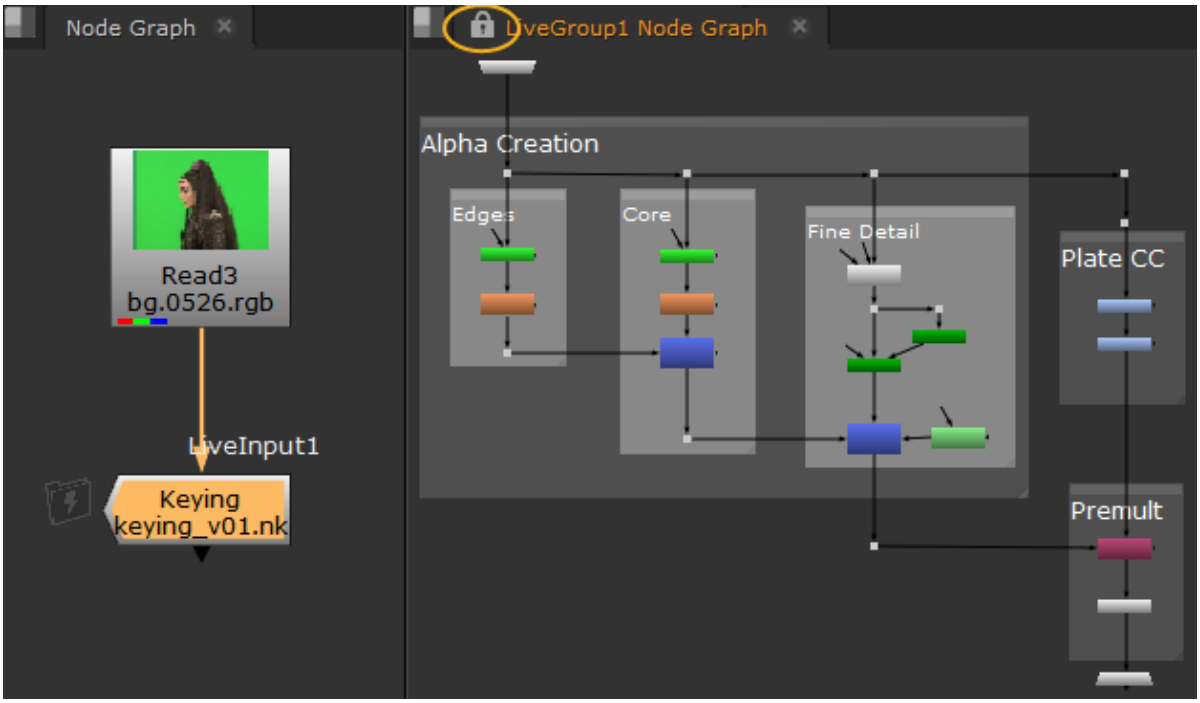
Importing an Existing Script into a LiveGroup

1. In Nuke's Toolbar, select **Other** > **LiveGroup**.
   A new Node Graph tab opens showing the empty LiveGroup.
2. Enter the location of the **.nk** script in the **file** control or browse to its location and click **Open**.

A warning displays, informing you that there are differences between the local script and the saved script.

3. Click **Yes** to overwrite the local script with the saved script.

   The LiveGroup icon turns gray and the LiveGroup tab updates to show the contents of the imported **.nk** script. The LiveGroup is locked as indicated by the padlock on the LiveGroup's tab and it behaves like regular Nuke Precomp node.

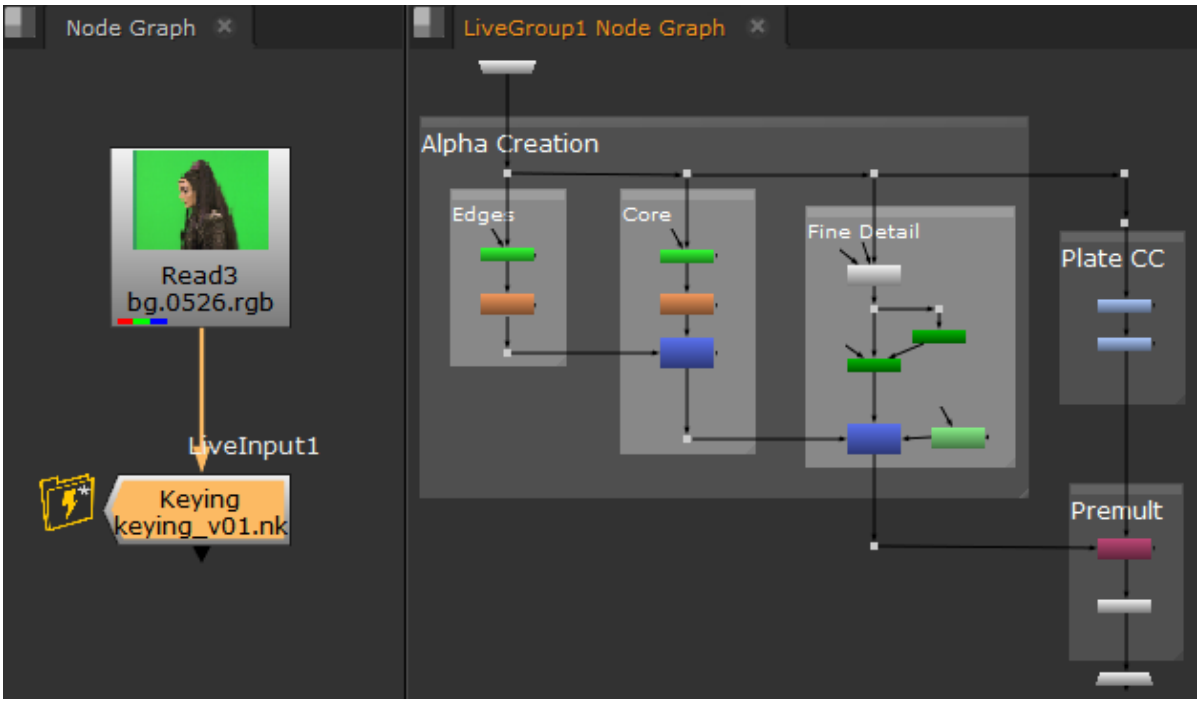


Note: Click **Open** in the LiveGroup's **Properties** to start a new Nuke session containing just the LiveGroup's contents.

Adding Nodes to a LiveGroup

1. Select all the nodes you want to nest inside the LiveGroup.
2. Do one of the following:
   - Press **Tab** in the Node Graph, type **LiveGroup** and press enter,
   - Press **Ctrl/Cmd**+**L**, or
   - Right-click and select **Edit** > **Node** > **Group** > **Collapse To LiveGroup**.

   The LiveGroup icon turns yellow and a LiveGroup tab shows the contents of the node. The LiveGroup is not locked and it behaves like regular Nuke Group node.



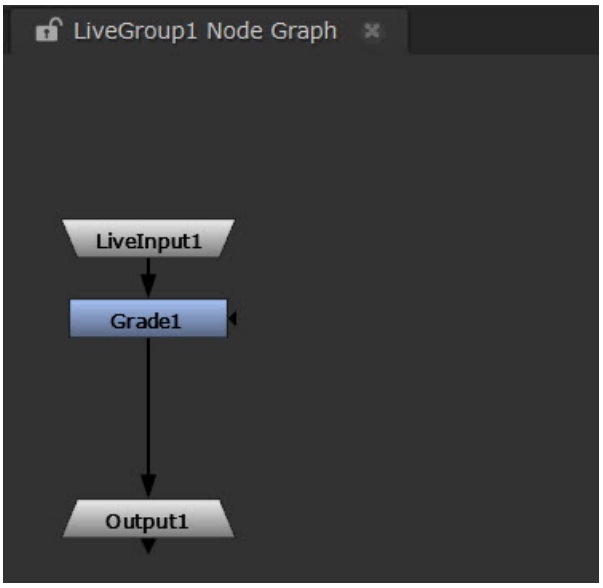
Note: It will remain unlocked until it is **Published**. See Editing and Publishing LiveGroups to learn how to publish LiveGroups.

Viewing Nodes Inside a LiveGroup

In the LiveGroup's **Properties** panel, click the **S** button in the top-right corner.



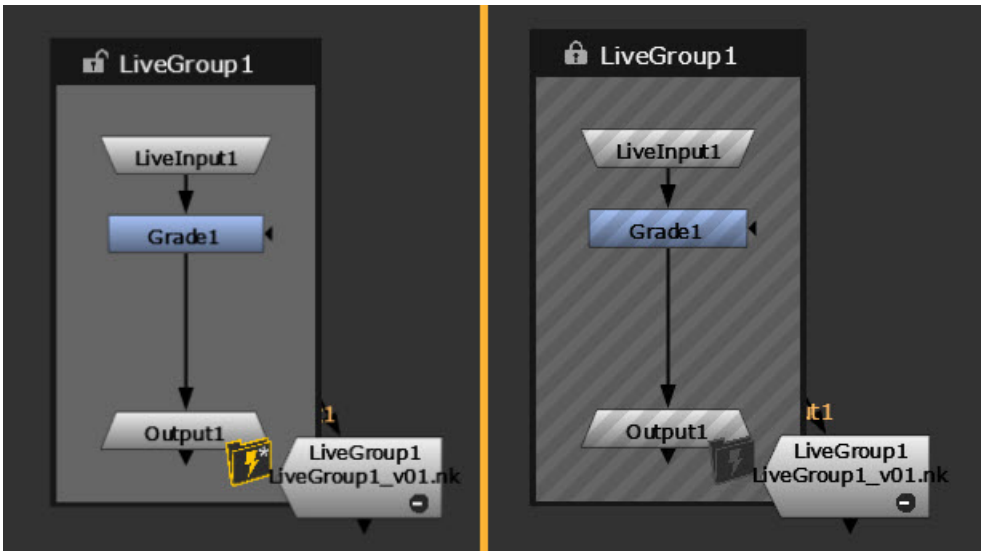
A new Node Graph tab opens containing the nested nodes. LiveGroups contain a LiveInput and Output node by default, which points to the source script so that artists can collaborate without depending on Read nodes in the LiveGroup.



See Referencing a Source Script Using LiveInputs below for more information.

You can also view the nested nodes in the main Node Graph by clicking the + icon on the LiveGroup node.



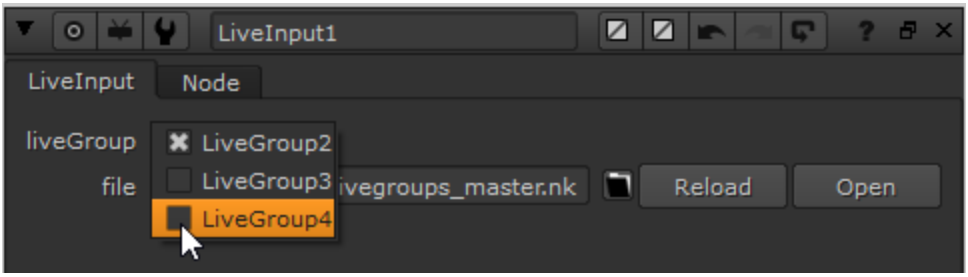


Left: Unlocked (Editable) LiveGroup. Right: Locked (Published) LiveGroup with diagonal line pattern.

Referencing a Source Script Using LiveInputs

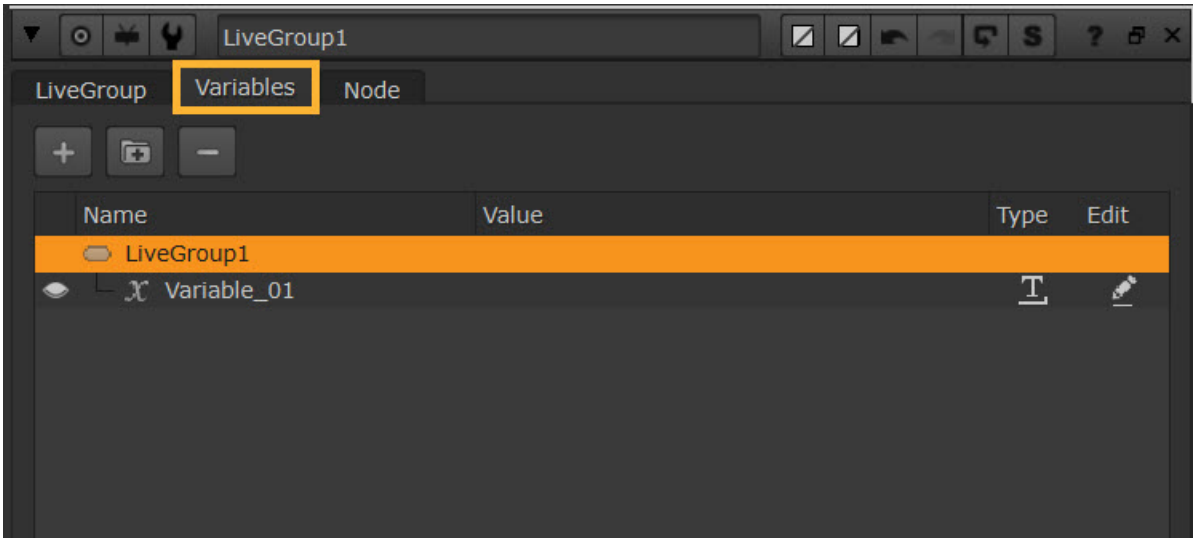
The LiveInput node can reference any LiveGroup within a script, eliminating the need to include the required assets in the sub-script. The only requirements are that the sub-script and source script have been saved and the LiveGroups you intend to connect to are pointing at the new sub-script.

1. Create a LiveInput node in the sub-script and then save the script.
2. In the source script, create a LiveGroup node and enter the sub-script name in the **file** control.
3. Save the source script.
4. In the sub-script, enter the source script name in the **file** control and click **Reload**.
5. Select the LiveGroup you want to reference from the **liveGroup** dropdown.



Using Variables with LiveGroups

There is also a Variables tab on the LiveGroup node, allowing you to work with Variables on shared scripts. See Using Graph Scope Variables in Your Workflows to learn more.



LiveGroup Management

Learn more about LiveGroup management and states at Managing LiveGroup Controls and Editing and Publishing LiveGroups.

Tip: LiveGroups support expression links between the source and sub-scripts, allowing you to easily update exposed control values that you know are likely to change, such as Camera transforms.

Managing LiveGroup Controls

You can expose controls from within your LiveGroup by picking existing knobs from nested nodes or by adding a new control, and even expression link these controls so that updates in the LiveGroup are passed to the source script.

Note: You cannot add or remove controls when a LiveGroup is locked. To unlock a LiveGroup, open the **Properties** panel **LiveGroup** tab and click **Make Editable**.

Picking and Adding Controls

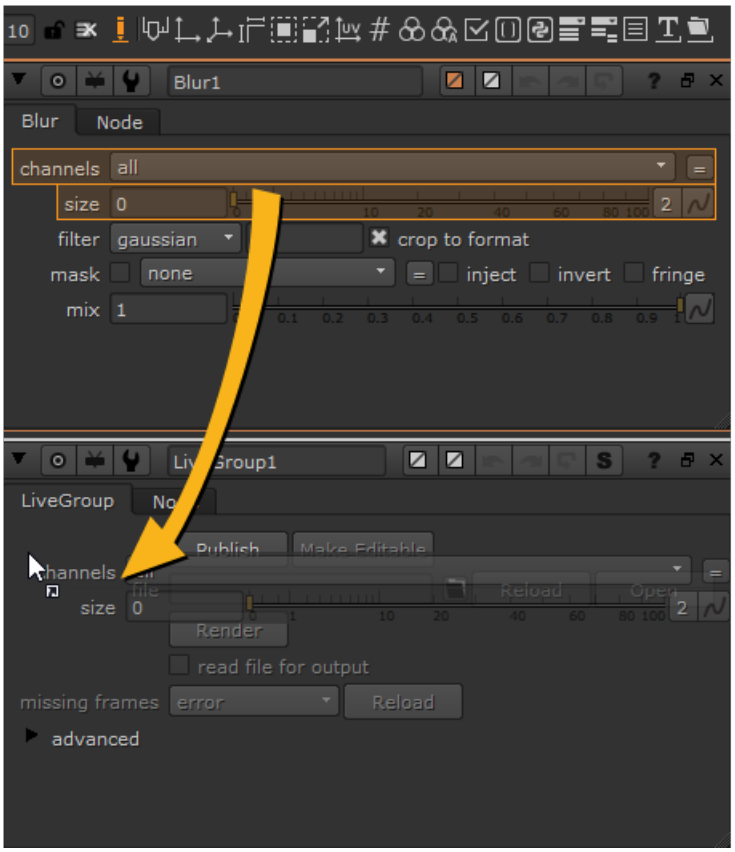
LiveGroups support the same method as gizmos for exposing controls in the **Properties** panel. Any control from a nested node within the LiveGroup can be exposed or you can add your own.

In Nuke's **Properties** panel, you can add controls using the Edit button or by right-clicking in the panel and selecting Adding Knobs Manually. Drag-and-drop knobs significantly reduce the time spent managing user knobs in a node's **Properties** panel when compared to the legacy **Manage User Knobs** method within Nuke.

To add knobs to a LiveGroup:

1. Double-click the LiveGroup to open its **Properties** panel.
2. Double-click the node containing the knobs you want to expose to open its **Properties** panel.
3. Click the edit button to enable drag-and-drop functionality.

4. Drag the target knob or knobs from the source node's **Properties** and drop them onto the LiveGroup's **Properties**.



5. Click the edit button to disable drag-and-drop functionality when you've added the required knobs.

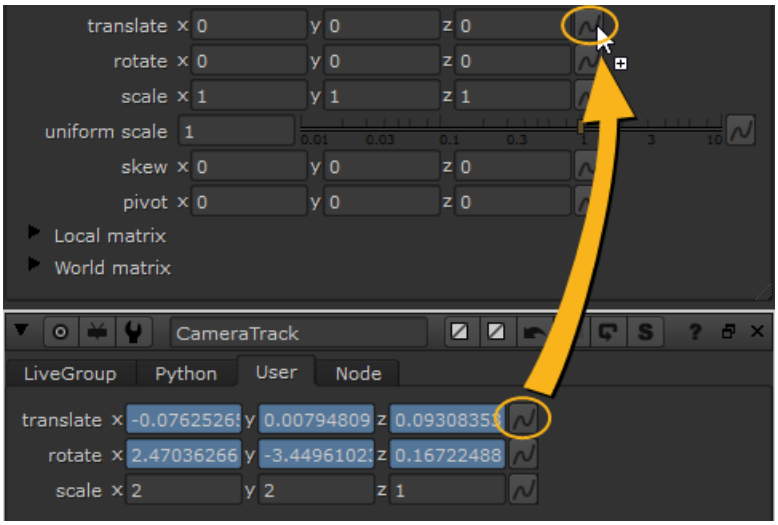
You can also expression link **User** knobs just like any other knob in Nuke, but with LiveGroups, you can expression link knobs from sub-scripts as well. See Expression Linking Controls for more information.

Expression Linking Controls

LiveGroups support expression links between the source and sub-scripts, allowing you to easily update exposed control values that you know are likely to change, such as Camera transforms.

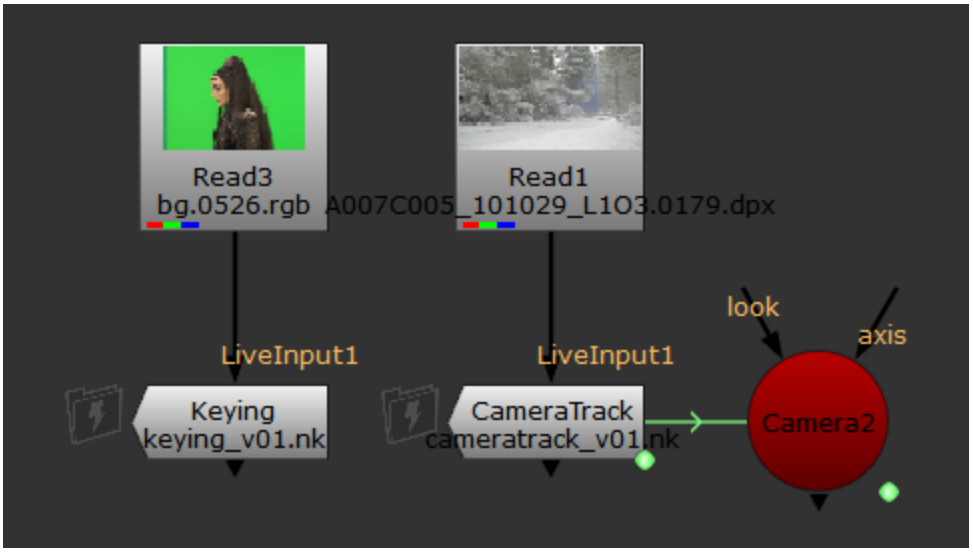
To expression link a sub-script control:

1. Double-click the LiveGroup containing the controls you want to link and navigate to the **User** tab.
2. Double-click the node you want to link the controls to, in this case a Camera.
3. Hold **Ctrl**/**Cmd** and drag the animation button from the LiveGroup to the node's control properties.



4. Repeat the process for all the controls you want to link from the sub-script.

   The Node Graph updates to show a green arrow representing the expression link and the direction of information flow.



Now, whenever the **CameraTrack** LiveGroup is reloaded, any updates to the camera translation and rotation are automatically applied in the source script.

Editing and Publishing LiveGroups

The state of a LiveGroup in the Node Graph controls how the source script and sub-scripts behave. LiveGroups have five states:

	The **Editable** state indicates that the LiveGroup is unlocked, but no edits have been made to the LiveGroup.
	The **Edited** state indicates that the LiveGroup has been edited in the current Nuke session.
	The **Edited with Overrides** state indicates that the LiveGroup has been edited in the current Nuke session and that the edits are overriding controls in the saved version of the LiveGroup. See Overriding LiveGroup Controls for more information.
	The **Published** state indicates that the LiveGroup has been saved to the location specified in the **file** control. LiveGroups in this state cannot be edited, but you can still adjust any controls exposed in the **Properties**.
	The **Published with Overrides** state indicates that the LiveGroup has been saved to the location specified in the **file** control and that one or more controls exposed in the **Properties** are overriding controls in the saved version of the LiveGroup. See Overriding LiveGroup Controls for more information.

Editing LiveGroups

The **Editable** and **Edited** states apply to new LiveGroups and LiveGroups that you have edited in the current Nuke session.



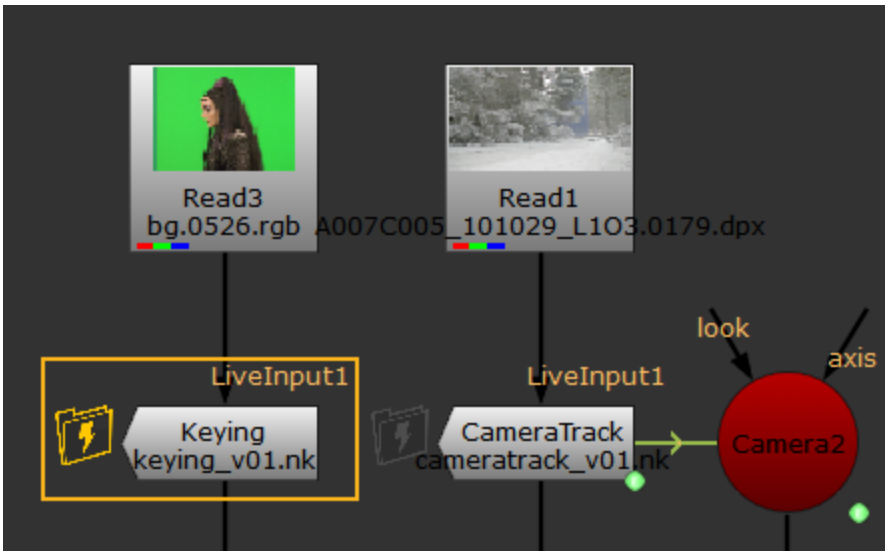
Note: You cannot add or remove controls when a LiveGroup is locked. To unlock a LiveGroup, open the **Properties** panel **LiveGroup** tab and click **Make Editable**.

A LiveGroup in the **Edited** state always writes its contents into the current script on save. If its **file** control points to a sub-script, no changes are made in the sub-script until the LiveGroup is **Published** to that location.

To change the state of a published LiveGroup to **Editable** without editing it:

1. Double-click the LiveGroup node to open its **Properties**.
2. Click **Make Editable**.

   The LiveGroup's state changes to **Editable** in the Node Graph and the contents of the group are added into the current script.



Versioning LiveGroup Scripts

Versions allow you to record the different stages of your workflow as you progress, quickly swapping the contents of LiveGroups in and out without overwriting existing work. You can publish different versions of LiveGroups in the same way as Nuke scripts.

To load a new version of a LiveGroup:

1. Select the required LiveGroup in the Node Graph,

   OR

   Double-click the LiveGroup to open its **Properties**.
2. Press **Alt**+**up arrow** or **Alt**+**down arrow** to load the next version.

   If no version of the current LiveGroup exists, Nuke displays a warning.

To save a new version of a LiveGroup:

1. Double-click the LiveGroup to open its **Properties**.
2. Enter the new version number in the **file** control,

   OR

   Press **Alt**+**up arrow** to increment the existing version number.



Note: Nuke displays a warning that the new version doesn't exist.

3. Click **Publish** to save the LiveGroup to the new version.

Publishing LiveGroups

The **Published** state only applies to LiveGroups that you have published to the location specified in the **file** control.

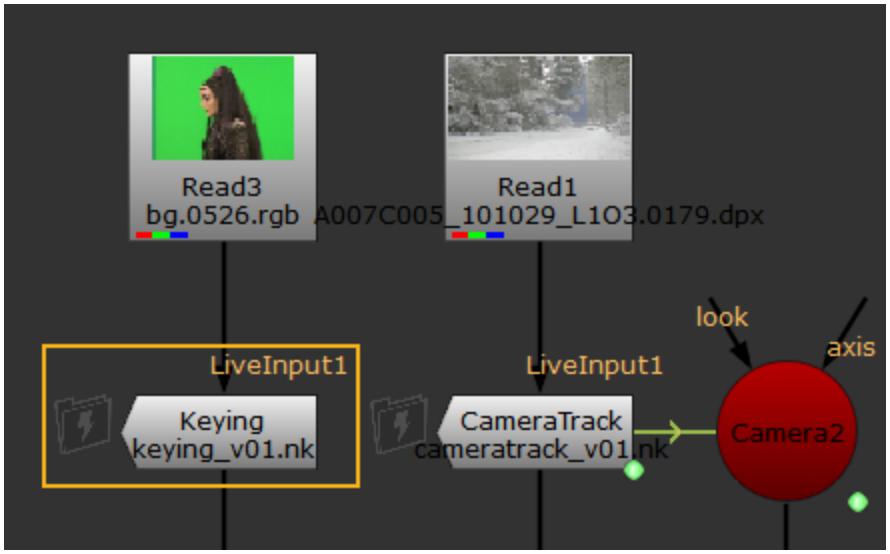
LiveGroups in the **Published** state never write their contents into the current script on save. To make changes to a **Published** LiveGroup, click **Make Editable**. It becomes **Editable** until you publish it again by clicking the **Publish** button.

Note: If you create and publish a LiveGroup Pythonically, you need to add **xpos** and **ypos** values to the nodes in the Node Graph to position them correctly before you publish the LiveGroup. See Nuke's Python Developer's Guide for more information.

To **Publish** a LiveGroup:

1. Double-click the LiveGroup node to open its **Properties**.
2. Check that the **file** control points to the required location and version up the file path if required.
3. Click **Publish**.

   The LiveGroup's state changes to **Published** in the Node Graph and the sub-script specified in the **file** control is updated.



Note: When you **Publish** a LiveGroup, the resulting **.nk** script retains the project settings from the source script. The source script's **Project Settings** are also used whenever the sub-script is loaded into the source script.

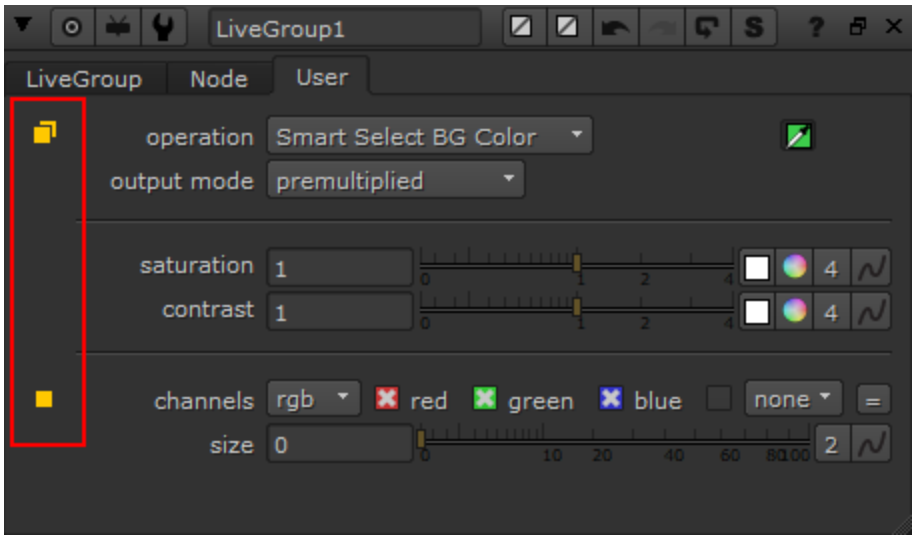
If you open the LiveGroup script in a separate instance of Nuke and change its **Project Settings** so that they no longer match the source script's settings, your changes DO NOT apply when the LiveGroup script is loaded into the source script. If you want to change the **Project Settings**, you should always do so in the source script rather than the sub-script.

Overriding LiveGroup Controls

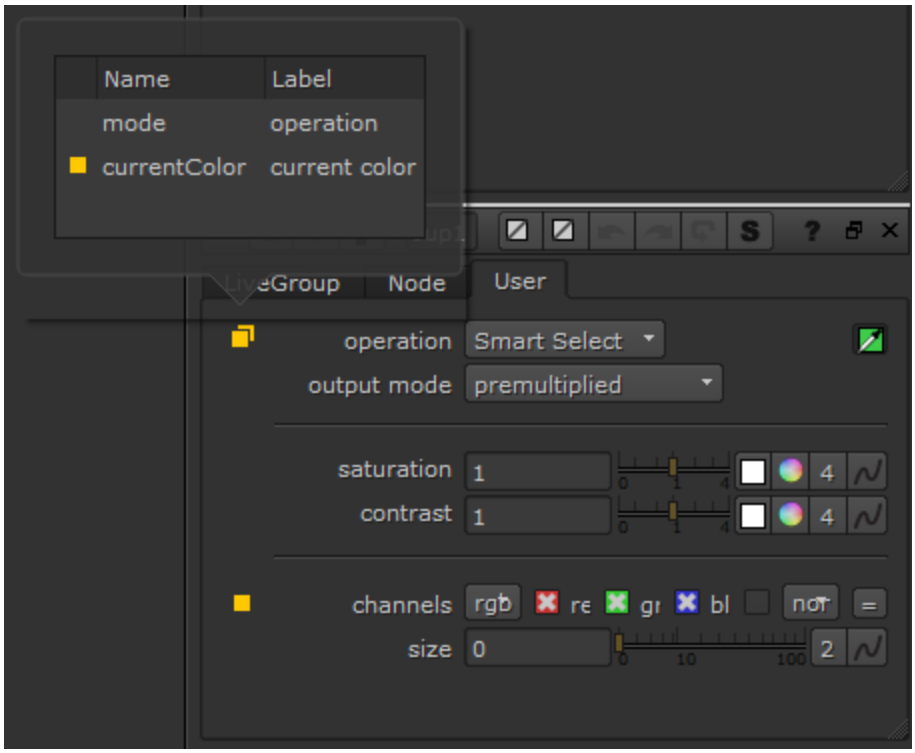
Published LiveGroups cannot be edited, unless you click **Make Editable** in the LiveGroups **Properties**, but the exposed controls can be overridden in scripts that include the LiveGroup. Override controls are marked with a yellow chip in the **Properties** panel, so you can quickly see which controls are affecting the output.

Note: Override controls are also indicated on the LiveGroup node in the Node Graph .

In the example, the **current color** and **channels** controls are overrides. The double yellow chip indicates that one of the controls on the top line is an override, in this case the **current color** control.



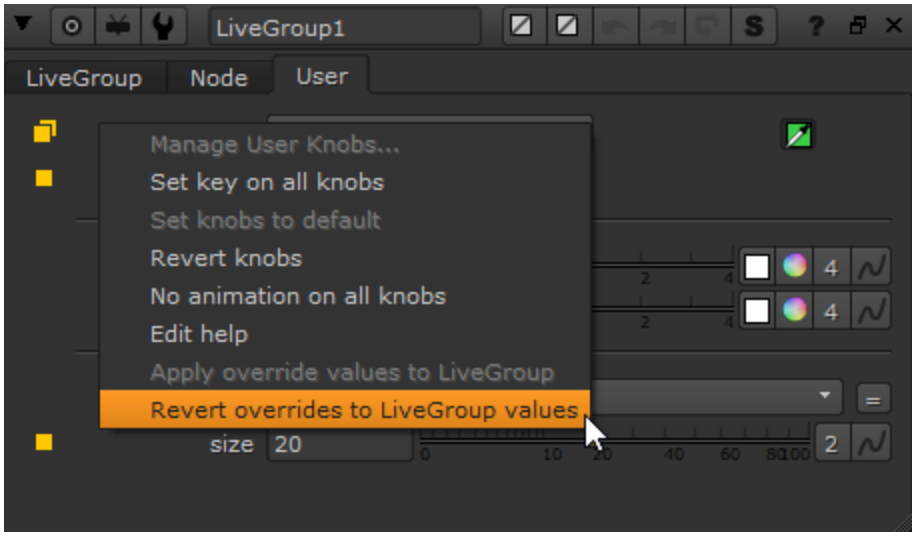
Click the yellow chip to list the overridden controls on that line.



Overrides can be reverted or applied to the LiveGroup controls, if the LiveGroup is in the editable state.

To revert overrides to the values stored in the LiveGroup:

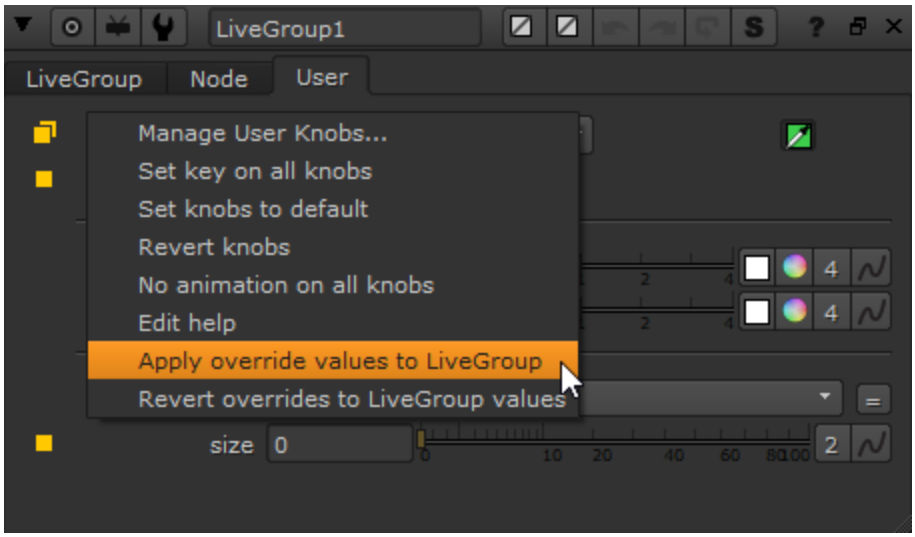
1. Open the LiveGroup **Properties** panel by double-clicking the node.
2. Click the custom tab on which the override controls are stored. In the example, the **User** tab.
3. Right-click the panel, making sure you're not hovering over a control, and select **Revert overrides to LiveGroup values**.



The controls revert to the values stored in the LiveGroup and the overrides are removed.

To apply the overrides to the LiveGroup controls:

1. Open the LiveGroup **Properties** panel by double-clicking the node.
2. Click **Make Editable** to unlock the LiveGroup.
3. Click the custom tab on which the override controls are stored. In the example, the **User** tab.
4. Right-click the panel, making sure you're not hovering over a control, and select **Apply override values to LiveGroup**.



   The override values are applied to the LiveGroup's controls and the overrides are removed.
5. Click **Publish** to overwrite the LiveGroup on disk or save a new version of the LiveGroup.

Configuring Nuke

These pages show visual effects supervisors how to configure Nuke for multiple artists, prior to the start of a project. These are the common application settings discussed:

- Command line operations
- Environment variables
- Gizmo, NDK plug-in, and Tcl script directories
- Python script directories
- OFX plug-in directories
- Favorite directories
- Cross-platform file paths
- Menu and Toolbar options
- Image formats
- Gizmos (Nuke group nodes or subscripts that allow only select modifications)
- Custom plug-ins (binary plug-ins made via the Nuke software developers kit)
- Generic Tcl ("**T**i**c**k**l**e") scripts
- Template scripts
- Common preferences
- Script's lookup tables (LUTs)
- Custom Viewer Processes



Note: If you copy and paste Python example scripts from this user guide into a text editor, line indentations may not be preserved. If this is the case, correct the indentations manually.

What Is a Terminal and How Do I Use One?

Many tasks in this chapter tell you to enter commands from a terminal or shell. This refers to a window where you can enter commands directly rather than making selections through a user interface.

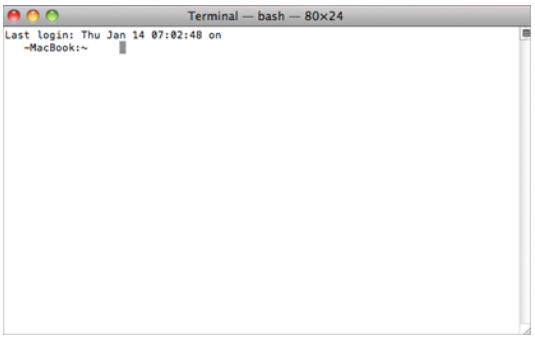
The following describes how to open such a window for your operating system.

- **Linux:** Click the right mouse button over the desktop and select **New Terminal** (or **Open Terminal**) from the right-click menu.
- **Windows:** From the **Start** menu, select **All Programs > Accessories > Command Prompt**.
- **Mac:** Click on the **Terminal** dock icon.

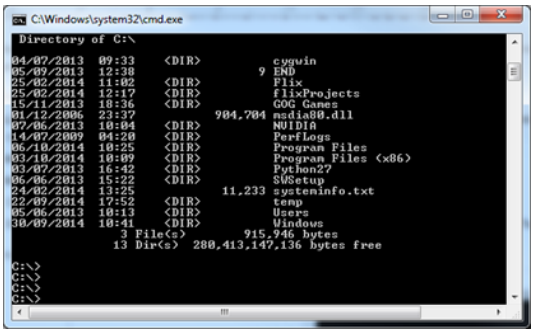
  OR

  Browse to the **Applications > Utilities** folder on your system hard drive, and double-click the **Terminal** icon.

Inside the terminal or shell, you'll see a command prompt, which looks similar to this:



Terminal prompt on Linux and Mac



Command prompt for Windows

Once you see the command prompt, you can enter commands to perform various tasks like listing the files in a directory or running programs. Here are some specific examples:

- On Linux or Mac, type **pwd** and press **Enter** to view the path of the current directory. On Windows, the equivalent command would be **cd**.
- On Linux or Mac, type **ls** and press **Enter** to view a list of files in the current directory. On Windows, the equivalent command would be **dir**.
- On Linux, Mac, and Windows, type **cd** followed by a full path name and press **Enter** to change directories.

Command Line Operations

Command line flags activate various options when you launch Nuke from a command line or Terminal, and provide additional functionality to Nuke. First let's discuss how to launch Nuke from a shell.

On Windows



Note: If you're using a terminal emulator, follow the steps under Mac or Linux, substituting the file paths as required.

Open a command line prompt and change directory as follows:

```
cd C:\Program Files\Nuke17.0v3\
```

To launch Nuke, type this command:

```
Nuke17.0.exe
```

Alternatively, you can set a **doskey** to point to Nuke and then you can launch Nuke from any directory:

```
doskey nuke="C:\Program Files\Nuke17.0v3\Nuke17.0.exe"
```

If you want to doskey NukeX, enter:

```
doskey NukeX="C:\Program Files\Nuke17.0v3\Nuke17.0.exe" --nukex
```

If you want to doskey Nuke Studio, enter:

```
doskey nukes="C:\Program Files\Nuke17.0v3\Nuke17.0.exe" --studio
```

On Mac

Open a command line prompt and change directory as follows:

```
cd /Applications/Nuke17.0v3/Nuke17.0v3.app/Contents/MacOS/
```

To launch Nuke, type this command:

```
./Nuke17.0
```

Alternatively, you can set an alias to point to Nuke and then you can launch Nuke from any directory. The procedure for this depends on what your default shell is. To get the name of the shell you are using, launch Terminal and enter **echo $SHELL**.

If you are using a bash shell, enter:

```
alias nuke='/Applications/Nuke17.0v3/Nuke17.0v3.app/Contents/MacOS/Nuke17.0'
```

Alternatively, if you are using a tcsh shell, enter:

```
alias nuke /Applications/Nuke17.0v3/Nuke17.0v3.app/Contents/MacOS/Nuke17.0
```

If you want to alias NukeX, enter:

```
alias NukeX /Applications/Nuke17.0v3/Nuke17.0v3.app/Contents/MacOS/Nuke17.0 --nukex
```

If you want to alias Nuke Studio, enter:

```
alias nukes /Applications/Nuke17.0v3/Nuke17.0v3.app/Contents/MacOS/Nuke17.0 --studio
```


Tip: You can add aliases to a **.cshrc** or **.bashrc** file in your home directory so that they are activated each time you open a shell. See your Systems Administrator for help setting this up.

On Linux

Open a command line prompt and change directory as follows:

```
cd /usr/local/Nuke17.0v3/
```

To launch Nuke, type this command:

```
./Nuke17.0
```

Alternatively, you can set an alias to point to Nuke and then you can launch Nuke from any directory. The procedure for this depends on what your default shell is. To get the name of the shell you are using, launch Terminal and enter **echo $SHELL**.

If you are using a bash shell, enter:

```
alias nuke='/usr/local/Nuke17.0v3/Nuke17.0'
```

Alternatively, if you are using a tcsh shell, enter:

```
alias nuke=/usr/local/Nuke17.0v3/Nuke17.0
```

If you want to alias NukeX, enter:

```
alias NukeX=/usr/local/Nuke17.0v3/Nuke17.0 --nukex
```

If you want to alias Nuke Studio, enter:

```
alias nukes=/usr/local/Nuke17.0v3/Nuke17.0 --studio
```

Tip: You can add aliases to a **.cshrc** or **.bashrc** file in your home directory so that they are activated each time you open a shell. See your Systems Administrator for help setting this up.

Using command line Flags

Now you can start experimenting with command line flags on launching Nuke. Here's one that displays the version number and build date.

```
nuke -version
```

If you have an **.nk** script, you can render it on the command line without opening the GUI version. Here's an example that renders a hundred frames of a Nuke script:

```
nuke -F 1-100 -x myscript.nk
```

Note how you can use the **-F** switch on the command line to indicate a frame range, separating the starting and ending frames with a dash.



Note: We recommend that you use the **-F** switch whenever defining a frame range on the command line, which must precede the script name argument.
However, for backwards compatibility, you can also use the old syntax. To do so, place the frame range at the end of the command and use a comma to separate the starting and ending frames. For example:

```
nuke -x myscript.nk 1,100
```

To display a list of command line flags (switches) available to you, use the following command:

```
nuke -help
```

Here's that list of command line flags in a table:

Switch/Flag	Action
-a	Formats default to anamorphic.
-c size (k, M, or G)	Limit the cache memory usage, where **size** equals a number in bytes. You can specify a different unit by appending **k** (kilobytes), **M** (megabytes), or **G** (gigabytes) after **size**.

Switch/Flag	Action
	Note: Nuke's actual cache memory usage is determined by which ever is smaller: the size value passed into this flag, or the "comp cache size (%)" set in Preferences > Performance > Caching. As such, this flag can't be used to increase Nuke's cache size above the "comp cache size (%)".
--cont	Nuke attempts to render subsequent frames in the specified range after an error. When **--cont** is not specified, rendering stops when an error is encountered.
--crashhandling 1 --crashhandling 0	Breakpad crash reporting allows you to submit crash dumps to Foundry in the unlikely event of a crash. By default, crash reporting is enabled in GUI mode and disabled in terminal mode. Use **--crashhandling** 1 to enable crash reporting in both GUI and terminal mode. Use **--crashhandling** 0 to disable crash reporting in both GUI and terminal mode. **Tip:** You can also use the NUKE_CRASH_HANDLING environment variable to control crash handling. See Environment Variables for more information.
-d <x server *name*>	This allows Nuke to be viewed on one machine while run on another. (Linux only and requires some setting up to allow remote access to the X Server on the target machine).
-f	Open Nuke script at full resolution. Scripts that have been saved displaying proxy images can be opened to show the full resolution image using this flag. See also **-p**.
-F	Frame numbers to execute the script for. All **-F** arguments must precede the script name argument. Here are some examples: • **-F 3** indicates frame 3. • **-F 1-10** indicates frames 1, 2, 3, 4, 5, 6, 7, 8, 9, and 10. • **-F 1-10x2** indicates frames 1, 3, 5, 7, and 9.

Switch/Flag	Action
	You can also use multiple frame ranges: `nuke -F 1-5 -F 10 -F 30-50x2 -x myscript.nk`
--gpu ARG	When set, enables GPU usage in terminal mode with an optional GPU index argument, which defaults to 0. Use **--gpulist** to display the selectable GPUs. **Note:** Overrides the GPU set in **Preferences** > **Performance** > **Hardware** when run in interactive mode.
--gpulist	Prints the selectable GPUs and their corresponding index for use with the **--gpu ARG** option.
-h	Display command line help.
-help	Display command line help.
-i	Use an interactive (nuke_i) RLM license key. This flag is used in conjunction with background rendering scripts using **-x**. By default **-x** uses a nuke_r license key, but **-xi** background renders using a nuke_i license key. **Note:** If you still use FLEXlm licenses and you're interested in making a move to RLM licensing, please contact sales@foundry.com to obtain a replacement license.
-l	New read or write nodes have the colorspace set to linear rather than default.
-m #	Set the number of threads to the value specified by **#**.
--multigpu	If have multiple GPUs of the same type installed, you can enable this preference to share work between the available GPUs for extra processing speed. This is a global preference and is applied to all GPU enabled nodes. **Note:** For more information on GPU support, see the Release Notes for your version of Nuke.
-n	Open script without postage stamps on nodes.

Switch/Flag	Action
--nocrashprompt	When crash handling is enabled in GUI mode, submit crash reports automatically without displaying a crash reporter dialog. **Tip:** You can also use the NUKE_NO_CRASH_PROMPT environment variable to control the crash prompt. See Environment Variables for more information.
--nukeassist	Launch Nuke Assist, which is licensed as part of a NukeX Maintenance package and is intended for use as a workstation for artists performing painting, rotoscoping, and tracking. Two complimentary licenses are included with every NukeX license.
-p	Open Nuke script at proxy resolution. Scripts that have been saved displaying full resolution images can be opened to show the proxy resolution image using this flag. See also **-f**.
-P	Measure your nodes' performance metrics and show them in the Node Graph. See Using Performance Timing for more information.
--pause	Initial Viewers in the script specified on the command line should be paused.
-Pf <filename>	Measure your nodes' performance metrics and write them to an XML file at render time. See Using Performance Timing for more information.
--nc	Runs Nuke in Nuke Non-commercial mode. See About Nuke Non-commercial for more information.
--priority p	Runs Nuke with a different priority, you can choose from: • **high** (only available to the super user on Linux/OS X) • **medium** • **low**
--python-no-root-knobdefaults	Prevents the application of knob defaults to the root node when executing a Python script.
-q	Quiet mode. This stops non-essential printing to the command-line and suppresses information pop-ups, such as loading autosaved scripts and local changes to LiveGroups.

Switch/Flag	Action
--remap	Allows you to remap file paths in order to easily share Nuke projects across different operating systems. This is the command line equivalent of setting the **Path Remaps** control in the **Preferences** dialog. The --remap flag takes a comma-separated list of paths as an argument. The paths are arranged in pairs where the first path of each pair maps to the second path of each pair. For example, if you use: nuke -t --remap "X:/path,Y:,A:,B:/anotherpath" • Any paths starting with **X:/path** are converted to start with **Y:** • Any paths starting with **A:** are converted to start with **B:/anotherpath** The --remap flag throws an error if: • it is defined when starting GUI mode, that is, without **-x** or **-t** • the paths do not pair up. For example, if you use: nuke -t --remap "X:/path,Y:,A:" **A:** does not map to anything, and an error is produced. The --remap flag gives a warning (but does not error) if you give it no paths. For example: nuke -t --remap "" **Note:** Note that the mappings are only applied to the Nuke session that is being started. They do not affect the Preferences.nk file used by the GUI.
-s #	Sets the stack size per thread, in bytes. The default value is 16777216 (16 MB). The smallest allowed value is 1048576 (1 MB). None of Nuke's default nodes require more than the default memory stack value, but if you have written a custom node that requests a large stack from the memory buffer, increasing the stack size can prevent stack overflow errors.
--safe	Running Nuke in **safe mode** stops the following loading at startup: • Any scripts or plug-ins in ~/.nuke

Switch/Flag	Action
	• Any scripts or plug-ins in $NUKE_PATH or %NUKE_PATH% • Any OFX plug-in (including FurnaceCore)
--sro	Forces Nuke to obey the render order of Write nodes so that Read nodes can use files created by earlier Write nodes.
-t	Terminal mode (without GUI). This allows you to enter Python commands without launching the GUI. A >>> command prompt is displayed during this mode. Enter quit() to exit this mode and return to the shell prompt. This mode uses a nuke_r license key by default, but you can get it to use a nuke_i key by using the -ti flag combo.
--tg	Terminal Mode. This also starts a QApplication so that PySide/PyQt can be used. This mode uses an interactive license, and on Linux requires an XWindows display session.
--topdown	Enables the new top-down rendering project setting to render the full frame at the cost of more memory. Although top-down rendering produces the full frame faster, it disables progressive rendering and uses more memory. You may want to use the classic method if you're only interested in the first few scan lines in the Viewer.  **Note:** The render mode is only saved as part of the script if you set **render mode** to **top-down** in the **Project Settings**. This ensures that the chosen render method is used the next time the script is opened. Using the **--topdown** command line argument does not save the render mode in the script.
--use-local-file-cache	Forces a command line render to use localized files.
-V level	Verbose mode. In the terminal, you'll see explicit commands as each action is performed in Nuke. Specify the **level** to print more in the Terminal, select from: • **0** (not verbose) • **1** (outputs Nuke script load and save) • **2** (outputs loading plug-ins, Python, Tcl, Nuke scripts, progress and buffer reports)
-v	This command displays an image file inside a Nuke Viewer. Here's an example:

Switch/Flag	Action
	`nuke -v image.tif`
--view v	Only execute the specified views. For multiple views, use a comma separated list: **`left,right`**
--version	Display the version information in the shell.

--workspace arg	Launch Nuke and activate the specified workspace automatically. Any named layout specified in the **Workspace** dropdown is valid, including custom workspaces. For example:
	`--workspace Compositing`
	`--workspace "Monitor Out"`
	`--workspace MySetup`
	
	 Tip: If the workspace name contains a space, you may need to add quotes around the name.

-x	eXecute mode. Takes a Nuke script and renders all active Write nodes.  **Note:** Nuke Non-commercial is restricted to encrypted **.nknc** scripts with **-x** from the command line, that is, using the syntax: **nuke -x myscript.nknc** This mode uses a nuke_r license key. To use a nuke_i license key, use **-xi**. This is the syntax: **nuke -xi myscript.nk** On Windows, you can press **Ctrl**+**Break** to cancel a render without exiting if a render is active, or exit if not. **Ctrl**/**Cmd**+**C** exits immediately. On Mac and Linux, **Ctrl**/**Cmd**+**C** always exits.

Switch/Flag	Action
	Note: If you still use FLEXlm licenses and you're interested in making a move to RLM licensing, please contact sales@foundry.com to obtain a replacement license.
-X *node*	Render only the Write node specified by *node*.
--	End switches, allowing script to start with a dash or be just - to read from stdin

General syntax

This is the general syntax for using these options when launching Nuke at the command prompt:

```
nuke <switches> <script> <argv> <ranges>
```

<switches> - modifies the behavior of Nuke when run from the command line. A list of switches is given in the table above. These are sometimes called flags.

<script> - the name of the Nuke script.

<argv> - an optional argument that can be used in Nuke. See the example below.

<ranges> - this is the frame range you want rendering.

Examples

Let's consider some practical examples.

To launch Nuke and open a script.

```
nuke myscript.nk
```

Crazy I know, but I've called my script, -myscript.nk, and the hyphen at the start of the filename has confused Nuke. To get round this if you don't want to rename your file use the double hyphen syntax:

```
nuke -- -myscript.nk
```

To display an image:

```
nuke -v polarbear.tif
```

To display several images:

```
nuke -v polarbear.tif whiteflower.psd mountains.cin
```

To display an image sequence (taxi.0001.tif, taxi.0002.tif,...,taxi.0050.tif):

```
nuke -v taxi.####.tif 1-50
```

To render frame 5 of a Nuke script:

```
nuke -F 5 -x myscript.nk
```

To render frames 30 to 50 of a Nuke script:

```
nuke -F 30-50 -x myscript.nk
```

To render two frame ranges, 10-20 and 34-60, of a Nuke script:

```
nuke -F 10-20 -F 34-60 -x myscript.nk
```

To render every tenth frame of a 50 frame sequence of a Nuke script:

```
nuke -F 1-50x10 -x myscript.nk
```

This renders frames 1, 11, 21, 31, 41.

In a script with two write nodes called WriteBlur and WriteInvert this command just renders frames 1 to 20 from the WriteBlur node:

```
nuke -X WriteBlur myscript.nk 1-20
```

Using [argv 0]

Let's use [argv] to vary the output file name. Launch the GUI version of Nuke and create a node tree that puts a checker into a Write node. Open the write node property panel by double clicking on it and in the file text field enter this filename:

[argv 0].####.tif

Save the script and quit Nuke. On the command line type:

```
nuke -x myscript.nk mychecker 1-5
```

This renders 5 frames (mychecker.0001.tif, mychecker.0002.tif, etc.).

You can add another variable to control the output image file type. The file text field needs this:

[argv 0].####.[argv 1]

and then render the script using this command:

```
nuke -x myscript.nk mychecker cin 1-5
```

to get mychecker.0001.cin, mychecker.0002.cin, etc.

The <argv> string can be any [argv n] expression to provide variable arguments to the script. These must be placed between the <*script*> and the <*ranges*> on the command line. You can include multiple expressions,

but each must begin with a non-numeric character to avoid confusion with the frame range control. For more information on expressions, see Expressions.

Using Python to convert TIFFs to JPEGs

This command line method converts 5 TIFF frames to JPEG.

```
nuke -t
>>> r = nuke.nodes.Read(file = ”myimage.####.tif”)
>>> w = nuke.nodes.Write(file = ”myimage.####.jpg”)
>>> w.setInput( 0, r )
>>> nuke.execute(“Write1”, 1,5)
>>> quit()
```

It’s a bit tedious typing these commands in line by line. So let’s put them in a text file called imageconvert.py and get Nuke to execute the Python script.

```
cat imageconvert.py
r = nuke.nodes.Read(file = ”myimage.####.tif”)
w = nuke.nodes.Write(file = ”myimage.####.jpg”)
w.setInput( 0, r )
nuke.execute(“Write1”, 1,5)
nuke -t < imageconvert.py
```

You can also pass in the Python script as a command line parameter. Doing this allows you to enter additional parameters after the script name to pass into your script. When you do so, note that sys.argv[0] is the name of the Python script being executed, and argv[1:] are the other parameters you passed in. One example of this is below. See the standard Python module optparse for other ways to parse parameters.

```
cat imageconvertwithargs.py
import sys
r = nuke.nodes.Read(file = sys.argv[1])
w = nuke.nodes.Write(file = sys.argv[2])
w.setInput(0, r)
nuke.execute("Write1", 1, 5)
nuke -t imageconvertwithargs.py myimage.####.tif myimage.####.jpg
```

Environment Variables

Environment variables are named variables used to store a value, such as a specific file path. They can be used to influence Nuke's behavior. For example, Nuke uses the information stored in them to define where to place certain files.

Setting Environment Variables

This section teaches you how to set environment variables, check if a particular environment variable exists, and display a list of set environment variables.

On Windows

1. Click the **Windows** button and type "environment variables" to display a list of options.
2. Click the **Edit the system environment variables** option.

   The **System Properties** dialog is displayed.
3. Click the **Environment Variables** button.

   The **Environment Variables** dialog is displayed.
4. Click the **New** button under either **User variables** or **System variables**, depending on whether you want to set the variable for the current user or all users.



Note: To set environment variables for all users, you need to have administrator privileges.

5. In the **Variable name** field, enter the name of the environment variable you want to set. For example, **PATH**.

   For a list of the environment variables that Nuke understands, see Nuke Environment Variables.
6. In the **Variable value** field, enter the value for the variable. For example, **C:\Program Files\Common Files\Nuke\13.2\plugins**
7. Click **OK**.



Note: When editing existing system variables, or adding or deleting either user or system variables, you may need to log off and on again before your changes to environment variables take effect.

On Mac

On Mac, you can use the **launchd.conf** file to set environment variables. You may need to create the **launchd.conf** file if it doesn't already exist in the **/etc/** directory.



Note: If you only need to set an environment variable for a single session, or you don't want to use the Mac **launchd.conf** file, you can also set variables using the method described in the On Linux section.

Environment variables set using the **launchd.conf** file are read both when Nuke is launched from the Nuke icon and when it's launched from the Terminal.

1. Open a Terminal window.
2. Create the **/etc/launchd.conf** file, if it doesn't already exist, and then add the environment variable(s) and value(s) to the file using the following format:

   ```
   setenv <VARIABLE> <VALUE>
   ```

   For example, to set two environment variables, NUKE_PATH and OFX_PLUGIN_PATH, to point to alternate locations:

   ```
   setenv NUKE_PATH /SharedDisk/Nuke/
   setenv OFX_PLUGIN_PATH /SharedDisk/OFX
   ```

   For a list of the environment variables that Nuke understands, see Nuke Environment Variables.

Tip: A handy command line tool for creating and editing files in the Terminal is **nano**. To start nano with the correct permissions, enter:
sudo nano /etc/launchd.conf

3. To force the OS to read the **launchd.conf** file at startup, enter:

   ```
   launchctl < /etc/launchd.conf; sudo launchctl < /etc/launchd.conf
   ```

4. Restart your Mac to apply the changes.

On Linux

1. The procedure for setting an environment variable depends on what your default shell is. To get the name of the shell you are using, launch a shell and enter **echo $SHELL**.
2. Depending on the output of the previous step, do one of the following:

- If your shell is a csh or tcsh shell, add the following command to the .cshrc or .tcshrc file in your home directory: **setenv VARIABLE value**. Replace **VARIABLE** with the name of the environment variable and **value** with the value you want to give it, for example **setenv NUKE_PATH /SharedDisk/Nuke**.
- If your shell is a bash or ksh shell, add the following command to the .bashrc or .kshrc file in your home directory: **export VARIABLE=value**. Replace **VARIABLE** with the name of the environment variable and **value** with the value you want to give it, for example **export NUKE_PATH=/SharedDisk/Nuke**.

For a list of the environment variables that Nuke understands, see Nuke Environment Variables.



Article: How to set environment variables
How to delete or unset environment variables

To Check if an Environment Variable Exists

From Inside Nuke

- Press **X** in the Node Graph, check that **TCL** is enabled, and enter:

```
getenv <VARIABLE>
```

OR

- Open the **Script Editor** and enter:

```
import os
print os.environ["VARIABLE"]
```

In both cases, VARIABLE should be replaced by the environment variable you're interested in. For example, NUKE_TEMP_DIR on Windows returns:
`C:/Users/<current_user>/AppData/Local/Temp/nuke`

In the Windows Environment

1. Select **Start** > **All Programs** > **Accessories** > **Command Prompt**.
2. In the command window that opens, enter **echo %VARIABLE%**. Replace **VARIABLE** with the name of the environment variable. For example, to check if NUKE_DISK_CACHE is set, enter **echo %NUKE_DISK_CACHE%**.

If the variable is set, its value is displayed in the command window.

In the Mac or Linux Environment

1. Launch Terminal or a shell.
2. Enter **echo $VARIABLE**. Replace **VARIABLE** with the name of the environment variable. For example, to check if NUKE_DISK_CACHE is set, enter **echo $NUKE_DISK_CACHE**.

If the variable is set, its value is displayed in the Terminal or shell window.



Article: How to get environment variable values

To Display a List of Set Environment Variables

On Windows

1. Select **Start** > **All Programs** > **Accessories** > **Command Prompt**.
2. In the command window that opens, enter **set**.

A list of all the environment variables that are set is displayed in the command window.

On Mac or Linux

1. Launch Terminal or a shell.
2. Enter **printenv**.

A list of all the environment variables that are set is displayed in the Terminal or shell window.



Article: How to list environment variables

Nuke Environment Variables

The following table lists the environment variables Nuke recognizes.

Environment Variable	Description
FN_DISABLE_LICENSE_DIALOG or FN_NUKE_DISABLE_TMPLIC_NOTIFY_DIALOG	By default, if you have installed a temporary license, Nuke displays a dialog at start-up alerting you to the number of days remaining. If you want to disable this behavior, you can set either of these environment variables to 1 to suppress the warning message about imminent license expiration.  **Note:** You still get a warning if no license is found, for example if you only have a Nuke license but you try to run NukeX.
FN_LICENSE_DIALOG_DAYS_LEFT_BEFORE_PROMPT	By default, if you have installed a temporary license, Nuke displays a dialog at start-up alerting you to the number of days remaining. If you want to disable this behavior until a set number of days before expiry, you can set this environment variables to the required number of days.  **Note:** You still get a warning if no license is found, for example if you only have a Nuke license but you try to run NukeX.
FN_NUKE_DISABLE_GPU_ACCELERATION	This variable disables Nuke's CUDA and OpenCL capabilities. When enabled, any GPUs installed locally are disabled and cannot be selected from **Preferences** > **Performance** > **Hardware** > **default blink device** dropdown. Any GPU accelerated nodes, such as Kronos and Denoise, default to processing on the CPU.
FN_SUBSCRIPTION_LICENSE_DIR	On Windows, user names containing non-ASCII characters can cause subscription licensing to fail. If a licensing error similar to the following displays: **Unable to create subscription license directory: C:\Users\Zoë Hernández\FoundryLicensing** Try changing the license directory to an alternate location using this environment variable.
foundry_LICENSE	The location of the Nuke RLM license file, if the following recommended location is not used: **On Mac and Linux**: `/usr/local/foundry/RLM`

Environment Variable	Description
	On Windows: *drive letter*:\Program Files\The Foundry\RLM **Note:** If you still use FLEXlm licenses and you're interested in making a move to RLM licensing, please contact sales@foundry.com to obtain a replacement license.
FOUNDRY_LICENSE_DEBUG	This variable prints additional licensing information to the command line or Terminal.
FOUNDRY_LICENSE_FILE	The location of the Nuke FLEXlm license file, if the following recommended location is not used: **On Mac and Linux**: /usr/local/foundry/FLEXlm **On Windows:** *drive letter*:\Program Files\The Foundry\FLEXlm **Note:** If you still use FLEXlm licenses and you're interested in making a move to RLM licensing, please contact sales@foundry.com to obtain a replacement license.
FOUNDRY_LOG_FILE	This variable specifies the location of Nuke Studio's logfile. If you don't specify a logfile, all output is to screen.
FOUNDRY_LOG_LEVEL	This variable sets the level of logging Nuke Studio produces during operation. There are four levels of detail, on a sliding scale from minimal to verbose: • **error** • **warning** • **message** • **verbose**

Environment Variable	Description
	Note: Setting the logging level to verbose can produce large log files when FOUNDRY_LOG_FILE is specified.
FRAMESERVER_LOG_DIR	This variable is used to specify a different location for the Frame Server to write log files to, if you'd like to keep them separate from the default NUKE_TEMP_DIR. See Using the Frame Server on Render Machines for more information.
HIERO_DISABLE_THUMBNAILS	Set this variable to stop Nuke Studio loading thumbnails.
HIERO_DISABLE_THUMBNAILS_CACHE	Set this variable to stop Nuke Studio caching thumbnails for improved access once loaded. **Note:** This variable does not clear the cache, you must remove cached files manually.
NUKE_AJA_CHANNEL	AJA cards take a 3G level signal (mostly for 12-bit 444 RGB) and combine it into a single 3G-B (B denotes B level, hence the 3G-B) stream through SDI3 by default. Set this environment variable to 1, 2, or 4 to output a single stream through SDI1, SDI2, or SDI4.
NUKE_ALLOW_GIZMO_SAVING	Nuke does not allow you to **Overwrite** and **Save as** gizmos by default, without copying the gizmo to a Group. Setting this environment variable to 1 enables this behavior, so artists don't need to copy a gizmo before editing it.
NUKE_CRASH_HANDLING	Breakpad crash reporting allows you to submit crash dumps to Foundry in the unlikely event of a crash. By default, crash reporting is enabled in GUI mode and disabled in terminal mode. When NUKE_CRASH_HANDLING is set to 1, crash reporting is enabled in both GUI and terminal mode. When NUKE_CRASH_HANDLING is set to 0, crash reporting is disabled in both GUI and terminal mode.

Environment Variable	Description
NUKE_DEBUG_ IMAGECACHE	When enabled, Comp Viewer image cache data is printed to the command line or Terminal. Information on disk space used, the number of files cached, and the cache location is displayed.
NUKE_DEBUG_MEMORY	When working on large images, Nuke may need to free up memory during rendering. When this happens and NUKE_DEBUG_MEMORY is set to 1, Nuke prints the following information to the console: Memory: over maximum usage, trying to reduce usage from 1 GB to 924 MB. If this variable is not set, you cannot see the debug memory messages. Note that here, KB, MB, GB, and TB mean units of 1000. For example, 1MB means 1,000,000 bytes.
NUKE_DISK_CACHE	The location where Nuke saves all recent images displayed in the Viewer. Ideally, this should be a local disk with the fastest access time available.
NUKE_DISK_CACHE_GB	The maximum size the disk cache can reach (in gigabytes).
NUKE_EXR_TEMP_DIR	On Linux, this is the location Nuke uses for temporary files while reading PIZ-compressed **.exr** files. This environment variable is only relevant on Linux. If this variable is not set, the location is determined by NUKE_TEMP_DIR.
NUKE_EXR_TEMP_NAME	Changes the naming convention of **.exr** temporary files during rendering. Setting the variable to **1** writes temporary **.exr** files as **<filename>.exr.tmp**, rather than **<filehash>.tmp** as in previous releases.
NUKE_FONT_PATH	The location that Nuke checks for available font files when the Text node properties panel is opened.
NUKE_IGNORE_ROTO_ INCOMPATIBILITY	This variable disables the warning dialog displayed when you open scripts containing pre-Nuke 8 RotoPaint nodes.
NUKE_INTERACTIVE	The **import nuke** function checks-out a nuke_r render license by default. If you want to use Nuke interactively, and you have an interactive license, set this environment variable to 1.

Environment Variable	Description
	See Nuke as a Python Module for more information.
NUKE_LEGACY_CHANNEL_SORTING	This variable disables the new channel sorting behavior, where the RGBA layer is sorted first. Enabling this variable causes Nuke to sort channels alphabetically.
NUKE_LOCALIZATION_NUMWATCHERS	Controls the number of threads used for querying the file system, such as checking if localized files are out of date, if they exist, and so on. The actual copying of files is done in a separate thread pool with only 1 thread, which is fixed.
NUKE_NO_VIEWER_GPU	Disables Nuke's Comp Viewer OpenGL hardware acceleration.
NUKE_OCIO_ROLES	Controls the state of the **Prioritize OCIO Roles** preference. This variable allows you to set the desired behavior at an environment level for all artists, rather than allowing individual artists to change the preference. • NUKE_OCIO_ROLES=0 Roles are hidden. • NUKE_OCIO_ROLES=1 Roles are prioritized, which is the same behavior as Nuke 12.2 releases. • NUKE_OCIO_ROLES=2 Roles are de-prioritized. **Note:** If the variable is set, the **Prioritize OCIO Roles** preference is disabled.
NUKE_PATH	The location where files related to Nuke customizations are stored. For more information, see Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts.
NUKE_TEMP_DIR	The location where Nuke saves any temporary files that do not have a particular place defined for them. You can find the current location of Nuke's temporary directory from within Nuke by pressing **X** on your keyboard, when the focus is on the Node Graph, and then running the following TCL command: `getenv NUKE_TEMP_DIR`
NUKE_TOPDOWN	When enabled, use the new top-down rendering project setting to render the full frame at the cost of more memory. Although top-down rendering produces the full frame faster, it disables progressive rendering and uses

Environment Variable	Description
	more memory. You may want to use the legacy method if you're only interested in the first few scan lines in the Viewer. **Note:** The render mode is only saved as part of the script if you set **render mode** to **top-down** in the **Project Settings**. This ensures that the chosen render method is used the next time the script is opened. Using the **NUKE_TOPDOWN** environment variable does not save the render mode in the script.
NUKE_ WINDOWMANAGER_ DEBUG	When enabled, data from Nuke's window manager is printed to the command line or Terminal.
OCIO	Set this variable to the location of your OCIO configuration file for color conversion. **Note:** If you plan to use the OCIO config file specified in the Preferences, ensure that the **Preferences** > **Project Defaults** > **Color Management** > **Export** > **use OCIO nodes when exportingto a Comp** checkbox is enabled.
OFX_PLUGIN_PATH	The location where Nuke looks for OFX plug-ins. For more information, see Loading OFX Plug-ins.
QT_AUTO_SCREEN_SCALE_ FACTOR	Controls whether or not automatic interface scaling for high resolution screens is enabled. On Windows, this variable is enabled (1) by default. On Linux distributions, scaling is currently disabled (0) by default. **Note:** macOS handles scaling automatically, so this variable is not required.
QT_COMPRESS_TABLET_ EVENTS	Due to recent updates to Qt, running Nuke on CentOS 7 Linux distributions with a tablet can cause lag when moving Roto shapes around the Viewer. Setting this environment variable compresses tablet events, eliminating the lag.

Environment Variable	Description
QT_SCALE_FACTOR	Sets the automatic interface scaling factor when QT_AUTO_SCREEN_SCALE_FACTOR is enabled. You can set scaling to **1**, **1.5,** or **2**. The recommended scaling factor is **1.5**. **Note:** Other scaling factors may work but have not been tested.
QT_SCREEN_SCALE_FACTORS	In multi-monitor setups, sets the interface scale independently by screen using the QT_SCREEN_SCALE_FACTORS variable. Scaling uses the same recommended factors, separated by ; (semicolon). For example, QT_SCREEN_SCALE_FACTORS="1.5;1" where the first monitor is higher resolution than the second.
QT_PLUGIN_PATH	The location where Nuke looks for custom Qt libraries if you don't want to use those shipped with Nuke. Setting this environment variable adds the custom path to Nuke's Qt library paths.
TIMELINE_DISABLE_PBO_UPLOADS	When enabled, the performance benefit from using pixel buffer objects (PBO) for texture uploads from RAM to the GPU is disabled. You can try disabling PBOs if you notice playback degradation.

Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts

On start-up, Nuke scans various directories for files that customize the behavior of Nuke. It looks for information on favorite directories, menu options, image formats, gizmos, NDK plug-ins, Python scripts, generic Tcl scripts, and preferences.

Warning: It's worth saying that you should edit Python files with care as mistakes could stop Nuke from running.
For more information on Python in Nuke, see The Script Editor and Python or the Python Developer's Guide (**Help** > **Documentation**).

To make your customizations available to all versions of a particular release, place them in the following directories:

- Linux: /usr/local/Nuke/<version>/plugins/
- Mac: /Library/Application Support/Nuke/<version>/plugins/
- Windows: C:\Program Files\Common Files\Nuke\<version>\plugins\

If you want Nuke to look for plug-ins somewhere else rather than in these default locations, you can also define a common plug-in path yourself. Thus, by defining the Nuke plug-in path, you can assign yourself a common shared directory from which to control Nuke for multiple artists. See Defining the Nuke Plug-in Path for more information.

Nuke also looks in specific sub-directories of your home directory and the Nuke application directory in the order shown by platform:



Note: If you place customizations in the application directories, they'll only be available for that release.

- Linux:

  /usr/local/Nuke17.0v3/plugins

  /usr/local/Nuke17.0v3/plugins/nukescripts

  /home/*login name*/.nuke
- Mac:

  /Applications/Nuke17.0v3/Nuke17.0v3.app/Contents/MacOS/plugins

  /Applications/Nuke17.0v3/plugins/nukescripts

  /Users/*login name*/.nuke
- Windows:

  drive letter:\Program Files\Nuke17.0v3\plugins\nukescripts or

  drive letter:\Program Files\Nuke17.0v3\plugins

  ~\.*nuke*



Note: On Windows, the **.nuke** folder can be found under the directory pointed to by the HOME environment variable. If this variable is not set (which is common), the .nuke directory will be under the folder specified by the USERPROFILE environment variable - which is generally of the form *drive letter*:\Documents and Settings*login name*\ or *drive letter*:\Users*login name*\

To find out if the HOME and USERPROFILE environment variables are set and where they are pointing at, enter %HOME% or %USERPROFILE% into the address bar in Windows Explorer. If the environment variable is set, the folder it's pointing at is opened. If it's not set, you get an error.

In addition to setting up your Python/plugin environment using the **init.py** and **menu.py** files, Nuke Studio and Hiero also allows you to run Python code automatically on start-up. The default location for this Python code is:

```
<STARTUP_PYTHON_PATH>/Python/Startup
```

Where by default, <STARTUP_PYTHON_PATH> is $HOME/.nuke as described in the OS breakdown above.



Article: Have a look at this Knowledge Base article for more detailed information.

Any Python **.py** modules or packages containing __init__.py found within the /Python/Startup directory is imported when the application launches, and added to the hiero.plugins namespace.

Defining the Nuke Plug-in Path

To define the plug-in path:

1. On each artist's machine, create an environment variable called **NUKE_PATH**.
2. Assign the **NUKE_PATH** environment variable to the path name of the directory where files related to Nuke customization will reside.

   For example, on Mac using a csh or tcsh shell:

   ```
   setenv NUKE_PATH /SharedDisk/Nuke
   ```

   Or, if you're using a bash or ksh shell:

   ```
   export NUKE_PATH=/SharedDisk/Nuke
   ```

Loading a plug-in (plugin_find()) searches the NUKE_PATH until the first plug-in is located, ensuring that only the most local plug-in is loaded. For example if the NUKE_PATH variable contains:

project_dir:studio_dir:company_dir

The path is searched in the following order until the first plug-in is located:

- ~/.nuke
- project_dir
- studio_dir
- company_dir
- nuke_dir

However, the NUKE_PATH environment variable is parsed in reverse order when loading **init.py** and **menu.py** and all discovered copies are used. This allows localized settings to override more global ones. So, with the directory hierarchy above, **init.py** scripts would execute as follows:

- nuke_dir/init.py
- company_dir/init.py
- studio_dir/init.py
- project_dir/init.py
- ~/.nuke/init.py

Loading OFX Plug-ins

On start-up, Nuke scans various directories for OFX plug-ins that bring additional functionality to Nuke. Paths to these directories vary between operating systems, but here are examples of where you may find them:

- **Linux:**

  /usr/OFX/Nuke

  /usr/OFX/Plugins

- **Windows:**

  C:\Program Files\Common Files\OFX\Nuke

  C:\Program Files\Common Files\OFX\Plugins

- **Mac:**

  /Library/OFX/Nuke

  /Library/OFX/Plugins

If you want Nuke to look for OFX plug-ins somewhere else you can. Just define the environment variable OFX_PLUGIN_PATH to point to the new directory.

For example, on Mac using a csh or tcsh shell:

```
setenv OFX_PLUGIN_PATH /SharedDisk/OFX
```

Or, if you're using a bash or ksh shell:

```
export OFX_PLUGIN_PATH=/SharedDisk/OFX
```

Defining Common Favorite Directories

Favorite directories can be accessed by artists with a single click from any Nuke file browser. Typically you would create these favorites for common directories on a project.

To Define a Common Set of Favorite Directories

To define an common set of favorite directories:

1. Create a file called **menu.py** in your plug-in path directory.

   For more information on plug-in path directories, see Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts.
2. Add an entry in the following format:

   ```
   nuke.addFavoriteDir('DisplayName', 'Pathname')
   ```

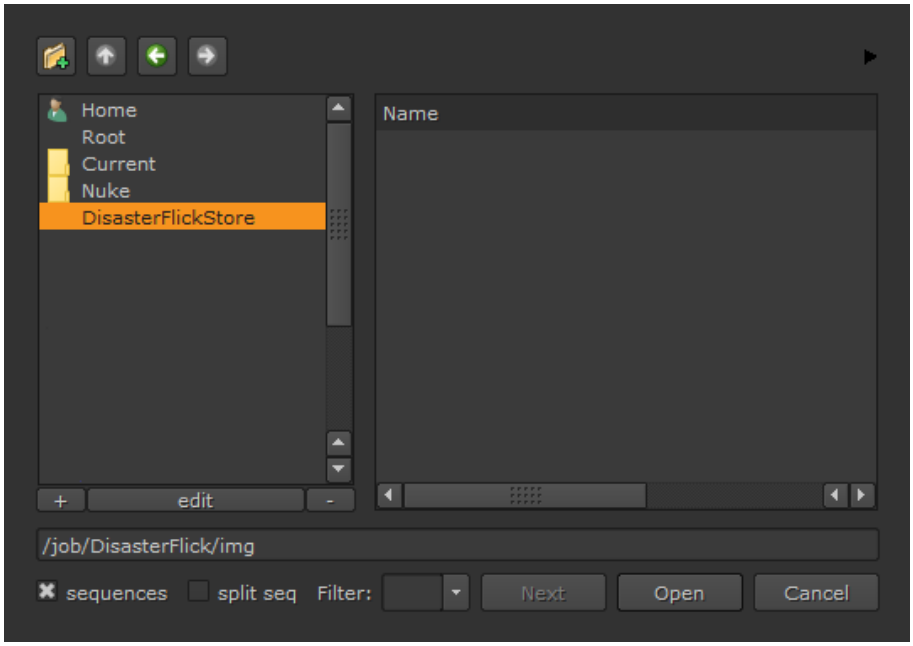
   - Replace **DisplayName** with the string you want as the display name for the directory link, for example **'Home'** or **'Desktop'**.
   - Replace **Pathname** with the path name to the directory to which the favorite should point.
3. In the above entry, you can also add the following optional arguments after **'Pathname'**:
   - **type**. This is an integer and a bit-wise operator, OR a combination of **nuke.IMAGE**, **nuke.SCRIPT** or **nuke.FONT**.
     - **nuke.IMAGE** restricts the favorite directory to appear only in the image file browser, which Nuke opens for file Read/Write operations.
     - **nuke.SCRIPT** restricts the favorite directory to appear in the script file browser, which appears when you select **File** > **Open Comp** or a similar menu selection to open or import script files.
     - **nuke.FONT** restricts the favorite directory to appear in the font browser.
   - **icon='Name'.** Replace **Name** with the name and file extension of the **.png** (or **.xpm**) image you wish to use as the icon for the favorite directory. This image must be stored in your Nuke plug-in path directory. It should be 24 x 24 pixels in size.

- **tooltip='My tooltip'**. Replace **My tooltip** with the string you wish to display as pop-up help.

Example 1

The following entry would create a favorite called **DisasterFlickStore** which appears on all File Browsers invoked from Read nodes and which points to the /job/DisasterFlick/img directory.

```
nuke.addFavoriteDir ('DisasterFlickStore', '/job/DisasterFlick/img',
nuke.IMAGE)
```

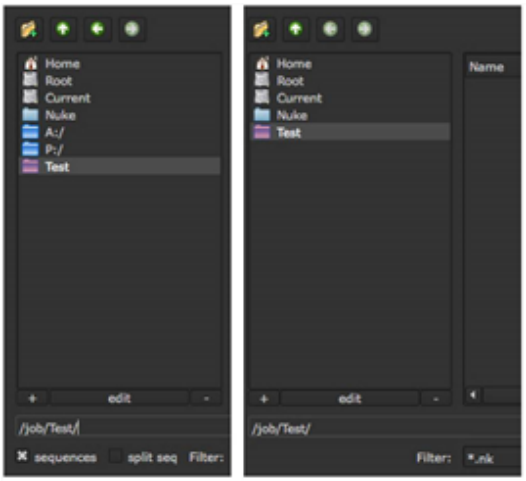


The result of example 1.

Example 2

The following entry would create a favorite called **Test**. It appears on all File Browsers invoked from Read nodes or by selecting **File** > **Open Comp** and points to the /job/Test directory. The entry also defines *Test Images and Scripts* as the tool tip for the favorite directory.

```
nuke.addFavoriteDir ('Test', '/job/Test', nuke.IMAGE | nuke.SCRIPT,
tooltip='Test images and Scripts')
```



The result of example 2.

Handling File Paths Cross-Platform

If your facility uses Nuke on several operating systems, you may want to configure Nuke to replace the beginnings of file paths so that scripts created on one platform also work on another.

For example, to ensure file paths created on Windows also work on Linux for a particular version of Nuke, you can do the following:

1. Create a file called **init.py** in your plug-in path directory if one doesn't already exist.
    - Linux: /usr/local/Nuke/<version>/plugins/
    - Mac: /Library/Application Support/Nuke/<version>/plugins/
    - Windows: C:\Program Files\Common Files\Nuke\<version>\plugins\

    For more information on plug-in path directories, see Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts.
2. Open the **init.py** file in a text editor and add an entry in the following format:

```
import platform

def filenameFix(filename):
   if platform.system() in ("Windows", "Microsoft"):
      return filename.replace( "/SharedDisk/", "p:\\" )
   return filename.replace( "p:\\", "/SharedDisk/" )
```

This way, the Windows file paths (beginning with **p:** in the above example) are replaced with the Linux file paths (beginning with **/SharedDisk/**) under the hood whenever a Nuke script is used on Linux. Otherwise, the Windows file paths are used.

Note that the file paths displayed in the graphical user interface (GUI) do not change. If you are using **p:** in a node control, this is still displayed as **p:**. However, on Linux, Nuke interprets **p:** as **/SharedDisk/**.

Setting Default Values for Controls

You can set default values for node controls by adding a simple line of Python to your **init.py** file. Once a default value is set, all controls with matching names default to this value. For example, you can set default values for file format specific controls in the Read, Write or other file format-dependent nodes. To set a default value, use the following statement:

```
nuke.knobDefault()
```


Tip: If you want to know the name of a particular knob, hover over the knob to display its tooltip. The knob name is in bold type.
You can also copy/paste a node into a text editor, but knobs with default values are not displayed. Changing the default value in the node's **Properties** panel adds it to the pasted text.

To specify file format specific defaults, use the class name, followed by the file format extension and the control name, all separated by periods. For example:

```
nuke.knobDefault("Read.exr.compression", "2")
```

Maybe you want the last frame value of the frame range controls in the **Project Settings** to default to frame 200. To do this, use the following statement:

```
nuke.knobDefault("Root.last_frame", "200")
```

Or you might want to set a different default font style in the Text node:

```
nuke.knobDefault("Text2.font", "{ Arial : Regular : arial.ttf : 0 }")
```

Defining Custom Menus and Toolbars

You can freely add custom menus and menu options as well as toolbars and toolbar options to the Nuke interface. Artists can then use these options to trigger gizmos and plug-ins stored in the plug-in path directory.

For example, to add a new menu in the default Toolbar with an option to trigger a gizmo called MyGizmo, you can do the following:

1. In your home directory, create a directory called **.nuke** (if it doesn't exist already). For more information on this directory, see Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts.
2. In the **.nuke** directory, create a file called **menu.py** if one does not already exist.
3. In a text editor, modify the file **menu.py**, adding the lines:

   ```
   toolbar = nuke.toolbar("Nodes")
   toolbar.addCommand( "Test/MyGizmo", "nuke.createNode('MyGizmo')")
   ```

   This adds a menu labeled "Test" to the default Nodes Toolbar with an item labeled "MyGizmo" that creates an instance of the node MyGizmo.



Note: Node **Class()** names occasionally change between major releases, such as Nuke 7 to Nuke 8. While these changes do not affect legacy scripts, you may not get the results you were expecting if a node class has been modified. The **toolbars.py** file, used to create Nuke's node toolbar, contains all the current node class names and is located in **<install_directory>/plugins/nukescripts/** for reference.

As an example, between Nuke 7 and Nuke 8, the Text node **Class()** changed from Text to Text2. In the **toolbars.py** file for the two releases, the entries for the Text node appear as follows:

```
m.addCommand("Text", "nuke.createNode(\"Text\")", icon="Text.png")
m.addCommand("Text", "nuke.createNode(\"Text2\")", icon="Text.png")
```

It's also possible to add items to other menus in Nuke and even create your own toolbars. The following sections cover these possibilities in detail.

To Add a Toolbar

To add a toolbar:

1. Create a file called **menu.py** in your plug-in path directory if one doesn't already exist.

   For more information on plug-in path directories, see Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts.

2. Open the **menu.py** file in a text editor and add an entry in the following format:

   ```
   t=nuke.toolbar("ToolbarName")
   t.addCommand("NewMenu", "PythonCode", "Shortcut", icon="IconName")
   ```

   - Replace **ToolbarName** with the name you want to give to the toolbar. This name appears in the content menus under **Windows** > **Custom** and above the toolbar on the title tab.
   - Replace **NewMenu** with the name of the menu you want to add to the toolbar.
   - Replace **PythonCode** with relevant Python code (usually **nuke.createNode**), and, if necessary, use the name of the gizmo, generic Python script, or plug-in file you want the menu option to invoke. For ease of use, place all such referenced files inside the plug-in path directory.

     If you like, you can also replace **PythonCode** by a Python callable.
   - Replace **Shortcut** with a keyboard shortcut, for example **Alt+A**, **Ctrl/Cmd+A**, or **Shift+A**. The letter **a** alone represents lower-case **a**. **F1** represents function key 1. You can combine the **Shift**, **Ctrl/Cmd**, and **Alt** keys as necessary. If you like, you can also use **#A** to represent **Alt+A, ^A** to represent **Ctrl/Cmd+A**, and **+A** to represent **Shift+A**.
   - Replace **IconName** with the name of the **.png** (or **.xpm**) image you wish to use as the menu icon. This image must be stored in your Nuke plug-in path directory. It should be 24 x 24 pixels in size.

3. In the above entry, you can also add the following optional arguments in the parenthesis after **"ToolbarName"**:
   - **True**. This is the default. When True, nuke.toolbar() calls the toolbar with the given name or creates it if it does not exist. For example, **t=nuke.toolbar("Extras", True)** would either call an existing toolbar called Extras or create one if it did not already exist.
   - **False**. When False, the toolbar is not created if it does not already exist and nuke.toolbar() returns **None**. You can use this to find out if a toolbar with a given name already exists. For example, **t=nuke.toolbar("Extras", False)** would either call an existing toolbar called Extras or return None if such a toolbar did not exist.

The new toolbar does not appear by default, but is listed under **Custom** in the content menus. From there, you can insert it in any pane. Once you are happy with the new toolbar and its position, save the layout (select **Workspace** > **Save Workspace**). Thereafter, the toolbar appears whenever Nuke is launched in the saved workspace.

You can build several toolbars for different tasks and save layouts that have one or another present for easy context switching.

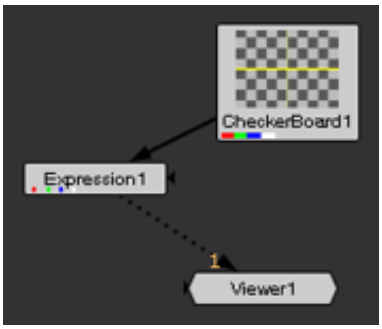
Example 1

The following entry creates a new toolbar called **Extras**. The toolbar includes an option called **Create VectorBlur** that creates a VectorBlur node. The entry also defines **v** as the keyboard shortcut for the VectorBlur node.

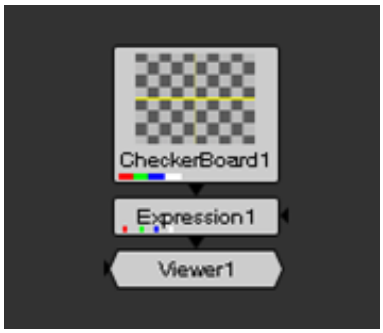
```
t=nuke.toolbar("Extras")
t.addCommand("Create VectorBlur", "nuke.createNode ('VectorBlur')", "v")
```

Example 2

In this example, we add an option called Autoplace to the toolbar created in example 1. This option places the selected nodes neatly one after another, as illustrated in the following images:



Before using Autoplace to tidy up the Node Graph.



After using Autoplace to tidy up the Node Graph.

The following entry adds the Autoplace option. It also defines **Alt**+**A** as the keyboard shortcut for this option.

```
def _autoplace():
n = nuke.selectedNodes()
for i in n:
nuke.autoplace(i)
t=nuke.toolbar("Extras")
t.addCommand("Auto&place", "_autoplace()", "Alt+a")
```

To Define a Menu or Toolbar Option

To define a menu or toolbar option:

1. If you haven't already done so, create a file called **menu.py** in your plug-in path directory. For more information on plug-in path directories, see Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts.
2. Open the **menu.py** file in a text editor and add an entry in the following format:

```
menubar=nuke.menu("MenuType")
m=menubar.addMenu("&NewMenu")
m.addCommand("&NewItem", "PythonCode", "Shortcut", icon="IconName", index=#)
```

   - Replace **MenuType** with the type of menu or toolbar you want to add an item to:

     Nuke adds an item to the application main menu bar.

     Animation adds an item to the menu on the Animation button  of all panels, and to the right-click menu of the Curve editor.

     Properties adds an item to the right-click menus of properties panels.

     Node Graph adds an item to the right-click menu of the Node Graph.

     Nodes adds an item to the default Toolbar.

     Viewer adds an item to the right-click menu of the Viewer.

     Pane adds an item to the content menus where it appears under **Custom**.
   - Replace **NewMenu** with the menu name. Using an existing menu name appends any new options to the existing menu. You can also add options to the default Menu bar and Toolbar.
   - Replace **NewItem** with the underlying item you want to add to the menu. You may precede any character with an **&** in order to flag it as keyboard shortcut trigger.
   - Replace **PythonCode** with relevant Python code (usually **nuke.createNode**) and, if necessary, use the name of the gizmo, generic Python script, or plug-in file you want the menu option to invoke. For ease of use, place all such referenced files inside the plug-in path directory.

     For more information on plug-in path directories, see Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts.

     If you like, you can also replace **PythonCode** by a Python callable. This has the advantage that you get informed about errors in your script on start-up instead of when the menu item is invoked. For an example of using a lambda function, see Example 3.
   - Replace **Shortcut** with a keyboard shortcut, for example **Alt+A**, **Ctrl/Cmd+A**, or **Shift+A**. The letter **a** alone represents lower-case **a**. **F1** represents function key 1. You can combine the **Shift**, **Ctrl/Cmd**, and **Alt** keys as necessary. If you like, you can also use **#A** to represent **Alt+A,** **^A** to represent **Ctrl/Cmd+A**, and **+A** to represent **Shift+A**.

Note: By assigning a keyboard shortcut, you can overwrite existing shortcuts. For example, if you assign the shortcut **Ctrl**/**Cmd**+**O** to a new menu item, it is no longer used for it's default purpose, which is opening a file. However, shortcuts are only overwritten in the main menu bar, the Toolbar, any user-created toolbars, and the menu you are adding the new menu item to. This means you can add a shortcut into the Node Graph, for example, without resetting the same shortcut in the Viewer. However, you cannot add a shortcut into the Node Graph without resetting the same shortcut in the main menu bar or the Toolbar.

- Replace **IconName** with the name of the .png (or .xpm) image you wish to use as the menu icon. This image must be stored in your Nuke plug-in path directory. It should be 24 x 24 pixels in size.
- Replace **#** with a number that represents the position of the item in the menu or toolbar. If you do not use an index keyword, the item is added in the end of the menu or toolbar.

Tip: You can also put the menu name in the addCommand call, like this:
nuke.menu("MenuType").addCommand("NewMenu/NewItem", "PythonCode("name")")

Example 1

The following entry creates a new menu and option called **Custom** > **Cue Render** in the menu bar. It inserts a gizmo called "cue_render." The entry also defines **Ctrl+R** as the keyboard shortcut for the gizmo.

```
menubar=nuke.menu("Nuke")
m=menubar.addMenu("&Custom")
m.addCommand("&Cue Render", "nuke.createNode('cue_render')", "Ctrl+R")
```



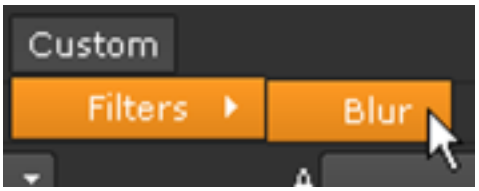
Example 2

For information on how to create a new menu in the default Toolbar with a menu item that triggers a gizmo, see the example under Defining Custom Menus and Toolbars.

Example 3

The following entry creates a menu and options called **Custom** > **Filters** > **Blur** in the menu bar. Selecting **Blur** inserts the Blur node.

```
menubar=nuke.menu("Nuke")
m=menubar.addMenu("&Custom")
m.addCommand("Filters/Blur", "nuke.createNode(\"Blur\")" )
```



You can also do the same with a lambda function:

```
menubar=nuke.menu("Nuke")
m=menubar.addMenu("&Custom")
m.addCommand("Filters/Blur", lambda: nuke.createNode("Blur") )
```

This way, you don't have to use the backslashes.

Defining Common Image Formats

You may wish to define image formats (image resolutions and corresponding pixel aspect ratios) for a given project. These appear as dropdown menus on Read and Reformat nodes.

Defining an Image Format

To define an image format:

1. If you haven't already done so, create a file called **menu.py** in your plug-in path directory.

   For more information on plug-in path directories, see Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts.
2. Add an entry in the following format:

   ```
   nuke.addFormat(" ImageWidth ImageHeight LowerLeftCorner LowerRightCorner
   UpperRightCorner UpperLeftCorner PixelAspectRatio DisplayName ")
   ```

   - Replace **ImageWidth**with the width (in pixels) of the image format.
   - Replace **ImageHeight**with the height (in pixels) of the image format.

- If you wish to define an image area that is smaller than the format resolution, replace **LowerLeftCorner, LowerRightCorner, UppperRightCorner, UpperLeftCorner** with the proper x and y coordinates (in pixels) for the lower left, lower right, upper right, and upper left corners of this smaller image area (optional).
- Replace **DisplayName**with display name of the format. This appears on all relevant dropdown menus.

Example

The following entry would create a new format called **full_aperature_anamorphic** with a resolution of 2048 by 1556 and a pixel aspect ratio of 2.0. (As the image corners are not explicitly defined, the image covers the entirety of the format area.)

```
nuke.addFormat (" 2048 1556 2.0 full_aperture_anamorphic ")
```

Creating and Accessing Gizmos

Nuke enables artists and technical directors to create gizmos, which are groups of Nuke nodes that may be reused by other artists. Studios commonly use gizmos to consistently apply certain color grading techniques, process incoming footage according to a particular conversion formula, and process outgoing footage in preparation for film printing.

A gizmo is a Group Node that you create and save in a separate **.gizmo** file in your Nuke plug-in folder. Nuke scripts can use this gizmo just like any other node type. Saved scripts only contain the name and control settings for the gizmo; the definition is in the gizmo file and it is read at the same time the script is loaded into Nuke. Thus, you can alter the implementation of the gizmo and change all the scripts that are using it.



Note: Unlike other nodes, gizmos cannot be cloned. For more information, see Working with Nodes.

Tip: You can register gizmos as soft effects in Nuke's Timeline environment. See Adding Effects on the Timeline for more information.

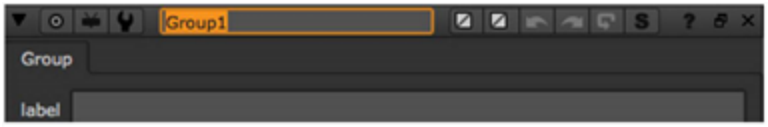
Creating Gizmos

A gizmo is a Group Node that you create and save in a separate **.gizmo** file in your Nuke plug-in folder. Nuke scripts can use this gizmo just like any other node type.

Using Nuke's **Export gizmo** command, you can export nodes inside a Group and explicitly control which controls may be edited by the artists to ensure the processes within the gizmo are consistently applied.

To create a gizmo:

1. Select the nodes you want to include in the gizmo.
2. Select **Other** > **Group** from the Toolbar (or press **Ctrl**/**Cmd**+**G**) to group the nodes.
3. You may want to rename the group by entering a new name in the Group properties panel title field. This step is optional, and has no effect on the saved gizmo. However, it is often a good idea to give the Group the same name you intend to use for the gizmo.



4. To expose which controls the artists can adjust, see Managing Gizmo Controls.
5. Click the **export as gizmo** button.
6. In the file browser that appears, click **Home**. Type **.nuke/** after the path displayed at the bottom of the file browser.
7. Enter a name after the path, and append a **.gizmo** extension after the name. This is the name of the command that is written to any saved script that's using the gizmo. It's a good idea, and common practice, to begin the name with a capital letter, because Nuke uses this as an indication that the command is a node or a gizmo.
8. Click **Save**.

   See Accessing Gizmos in Nuke for information on how to access your gizmo in Nuke.



Note: Nuke does not allow you to **Overwrite** and **Save as** gizmos by default, without copying the gizmo to a Group. If you want to allow this behavior, so artists don't need to copy a gizmo before editing it, set the NUKE_ALLOW_GIZMO_SAVING environment variable to **1**. See Environment Variables for more information.

Managing Gizmo Controls

You can add controls to your gizmo by picking and editing from the default controls that exist for the nodes inside your Group node or by adding a control you have created yourself to your gizmo **Properties** panel.

In Nuke's **Properties** panel, you can add controls using the Adding Knobs Using Drag-and-Drop edit button or by right-clicking in the panel and selecting Adding Knobs Manually. Drag-and-drop knobs significantly reduce the time spent managing user knobs in a node's **Properties** panel when compared to the legacy **Manage User Knobs** method within Nuke.

Adding Knobs Using Drag-and-Drop

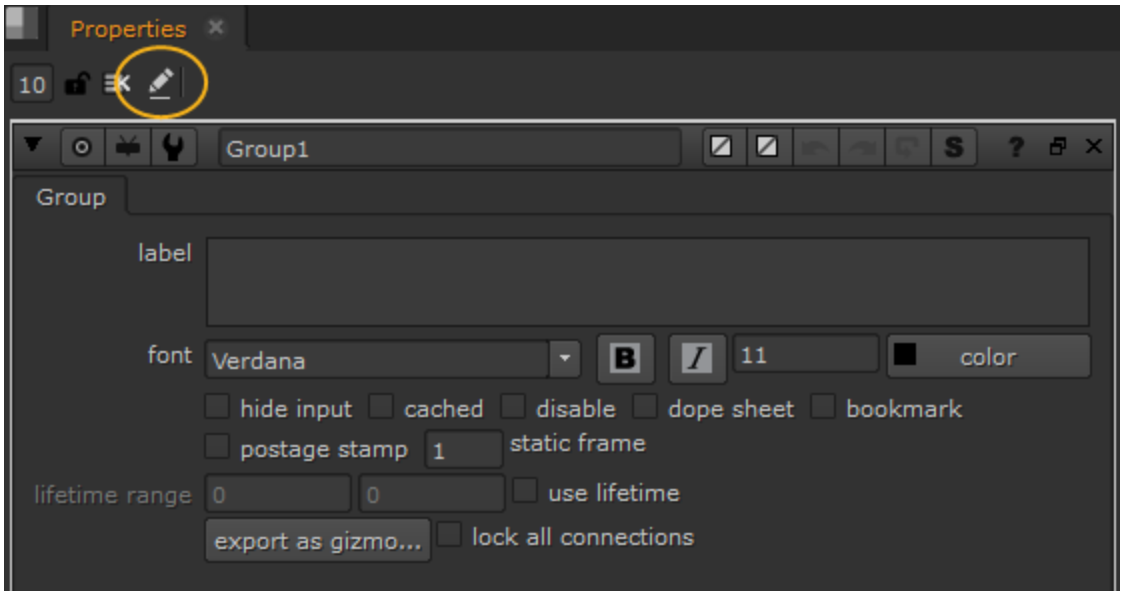
You can drag-and-drop knobs between open node **Properties** panels or add your own using the knob icons listed at the top of the panel. You can also order, hide, customize, and delete knobs within the panel.

Note: The drag-and-drop interface is also useful for managing LiveGroup knobs. See Using the LiveGroup Node for more information.

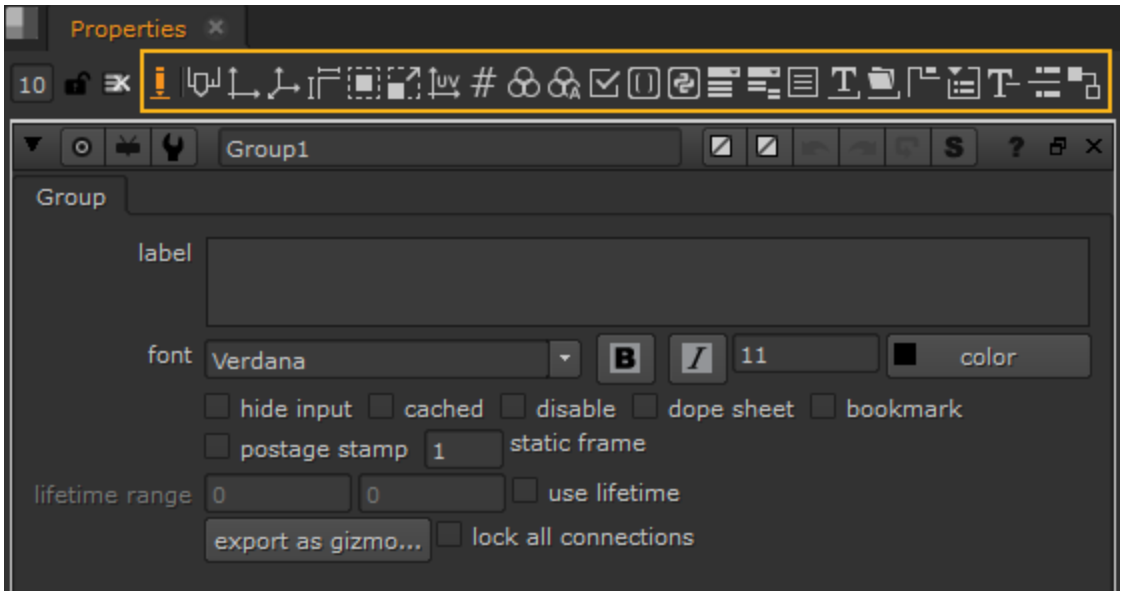
Tip: You can float the **User Knob Editor** by clicking the content menu at the top of the pane and selecting **Windows** > **User Knob Editor**.

To add knobs to a Group:

1. Click the edit button at the top of the Group **Properties** panel.

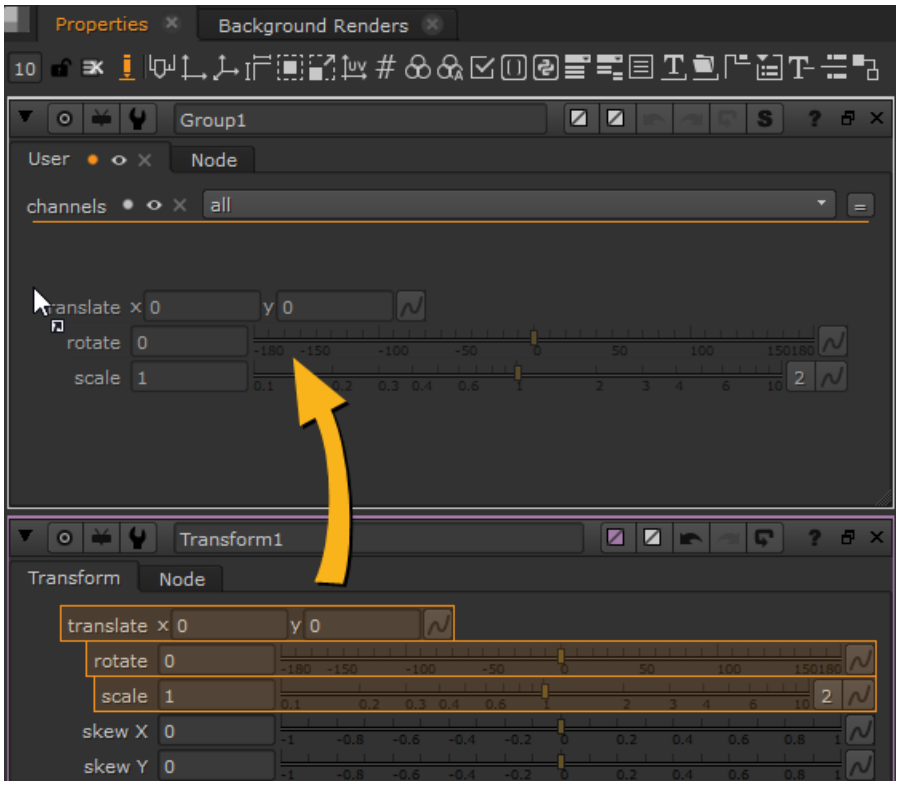


The **User Knob Editor** is displayed and the node's properties are now editable.



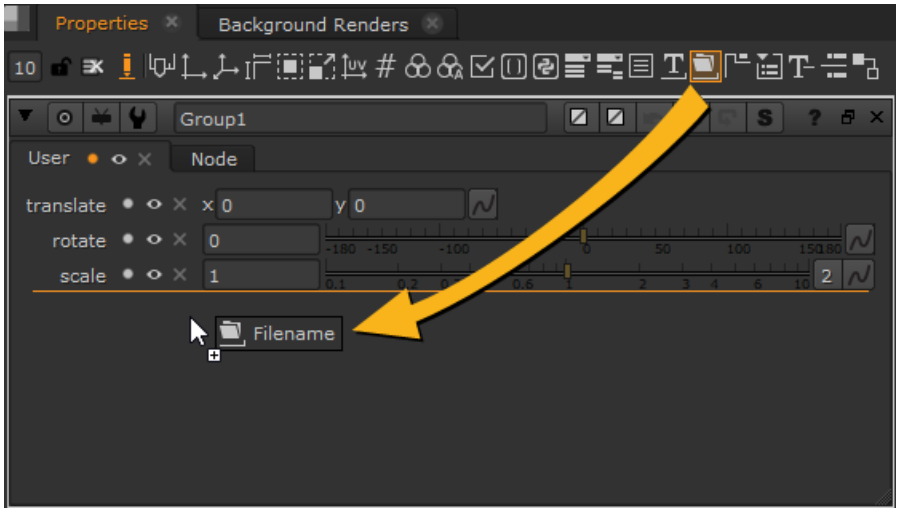
2. To add a knob, either:

   Open the **Properties** panel of the node containing the knob(s) you want to expose and drag-and-drop the knob(s) onto the Group node's properties,

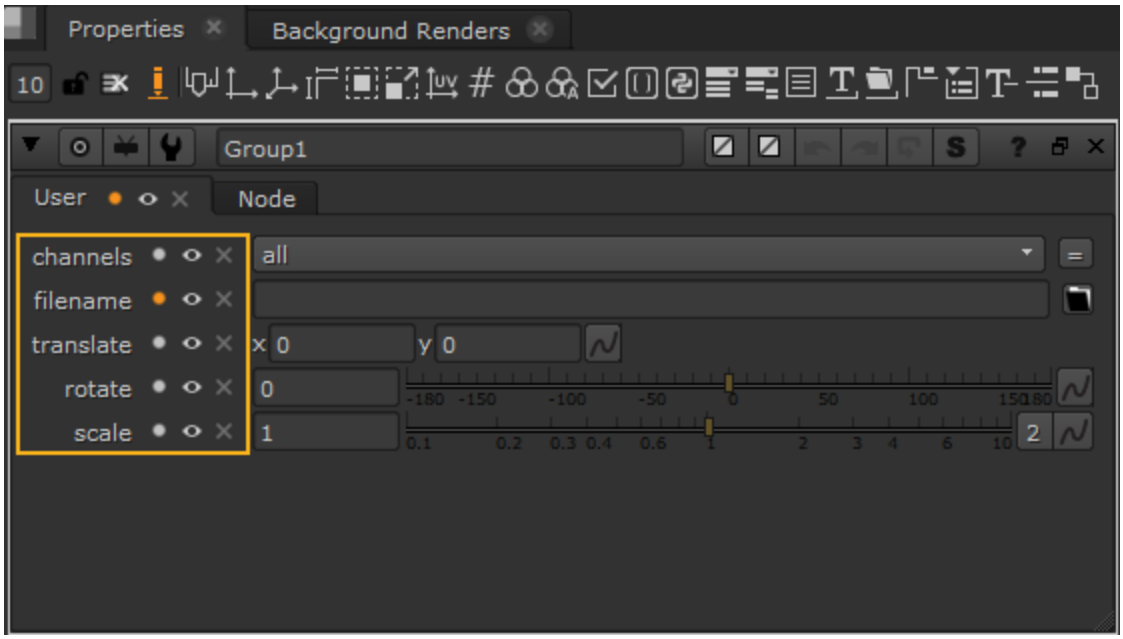


OR

Drag-and-drop a new knob from the **User Knob Editor** onto the Group node's properties.



Knobs added from nodes in the Group are marked with a gray dot and new knobs are marked with an orange dot.



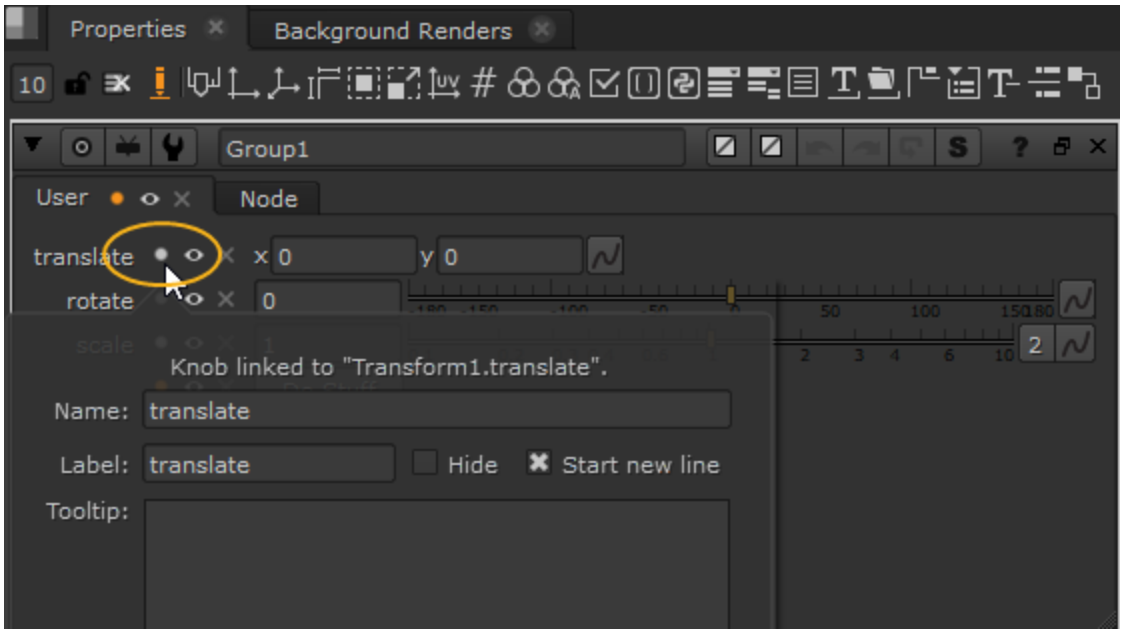
See Organizing Knobs for information on how to rearrange knobs in the panel.

3. Click the edit button at the top of the Group **Properties** panel to exit edit mode.

   See Accessing Gizmos in Nuke for information on how to access your gizmo in Nuke.

Editing Knobs

Knobs added to the **Properties** panel are edited by clicking the gray or orange dot next to the knob label.

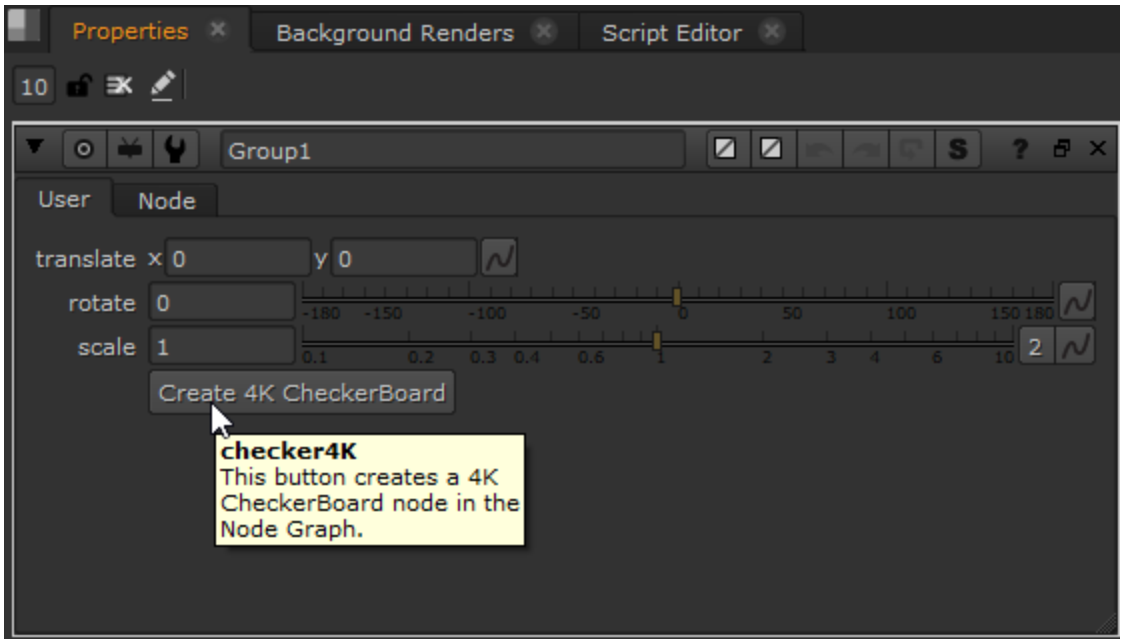


The components you can edit depend on the knob type. For example, a **translate** knob from a node within the Group allows you to edit the following:

- **Name** - the scripting name used to call the knob from scripts and expressions. This field is alphanumeric and cannot contain spaces.
- **Label** - the display name for the knob in the Nuke interface.
- **Tooltip** - the information displayed when you hover the pointer over the knob.

Tip: The Nuke Reference Guide lists the labels and knob names for the majority of Nuke nodes.

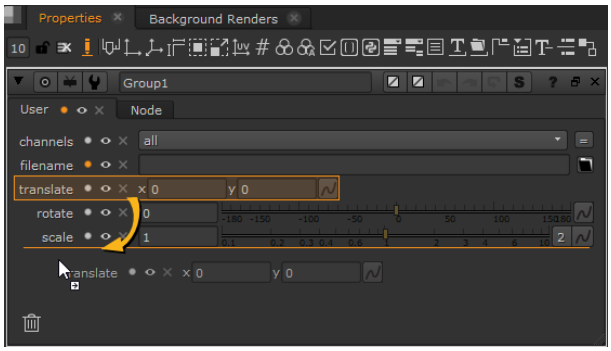
A more complex example is the **Python Script Button** knob from the **User Knob Editor**. Adding a Python knob to the **Properties** panel allows you to edit an additional component, **Script**. The **Script** field adds Python to the knob, which is executed when you click the button. For example:

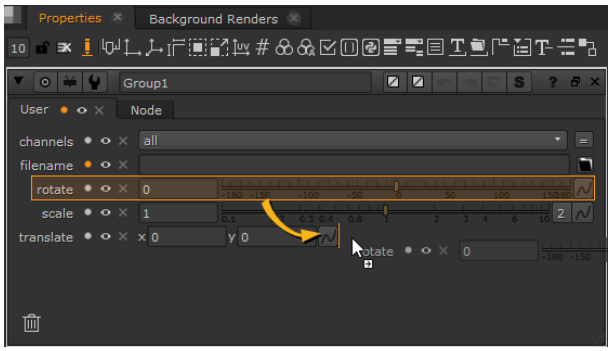


Click the edit button at the top of the Group **Properties** panel to exit edit mode. See Accessing Gizmos in Nuke for information on how to access your gizmo in Nuke.

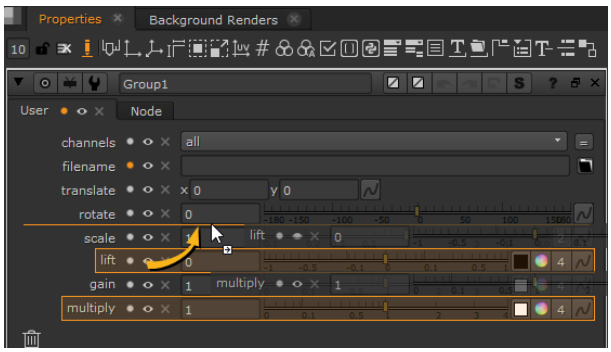
Organizing Knobs

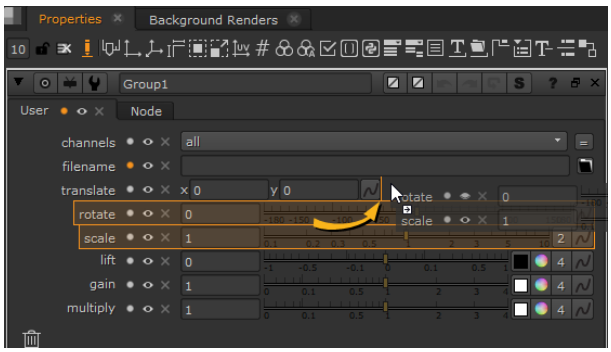
You can re-order, hide, and remove knobs quickly using the drag-and-drop interface. When you're dragging knobs around the **Properties** panel, an orange guide line indicates where the knob is placed when dropped.



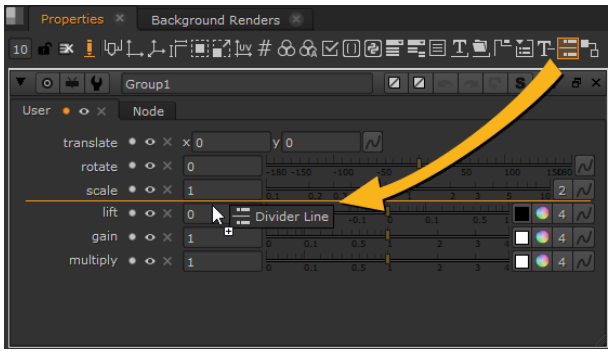


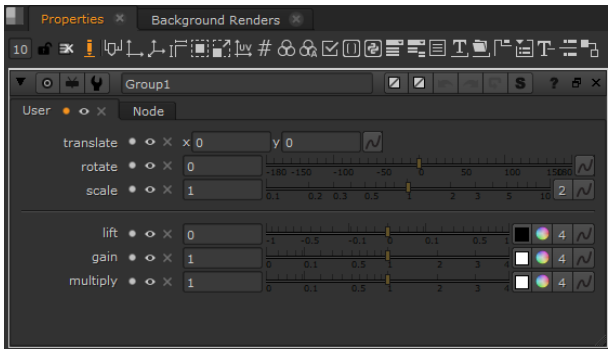
You can drag-and-drop multiple knob selections in the same way. Hold **Ctrl/Cmd** and click to select multiple knobs in the **Properties** panel.





You can add dividing lines between controls to delineate between groups of knobs, such as transform and color knobs. To add a divider, drag-and-drop a **Divider Line** knob between the controls you want to separate.





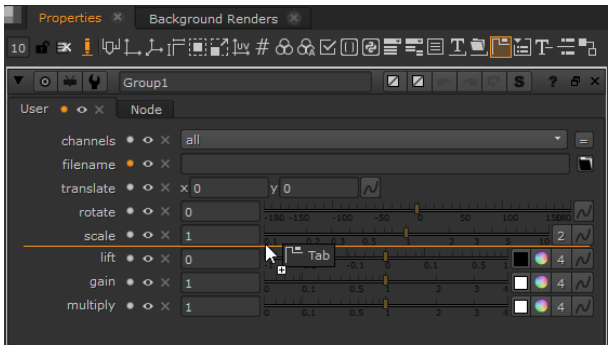
Click the edit button  at the top of the Group **Properties** panel to exit edit mode. See Accessing Gizmos in Nuke for information on how to access your gizmo in Nuke.

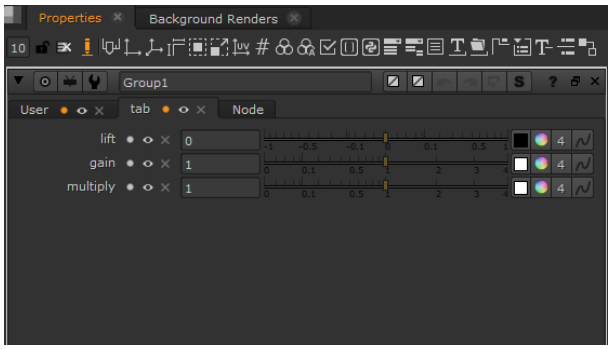
Adding Tabs

You can add tabs in the **Properties** panel to organize knobs into groups, just like Nuke's existing nodes. To add a tab:

1. Drag a **Tab** knob from the **User Knob Editor** and drop it into the **Properties**.

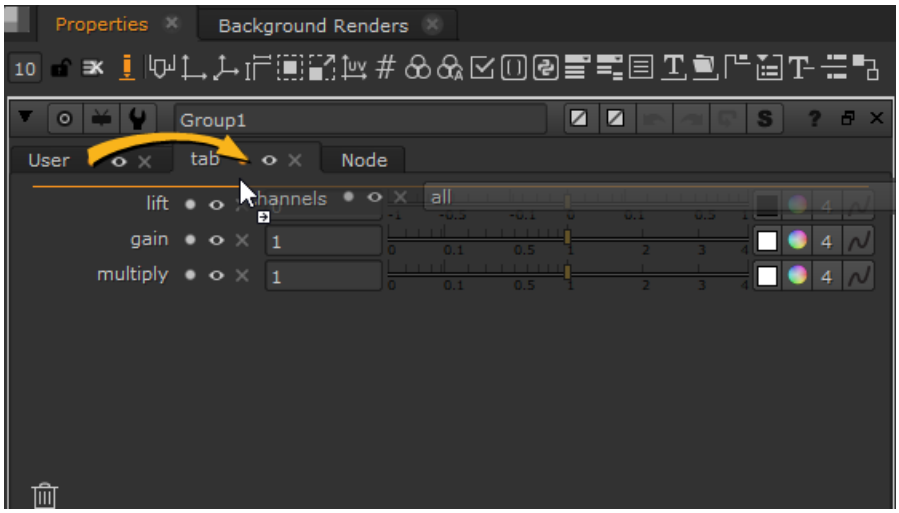
   Dropping a **Tab** knob above other knobs moves those knobs to the new tab.





Tip: If you want to create an empty tab, drop the **Tab** knob at the bottom of the knob list.

2. Drag-and-drop knobs between tabs to organize them as required.



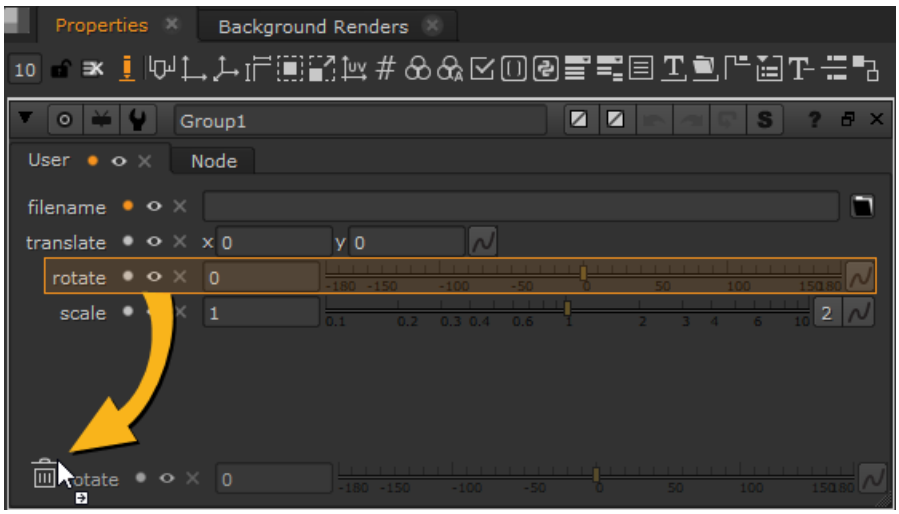
Click the edit button at the top of the Group **Properties** panel to exit edit mode. See Accessing Gizmos in Nuke for information on how to access your gizmo in Nuke.

Hiding and Removing Knobs

You can hide knobs by clicking the icon next to the knob. Hiding a knob doesn't hide it in edit mode, only when you've finished exposing knobs and clicked the edit button .

To remove a knob you can:

- Select the knob and press **backspace** or **delete**,
- Click the button next to the knob, or
- Drag the knob to the bottom-left of the **Properties** panel and drop it into the trash icon.



Click the edit button at the top of the Group **Properties** panel to exit edit mode. See Accessing Gizmos in Nuke for information on how to access your gizmo in Nuke.

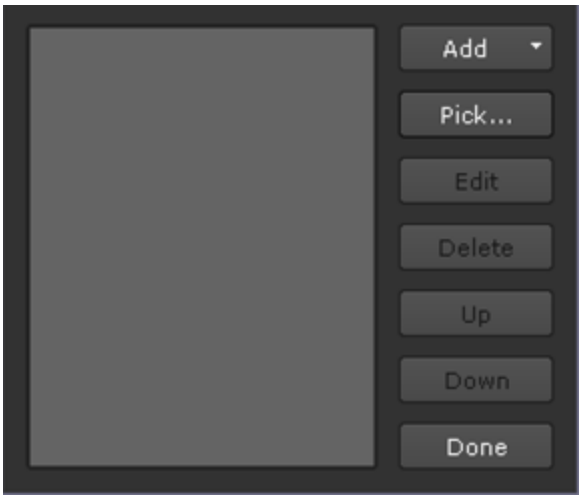
Adding Knobs Manually

You can manually add controls to your Group node before exporting it as a gizmo in two different ways:

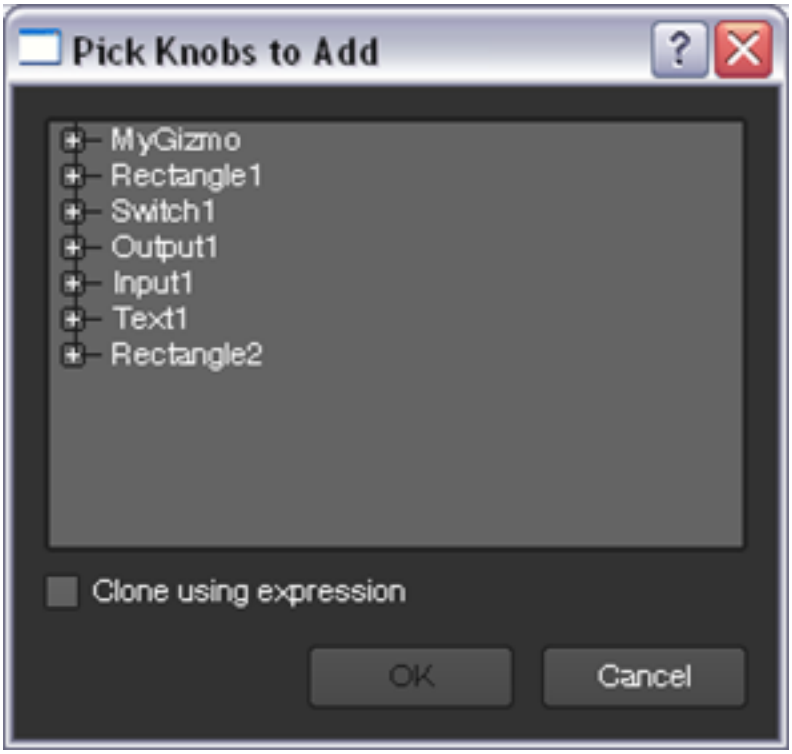
- by picking and editing a control from the default controls that exist for the nodes inside your Group node. For example, if the Group node contains a Grade node, you can add any of the Grade node controls to your gizmo **Properties** panel.
- by adding a control you have created yourself to your gizmo **Properties** panel.

To Pick Existing Controls

1. Right-click on the dark gray background of the Group properties panel and select **Manage User Knobs**. The following dialog opens.



2. To pick a control you want to give the users control of, click on the **Pick** button. This opens a dialog that lists all the nodes that the group contains.



Expand the list items as necessary to see the controls you can include in the gizmo controls. Select a control and click **OK**. You can also select multiple controls by **Ctrl**/**Cmd**+clicking on them, or pick a range of controls by **Shift**+clicking.

At this point, a new tab called **User** appears in the Group properties panel. The control you selected has been added to this tab. In our example, we selected the **size** control of the Text node.

The control has also been added to the Manage User Knobs dialog, ready for you to edit.

3. To edit the control you added, open the **Manage User Knobs** dialog and select the control from the list. Click **Edit**.



In most cases, you can edit the following:

- **Name** - Give the new control a unique name here. You need to use this name whenever you want to reference the control from scripts or via expressions. The name can only contain letters and digits. Spaces or punctuation are not allowed. This field cannot be left empty.
- **Label** - Whatever you enter here appears to the left of the control in the gizmo properties panel (or, in the case of buttons, on the button). If you leave this empty, whatever is in the **Name** field is also used as the label.

  In the **Label** fields of check boxes, Tcl script buttons, and Python script buttons, you can also use HTML. For example, to have your text appear in bold, you can enter **text**.

  To add an icon to your check box or Tcl/Python button using HTML, you can enter **** in the **Label** field. This adds the Nuke color wheel icon. You can also use your own icons in the same way as long as you save them in your plug-in path directory. Most common image formats work, but we recommend using .png files.

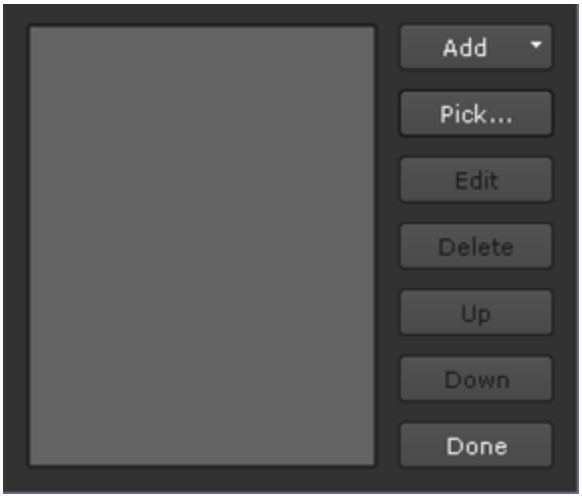
  Note that the HTML has been changed to a slightly non-standard form where newlines are significant. If there is a newline character in your data, a new line is displayed in the label.
- **Hide** - Check this to hide the control from the users. This can be useful if you want to make a new control to contain a complex expression that you can then refer to repeatedly by other controls.
- **Start new line** - Uncheck this if you want the control to appear on the same line as the previous control in the gizmo properties panel.
- **Tooltip** - Enter a short help text here. It appears, along with the Name, in a pop-up tool tip when the user points the mouse at the control. If you do not provide a tool tip, whatever is in the **Name** field is used as the tool tip.

4. If necessary, repeat the previous three steps to add more controls to your Group node (future gizmo).

5. In the Group node properties panel, the controls are listed in the same order as they are in the **Manage User Knobs** dialog. To move a control up or down in the properties panel, select it in the dialog and use the **Up** and **Down** buttons.
6. Click the **export as gizmo** button.
7. In the file browser that appears, click **Home**. Type **.nuke/** after the path displayed at the bottom of the file browser.
8. Enter a name after the path, and append a **.gizmo** extension after the name. This is the name of the command that is written to any saved script that's using the gizmo. It's a good idea, and common practice, to begin the name with a capital letter, because Nuke uses this as an indication that the command is a node or a gizmo.
9. Click **Save**.

To Create New Controls:

1. Right-click on the dark gray background of the Group properties panel and select **Manage User Knobs**. The following dialog opens.



2. To add a new control, tab, static text, or divider line to the Group (gizmo) controls, click **Add** on the **Manage User Knobs** dialog and select the option you want to add. This opens a dialog where you can edit the control, tab or static text you added. In most cases, you can edit the following:
   - **Name** - Give the new control a unique name here. You need to use this name whenever you want to reference the control from scripts or via expressions. The name can only contain letters and digits. Spaces or punctuation are not allowed. This field cannot be left empty.
   - **Label** - Whatever you enter here appears to the left of the control in the gizmo properties panel. If you leave this empty, whatever is in the **Name** field is also used as the label.

     In the **Label** fields of check boxes, Tcl script buttons, and Python script buttons, you can also use HTML. For example, to have your text appear in bold, you can enter **text**.

To add an icon to your check box or Tcl/Python button using HTML, you can enter **** in the **Label** field. This adds the Nuke color wheel icon. You can also use your own icons in the same way as long as you save them in your plug-in path directory. Most common image formats can work, but we recommend using .png files.

Note that the HTML has been changed to a slightly non-standard form where newlines are significant. If there is a newline character in your data, a new line is displayed in the label.

- **Hide** - Check this to hide the control from the users. This can be useful if you want to make a new control to contain a complex expression that you can then refer to repeatedly by other controls.
- **Start new line** - Uncheck this if you want the control to appear on the same line as the previous control in the Group properties panel.
- **Tooltip** - Enter a short help text here. It appears, along with the Name, in a pop-up tool tip when the user points the mouse at the control. If you do not provide a tool tip, whatever is in the **Name** field is used as the tool tip.

3. Use an expression to link the control you just created to a node and its control inside the Group node. This is important, because for the new control to do anything, you need to refer to it using an expression in some other control on a node inside the Group. For more information, see the examples below and refer to Expressions.
4. If necessary, repeat the previous four steps to add more controls to your Group node (future gizmo).
5. In the Group node properties panel, the controls are listed in the same order as they are in the **Manage User Knobs** dialog. To move a control up or down in the properties panel, select it in the dialog and use the **Up** and **Down** buttons.
6. Click the **export as gizmo** button.
7. In the file browser that appears, click **Home**. Type **.nuke/** after the path displayed at the bottom of the file browser.
8. Enter a name after the path, and append a **.gizmo** extension after the name. This is the name of the command that is written to any saved script that's using the gizmo. It's a good idea, and common practice, to begin the name with a capital letter, because Nuke uses this as an indication that the command is a node or a gizmo.
9. Click **Save**.

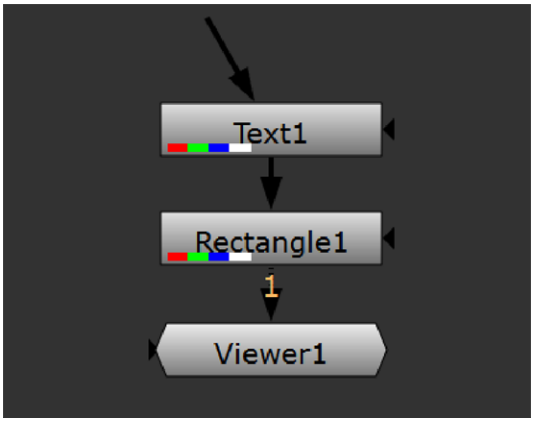
To Delete Controls:

1. Right-click on the dark gray background of the Group properties panel and select **Manage User Knobs**.
2. In the dialog that opens, select the controls that you want to delete from the list and click **Delete**.
3. To delete an entire tab, select all controls on the tab as well as the tab name and click **Delete**.

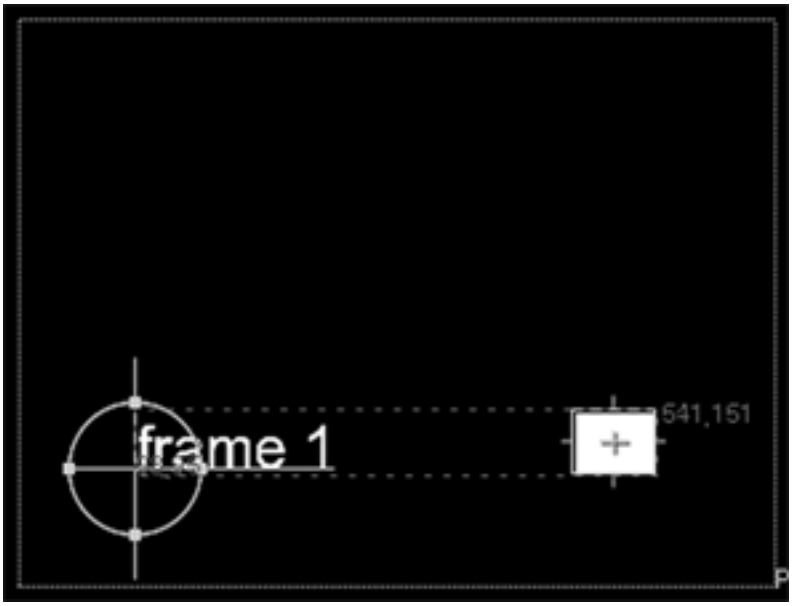
Examples

Below are some examples on how to create new controls for gizmos. To try them out, do the following preparations:

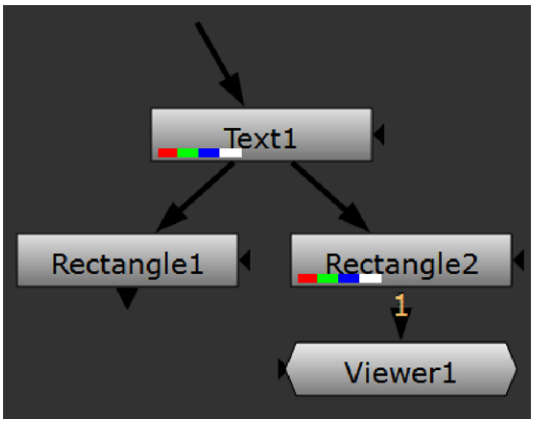
1. Select **Draw** > **Text** and **Draw** > **Rectangle**. Create the following setup:



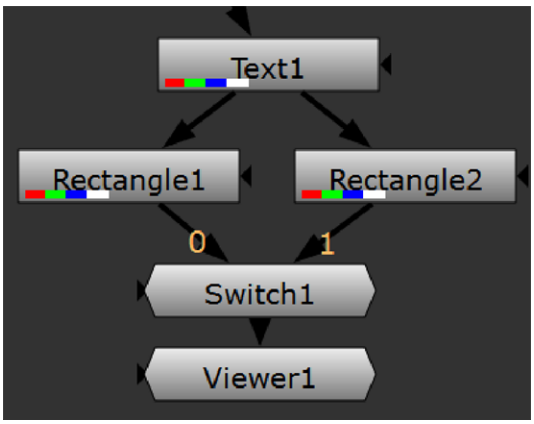
2. Double-click on the Rectangle1 node.
3. In the Viewer, resize and reposition the rectangle until it looks like the following:



4. In the Rectangle1 properties panel, go to the **Color** tab. Click on the **4** button to display multiple values rather than the slider. Enter **1** as the value for **r**, and **0** as the value for **b**, **g** and **a**. This changes the color of the rectangle from white to red.
5. Copy the Rectangle1 node and paste it into the same script. Create the following connections:



6. Double-click on the Rectangle2 node and change the color of the rectangle from red to green (r 0, g 1, b 0, a 0).
7. Select **Merge** > **Switch** to add a Switch node. Create the following connections:

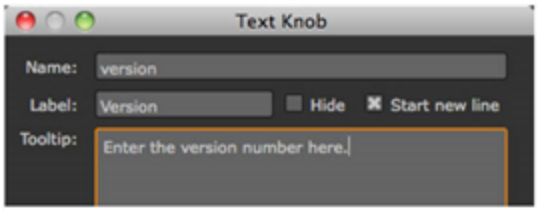


8. Select the Text1, Rectangle1, Rectangle2 and Switch1 nodes and press **Ctrl/Cmd+G** to group them. This group is the gizmo we add controls to in the following examples.
9. Delete the original four nodes from the **Node Graph** tab.
10. Select the Group node and append a Viewer to it.

Example 1

In this example, we add a control called **Version** to the Group node controls. This control is an input field. Whatever is entered in the field is called by the Text1 node and displayed in the Viewer when you view the output of the group.

1. Open the Group properties panel and right-click on the dark gray background. Select **Manage User Knobs**.
2. In the dialog that opens, select **Add** > **Text input Knob** to add a text input field control to your Group properties panel.
3. Enter *version* as the **Name** for the control, *Version* as the **Label**, and *Enter the version number here* as the **Tooltip**. Click **OK** and **Done** to close the dialogs.

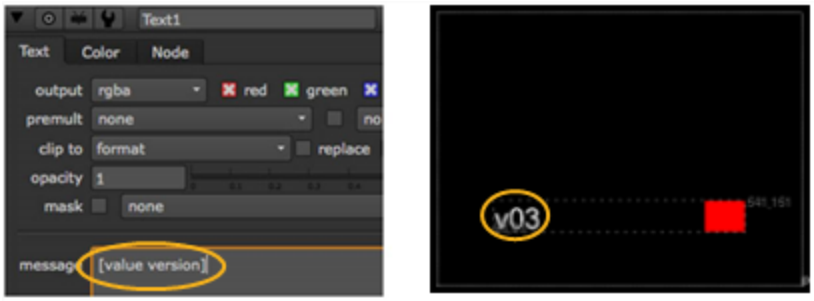


This step created a tab called **User** in the Group node controls. All the controls you add or pick are added on this tab by default.

As you can see, the **Version** control is now there.



4. On the **Group1 Node Graph** tab, double-click the Text1 node to open its controls. In the **message** field, enter the following expression: **[value version]**. This expression calls the control named *version* that you created in the previous step. Therefore, whatever is entered in the **Version** field of the Group node (for example, **v03**), appears as a result of the Text1 node.

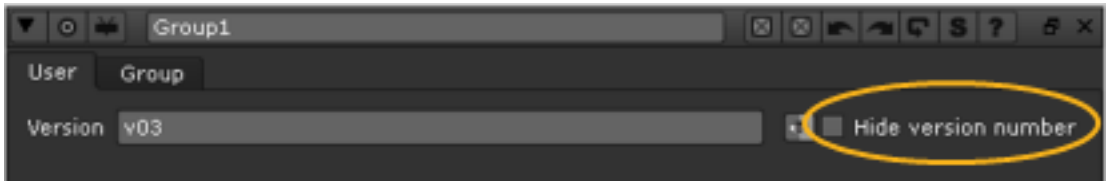


Example 2

This example teaches you to create a checkbox control that the users can use to specify whether they want to display or hide the version number added in the previous example.

1. In the Group properties panel, right-click on the dark gray background and select **Manage User Knobs**.
2. In the dialog that opens, select **Add** > **Check Box** to add a checkbox control to your Group properties panel.
3. Enter *hideversion* as the **Name** for the control, *Hide version number* as the **Label**, and *Check this to hide the version number* as the **Tooltip**.
4. To have the new control appear next to the Version control (created in the previous example) rather than below it on its own line, uncheck **Start new line**. Click **OK** and **Done** to close the dialogs.

   The control you created appears in the Group properties panel now.

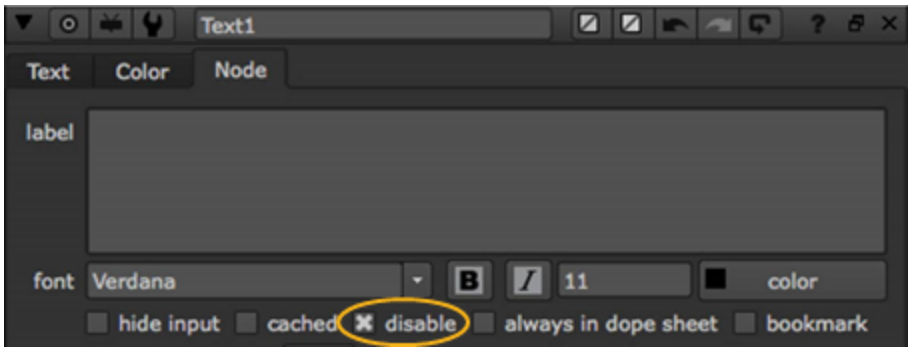


5. In the Text1 controls, go to the **Node** tab. Right-click on the **disable** control and select **Add expression**.



6. In the **Expression** field, enter **hideversion** (or, if you want to make it clear that the control is in the enclosing group, you can also use **parent.hideversion**). This calls the control you created in steps 2 and 3. Click **OK**.

   From now on, whenever **Hide version number** is checked in the Group controls, the Text1 node is disabled and you cannot see the version number it would otherwise create.







Example 3

In this example, we add a control labeled **Status** to the Group controls. This control is a dropdown menu with two options: **Finished** and **Unfinished**. When **Finished** is selected, the green rectangle is displayed. When **Unfinished** is selected, you'll see the red rectangle instead.

1. In the Group properties panel, right-click on the dark gray background and select **Manage User Knobs**.
2. In the dialog that opens, select **Add** > **Pulldown Choice** to add a dropdown menu control to your Group properties panel.
3. Enter *status* as the **Name** for the control and *Status* as the **Label**. In the **Menu Items** field, list the items you want to appear in the dropdown menu - in this case, *Finished* and *Unfinished*.



Finally, enter *Select the production status here* as the **Tooltip**. Click **OK** and **Done** to close the dialogs.

The **Status** control should now appear in the Group controls.

4. On the **Group1 Node Graph** tab, double-click the Switch1 node to open its controls. Right-click on the **which** field and select **Add expression**.
5. In the dialog that opens, enter the following expression: **status==0** (or, **parent.status==0**). This expression calls the control named *status* that you created earlier in this example. For the dropdown menus, the first item is **0**, the next **1**, the next **2**, and so on.

   From now on, whenever **Finished** is selected under **Status**, the green rectangle is shown. When **Unfinished** is selected, the red rectangle is shown.

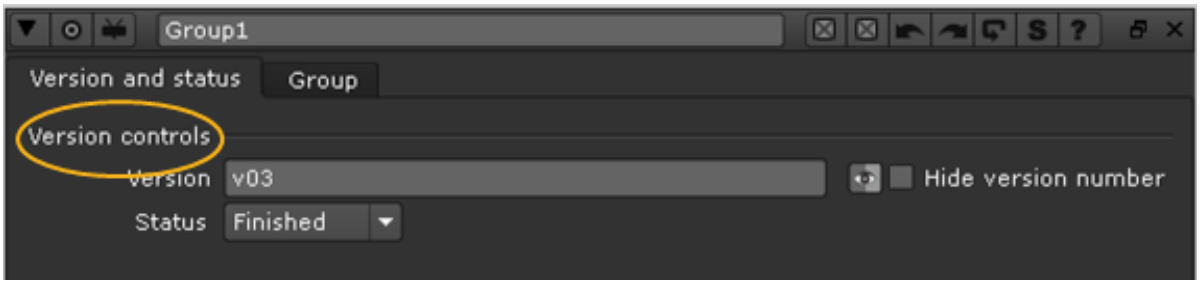
Example 4

This example teaches you how to visually group and rearrange the controls you created for the Group properties panel. You can do this by renaming the **User** tab, and using static text and divider lines to group the controls on the tab.

First, we'll rename the **User** tab in the Group properties panel:

1. In the Group properties panel, right-click on the dark gray background and select **Manage User Knobs**.
2. In the dialog that opens, select **User** and click **Edit**.
3. In the **Label** field, enter a new name for the tab, for example, *Version and status*. Click **OK** and **Done** to close the dialogs.

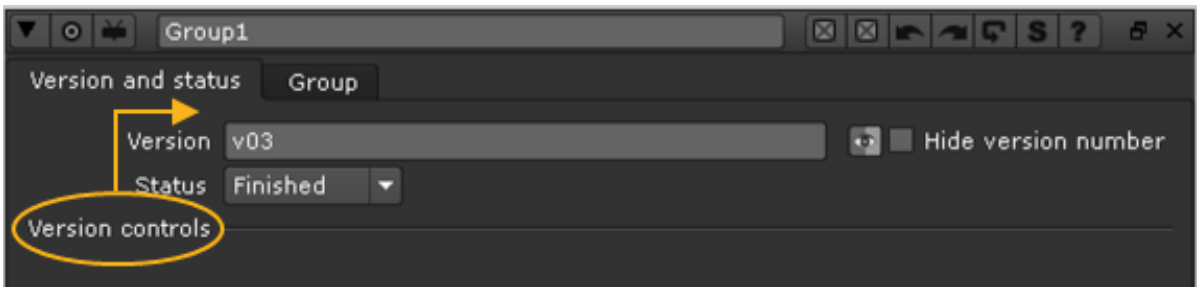
   If you now look at the Group controls, you'll notice that the **User** tab has been renamed to **Version and status**.



Next, we'll group the two version controls of the Group node under a title called *Version controls*:

1. In the Group properties panel, right-click on the dark gray background and select **Manage User Knobs**.
2. In the dialog that opens, select **Add** > **Text** to add text to your Group properties panel.
3. Enter *versioncont* as the **Name** for the control and *Version controls* as the **Label**. Click **OK** and **Done** to close the dialogs.

   This adds the text *Version controls* to the Group properties panel. However, the text does not appear where we want it to appear: on top of the **Version** and **Hide version number** controls. Let's move it up.



4. Right-click on the Group properties panel again and select **Manage User Knobs**.
5. Select **[Version controls]** from the list and click **Up** three times. Click **Done**.

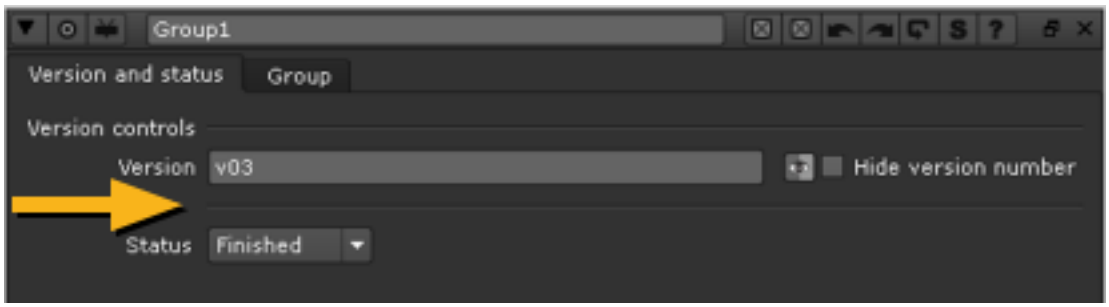
   The text should now appear on top of the Group properties panel, above the version controls.



Finally, we'll add a divider line between the version controls and the **Status** control:

1. In the Group properties panel, right-click on the dark gray background and select **Manage User Knobs** again.
2. In the dialog that opens, select **Add** > **Divider Line** to add a line to divide the controls in your Group properties panel.
3. Select the line from the Manage User Knobs dialog, where it is shown as **unnamed**.
4. Click the **Up** button once to move the line between the **Hide version number** and **Status** controls. Click **Done**.

   If you now open the Group controls, you'll notice that a there's a line between these controls.



5. See Accessing Gizmos in Nuke for information on how to access your gizmo in Nuke.

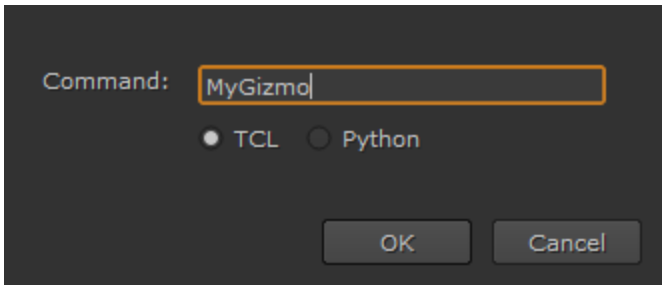
Accessing Gizmos in Nuke

There are several ways to access your gizmos once you've saved them into your **.nuke** folder or NUKE_PATH. See Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts for more information on where Nuke looks for your gizmos.

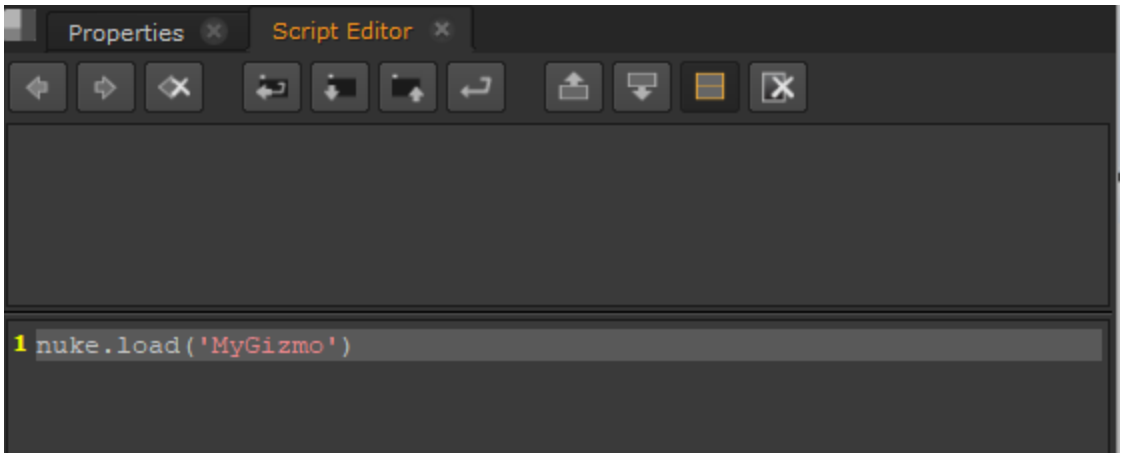
If you're going to use a gizmo often, across multiple sessions, you can add it to a Nuke menu to provide quick access. See Defining Custom Menus and Toolbars for more information.

Otherwise, you can quickly add it per-session by:

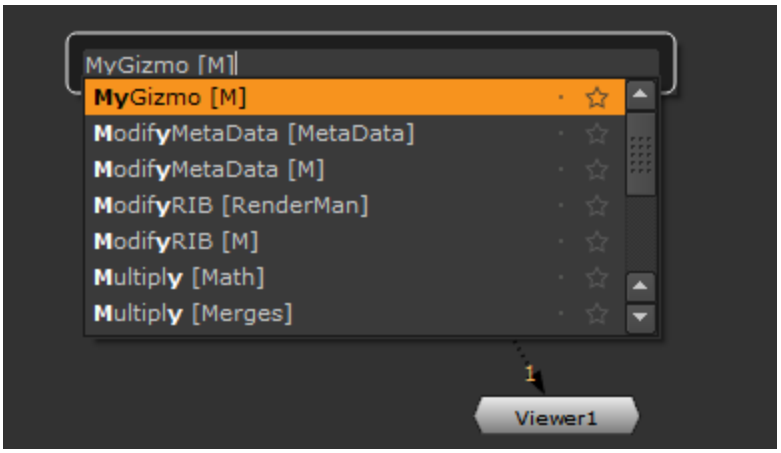
- pressing **X** on the Node Graph or **Properties** panel and entering the gizmo name (without the extension) as a TCL command in the dialog that opens,



- opening the Script Editor and entering **nuke.load('gizmo name')** where **gizmo name** is the name of the gizmo without the extension, or



- selecting **Other** > **All plugins** > **Update** from the node toolbar and then adding the gizmo using the **Tab** menu in the Node Graph.



Sourcing Custom Plug-ins and Generic Tcl Scripts

Nuke Custom Plug-ins

The Nuke developer's kit (NDK) allows developers to create and compile their own binary plug-ins.

To Source a Custom Plug-in

1. Place the plug-in file in the plug-in path directory. Its name should include a **.dll** (on Windows), **.so** (on Linux) or **.dylib** (on Mac) extension.

   For more information on plug-in path directories, see Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts.
2. Create a menu option referencing the plug-in file (see Defining Custom Menus and Toolbars).

   Or instruct artists to invoke the plug-in by opening the Script Editor and entering **nuke.load ("*plug-in name*")** where ***plug-in name*** stands for the name of the plug-in without the extension.

Sourcing Tcl Procedure

A Nuke script or gizmo is in fact a Tcl procedure (script). Thus, Nuke also allows you to hand code generic Tcl procedures to automate Nuke in various ways.

To Source a Generic Tcl Procedure

1. Place the Tcl procedure file in the plug-in path directory. Its name should include a **.tcl** extension.

   For more information on plug-in path directories, see Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts.
2. Create a menu option referencing the plug-in file (see Defining Custom Menus and Toolbars).

   Or instruct artists to invoke the Tcl script by opening the Script Editor and entering **nuke.load ("procedure/script file name")** where **procedure/script file name** stands for the name of the procedure of script file without the extension.

Tip: For some code samples of useful Nuke Tcl procedures, look inside the **[Nuke directory]/plugins** directory.

Template Scripts

You can create a template script that is loaded instead of an empty script every time you launch Nuke or select **File** > **New Comp** or **File** > **Close Comp**. This allows you to save lookup table (LUT) setups and favorite arrangements of nodes, for example.

Creating and Using a Template Script

To create and use a template script:

1. Create the script you want to use as a template.
2. Select **File** > **Save Comp As**. Navigate to **~/.nuke**. The tilde (**~**) stands for your home directory and the full stop (**.**) for a hidden folder.
3. Name your script **template.nk** and click **Save**.

The next time you launch Nuke or select **File** > **New Comp** or **File** > **Close Comp**, Nuke loads the template from **~/.nuke/template.nk**.

Tip: If you're not sure of the location of your home directory, on Linux and Mac you can open a terminal window and type **echo $HOME**. The terminal returns the pathname to your home directory.

On Windows, you can find the **.nuke** directory under the directory pointed to by the HOME environment variable. If this variable is not set (which is common), the .nuke directory will be under the folder specified by the USERPROFILE environment variable. To find out if the HOME and USERPROFILE environment variables are set and where they are pointing at, enter %HOME% or %USERPROFILE% into the address bar in Windows Explorer. If the environment variable is set, the folder it's pointing at is opened. If it's not set, you get an error.

Here are examples of what the path name may be on different platforms:
Linux: /home/login name
Mac: /Users/login name
Windows: drive letter:\Documents and Settings\login name or drive letter:\Users\login name

Defining Common Preferences

The Nuke Preferences dialog (**Edit** > **Preferences**) allows any user to make myriad behavior and display adjustments to the interface. However, you may wish to assign certain default preferences for artists.

Defining Default Preferences

To define default preferences:

1. Select **Edit** > **Preferences** to display the **Preferences** dialog.
2. Modify the controls within the dialog as necessary. For descriptions of what the controls do, see Appendix A: Preferences.
3. Click **OK**. Nuke writes the modified preferences to a file called **preferences17.0.nk**, which is stored inside your [home directory]/.nuke directory.

Tip: If you're not sure of the location of your home directory, on Linux and Mac you can go to a terminal window and type **echo $HOME**. The terminal will return the path name to your home directory.

On Windows, you can find the .nuke directory under the directory pointed to by the HOME environment variable. If this variable is not set (which is common), the .nuke directory will be under the folder specified by the USERPROFILE environment variable. To find out if the HOME and USERPROFILE environment variables are set and where they are pointing at, enter %HOME% or %USERPROFILE% into the address bar in Windows Explorer. If the environment variable is set, the folder it's pointing at is opened. If it's not set, you get an error.

Here are examples of what the path name may be on different platforms:
Linux: /home/login name
Mac: /Users/login name
Windows: drive letter:\Documents and Settings\login name or drive letter:\Users\login name

4. Move the resulting **preferences17.0.nk** file into your Nuke plug-in path directory.

   For more information on plug-in path directories, see Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts.

Your preferences now act as the defaults for your artists. However, should they make changes using the **Preferences** dialog, these changes will override your defaults.

Deleting (and Resetting) the Preferences

To delete (and reset) the preferences:

1. Open a terminal (or shell) as described for your operating system at the beginning of this chapter.
2. Using the prompt, go to the .nuke directory, under your home directory.
3. Enter **pwd** to display and verify the path.

   You should see something similar to

   • /users/*login name*/.nuke (on Linux),

   • /Users/*login name*/.nuke (on Mac) or

   • *drive letter*:\Documents and Settings*login name*\.nuke or *drive letter*:\Users*login name*\.nuke

   This is not always the case, however, because on Windows the **.nuke** folder can be found under the directory pointed to by the HOME environment variable or (if HOME is not set) the USERPROFILE environment variable.

   To find out if the HOME and USERPROFILE environment variables are set and where they are pointing at, enter %HOME% or %USERPROFILE% into the address bar in Windows Explorer. If the environment variable is set, the folder it's pointing at is opened. If it's not set, you get an error.
4. Enter **rm preferences17.0.nk** to delete the preference file.
5. Close the terminal or shell.

   The next time you launch Nuke, it rebuilds the file with the default preferences.

Altering a Script's Lookup Tables (LUTs)

A script's lookup tables are curves that control the conversion between file or device color spaces and Nuke's internal color space. In the Curve Editor, the x axis represents the input pixel values and the y axis the output pixel values (normalized to the 0-1 range). When applying LUTs, Nuke looks up the input value along the x axis to determine what the y value is to output.

Nuke provides many nuke-default LUTs, including: **linear**, **sRGB**, **rec709**, **Cineon**[1]**, Gamma1.8, Gamma2.2, Panalog**[2], **REDLog**[3]**, ViperLog**[4]**, AlexaV3LogC**[5]**, PLogLin**[6]**, SLog**[7]**,** and **REDSpace**[8]**.**

You can also create an unlimited number of additional LUTs and edit or remove existing LUTs in the script's settings.

By default, Nuke uses certain LUTs for certain file types or devices. In most cases, you do not need to touch these defaults. However, there may occasionally be cases when changing the defaults is necessary: for example, if your material has been shot with a camera that records in a custom color space, such as Panalog. In those cases, you can change the defaults in the script's settings so that you don't need to change the color space on each Read or Write node.

If you do not want to use the default LUT for reading or writing certain individual images, you can select the LUT to use in the corresponding Read or Write node's controls.

Managing Nuke's Native LUTs

The **Color** tab in the Nuke **Project Settings** allows you to view, add, edit, and delete Nuke's native LUTs.

Note: You can't adjust LUTs that appear under the OCIO **color management** scheme. See OCIO Color Management for more information.

Article: See this Support Knowledge Base article for more detailed information.

---

1 The Cineon conversion is implemented as defined in Kodak's Cineon documentation.

2 The Panalog LUT is based on a log2lin conversion with a blackpoint of 64, whitepoint of 681, and a gamma of 0.89.

3 The REDLog LUT is based on a log2lin conversion with a blackpoint of 0, whitepoint of 1023, and a gamma of 1.022.

4 The ViperLog LUT is based on a log2lin conversion with a blackpoint of 0, whitepoint of 1023, and a gamma of 1.0.

5 The Alexa LogC LUT uses the formula provided by ARRI.

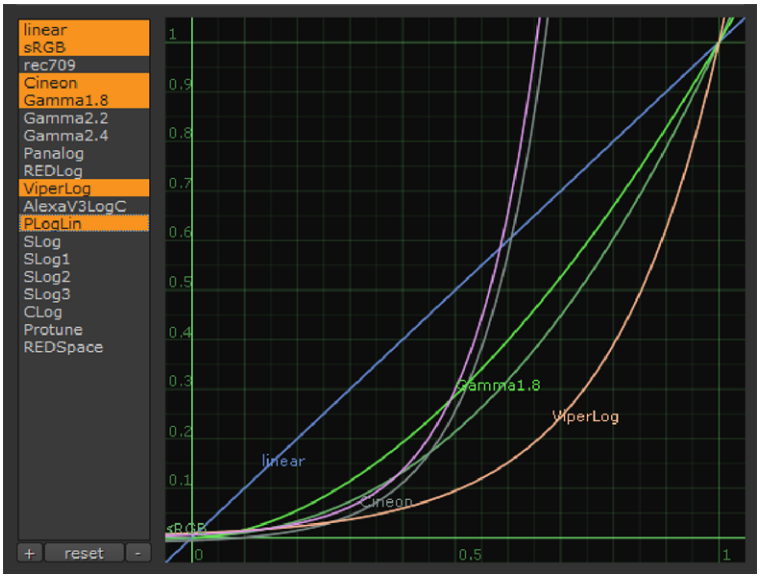
6 The PLogLin LUT uses the default values for the formula, mapping log 0.43457 (code value 445 on a 1024 scale) to linear 0.18 (mid-gray) assuming a negative gamma of 0.6 and a density per code value of 0.002. (This does factor in the fact that value ranges for a log image in Nuke are still scaled to 0-1 range.)

7 The Sony S-Log LUT comes from the Sony S-Log Whitepaper Version 1.110. For more information, see http://www.sony.co.uk/res/attachment/file/66/1237476953066.pdf.

8 The REDSpace LUT is implemented as defined in a curve provided by RED.

To Display LUT Curves

1. Select **Edit** > **Project Settings** to open the settings for the script.
2. Go to the **Color** tab.
3. From the list on the left, select the LUT you want to display in the curve editor. To select several LUTs, press **Ctrl** (Mac users press **Cmd**) while selecting the LUTs. All the selected LUTs are shown in the curve editor at the same time.



To Create a New LUT

1. Select **Edit** > **Project Settings** to open the settings for the script.
2. Go to the **Color** tab.
3. Click the plus button (**+**). A dialog opens.
4. Enter a name for the new LUT and click **OK**.
5. Adjust the lookup curve to suit your needs. Click on the curve to select it. **Ctrl/Cmd**+**Alt**+**click** to add points on the curve, and drag the points to a new position. To change the shape of the curve, adjust the tangent handles.

The new LUT is now available in the global LUT settings, the **Input Transform** and **output transform** dropdown menus of Read and Write nodes' properties panels, and the Viewer controls.

To Edit LUTs

1. Select **Edit** > **Project Settings** to open the settings for the script.

2. Go to the **Color** tab.
3. From the list on the left, select the LUT you want to edit.
4. Adjust the lookup curve to suit your needs. Click on the curve to select it. **Ctrl/Cmd**+**Alt**+**click** to add points on the curve, and drag the points to a new position. To change the shape of the curve, adjust the tangent handles.

   To use the usual editing commands, such as copy and paste, right-click on the curve editor and select **Edit**. Then, select the editing command you want to use, just like you would on any curve editor.



Note: Renaming existing LUTs is currently not possible. If you want to rename a LUT, you need to add and name a new LUT, copy the information from the old LUT into the new one, and then remove the old LUT.

To Reset the LUT Curves Back to Their Initial Default Shapes

1. Select **Edit** > **Project Settings** to open the settings for the script.
2. Go to the **Color** tab.
3. From the list on the left, select the LUT you want to reset. To select several LUTs, press **Ctrl**/**Cmd** while selecting the LUTs.
4. Click **reset**.

To Remove LUTs

1. Select **Edit** > **Project Settings** to open the settings for the script.
2. Go to the **Color** tab.
3. From the list on the left, select the LUT you want to remove. Only remove LUTs that you have, for example, created by accident and are not using in your script. To remove the LUT, click the minus button (**-**).

The LUT is removed from the LUT settings, and the **Input Transform** and **output transform** dropdown menus of Read and Write nodes' properties panels.



Note: If you remove a LUT that is used in a node, the node continues to refer to the LUT by name and raises an error.

Selecting the LUT to Use

To select the LUT to use when reading or writing an image:

1. Double-click to open the Read or Write node's properties panel.
2. From the **Input Transform** or **output transform** dropdown menus, select the LUT you want to use. To use the default LUT defined in Nuke's settings for the image type in question, select **default**.

Default LUT Settings

By default, Nuke uses the following LUTs in the following cases:

LUT	File Type / Device	Default LUT
working space	This determines what colorspace files should be converted to when read and from when written- it's the colorspace used by Nuke under the hood. In earlier releases of Nuke, this colorspace was hidden because **linear** was always chosen as the **working space**. You may find that some operations work better in colorspaces other than **linear**. For example, some transforms work better in the **CLog** colorspace. **Note:** You can only change the **working space** if you're using OCIO **color management**.	linear
monitor	This is used for postage stamps, OpenGL textures, the color chooser display, and all other non-Viewer image displays.	sRGB
8-bit files	This is used when reading or writing image files that contain 8-bit data. Also used by the Merge node's sRGB switch, and to convert Primatte inputs into sRGB and outputs from sRGB.	sRGB
16-bit files	This is used when reading or writing image files that contain 16-bit integer data (not half float).	sRGB
log files	This is used when reading or writing **.cin** or **.dpx** files.	Cineon
float files	This is used when reading or writing image files that contain floating-point data.	linear

To Change the Default LUT Settings

1. Select **Edit** > **Project Settings** to open the settings for the script.
2. Go to the **Color** tab.
3. From the dropdown menus under **Default LUT settings**, select the LUTs you want to use by default for each file type or device.

The new defaults are now used for any LUT setting where you have not selected a specific LUT. Any controls you have set to a specific LUT (that is, not set to **default**) continues to use the selected LUT, and only those set to **default** are affected.

Example Cases

Below are some examples of situations where you might need to alter the default LUTs.

Working in Video Colorspace

Emulating compositor software that works in video color space is not recommended. However, if you do need to do so, do the following:

1. Select **Edit** > **Project Settings** and go to the **Color** tab.
2. Under **Default LUT settings**, change the **monitor**, **8-bit files**, and **16-bit files** values to **linear**.

This prevents Nuke from converting from sRGB into linear. Nuke's nodes still assume linear data, but the image processing is applied to your unlinearized video color space images.

Linear Data in 16-Bit Files

Some facilities use linear data in 16-bit files. If this is the case in your facility, do the following:

1. Select **Edit** > **Project Settings** and go to the **Color** tab.
2. Under **Default LUT settings**, change the **16-bit files** value to **linear**.

Cineon Displays

Some facilities have adjusted their monitor electronics to correctly display Cineon data. If this is the case in your facility, do the following:

1. Select **Edit** > **Project Settings** and go to the **Color** tab.
2. Under **Default LUT settings**, change the **monitor** value to **Cineon**.

Color Management

Although true color management requires using other nodes, it may be useful to approximate it with a LUT that is used for the **monitor** setting. This way, texture maps and postage stamps resemble the final display more accurately.

If your color management is creating a monitor-corrected image, you'll want to set **monitor** to **sRGB** so you get reasonably monitor-correct output on non-Viewer images.

OCIO Color Management

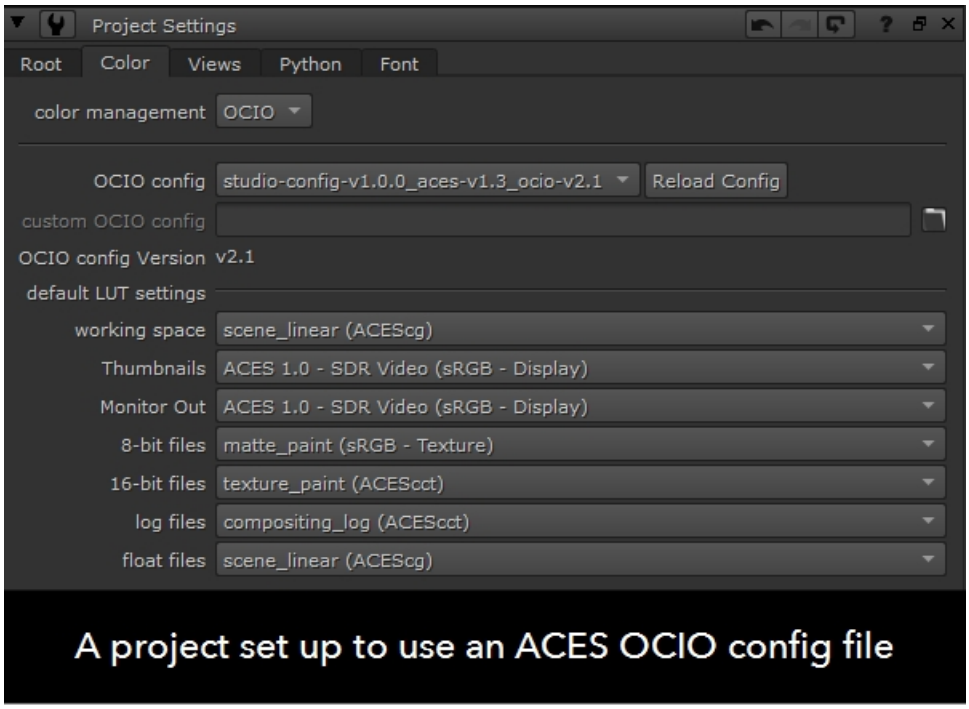
Nuke uses OpenColorIO for color management. All of the color transforms in Nuke, whether those shipped with the application or custom colorspaces, are defined in OCIO config files.



Note: There are a number of NDK examples covering OCIO workflows that ship with the Nuke developer documentation, which can be found here:
`<install_directory>/Documentation/NDKExamples/Plugins/00b_8source.html`

Depending on the OCIO config file that you are working with, there are a number of color transform options and roles (aliases to transforms) that you can set in Nuke. There are also default options, which change depending on what file type you are working with. When the default option is selected, the color transform that Nuke has set for it is listed in brackets.

Tip: Use the options in **Preferences** > **Project Defaults** > **Color Management** to apply them to all new projects.



A project set up to use an ACES OCIO config file

The **color management** dropdown determines whether Nuke uses the color transforms read from the configuration specified or the **Nuke** native transforms. Selecting **OCIO** makes the relevant OCIO transforms available to the Read and Write nodes in scripts on a per project basis.

All OCIO configurations except **nuke-default** automatically switch this control to **OCIO**.

1. Set the OpenColorIO Config you want to use for this project.

   Nuke ships with a number of default configurations, but you can use a custom OCIO config file by selecting **custom** from the **OCIO config** dropdown and then entering the file path.
2. Changing the configuration updates the **default LUT settings** accordingly. If the selected configuration is invalid for certain transforms, a warning displays.

   The **working space** transform determines what colorspace files should be converted to (Read) and from (Write) - it's the colorspace used by Nuke under the hood.

   > **Note:** In earlier releases of Nuke, this colorspace was hidden because **linear** was always chosen as the **working space**. You may find that some operations work better in colorspaces other than **linear**. For example, some transforms work better in the **CLog** colorspace.

3. You can use **default LUT settings** dropdown menus to override how clips in the Viewer, thumbnails, and so on are converted to and from the **working space**.

When **nuke-default** is selected, Reads and Writes work the same as in legacy versions of Nuke, with no integrated OCIO transforms. When **OCIO** is selected:

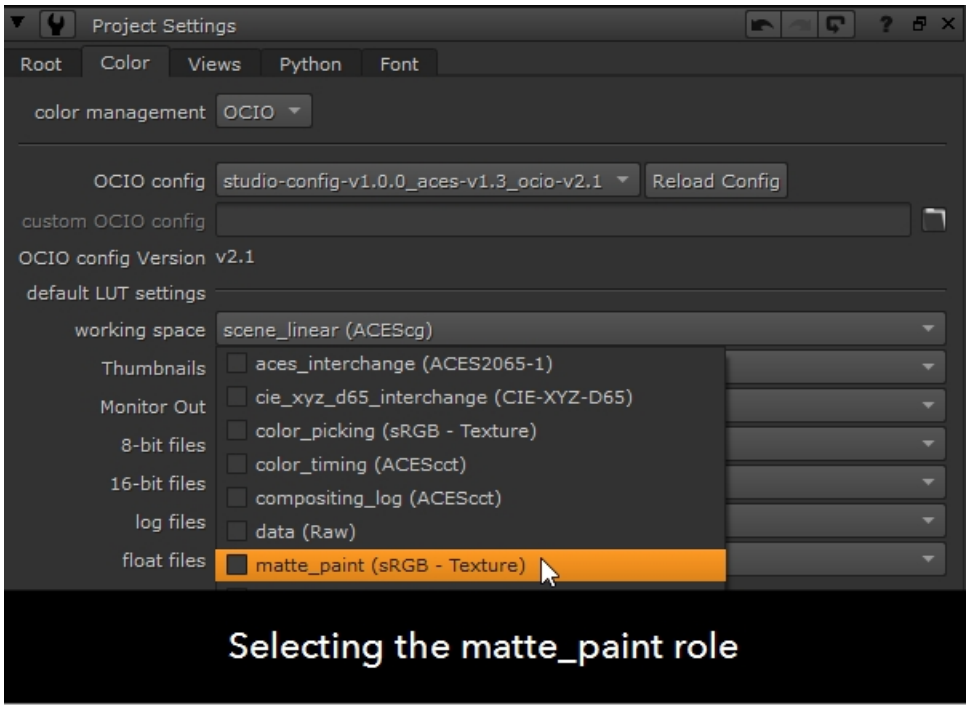
- Reads and Writes use OCIO transforms, with no Nuke built-in transforms applied to the image.
- Read and Write colorspace controls are populated with the list of colorspaces defined in your currently selected OCIO config.
- The **default LUT settings** dropdowns are also populated with the list of colorspaces or display transforms defined in your OCIO config. The default value for each menu match the defaults in a project with the same config. These defaults can be overridden using Python callbacks. See the following path for the default implementation that ships with Nuke:

  `<install_dir>/plugins/nuke/colorspaces.py`
- The **working space** dropdown sets the colorspace that Nuke uses internally for its image processing. This automatically sets the **output transform** colorspace of Write nodes, and the **Input Transform** colorspace for Read nodes. This defaults to the scene **linear** role defined in your OCIO config. For example, selecting an ACES config sets the **working space** to **scene_linear (ACEScg)**.
- Nuke Studio-created comps no longer contain automatically injected OCIOColorspace nodes. Instead, OCIO Color Management is automatically set in the comp's **Project Settings**, and the correct OCIO colorspace is set directly into the Read and Write nodes.

Adding OCIO Roles

OCIO roles allow you to set custom role names for different colorspaces to make it easier for artists to instinctively know which transform to use for any shot. For instance, if an element is coming from your matte painting department and should always be brought into Nuke as sRGB, you can create a **matte painting** role, which is associated with the sRGB colorspace for your artist to select.

OCIO roles are the primary method for selecting colorspaces. All of the colorspaces in the OCIO config file are still accessible, but they have been grouped together into a **Colorspaces** menu under the roles.



OCIO roles are stored in **config** files, some of which ship with Nuke in the following directory:
`<install_dir>/plugins/OCIOConfigs/configs/`

For example, the aces-v1.3 **config** file includes the following roles:

```
roles:

 aces_interchange: ACES2065-1
 cie_xyz_d65_interchange: CIE-XYZ-D65
 color_picking: sRGB - Texture
 color_timing: ACEScct
 compositing_log: ACEScct
 data: Raw
 matte_paint: sRGB - Texture
 scene_linear: ACEScg
 texture_paint: ACEScct
```

The **!<Colorspace>** describes colorspace **name** and **family**. The name and family define which colorspace is associated with the role. For example, the **scene_linear** role uses the **ACEScg** colorspace:

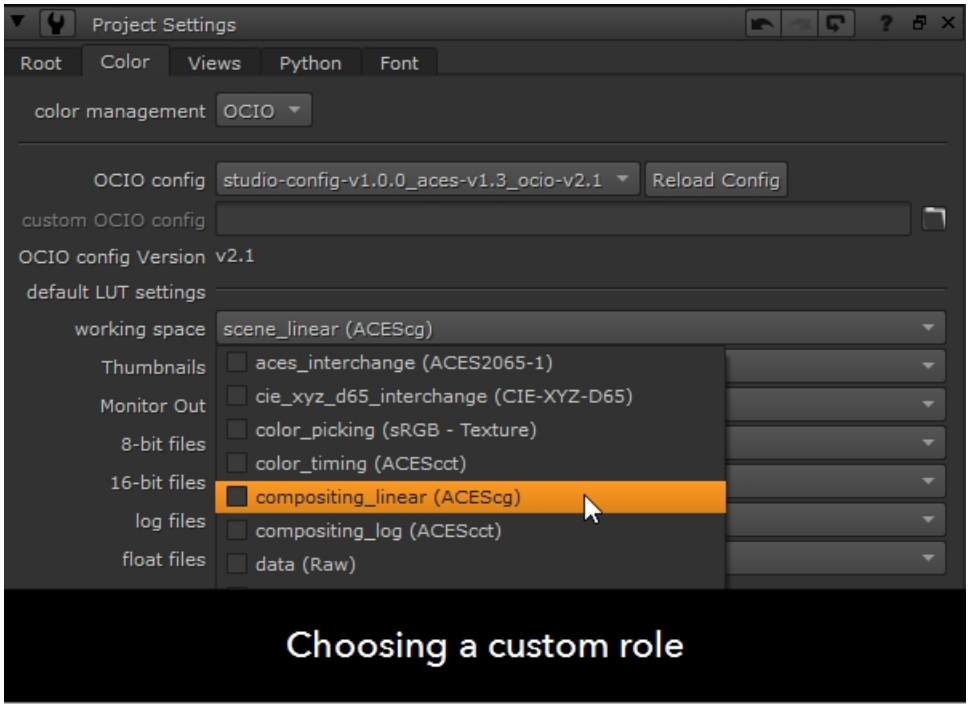
```
- !<ColorSpace>

 name: ACEScg
 aliases: [ACES - ACEScg, lin_ap1]
 family: ACES
 equalitygroup: ""
 bitdepth: 32f
 description: |
   Convert ACEScg to ACES2065-1
```

```
  ACEStransformID: urn:ampas:aces:transformId:v1.5:ACEScsc.Academy.ACEScg_to_
  ACES.a1.0.3

isdata: false
categories: [file-io, working-space]
encoding: scene-linear
allocation: uniform
to_scene_reference: !<BuiltinTransform> {style: ACEScg_to_ACES2065-1}
```

You can edit these files to add roles or create your own custom config and then point Nuke to the file using the **Project Settings** > **Color** > **custom OCIO config** control.

To add a role to a config file:

1. Open the required config file or create a custom config.
2. Enter the **role**, **name**, and **family** of the role under the **roles:** line. For example:
   `compositing_linear: ACES - ACEScg`
3. Save the file and open Nuke.
4. Open the **Project Settings** and click the **Color** tab.
5. You can now pick your role from the **default LUT settings**. For example, you can set the working space to **compositing_linear** if you want to work in the ACEScg colorspace.



Choosing a custom role

Define Color Space Mappings Using OCIO Aliases

Nuke supports the use of aliases in an OCIO config and checks these when attempting to set the input transform for a given image or sequence. The aliases feature in OCIO allows config authors to define a set of alternate names for a color space, which can then be used in place of the primary color space name. For example, it may be useful to define a more suitable name for file paths, or a color space name might have changed between versions of a config, so it may be necessary to add an alias for the older version of the name for backwards compatibility. Additionally if an application tries to set defaults based on a specific name then aliases can be used to ensure a suitable color space is found.

Below is part of a ColorSpace definition from an ACES 1.3 OCIO config.

```
- !<ColorSpace>

 name: ARRI LogC3 (EI800)
 aliases: [arri_logc3_ei800, Input - ARRI - V3 LogC (EI800) - Wide Gamut,
 logc3ei800_alexawide]
 family: Input/ARRI
 equalitygroup: ""
 bitdepth: 32f
 description: |
 Convert ARRI LogC3 (EI800) to ACES2065-1
```

Here you can see the **name:** attribute is defined as **ARRI LogC3 (EI800)**. This is the primary name for the color space.

Directly below is the **aliases:** attribute. This is where the alternate names are defined, in this case:

- `arri_logc3_ei800`
- `Input - ARRI - V3 LogC (EI800) - Wide Gamut`
- `Logc3ei800_alexawide`

The first and third aliases **(arri_logc3_ei800** and **logc3ei800_alexawide)** could be an example of a suitable name for including in a file path.

The second alias **(Input - ARRI - V3 LogC (EI800) - Wide Gamut)** is the name that was used for this color space in ACES 1.2. This is useful when switching from an ACES 1.2 config to ACES 1.3. The Nuke script will likely reference the older name but the alias means the input transform should be set correctly without any user intervention required.

Adding an alias to a color space in an OCIO config can be as simple as appending it to the end of the comma separated list. For example:

```
- !<ColorSpace>
```

```
name: ARRI LogC3 (EI800)
aliases: [arri_logc3_ei800, Input - ARRI - V3 LogC (EI800) - Wide Gamut,
logc3ei800_alexawide, AlexaV3LogC]
...
```

Here we've added the alias **AlexaV3LogC**. This is the color space name that Nuke looks for in the first instance when it detects an ARRI V3 LogC file.



Note: In versions of Nuke prior to 14.1 the example above would typically result in an error, because there isn't a matching color space name in the ACES config.

From Nuke 14.1, the following aliases are included in our ACES 1.3 configs:

```
- Aliases (Nuke > ACES):

 AlexaV3LogC > ARRI LogC3 (EI800)
 ARRILogC4 > ARRI LogC4
 Blackmagic Film Generation 5 > BMDFilm WideGamut Gen5
 Log3G10 > Log3G10 REDWideGamutRGB
 SLog3 > S-Log3 S-Gamut3
 Gamma1.8 > Gamma 1.8 Rec.709 - Texture
 Gamma2.2 > Gamma 2.2 Rec.709 - Texture
 Gamma2.4 > Gamma 2.4 Rec.709 - Texture
```

The color space names that Nuke currently looks for can be seen in the **nuke-default** OCIO config.

For more information on aliases and authoring configs please see the official OCIO documentation.

Correcting Out of Gamut Errors Using the ACES Gamut Compression Look

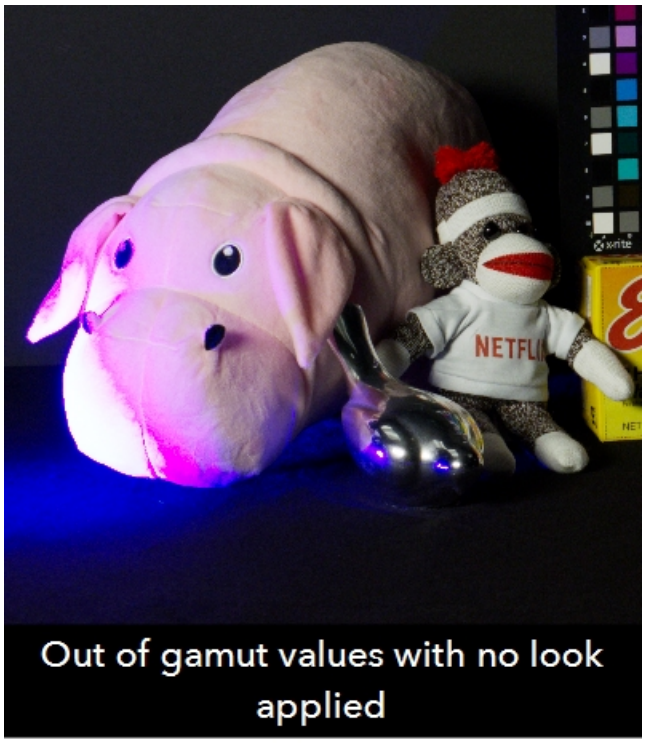
A common side effect of using ACES is artifacts that result from out of gamut values in source images. This typically affects blue light, such as that found in police car sirens and other light sources. The ACES 1.3 OCIO config files that ship with Nuke contain a look designed to correct this error.

To apply the ACES 1.3 Reference Gamut Compression look:

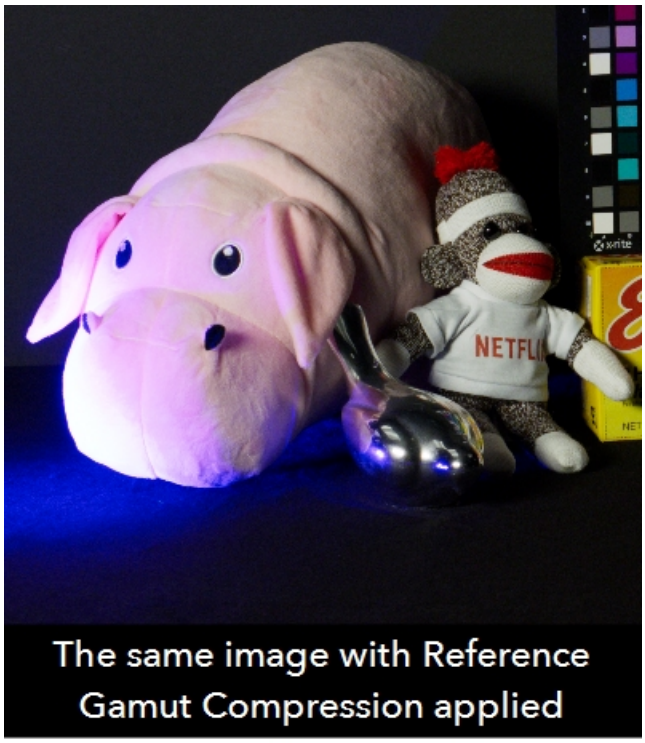
1. Go to **Edit** > **Project Settings** or press **S** in the Node Graph to display the **Project Settings**.
2. Click the **Color** tab and then select **OCIO** from the **color management** dropdown.
3. Select an ACES 1.3 config in the **OCIO config** dropdown.
4. Select the Read node in the Node Graph to which you want to apply the look.
5. Press **Tab** and start typing **OCIO** to display a list of available OCIO nodes.
6. Select **OCIOLookTransform** from the list to add the node to the graph.

The **ACES 1.3 Reference Gamut Compression** look is applied by default, because it is the only look defined in the configs that ship with Nuke.

Tip: If you're using a custom config or you've added more looks to the ACES config yourself, you can select the **ACES 1.3 Reference Gamut Compression** manually using the **look** dropdown.



Out of gamut values with no look applied



The same image with Reference Gamut Compression applied

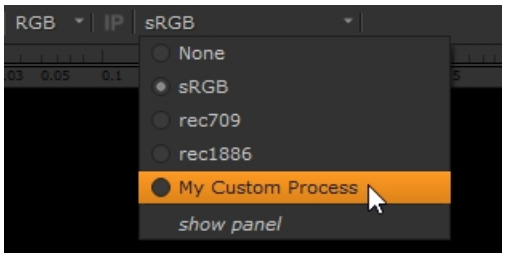
Creating Custom Viewer Processes

Using look-up tables (LUTs) in Viewer Processes, you can adjust individual Viewer displays to simulate the way the image looks on output to film or some video display device. Nuke includes some predefined Viewer Process gizmos, but you can also add your own processes by registering a node or gizmo as a Viewer Process using the **nuke.ViewerProcess.register()** class method. See the Nuke Python Reference Guide for more information.

You can register as many custom Viewer Processes as you like. If you want to use one of the 1D LUTs listed in the Project Settings in your Viewer Process, you can use the built-in gizmo called ViewerProcess_1DLUT.

Tip: There are a couple of commented out examples in the installed **init.py** file demonstrating how to use a 3D LUT for a Viewer Process. You can find this file in the following location:
On Windows:
drive letter:\Program Files\Nuke17.0v3\plugins
On Mac:
/Applications/Nuke17.0v3/Nuke17.0v3.app/Contents/MacOS/plugins
On Linux:
/usr/local/Nuke17.0v3/plugins

All available Viewer Processes (both custom and predefined ones) can be applied from the Viewer Process dropdown menu in the Viewer controls. Both predefined and custom Viewer Processes can be applied from the Viewer Process dropdown menu.



Note that Viewer Processes are part of a built-in, fixed pipeline of nodes that are applied to images before they are displayed in the Viewer. This pipeline is either:

- gain > Input Process > Viewer Process > gamma > dither > channels > cliptest (if **viewer input order** has been set to **before viewer process** in the Viewer settings)

OR

- gain > Viewer Process > Input Process > gamma > dither > channels > cliptest (if **viewer input order** has been set to **after viewer process** in the Viewer settings).

However, depending on what the Viewer Process is doing, this may not be the correct order. Therefore, if your Viewer Process (or an Input Process) has controls that also exist for the Viewer, such as controls named **gain**, **gamma**, or **cliptest**, then the Viewer drives them from the corresponding Viewer controls and does not do that image processing itself. This allows you to implement these controls in your Viewer Process using whatever nodes and order you want. If your Viewer Process does not have these controls (and they are not found on any Input Process in use either), then the Viewer applies the effects in its normal way according to the built-in pipeline.

In the built-in pipeline, dither is applied to diffuse round-off errors in conversion of floating point data to the actual display bit depth. Although the cliptest is drawn at the end, it is computed on the image as input to the Viewer.

Using a Gizmo as a Custom Viewer Process

To create a custom Viewer Process, you would typically create a gizmo that includes some color correction like a look-up table (LUT) and register it as a Viewer Process using Python. (For more information on gizmos, see Sourcing Custom Plug-ins and Generic Tcl Scripts.)

If you want to use one of the 1D LUTs listed in the Project Settings in your Viewer Process, you do not need to create a custom gizmo. Instead, you can simply register a built-in gizmo called ViewerProcess_1DLUT. This gizmo takes a parameter for which LUT to use, but does not allow it to be edited. For more information, see To Register a LUT in the Project Settings as a Viewer Process.

If you want anything more complex than a 1D LUT that can be found on the **LUT** tab of the Project Settings, you need to create your own gizmo and register that. For more information, see To Create a Viewer Process Gizmo and To Register a Custom Viewer Process.

To Register a LUT in the Project Settings as a Viewer Process

1. Create a file called **init.py** in your plug-in path directory if one doesn't already exist. For more information on plug-in path directories, see Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts.
2. To register one of the LUTs in the Project Settings as a Viewer Process, use, for example, the following function in your **init.py**:

   ```
   nuke.ViewerProcess.register("Cineon", nuke.createNode, ("ViewerProcess_
   1DLUT", "current Cineon"))
   ```

   This registers a built-in gizmo called ViewerProcess_1DLUT as a Viewer Process and sets it to use the Cineon LUT. The registered Viewer Process appears in the Viewer Process dropdown menu as *Cineon*.

   Note that you can set the built-in gizmo to use any 1D LUT in the Project Settings. For example, to set it to use the Panalog LUT, use the following function:

   ```
   nuke.ViewerProcess.register("Panalog", nuke.createNode, ("ViewerProcess_
   1DLUT", "current Panalog"))
   ```

To Create a Viewer Process Gizmo

1. Create the node(s) you want to use as a Viewer Process. For example, you can use a ColorLookup, Vectorfield (3D LUT), or Colorspace node.
2. Select the node(s) you want to include in the Viewer Process and select **Other** > **Group**.
3. To select which controls the users of your Viewer Process can adjust, right-click on the dark gray background of the Group properties panel and select **Manage User Knobs**. For more information on how to add controls to your gizmo, see Creating and Accessing Gizmos.

   If you expose controls with the same name as the controls in the Viewer (such as **gain** or **gamma**), then the controls in the Viewer are used to drive these. However, if an Input Process that exposes the same controls is also in use, the Input Process takes precedence and the Viewer controls drive it, ignoring the same-named Viewer Process control(s). For more information on Input Processes, see Using the Viewer Controls > **Input Process and Viewer Controls**.
4. Once you are happy with the modified Viewer Process group, export it to a gizmo by clicking **export as gizmo** on the **Node** tab of the group controls.
5. In the file browser that appears, click **Home**. Type **.nuke/** after the path displayed at the bottom of the file browser. Enter a name after the path, and append a **.gizmo** extension after the name. The name should begin with a capital letter. Finally, click **Save**.
6. Proceed to registering the gizmo as a custom Viewer Process, described below.

Tip: If you like, you can test your Viewer Process gizmo as an Input Process before registering it. Do the following:
1. In the top right corner of the Viewer, set the Viewer Process dropdown menu to **None**.
2. Select the gizmo in the Node Graph.
3. To toggle the Input Process on or off, click the IP button in the Viewer controls. If you are happy with the result, proceed to registering the gizmo as a Viewer Process.

For more information on Input Processes, see Using the Viewer Controls > **Input Process and Viewer Controls**.

Tip: If you want to view or modify the internals of an existing Viewer Process, you can do the following:
1. Select the Viewer Process that you want to modify from the Viewer Process dropdown menu.
2. Select **Edit** > **Node** > **Copy Viewer Process to Node Graph**. This inserts the Viewer Process gizmo you selected in the Node Graph.
3. Double-click on the gizmo to open its controls. Go to the **Node** tab and click **copy to group**. This gives you an editable group version of the gizmo.
4. In the Group controls, click the **S** button to show the internals of the group. They are shown on a new tab in the Node Graph.
5. Make your changes and export the group to a gizmo by clicking **export as gizmo** on the **Node** tab of the group controls.

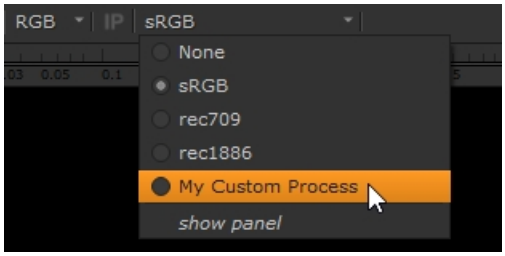
Tip: If you use the ViewerLUT node in a Viewer Process gizmo, you can toggle **rbg_only** in the ViewerLUT controls to define whether the LUT is applied to all channels or only the red, green, and blue channels. You can also expose this control in the Viewer Process gizmo's controls, so that users can set it themselves.

To Register a Custom Viewer Process

1. Create a file called **init.py** in your plug-in path directory if one doesn't already exist. For more information on plug-in path directories, see Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts.
2. To register a gizmo or a node as a Viewer Process, use the following function in your **init.py**:
   `nuke.ViewerProcess.register()`

   For example, to register a gizmo called MyProcess.gizmo as a Viewer Process and have it appear in the Viewer Process dropdown menu as *My Custom Process*, you would enter the following:
   `nuke.ViewerProcess.register("My Custom Process", nuke.Node, ("MyProcess", ""))`

   Your Viewer Process should now appear in the Viewer controls.



   If you need to unregister a Viewer Process, you can use **nuke.ViewerProcess.unregister()**. For example:
   `nuke.ViewerProcess.unregister("My Custom Process").`

To get help on the use of these statements, you can enter **help (nuke.ViewerProcess)** in the Script Editor.

Tip: You can also pass arguments to **nuke.ViewerProcess.register()**. For example, to register a Blur node with its **size** knob set to 10, you would enter the following:
nuke.ViewerProcess.register("Blur", nuke.createNode, ("Blur", "size 10"))

Tip: You can easily register any LUT defined in the project settings as a Viewer Process. For how to do this, see the installed **menu.py** file where the built-in Viewer Processes are registered. You can find **menu.py** in the following location:

On Windows:
drive letter:\Program Files\Nuke17.0v3\plugins

On Mac:
/Applications/Nuke17.0v3/Nuke17.0v3.app/Contents/MacOS/plugins

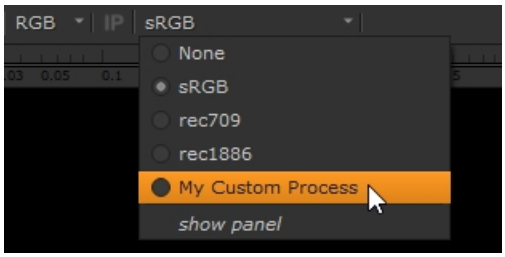
On Linux:
/usr/local/Nuke17.0v3/plugins

Applying Custom Viewer Processes to Images

In the Viewer controls, you can apply a custom Viewer process to images displayed in the Viewer and open the controls for the currently active Viewer process.

To Apply Your Custom Viewer Process to Images Displayed in a Viewer

Select the process from the Viewer Process dropdown menu in the Viewer controls.



To View the Controls of the Currently Active Viewer Process

In the Viewer controls, select **show panel** from the Viewer Process dropdown menu.

This opens the Viewer Process' properties panel. Any controls with the same name as the controls in the Viewer (such as **gain** or **gamma**) can only be adjusted using the Viewer controls. If these controls are also exposed on an Input Process and the Input Process has been activated, the Viewer controls drive the Input Process controls and the Viewer Process controls are disabled.

For more information on Input Processes, see Using the Viewer Controls > **Input Process and Viewer Controls**.

Expressions

This topic is intended as a primer on how to apply expressions (programmatic commands) to Nuke parameters. It explains how to perform some common tasks with expressions (for example, how to link the values of one parameter to another), and concludes with a table all the functions that you may include as part of an expression.

Quick Start

Here's a quick overview of the workflow:

1. You enter Nuke expressions in the Expression dialog, which you can open either by pressing the equals sign (**=**) on a parameter or by right-clicking on it and selecting **Add expression**.
2. In the Expression dialog, enter text that either references values from other parameters (creating a linking expression - see Linking Expressions) or applies mathematical functions of some kind to the current values (see Adding Mathematical Functions to Expressions). An example of the former would be **parent.Transform1.rotate**, which indicates that this control takes its values from the parent control, Transform node's **rotate** control.
3. If necessary, you can also convert expressions between scripting languages (that is, between Nuke expressions, Tcl, and Python). See Converting Expressions Between Scripting Languages.

Linking Expressions

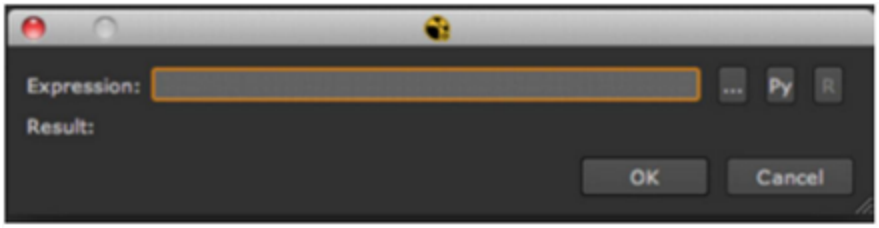


Note: Link Nodes are a quicker and more efficient way to link nodes than using expressions. Link Nodes enable you to make a linked copy of a node, and have changes updated in both nodes, whilst optionally being able to override any of the knobs. See Using Link Nodes.

Through expressions, you can link the parameters from one node and control the values of the parameters in other nodes. When creating a linking expression, type the elements listed in the table below; remember to separate each element with a period.

Element	Description
Node name	The node with the source parameter (i.e., **Transform1**).
Parameter name	The name of the parameter with the source value (for example, **translate**).The name is defined internally, and may not match the parameter's label that appear in the Nuke interface. If necessary, hover over the parameter's field with your mouse pointer and its name appears in the pop-up tool tip.
Child parameter name (optional)	Some parameters include child parameters, such as the fields for x and y axes, or red, green, and blue color channels. Child parameter names do match the label that appears before the parameter's field (for example, **x**).
Time (optional)	By default, linking expressions pull values from the current frame number, but you can read values from other frames, either statically or dynamically (that is, with a temporal offset). If you want to read in a static value for a given frame, you just type that frame number inside a set of parenthesis (for example, **(10)**). If you want to read in dynamic values but with an offset in time, type **t**, the variable for time, followed by a **+** (for a forward offset) or **-** (for a backward offset), followed by a number representing the number of frames worth of offset. For example, typing **(t-2)** would capture values that are two frames back from the current frame.

Thus, to create a linking expression that pulls the value from a Transform node's x translation field at the tenth frame, you would type **=** on a parameter to open the expression dialog, and then enter **Transform1.translate.x(10)** in the dialog's **Expression** field.



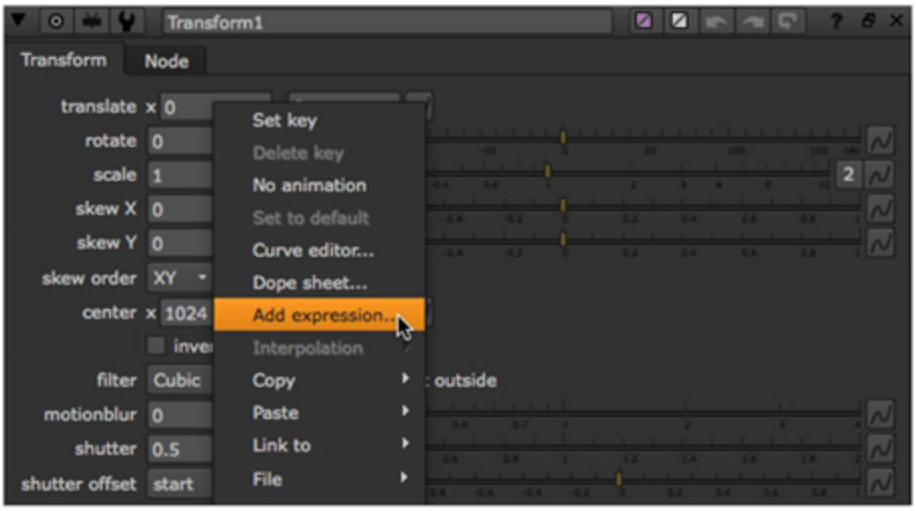
The steps below recap the process for creating a linking expression.

Referencing Values from Another Parameter

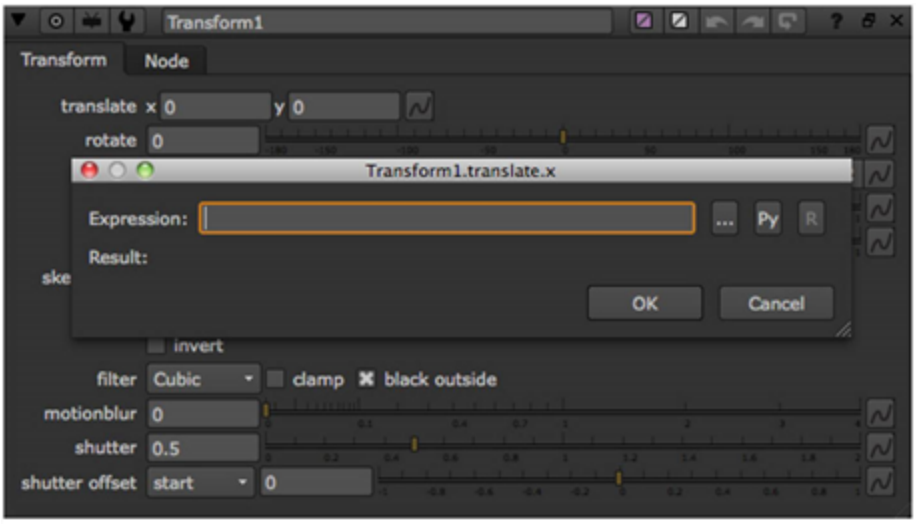
To Reference Values from Another Parameter - Method 1:

1. Click on the destination parameter (the one which receives values from another parameter).

2. To display the expression dialog, right-click on the parameter and select **Add expression**,



OR type **=** in the parameter field.

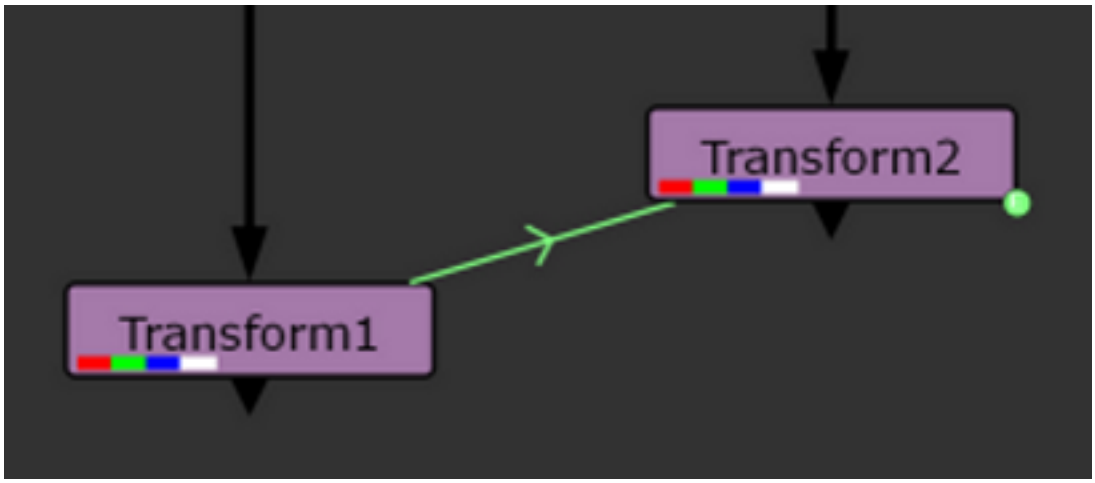


3. In the dialog that opens, type the name of the node containing the source parameter and a period. (Each node prominently displays its name on its face.)
4. If you want to enter a multi-line expression, you can click the multi-line edit field button .
5. Follow the name of the node by the source parameter's name and a period. (If you don't know the parameter's name, you can hover over its field in order to see it displayed in a tool tip.)
6. Optionally, type the child parameter's name and a period.
7. Optionally, type a frame number or offset variable in brackets (for example, **(2)** or **(t-2)**) in order to specify the frame or range of frames from which you pull values.
8. Next to the expression entry field, you can click the **Py** button to automatically make your expression a Python callback. You can also toggle the **R** button to have your expression interpreted as an expression

or as a series of statements. For example, with the multi-line edit mode and the Python mode on, you could enter the following expression, and get 15 as the resulting value:

```
-execlocal

def example():

  a = 5
  return a


def example2():

  b = 10
  return b


ret = example()+example2()
```

9. Click **OK**. This links the parameters, which turn blue. In the Node Graph, a green arrow appears between the nodes to indicate that they are linked via an expression.



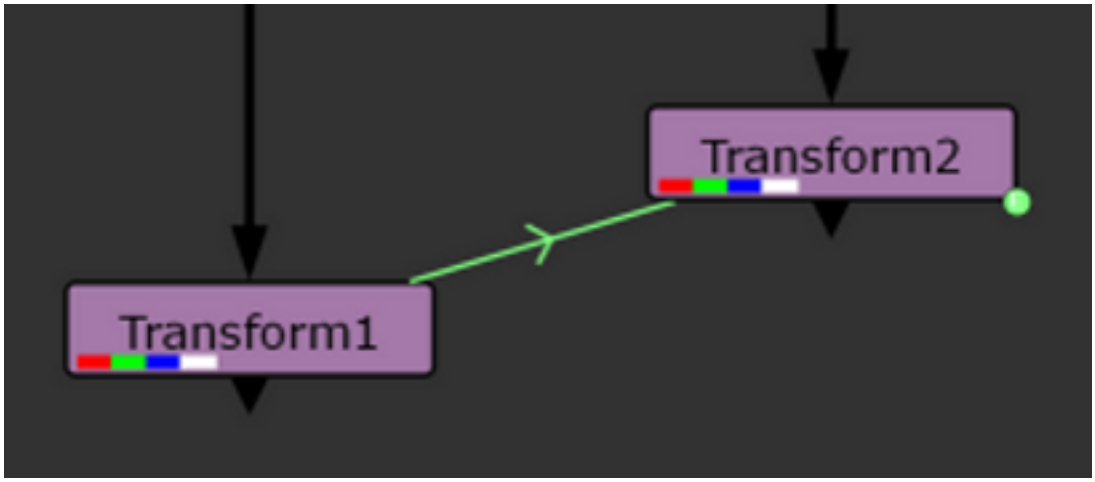
Note: Expression links between nodes are not shown if the node is cloned. For clarity, only the clone link is drawn in the Node Graph. See Cloning Nodes for more information.

10. To edit the expression later on, right-click on the parameter and select **Edit expression** (or press **=** on the parameter). You can also click the animation button and select **Edit expression** to edit expressions for all the parameters next to the button.



To Reference Values from Another Parameter - Method 2:

1. **Ctrl/Cmd**+drag the parameter that has the values you want to use on top of the parameter that receives these values. This links the parameters, which turn blue. In the Node Graph, a green arrow appears between the nodes to indicate that they are linked via an expression.



Note: Expression links between nodes are not shown if the node is cloned. For clarity, only the clone link is drawn in the Node Graph. See Cloning Nodes for more information.

   To view or edit the expression, right-click on the parameter and select **Edit expression**.

2. If you want to link several parameters at the same time, **Ctrl/Cmd**+drag the animation button next to the source parameters on top of the animation button next to the destination parameters.

   To view or edit the expressions used to link the parameters, click the animation button and select **Edit expression**.

Linking Channels and Formats Using Expressions

You can also create expression links to connect channel, layer, and format controls with other controls in various nodes. Since these controls aren't meant to be animated, you can't use the full range of Nuke

expressions, nor can you use Python or Tcl languages. You can link controls using the **Link menu**  next to the control on the properties panel:

1. Click the **Link menu** and select **Set link**. An **Expression** dialog opens.
2. Enter your expression in the Expression field and click **OK**.
3. You can edit an existing link by clicking the **Linkmenu** and selecting **Editlink**.
4. You can also **Ctrl/Cmd**+drag the **Link menu** to another control to create a link between the two.
5. To remove a link, click the **Linkmenu** and select **Removelink**.

Adding Mathematical Functions to Expressions

You can incorporate mathematical functions into parameters. For example, you might negate an expression in order to invert a tracking curve which you wish to use to stabilize an element (such an expression might resemble the following: **-(Transform1.translate.x)**).

You can also rely on a function to add more complex mathematical operation to your expressions. The table below list all the functions which you may incorporate into Nuke expressions.

Function	Purpose	Operator Usage	Related Functions	DeepExpression Compatible
abs (x)	Returns the absolute value of the floating-point number x.	x	See also: fabs.	●
acos (x)	Calculates the arc cosine of x; that is the value whose cosine is x.	If x is less than -1 or greater than 1, acos returns nan (not a number).	See also: cos, cosh, asin, atan.	●
asin (x)	Calculates the arc sine of x; that is the value whose sine is x.	If x is less than -1 or greater than1, asin returns nan (not a number).	See also: sin, sinh, acos, atan.	●

Function	Purpose	Operator Usage	Related Functions	DeepExpression Compatible
atan (x)	Calculates the arc tangent of x; that is the value whose tangent is x. The return value is between -PI/2 and PI/2.	x	See also: tan, tanh, acos, asin, atan2.	●
atan2 (x, y)	Calculates the arc tangent of the two variables x and y. This function is useful to calculate the angle between two vectors.	x, y	See also: sin, cos, tan, asin, acos, atan, hypot.	●
ceil (x)	Round x up to the nearest integer.	x	See also: floor, trunc, rint.	●
clamp (x, min, max)	Return x clamped to [min ... max].	x, min, max	See also: min, max.	●
clamp (x)	Return x clamped to [0.0 ... 1.0].	x	See also: min, max.	●
cos (x)	Returns the cosine of x.	x in radians	See also: acos, sin, tan, cosh.	●
cosh (x)	Returns the hyperbolic cosine of x, which is defined mathematically as (exp(x) + exp(-x)) / 2.	x	See also: cos, acos, sinh, tanh.	●
curve (frame)	Returns the y value of the animation curve at the given frame.	optional: frame, defaults to current frame.	See also: value, y.	
degrees (x)	Convert the angle x from radians into degrees.	x	See also: radians.	●
exp (x)	Returns the value of e (the base of natural logarithms) raised to the power of x.	x	See also: log, log10.	●
exponent (x)	Exponent of x.	x	See also:	●

Function	Purpose	Operator Usage	Related Functions	DeepExpression Compatible
			mantissa, ldexp.	
fBm (x, y, z, octaves, lacunarity, gain)	Fractional Brownian Motion. This is the sum of octaves calls to noise(). For each of them the input point is multiplied by pow(lacunarity,i) and the result is multiplied by pow (gain,i). For normal use, lacunarity should be greater than 1 and gain should be less than 1.	x, y, z, octaves, lacunarity, gain	See also: noise, random, turbulence.	●
fabs (x)	Returns the absolute value of the floating-point number x.	x	See also: abs.	●
false ()	Always returns 0		See also: true.	
floor (x)	Round x down to the nearest integer.	x	See also: ceil, trunc, rint.	●
fmod (x, y)	Computes the remainder of dividing x by y. The return value is x - n y, where n is the quotient of x / y, rounded towards zero to an integer.	x, y	See also: ceil, floor.	●
frame ()	Return the current frame number.		See also: x.	
from_byte (color component)	Converts an sRGB pixel value to a linear value.	color_ component	See also: to_sRGB, to_rec709f, from_ rec709f.	●
from_rec709f (color component)	Converts a rec709 byte value to a linear brightness	color_ component	See also: form_ sRGB, to_rec709f.	●
from_sRGB	Converts an sRGB pixel value	color_	See also: to_sRGB,	●

Function	Purpose	Operator Usage	Related Functions	DeepExpression Compatible
(color component)	to a linear value.	component	to_rec709f, from_rec709f.	
hypot (x, y)	Returns the sqrt(x*x + y*y). This is the length of the hypotenuse of a right-angle triangle with sides of length x and y.	x, y	See also: atan2.	●
int (x)	Round x to the nearest integer not larger in absolute value.	x	See also: ceil, floor, trunc, rint.	●
ldexp (x, exp)	Returns the result of multiplying the floating-point number x by 2 raised to the power exp.	x, exp	See also: exponent.	●
lerp (a, b, x)	Returns a point on the line f(x) where f(0)==a and f(1)==b. Matches the lerp function in other shading languages.	a, b, x	See also: step, smoothstep.	●
log (x)	Returns the natural logarithm of x.	x	See also: log10, exp.	●
log10 (x)	Returns the base-10 logarithm of x.	x	See also: log, exp.	●
logb (x)	Same as exponent().	x	See also: mantissa, exponent.	●
mantissa (x)	Returns the normalized fraction. If the argument x is not zero, the normalized fraction is x times a power of two, and is always in the range 1/2 (inclusive) to 1	x	See also: exponent.	●

Function	Purpose	Operator Usage	Related Functions	DeepExpression Compatible
	(exclusive). If x is zero, then the normalized fraction is zero and exponent() Returns zero.			
max (x, y, ...)	Return the greatest of all values.	x, y, (...)	See also: min, clamp.	●
min (x, y, ...)	Return the smallest of all values.	x, y, (...)	See also: max, clamp.	●
mix (a, b, x)	Same as lerp().	a, b, x	See also: step, smoothstep, lerp.	●
noise (x, y, z)	Creates a 3D Perlin noise value. This produces a signed range centerd on zero. The absolute maximum range is from -1.0 to 1.0. This produces zero at all integers, so you should rotate the coordinates somewhat (add a fraction of y and z to x, etc.) if you want to use this for random number generation.	x, optional y, optional z	See also: random, fBm, turbulence.	●
pi ()	Returns the value for pi (3.141592654...).			
pow (x, y)	Returns the value of x raised to the power of y.	x, y	See also: log, exp, pow.	●
pow2 (x)	Returns the value of x raised to the power of 2.	x, y	See also: pow.	●
radians (x)	Convert the angle x from degrees into radians.	x	See also: degrees.	●
random (x, y, z)	Creates a pseudo random value between 0 and 1. It	optional x, optional y,	See also: noise, fBm, turbulence.	●

Function	Purpose	Operator Usage	Related Functions	DeepExpression Compatible
	always generates the same value for the same x, y and z. Calling random with no arguments creates a different value on every invocation.	optional z		
rint (x)	Round x to the nearest integer.	x	See also: ceil, floor, int, trunc.	●
sin (x)	Returns the sine of x.	x in radians	See also: asin, cos, tan, sinh.	●
sinh (x)	Returns the hyperbolic sine of x, which is defined mathematically as (exp(x) - exp(-x)) / 2.	x	See also: sin, asin, cosh, tanh.	●
smoothstep (a, b, x)	Returns 0 if x is less than a, returns 1 if x is greater or equal to b, returns a smooth cubic interpolation otherwise. Matches the smoothstep function in other shading languages.	a, b, x	See also: step, lerp.	●
sqrt (x)	Returns the non-negative square root of x.	x	See also: pow, pow2.	●
step (a, x)	Returns 0 if x is less than a, returns 1 otherwise. Matches the step function other shading languages.	a, x	See also: smoothstep, lerp.	●
tan (x)	Returns the tangent of x.	x in radians	See also: atan, cos, sin, tanh, atan2.	●
tanh (x)	Returns the hyperbolic tangent of x, which is defined	x	See also: tan, atan, sinh, cosh.	●

Function	Purpose	Operator Usage	Related Functions	DeepExpression Compatible
	mathematically as sinh(x) / cosh(x).			
to_byte (color component)	Converts a floating point pixel value to an 8-bit value that represents that number in sRGB space.	color_component	See also: form_sRGB, to_rec709f, from_rec709f.	●
to_rec709f (color component)	Converts a floating point pixel value to an 8-bit value that represents that brightness in the rec709 standard when that standard is mapped to the 0-255 range.	color_component	See also: form_sRGB, from_rec709f.	●
to_sRGB (color component)	Converts a floating point pixel value to an 8-bit value that represents that number in sRGB space.	color_component	See also: form_sRGB, to_rec709f, from_rec709f.	●
true ()	Always Returns 1.	See also: false		
trunc (x)	Round x to the nearest integer not larger in absolute value.	x	See also: ceil, floor, int, rint.	●
turbulence (x, y, z, octaves, lucanarity, gain)	This is the same as fBm() except the absolute value of the noise() function is used.	x, y, z, octaves, lucanarity, gain	See also: fBm, noise, random.	●
value (frame)	Evaluates the y value for an animation at the given frame.	optional: frame, defaults to current frame.	See also: y, curve.	
x ()	Return the current frame number.		See also: frame.	
y (frame)	Evaluates the y value for an animation at the given frame.	optional: frame, defaults to current frame.	See also: value, curve.	

Converting Expressions Between Scripting Languages

Depending on where you need to use an expression, you might find that you want to, for example, convert Nuke expressions to Tcl expressions or embed Python functions in a Nuke expression. The different languages are used in different parts of Nuke:

- Python can be used in the Script Editor, in the Script Command (**File** > **Comp Script Command**) and in scripts run when Nuke starts (such as **init.py** and **menu.py**).For more information, see the Nuke Python documentation (**Help** > **Documentation**).
- Tcl can be used in most string knobs (where text other than just numbers can be entered), in the Script Command dialog (**File** > **Comp Script Command**), to open some compatibility start up scripts (such as **init.tcl** and **formats.tcl**).
- Nuke expressions can be used on the Add Expression dialog with most knobs in Nuke and expression entry field in the Expression node.

You can use the following functions to use different types of expressions together:

- **nuke.expression()** to use a Nuke expression in Python code.
- **expression** to use a Nuke expression in Tcl.
- **nuke.tcl()** to run Tcl code in Python.
- **python** to run Python code in Tcl.
- **[ ]** (square brackets) to embed Tcl in a Nuke expression (or a string knob).
- **[python {...}]** to embed Python in a Nuke expression.

Tip: Note that putting braces (**{ }**) around Python code when embedding it in Tcl may make the process a bit easier, because this prevents Tcl from performing its own evaluation of the Python code before passing it through to the Python interpreter. For example: [python {"hello " + "world"}]

Tip: Note that the "python" Tcl command by default evaluates a single line of code and returns the result. Use the "-exec" option (for example, "python -exec") if you want to run multiple lines. Please refer to the Nuke Tcl Scripting documentation (**Help** > **Documentation** > **TCL Scripting**) for further information.

The Script Editor and Python

Nuke ships with a comprehensive Python application programming interface (API), enabling you to perform user interface actions using Python scripting. This chapter describes how you can use the Script Editor for executing Python scripts and directs you to sources of more information on Python.

Quick Start

Here's a quick overview of the workflow:

1. Enter Python statements in Nuke's Script Editor to perform the required actions.
2. Save your script with the extension **.py** in a directory that is contained in the sys.path variable.
3. Later, when you want to execute the same statement sequence, import the .py file into Nuke's Script Editor. Nuke executes the statements in the specified order.

If you need more information on using Python in Nuke, you can always turn to the Nuke Python Developer's Guide (**Help** > **Documentation**).

Tip: You can import Nuke as a module into a third-party Python 3.11.11 interpreter. See Nuke as a Python Module for more information.



Tip: You can also run an interactive Python session on the command line with **nuke -t**.

Using the Script Editor

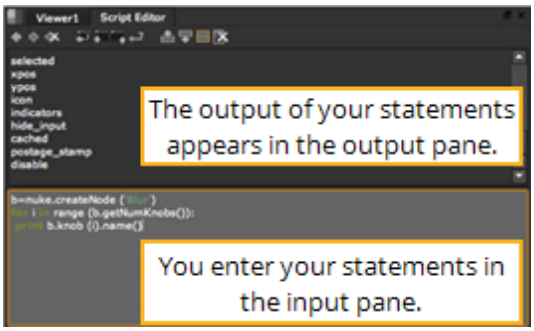
If you're not using a third-party Python interpreter, you can type Python scripts into Nuke's Script Editor.

Opening the Script Editor

To open the Script Editor, click on one of the content menus and select **Script Editor** from the menu that opens.

Input and Output Panes

The Script Editor is divided into two parts, as shown in the figure below. You use the lower part (input pane) to type in and execute your Python statement, and when you have done so, statements and their outputs appear in the upper part of the editor (output pane). Successfully executed statements are followed by a hash mark (**#**).



The two parts of the Script Editor.

To hide the output or input pane, click the **Show input only** or **Show output only** button  on top of the Script Editor.

To show both panes again, click the **Show both input and output** button .

Entering a Statement

To enter a statement in the Script Editor:

1. Click on the input pane of the editor to insert the cursor there.
2. Type in your statement. To use the usual editing functions, such as copy and paste, right-click on the editor and select the desired function.

   When entering the statement, you'll notice that any words that are Python's keywords (such as *print* and *import*) turn green, while strings (basically, anything in quotation marks) become either red or cyan. Comments are shown in yellow.



If you like, you can change these colors and the font on the **Script Editor** tab of the Preferences dialog. To open the preferences, press **Shift**+**S**.

Tip: You can also use auto-complete to help you with entering Python statements. Start writing a command and press the **Tab** key. If there's only one way to end your command, Nuke auto-completes it straight away. If there are several possible completions, Nuke gives you a pop-up menu listing them. If there's no known way to complete your command, nothing happens. Even if your command is automatically completed, it is not executed automatically, just in case you don't like surprise side effects.

3. If your statement includes several lines or you want to enter several statements at once, press **Return** to move to the next line.
4. To execute the statement, click the **Run the current script** button on the top of the Editor, or press **Ctrl/Cmd**+**Return**.

Tip: You can also execute statements by pressing **Ctrl/Cmd**+**Enter** on the numeric keypad.

By default, successful statements disappear from the input pane, and appear in the output pane. However, if you want all statements to stay in the input pane after they are executed, you can do the following:

1. Press **Shift**+**S** to open the Preferences dialog.
2. Go to the **Script Editor** tab.
3. Uncheck **clear input window on successful script execution**.
4. Click **Close** to save the preference for the current project only, or **Save Prefs** to save the preference for the current and future projects.

If you enter an invalid statement, Nuke produces an error in the output pane of the Script Editor, leaving the invalid statement in the input pane. Correct the statement and execute it again until you get it right.

Note: Sometimes you may get an error if you copy and paste statements into the Script Editor from another source, like an e-mail. This may be caused by the mark-up or encoding of the source you copied the statement from. To fix the problem, re-enter the statement manually.

If you want to have all executed Python commands appear in the output pane of the Script Editor, open the Preferences dialog (press **Shift**+**S**), go to the **Script Editor** tab, and check **echo all commands to output window**. This applies to both commands executed by yourself and by Nuke. For example, if you select a node from the Toolbar, the corresponding Python command is displayed in the output pane. This does not apply to all actions you take in the graphical user interface, however, but only those that are performed by executing Python script commands.

To only execute part of a script, enter the script in the input pane and select the part you want to execute. Press **Ctrl/Cmd**+**Return**. Nuke runs the selected part of the script, leaving the script in the input pane.

To repeat a statement, click the **Previous Script** button  on top of the Editor to move back to the previous statement. You can do this until you reach the statement you want to repeat. To execute the statement again, press **Ctrl/Cmd**+**Enter**.

To increase the indentation in the input window, press **Tab**.

To decrease the indentation in the input window, press **Shift**+**Tab**.

Moving Through and Clearing the Script History

In addition to stepping backwards through the history of your script, you can also step forwards. Click the **Next script** button  to move forward through your statements.

To clear the history, click the **Clear history** button. 

Clearing the Output Pane

Click the **Clear output window** button  (or press **Ctrl/Cmd**+**Backspace**).

Automating Procedures

Okay, so you know how to use the Script Editor to type in a sequence of Python statements that take care of a procedure. But so far, you've still sat by your computer typing the statements in. It's time to automate the

procedure. All you need to do is save your statements, and when you want to use them again later, import them into the Script Editor.

Saving Statements in a Python Module

To save statements in a Python module:

1. On the top of the Script Editor, click the **Save a script** button .
2. Save the script with the extension **.py** (for example **firstmodule.py**) in a directory that is contained in the sys.path variable. (To see these directories, enter **print sys.path** in the Script Editor. To add a directory to the sys.path variable, enter **sys.path.append ("*directory*")** where ***directory*** represents the directory you want to add.)

You have now created your first Python module.

Opening a Python Script in the Script Editor

To open a Python script in the Script Editor:

1. Click the **Load a script** button . on top of the Script Editor. The **Script to open** dialog opens.
2. Navigate to the Python module that contains the script you want to open and click **Open**.

Nuke opens the script in the input pane of the Script Editor, but does not execute it.

Importing and Executing a Python Script

To import and execute a Python script:

1. On top of the Script Editor, click the **Source a script** button . The **Script to open** dialog opens.
2. Navigate to the Python module that contains the script you want to import and click **Open**.

OR

In the input pane, enter:

import *module*

Where ***module*** represents the name of your Python module without the file extension, for example:

import firstmodule

Nuke imports the Python module and performs the procedure defined in the module.

Note: Importing the module is done according to Python's default rules. During the import, the module is searched in the following locations and order:
1. In the current directory.
2. In the directories contained in the **PYTHONPATH** environment variable, if this has been defined. (To view these directories, enter **echo $PYTHONPATH** in a command shell.)
3. In an installation-dependent default directory.
During the search, the variable **sys.path** is initialized from these directories. Modules are then searched in the directories listed by the **sys.path** variable. To see these directories, execute the statement **print sys.path** in the Script Editor.

Nuke as a Python Module

You can import Nuke as a module into a third-party Python 3.11.11 interpreter, granting full access to the Nuke Python-API, but from within a native Python interpreter instead of Nuke.

Note: Importing Nuke as a Python module is unavailable on macOS. Instead, use the Python build that ships with Nuke, which can be found here:
`/Applications/Nuke17.0v3/Nuke17.0v3.app/Contents/MacOS/python.app`

Note: Foundry cannot provide customer support for third-party Python interpreters.

Windows Setup

1. To import the Nuke module into a third-party Python interpreter, you can add C:\Program Files\Nuke<version> to your **PATH** environment variable. See Environment Variables for more information.
2. In Version 15.1 and higher you also need to ensure the USD dependencies can be found. This can be done by setting the following environment variables:
   a. **USG_SHIMLIB_PATH** to C:\Program Files\Nuke<version>
   b. **USG_USD_LIB_PATH** to C:\Program Files\Nuke<version>\FnUSD\lib
   c. **USG_USD_PLUGIN_PATH** to C:\Program Files\Nuke<version>\FnUSD\plugin\usd

3. Add the file path for Nuke's site-packages directory to the usrlocal.pth file in your Python 3.11.11 install. This is done by adding **C:\Program Files\Nuke<version>\lib\site-packages** to the **usrlocal.pth** file. Alternatively, you can add **C:\Program Files\Nuke<version>\lib\site-packages** python to your **PYTHONPATH** environment variable.



Tip: You can also use relative paths to the directory containing the usrlocal.pth file.



Note: If you encounter problems with FnUsdShim.X.Y.dll not being found then try setting **USG_SHIMLIB_NAME** to **C:\Program Files\Nuke<version>\FnUsdShim.X.Y.dll** . X and Y can be found by examining the name of that dll in the directory. Then delete the **USG_SHIMLIB_PATH** environment variable.

Linux Setup

1. To import the Nuke module into a third-party Python interpreter, you can add <path_to_nuke> to your **LD_LIBRARY_PATH** environment variable. See Environment Variables for more information.
2. Add the file path for Nuke's site-packages directory to the usrlocal.pth file in your python 3.11.11 install site-packages directory. For Python 3.11.7, for example, this is done by adding **<path_to_nuke>/lib/python3.11/site-packages** to the **usrlocal.pth** file. Alternatively, you can add **<path_to_nuke>/lib/python3.11/site-packages** python to your **PYTHONPATH** environment variable.

Usage

At the Python prompt, use the **import nuke** declaration to make Nuke's Script Editor functions and commands (such as **nuke.nodes.Blur()** to add a Blur node) available in your chosen Python interpreter.

The **import nuke** function checks-out a nuke_r render license by default. If you want to use Nuke interactively, and you have a nuke_i license, set the NUKE_INTERACTIVE environment variable to 1.

For more information, see Nuke as a Python Module in the Nuke Developer Guide.

Getting Help

In the scope of this user guide, it's not possible to go into detail with Python and all the scripts available. However, there are several sources of more information that you may find useful if you need help using Python.

More Documentation

In the Python Developer's Guide, you'll find everything you need in terms of Python examples and ways of using Python for various purposes in Nuke. You'll find the Python Developer's Guide when you click **Help** > **Documentation** > **Python Developers Guide** in Nuke.

You may also want to read the Nuke Python API reference documentation under **Help** > **Documentation** > **PythonScriptingReference**.

Viewing More Examples

We only described a few examples of Python scripts in this chapter, but there are more. You can find them in the following location:

- **On Windows**:

  drive letter:\Program Files\Nuke17.0v3\plugins\nukescripts
- **On Mac**:

  /Applications/Nuke17.0v3/Nuke17.0v3.app/Contents/MacOS/plugins/nukescripts
- **On Linux**:

  /usr/local/Nuke17.0v3/plugins/nukescripts

To view an example, select one of the **.py** files and open it in any text editor.

Using the Help Statement

Possibly the quickest way of getting help on the available commands is to enter the following in the Script Editor:

help (nuke)

This statement lists many of the available Python commands in an alphabetical order together with their descriptions.

You can also get help on more specific things. For example, the following statement gives you a description of what the **setValue()** method does:

help (nuke.Knob.setValue)

Python on the Web

To read more about Python, check out its documentation, or interact with other Python users. Visit the Python programming language official website at http://www.python.org/.

Timeline Editing in Nuke Studio and Hiero

Nuke Studio's timeline is designed to provide shot management, conform, and playback capabilities for people creating visual effects and delivers visual effects sequences without resorting to other third party applications. These are the topics covered:

- Using Tags tells you how to quickly sort or filter source clips and shots for better visibility, organization, and export.
- Viewing Metadata describes how to examine extra clip data and filter your project to find the footage you need.
- Conforming Sequences describes the process of matching up the footage from a shoot with the required edit decisions to create a meaningful timeline.
- Managing Timelines describes the timeline interface and how to add and manage footage on timelines.
- Adding Effects on the Timeline tells you how to add real-time GPU effects to timeline shots.
- Create Comp explains the difference between comps and regular shots and how to create and manage them.
- Annotations allow you to add editorial comments to your timeline output, enabling collaborative work between Nuke Studio and other Nuke workstations.
- Sync Review allows multiple participants to collaborate on a single project in Nuke Studio, Hiero, or HieroPlayer.
- Timeline Editing Tools describes how you manipulate your shots directly in the timeline using a series of modal editorial tools that complement the **Multi Tool**.
- Versions and Snapshots describes how to record the different states of your workflow as you progress using versions and snapshots.
- Exporting from Nuke Studio deals primarily with Nuke Studio's shot management and export functionality when you're farming out shots or sequences to other artists. It also deals with presets, which dictate how **Create Comp** passes data between the Timeline and Compositing environments.
- Contact Sheet in Nuke Studio describes how to review and compare shots in the Contact Sheet Viewer.

Using Tags

Tags are used by Nuke Studio to quickly sort or filter clips and shots for better visibility, organization, and export. Tags are used to mark shots of a particular type or content as you organize your project. The default tags supplied include **Approved**, **Note**, **Reference**, and other general purpose tags. You can also create custom tags by right-clicking in the **Tags** tab or by pressing **Ctrl**/**Cmd**+**Y** in the **Tags** tab. You can apply tags to clips, shots and Comp shots, individual frames, sequences, and tracks.

Clip and shot tags and notes can be added to exports using the burn-in feature. See Adding Burn-in Text to Exports for more information.

Tags can also be converted to Final Cut Pro or Premiere markers during export to XML. See Exporting Sequences as EDL, OTIO, and XML for more information.

Using Quick Tags

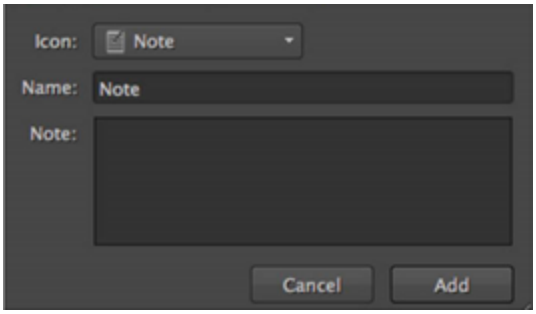
Quick tags allow you to add tags, depending on context, by right-clicking a selection and then choosing the type of tag to apply. If you're tagging a large amount of media, you might find it more convenient to use the drag-and-drop methods described later on.

Quick tags are accessible from bins, spreadsheets, Viewers, and timelines for single or multiple selections.

1. Select the target clips or sequences.
2. Right-click a highlighted selection, go to **Tags**, and choose the required action, dependent on context.

   For example, bin items only allow you to **Tag Selection**, whereas shots allow you to **Tag Shot Selection**, **Tag Tracks**, or **Tag this Sequence**

   Once you've selected the tag type, the **Add Tag** dialog displays.



3. Select the icon to represent the tag using the **Icon** dropdown.

4. Enter a **Name** and **Note** as required.
5. Click **Add** to mark your selections with the chosen tag.

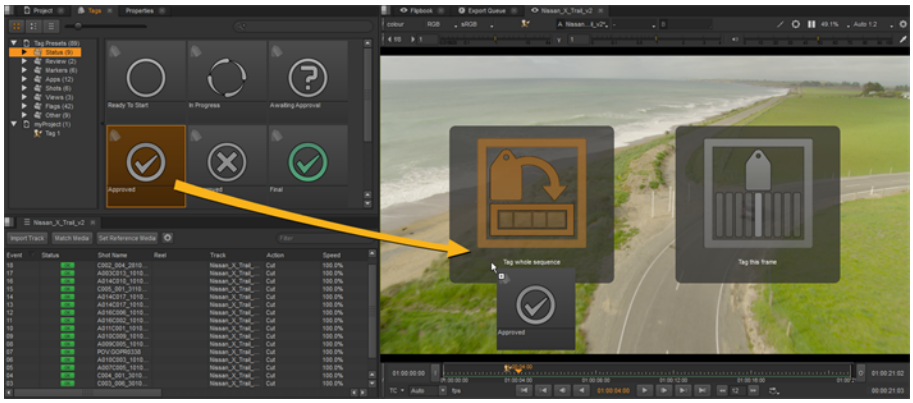
   See Creating Custom Tags and Removing Tags for more information.

Tagging Using the Viewer

To apply a tag using the Viewer:

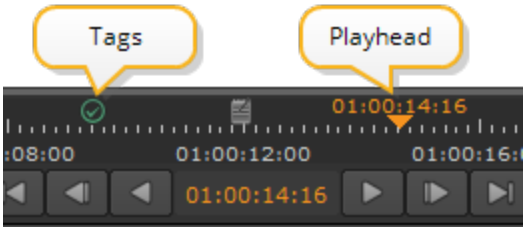
1. Click the **Tags** tab, or navigate to **Window** > **Tags**.

   The **Tags** panel displays.
2. Drag-and-drop the required tag from the **Tags** panel to the Viewer.



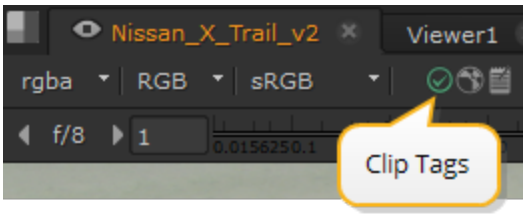
Depending on whether you're looking at a clip or sequence, drop the tag on **Tag this frame**, **Tag whole clip**, or **Tag whole sequence** as required.

Tags applied to frames appear above the playback tools in the Viewer frame slider.



Tip: You can use **Alt**+**Shift**+left and right arrows to skip to the previous or next tag on the current clip. You can also reposition tags by dragging them along the Viewer timeline.

Tags applied to clips are displayed above the Viewer.



Tagging Shots

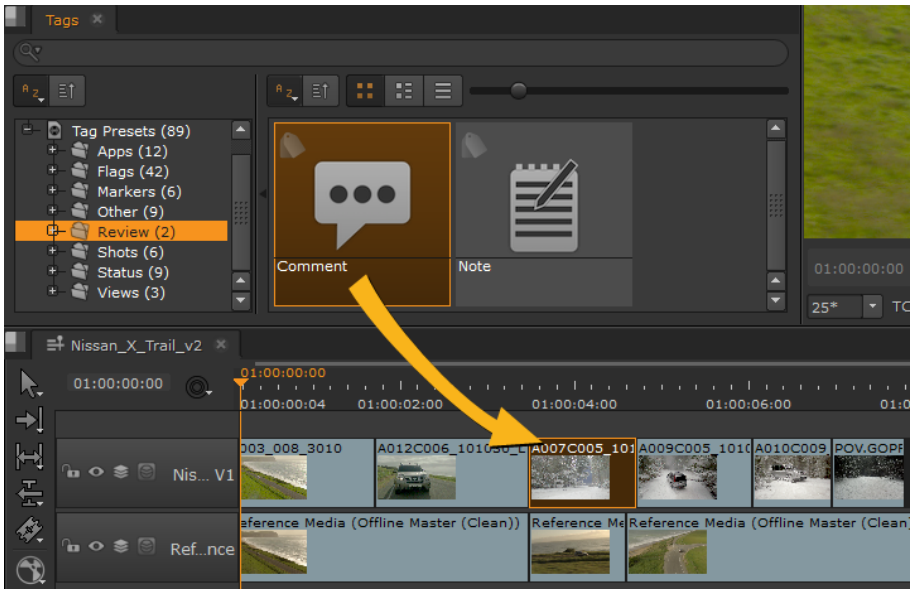
To apply a tag to a shot on the timeline:

1. Click the **Tags** tab, or navigate to **Window** > **Tags**.

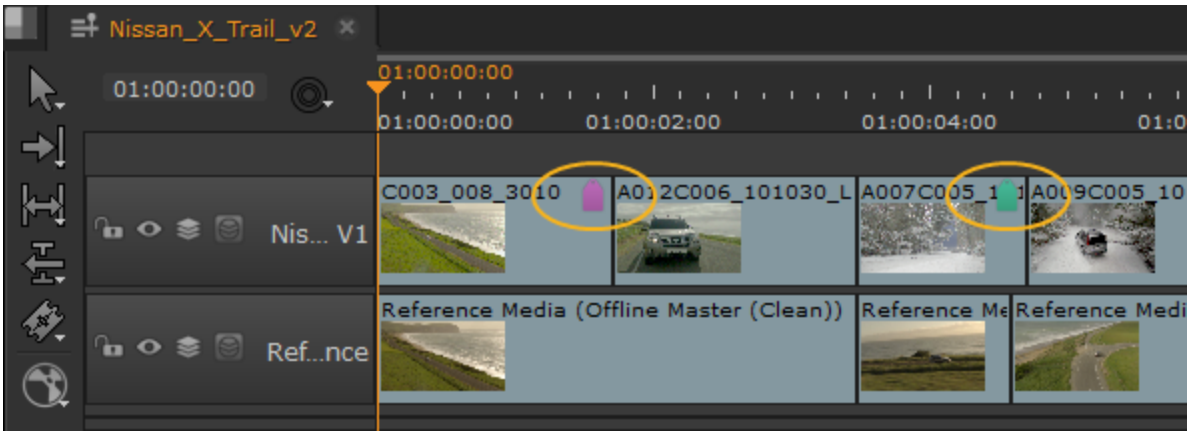
   The **Tags** panel displays.

2. Drag-and-drop the required tag from the **Tags** panel to the timeline.

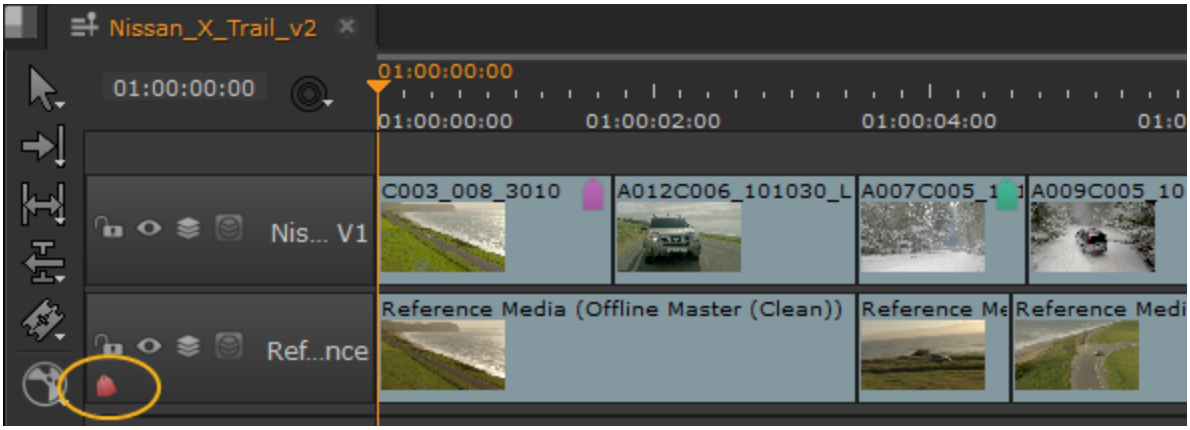
   Depending on where the tag is dropped, you'll mark a shot (or items if you make multiple selections) or a track.



Tags applied to shots appear on the right of the selected item(s) on the timeline.



Tags applied to tracks appear in the track header on the left of the timeline.



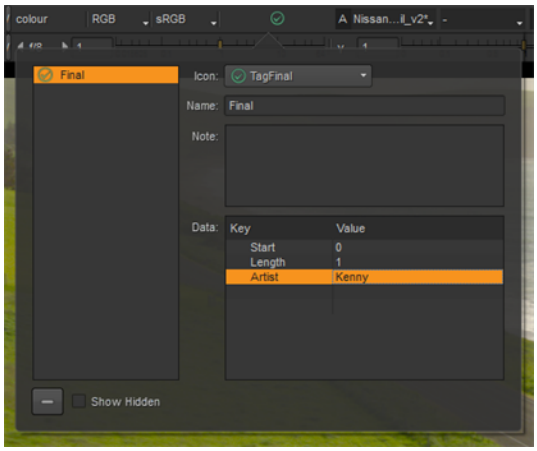
Adding Notes to Tags

In some cases, a simple tag on a frame or clip may not contain all the information that you wish to pass on to the next stage of production. Adding notes to a tag can provide that extra detail.

Warning: To delete a note, don't click the - button because this refers to the tag. Instead, simply delete the notes in the window and click outside the note dialog.

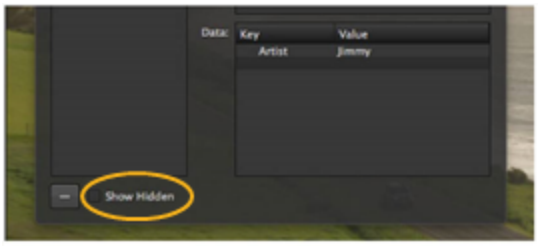
1. Add notes to tags by clicking on the required tag and entering text or editing the metadata keys and values.

   The example shows a note and metadata key "Artist" added to a clip tag, but you can add notes to frame and timeline tags in the same way.



2. Click outside the dialog to save the note.

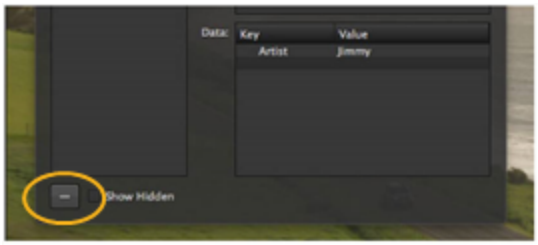
Nuke Studio allows you to "hide" tags using the Python API. Hidden tags are not displayed in the interface, unless you enable **Show Hidden** in the Tags popup, but the notes and metadata are still accessible.



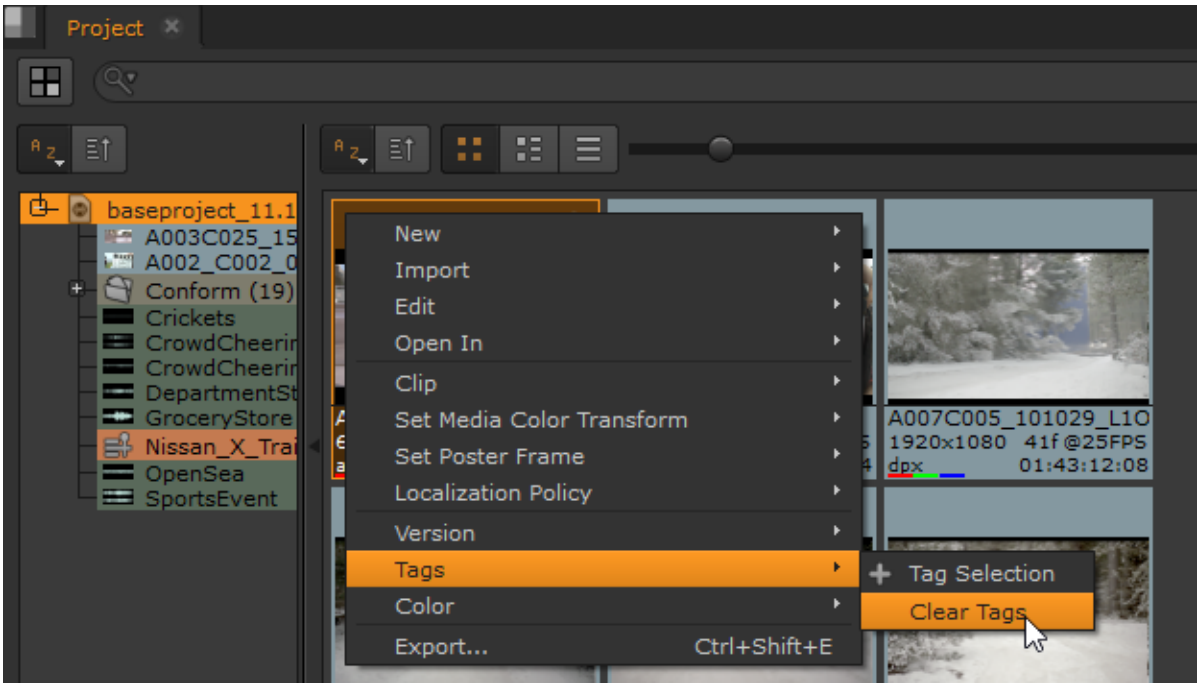
Navigate to **Help** > **Documentation** > **Hiero Python Developer's Guide** for more information on creating hidden tags.

Removing Tags

To remove a tag from a frame or shot, click the tag and then click .



You can remove all tags from a source clip or selection of clips by right-clicking your selections in the bin and choosing **Tags** > **Clear Tags**.



To remove a tag from a track or shot, click on a tag icon and select the required tag to remove.

Click  to remove your selection.

Creating Custom Tags

You may find that you require a specific tag or suite of tags that are not provided by default. Creating custom tags allows you to really control the organization of your media, and you can even create your own tag icons.

Note: Custom tags can only be created in the **Tags** panel.

To create a custom tag:

1. Click the **Tags** tab, or navigate to **Window** > **Tags**.
2. Select your project and navigate to **Project** > **New Tag**, or press **Ctrl/Cmd**+**Y**.
   The new tag is placed in the selected project.
3. Double-click the tag to open the **Edit Tag** dialog box.
4. Click the **Icon** dropdown menu to select an icon for the custom tag.



Tip: You can import your own image for the tag by selecting **Custom** to open the browser.

5. Enter a description for the tag in the **Name** field.
6. Click **OK** to save your changes.

Sharing Custom Tags

Custom tags can be shared between artists by saving them in a project in a shared network location. Shared tags are loaded at startup and appear in the **Tags** tab, below the standard tags that ship with Nuke Studio.

1. Create the required custom tags as described in Creating Custom Tags.
2. Save the **.hrox** project in a folder called **Templates** in a shared network location. For example:
   /SharedDisk/NukeStudio/Templates/SharedTags.hrox



Note: If your custom tags use custom icons, save the icon files in the same directory as the **.hrox** project.

3. Create a Python file containing the following lines, to direct Nuke Studio to the shared location:
   ```
   import hiero.core
   hiero.core.addPluginPath("/SharedDisk/NukeStudio")
   ```
4. Save the **.py** file in the **~/.nuke/Python/Startup/** directory on all the machines running Nuke Studio that require access to the tags.



Note: You may need to create the **.nuke/Python/Startup/** path manually.

   The location of the **.nuke** directory differs by platform. See Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts for more information.

5. Launch Nuke Studio and navigate to **Window** > **Tags** or switch to a workspace that contains the **Tags** tab.
   The custom tags are listed under the project name below the standard tags that ship with Nuke Studio.

Filtering and Flagging Media Using Tags

You can search for clips containing certain tags, for example, if you wanted to find all clips that you tagged as **Approved**.

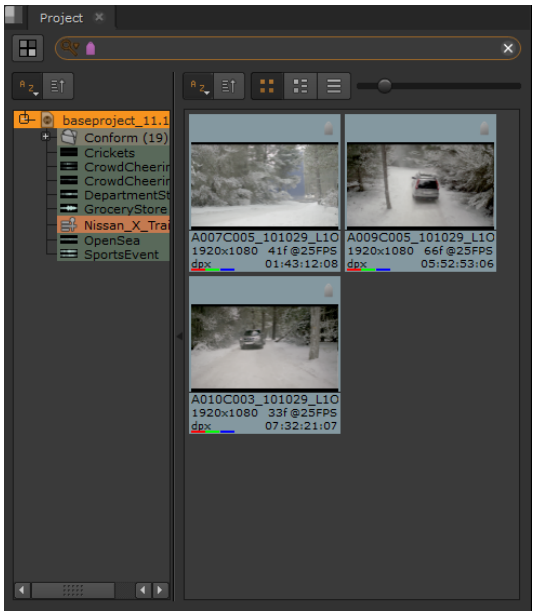
There are two types of tag search you can perform: **Filter** and **Flag**. Select the desired search type by clicking the magnifier icon in the **Project** tab.

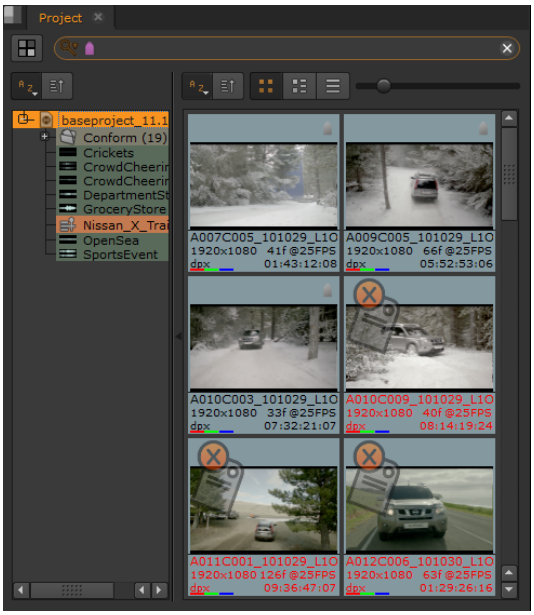
- **Filter** - displays all objects that contain the specified tag. This is the default search method.
- **Flag** - displays all objects and marks the items that don't match the search tag.

Drag the required tag from the **Tags** panel into the search box and select the bin or bins you want to Filter or Flag.

Tip: If you have more than one search criteria, click the icons in the search box to display a brief description of the icons.
Filters and flags persist until you change the search criteria or click the **x** icon in the search box.

The following examples show **Filtering** a bin to display only clips with the **Notes** tag applied and **Flagging** all clips that don't have the **Notes** tag applied.





Filtering ...

... Flagging.

Viewing Metadata

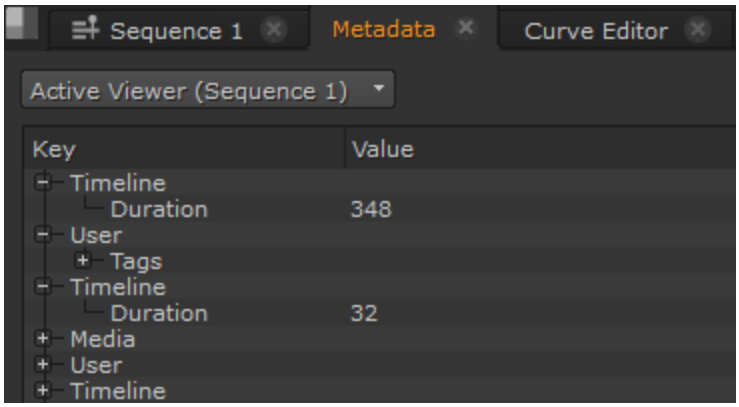
Metadata is information that describes media content, separate from the clip itself, in the form of a table on the **Metadata** tab. Types of metadata include **Duration**, **File Size**, and the **Path** to the location of the source media.

Source Clip and Shot Metadata

To view metadata for a source clip, right-click the clip and select **Open In** > **Metadata View**, or press **Alt**+**D**.

To view metadata for a shot, select the **Metadata** tab in the timeline panel and click on the item to examine.

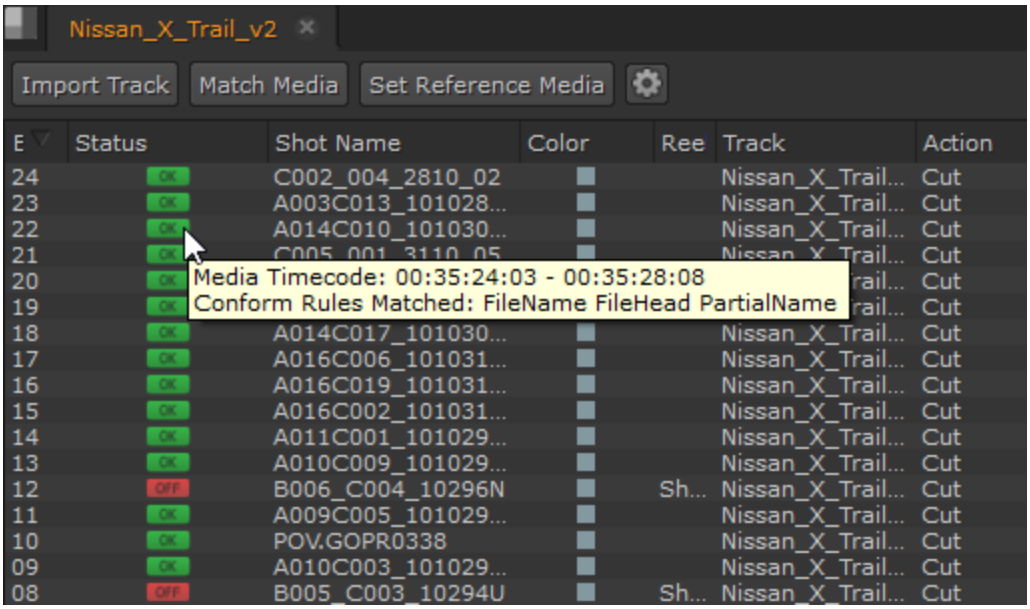
Tip: You may have to add the **Metadata** tab manually by clicking the icon and selecting **Window** > **Metadata**.



Conforming Sequences

Conforming describes the process of matching up the footage from a shoot with the required edit decisions to create a meaningful timeline. Nuke Studio accepts sequences either from EDLs (edit decision lists), OTIO (OpenTimelineIO), AAFs (advanced authoring format), or Final Cut Pro XML files from a specified directory structure containing the source media files. Nuke Studio attempts to conform the media, warning you if there are missing media.

Nuke Studio conforms EDLs into single tracks, and OTIOs, AAFs and XMLs into multi-track timelines. You can either conform into a brand new timeline, or into an existing timeline by adding new tracks. For example, when conforming multiple EDLs into the same timeline, you would add new tracks for each EDL sequence conformed.



Timeline Environment Project Settings

A good place to start work is by defining default **Project Settings** before importing sequences, particularly in the case of EDLs as they may not contain frame rate information. **Project Settings** only apply to the current project and override **Preferences** settings.

Note: You can modify **Project Settings** later on, for example, when you're ingesting media.

Tip: You can create templates for Nuke Studio by creating **.hrox** files in the **.nuke/Templates** folder. This helps if you have particular **Project Settings**, such as Colorspaces, footage, and folder structures, that you wish to reuse across multiple projects. See Creating and Using Project Templates.

To define Project Settings:

1. Navigate to **Project** > **Edit Settings**.

   The **Project Settings** dialog displays.
2. Click the **General** sub-menu to set the project **Name**.
3. Enter a **Project Directory** if required. This is the location of the **.hrox** project file and can be used as the root of the project if you want to use relative paths to source clips. See About Clips and Shots for more information.

   If you want this setting to apply to all new projects, use the **Preferences** > **Project Defaults** > **General** panel settings.

Tip: Click **Hrox Directory** to automatically enter an expression that evaluates to the **.hrox** location.

4. Set the **Poster Frame** used by Project bin clips or use the default **First** frame. See Setting Poster Frames for more information.
5. Click the **Sequence** sub-menu to set the default **Output Resolution**, **Frame Rate**, and **Start Timecode** for new timelines in the current project, and set clip formatting when new clips are added to the timeline.

6. Click the **Views** sub-menu to set up multi-view or stereo projects. See Stereoscopic and Multi-View Projects for more information.
7. Click the **Color Management** sub-menu to manage the display and file colorspaces for this project.

   See Color Management Settings for more information.

Tip: Use the options in **Preferences** > **Project Defaults** > **Color Management** to apply them to all new projects.

8. Click the **RED Settings** sub-menu to define the **Default Video Decode Mode** for new R3D files in the current project. This setting overrides the **Preferences** > **Behaviors** > **File Handling** > **default red clip video decode mode** control for existing projects. SeeAppendix A: Preferences for more information.



Note: Changing this setting does not change the decode setting for R3D media that has already been imported.

   The dropdown contains a sliding resolution scale from **FullPremium** to **SixteenthGood**, but bear in mind that higher resolutions are slower to decode.
9. Lastly, click the **Export/Roundtrip** sub-menu to select:
   - **External Media Track Name** - sets the default name of the track created when exported media is brought back into Nuke Studio.
   - **Export Directory** - sets whether the **Project Directory**, if specified, or a custom directory is used for exports. If no **Project Directory** is specified, the project root in the **Export** dialog is used.

     If you want this setting to apply to all new projects, use the **Preferences** > **Project Defaults** > **General** panel settings.
   - **Custom Export Directory** - when **Export Directory** is set to custom, enter the required custom export directory.
   - **Shot Preset** - sets the default preset to use when you select **Create Comp** from the timeline. See Create Comp for more information.

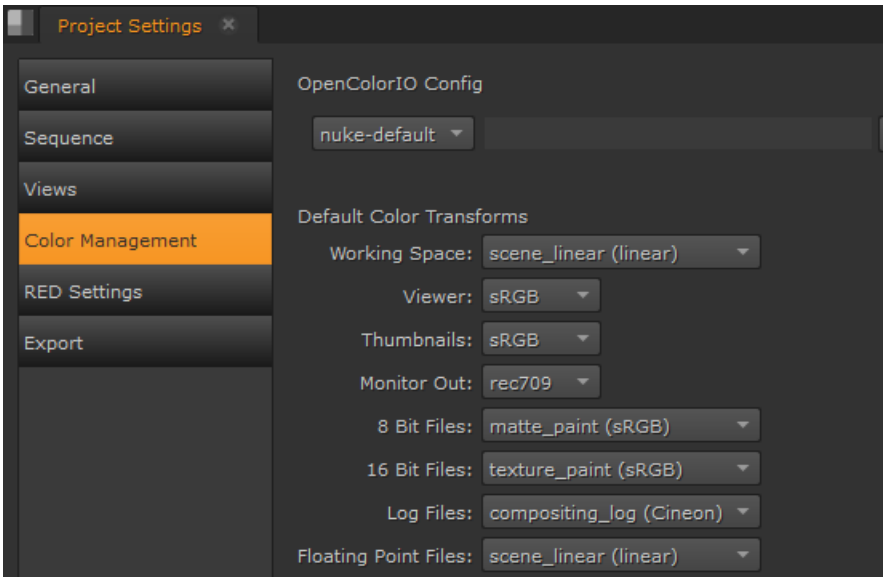
Color Management Settings

Nuke Studio uses OpenColorIO for color management. All of the colorspaces in Nuke Studio, whether those shipped with the application or custom colorspaces are defined in OCIO config files.

Depending on the OCIO config file that you are working with, there are a number of colorspace options and roles (aliases to colorspaces) that you can set in Nuke Studio. There are also default options, which change

depending on what file type you are working with. When the default option is selected, the colorspace that Nuke Studio has set for it is listed in brackets.

Tip: Use the options in **Preferences** > **Project Defaults** > **Color Management** to apply them to all new projects.

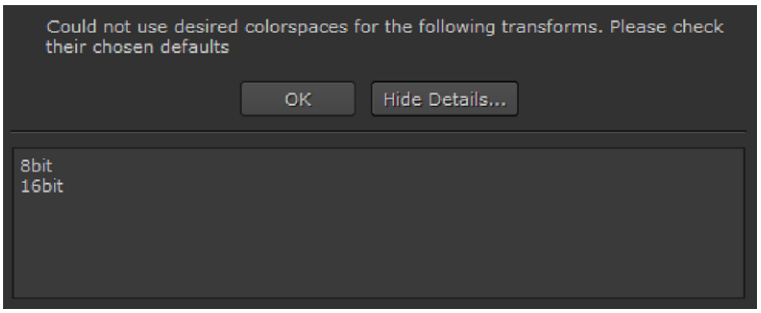


1. Set the OpenColorIO Config you want to use for this project.

   Nuke Studio ships with a number of default configurations, but you can:

   - use a custom OCIO config file by selecting **custom** from the **OpenColorIO Config** dropdown and then entering the file path, or
   - add your own config to your **.nuke** file. See Adding OCIO Configurations for more information.

   Changing the configuration updates the **Default Color Transforms** accordingly. If the selected configuration is invalid for certain transforms, a warning displays. For example, if you choose the shipped **iff** configuration, the **8-bit** and **16-bit** transforms are not compatible.



   In this case, the non-compatible transforms are set to the **raw** colorspace.

2. The **Working Space** transform determines what colorspace files should be converted to, on import, and from, during export - it's the colorspace used by the Timeline environment under the hood.

Note: In earlier releases of Nuke Studio, this colorspace was hidden because **linear** was always chosen as the **Working Space**. You may find that some operations work better in colorspaces other than **linear**. For example, some transforms work better in the **CLog** colorspace.

3. You can use **Default Color Transforms** dropdown menus to override how clips in the Viewer, thumbnails, and so on are converted to and from the **Working Space**.
4. The **Nuke Script Project Settings** dropdown determines whether Nuke Studio uses the LUTs read from the configuration specified or the **Nuke** native LUTs during export. Selecting **OCIO** makes the relevant OCIO LUTs available to the Read and Write nodes in scripts on a per project basis.

   All configurations except **nuke-default** automatically switch this control to **OCIO**.

When the **Nuke** is selected, Reads and Writes work the same as in previous versions of Nuke, with no integrated OCIO transforms. When **OCIO** is selected:

- Reads and Writes use OCIO transforms, with no Nuke built-in LUTs applied to the image.
- Read and Write colorspace controls are populated with the list of colorspaces defined in your currently selected OCIO config.
- The **default LUT settings** dropdowns are also populated with the list of colorspaces or display transforms defined in your OCIO config. The default value for each menu match the defaults in a Nuke Studio project with the same config. These defaults can be overridden using Python callbacks. See the following path for the default implementation that ships with Nuke:

  `<install_dir>/plugins/nuke/colorspaces.py`
- The **Working Space** dropdown allows you to change the colorspace that Nuke uses internally for its image processing. This automatically sets the input colorspace of Write nodes and Viewer Processes, and the output colorspace for Read nodes. This defaults to the scene **linear** role defined in your OCIO config.
- Nuke Studio-created comps no longer contain automatically injected OCIOColorspace nodes. Instead, OCIO Color Management is automatically set in the comp's **Project Settings**, and the correct OCIO colorspace is set directly into the Read and Write nodes.

Adding OCIO Configurations

You can add your own OCIO configurations to Nuke Studio as they become available, such as new versions of ACES. You can also add legacy configs for backward compatibility.

1. Navigate to the location of your **.nuke** file as shown by platform. You may have to create a **.nuke** folder if it doesn't exist.
   - Linux: /users/login name/.nuke
   - Mac: /Users/login name/.nuke
   - Windows: ~\.nuke



Note: On Windows, the **.nuke** folder can be found under the directory pointed to by the HOME environment variable. If this variable is not set (which is common), the **.nuke** directory is under the folder specified by the USERPROFILE environment variable - which is generally of the form *drive letter:*\Documents and Settings\login name\ or *drive letter:*\Users\login name\

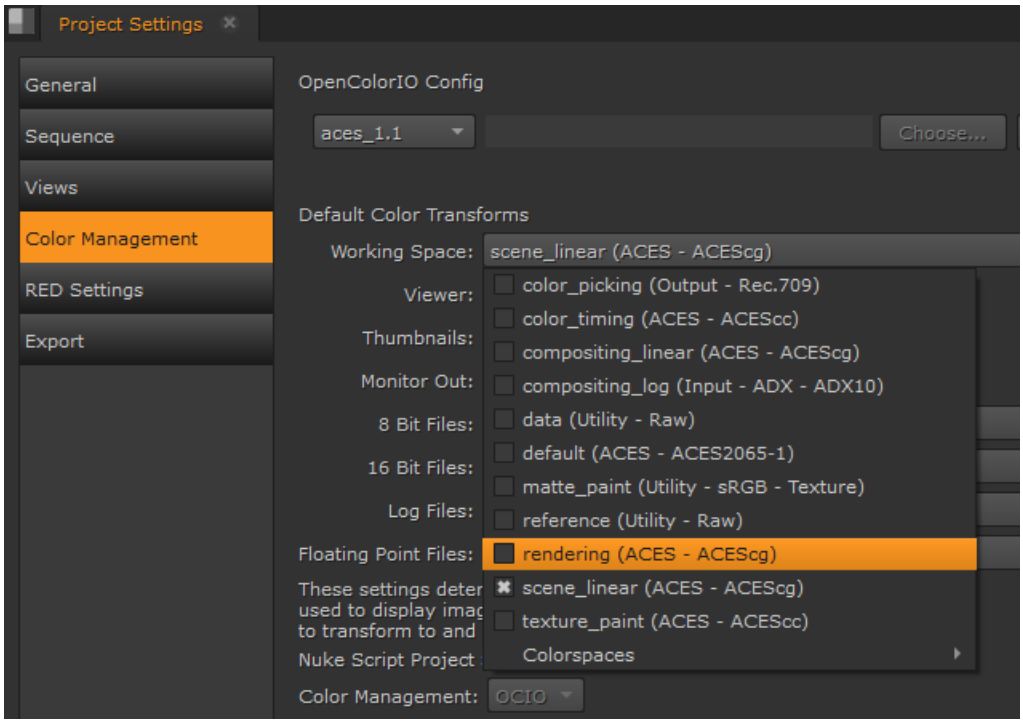
To find out if the HOME and USERPROFILE environment variables are set and where they are pointing at, enter **%HOME%** or **%USERPROFILE%** into the address bar in Windows Explorer. If the environment variable is set, the folder it's pointing at is opened.

2. Recreate the following structure within your **.nuke** folder:
   `~/plugins/OCIOConfigs/configs/<config name>`
3. Copy the contents of the config into the **config name** named folder. There should be a **luts** folder and **.ocio** file at the bare minimum.
4. If Nuke Studio is already running, relaunch the application to apply the change.
5. You can now select your configuration from the **Project Settings** > **Color Management** > **OpenColorIO Config** dropdown.

Adding OCIO Roles

OCIO roles allow you to set custom role names for different colorspaces to make it easier for artists to instinctively know which LUT to use for any shot. For instance, if an element is coming from your matte painting department and should always be brought into Nuke as sRGB, you can create a **matte painting** role, which is associated with the sRGB colorspace for your artist to select.

OCIO roles are the primary method for selecting colorspaces. All of the colorspaces in the OCIO config file are still accessible, but they have been grouped together into a **Colorspaces** menu under the roles.



OCIO roles are stored in **config** files, some of which ship with Nuke in the following directory:

```
<install_dir>/plugins/OCIOConfigs/configs/
```

For example, the aces_1.1 **config** file includes the following roles:

```
roles:
  color_picking: Output - Rec.709
  matte_paint: Utility - sRGB - Texture
  scene_linear: ACES - ACEScg
  texture_paint: ACES - ACEScc
```

The first part of the role defines the name of the **role** displayed in Nuke and second part describes colorspace **family** and **name**. The family and name define which colorspace is associated with the role. For example:

```
- !<ColorSpace>
  name: ACES - ACEScg
  family: ACES
  equalitygroup: ""
  bitdepth: 32f
  description: |
    The ACEScg color space

    ACES Transform ID : ACEScsc.ACEScg_to_ACES
  isdata: false
  allocation: lg2
  allocationvars: [-8, 5, 0.00390625]
  to_reference: !<MatrixTransform> {matrix: [0.695452, 0.140679, 0.163869, 0,
```

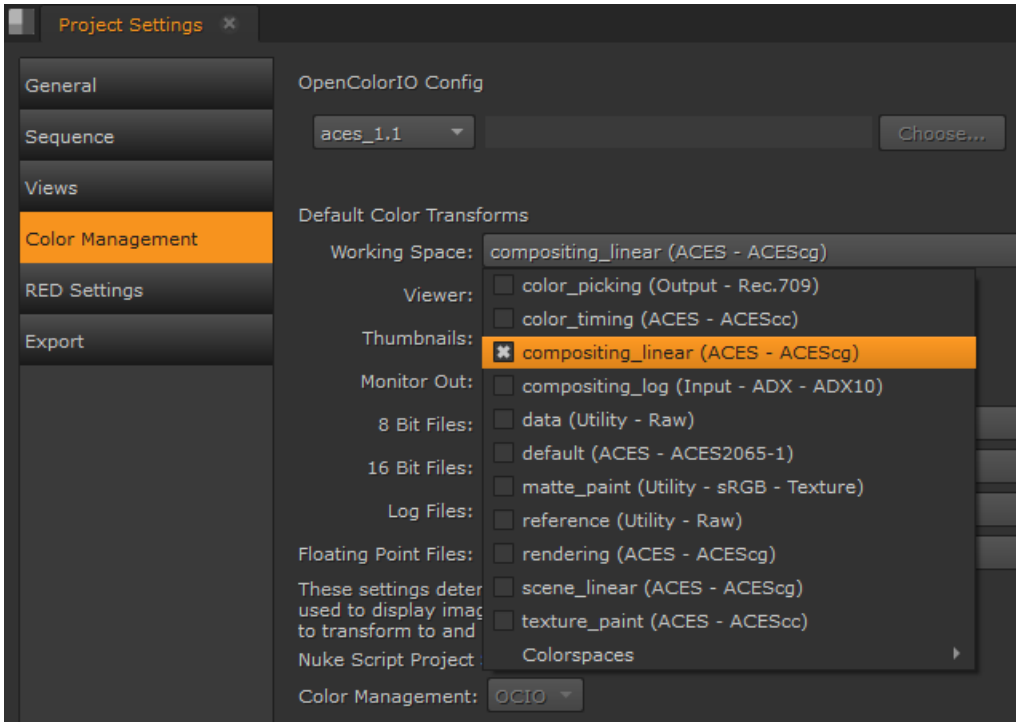
```
0.0447946, 0.859671, 0.0955343, 0,
  -0.00552588, 0.00402521, 1.0015, 0, 0, 0, 0, 1]}
```

Note: The LUT specified must exist in the **luts** directory in the same location as **config** file for the role to pick up the required colorspace.

You can edit these files to add roles or create your own custom config and then point Nuke to the file using the **Project Settings** > **Color Management** > **OpenColorIO Config** field.

To add a role to a config file:

1. Open the required config file or create a custom config.
2. Enter the **role** , **family**, and **name** of the role under the **roles:** line. For example:
   `compositing_linear: ACES - ACEScg`
3. Save the file and open Nuke.
4. Open the **Project Settings** and click the **Color Management** tab.
5. You can now pick your role from the default LUT settings. For example, you can set the working space to **compositing_linear** if you want to work in the ACEScg colorspace.

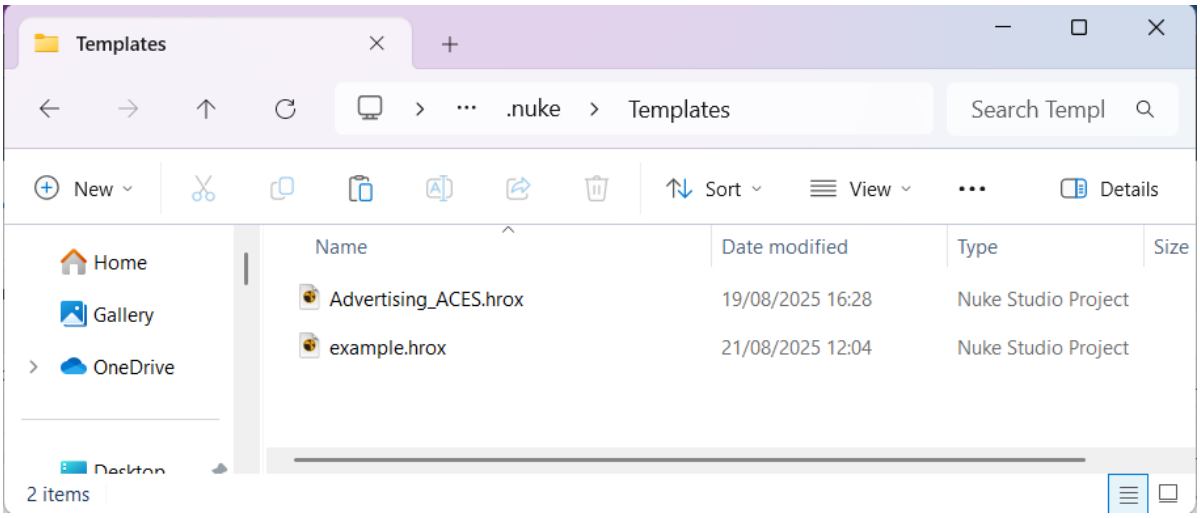


Creating and Using Project Templates

You can create templates for Nuke Studio by creating **.hrox** files in the **.nuke/Templates** folder. This helps if you have particular Timeline Environment Project Settings, such as Colorspaces, footage, and folder structures, that you wish to reuse across multiple projects.

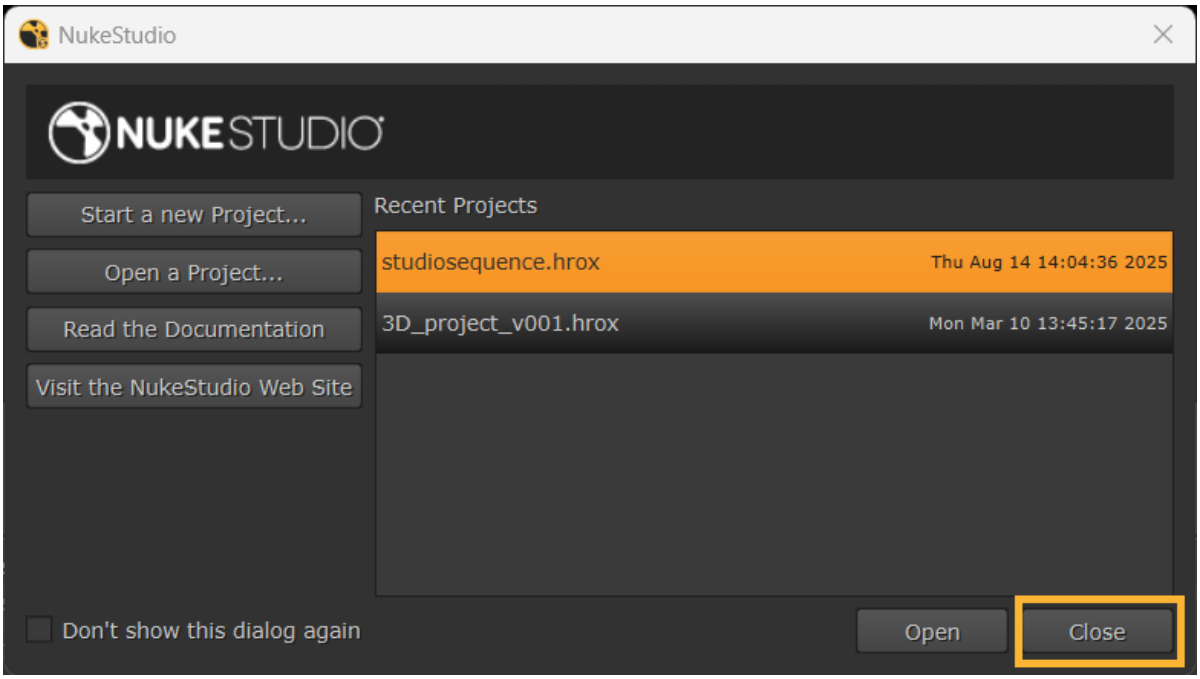
Creating Templates

1. Set up a Nuke Studio project with your desired Timeline Environment Project Settings, footage, and folder structures.
2. Create a folder named **Templates** under your **.nuke** directory, if you don't already have one.
3. Save your Nuke Studio project into the **.nuke/Templates** folder, with a name for the template. E.g. *Advertising_ACES.hrox*.

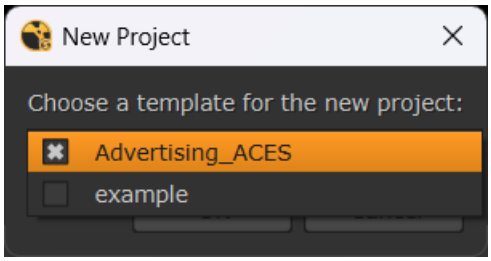


Using Templates

1. When you start Nuke Studio, close the opening dialogue window via the Close button.



2. A New Project dialogue box will open allowing you to select a template from the dropdown.



3. A new project will open with your saved setup to reuse.

Video: See this in practice - Creating Project Templates.

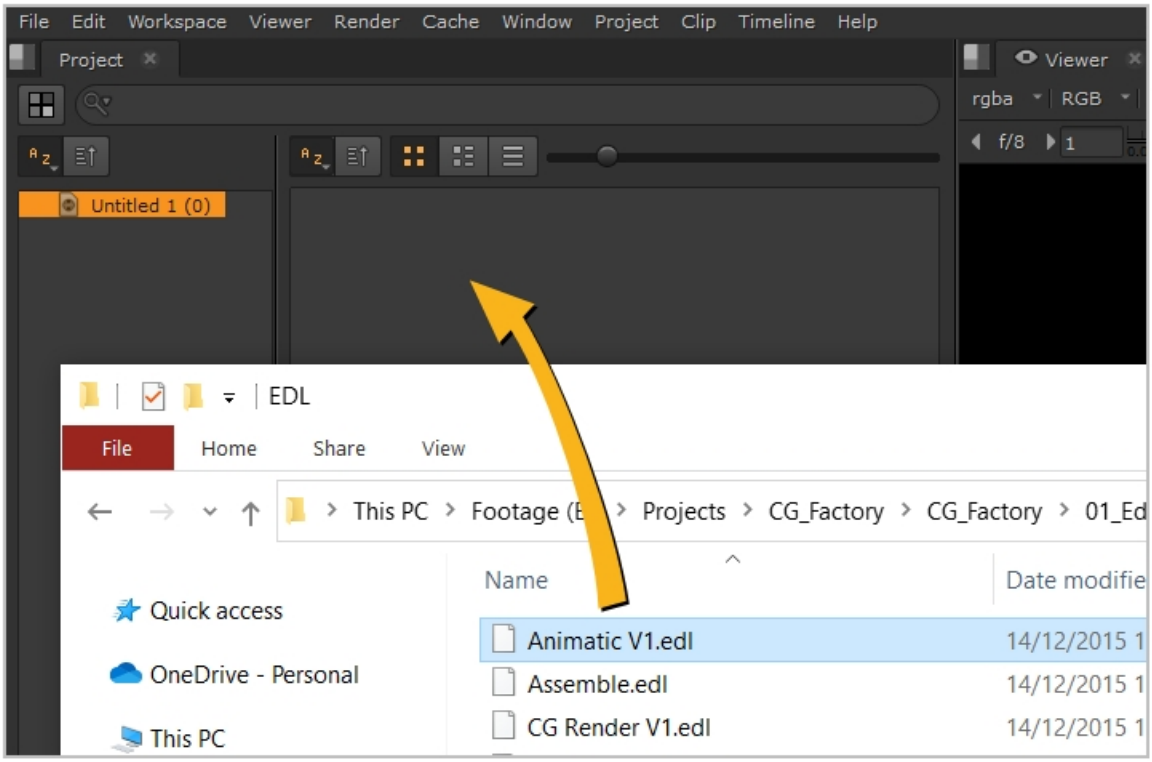
Importing Sequences

Nuke Studio allows you to import your EDL, OTIO, XML, or AAF sequences in one of two ways, depending on your preferences. Either:

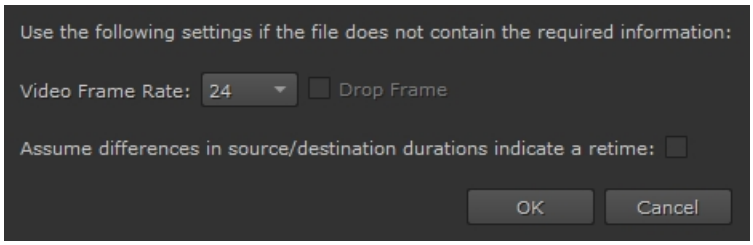
- Navigate to **File** > **Import EDL/XML/AAF** or **Import OTIO** and then use the browser to locate the EDL, OTIO, XML, or AAF, and then select the file and click **Open** to import the sequence,

  OR

- Drag-and-drop the EDL, OTIO, XML, or AAF files directly from a file browser into the interface.
- You can also import an OTIO as an additional track in an existing timeline by right-clicking on the track and selecting **New Track** > **New Track(s) from OTIO**.



If you're importing an EDL, bear in mind that there is no guaranteed frame rate information included in the file, so an **Import Options** dialog displays.



1. Select the correct frame rate and use the following check boxes, if required:
   - **Drop Frame** - when enabled, the EDL is assumed to contain drop file information. SeeTimeline Playback Tools for more information.
   - **Assume differences in source/destination durations indicate a retime** - when enabled, any disparity between the source clip (Src) and shot (Dst) duration is treated as a retime.
2. Click **OK** to import.

   OTIO, XML and AAF files imported into Nuke Studio support transform, crop, and retime edit decisions implemented in third-party applications, such as Adobe Premiere, Apple Final Cut Pro, and Avid Media

Composer. The information in the **.otio**, **.xml**, or **.aaf** is interpreted using soft effects, such as Transform and Crop. Non-linear retimes are represented by TimeWarp effects. Constant linear retimes are handled in the same way as in previous versions of Nuke Studio. See Notes on AAF Sequences for more information.

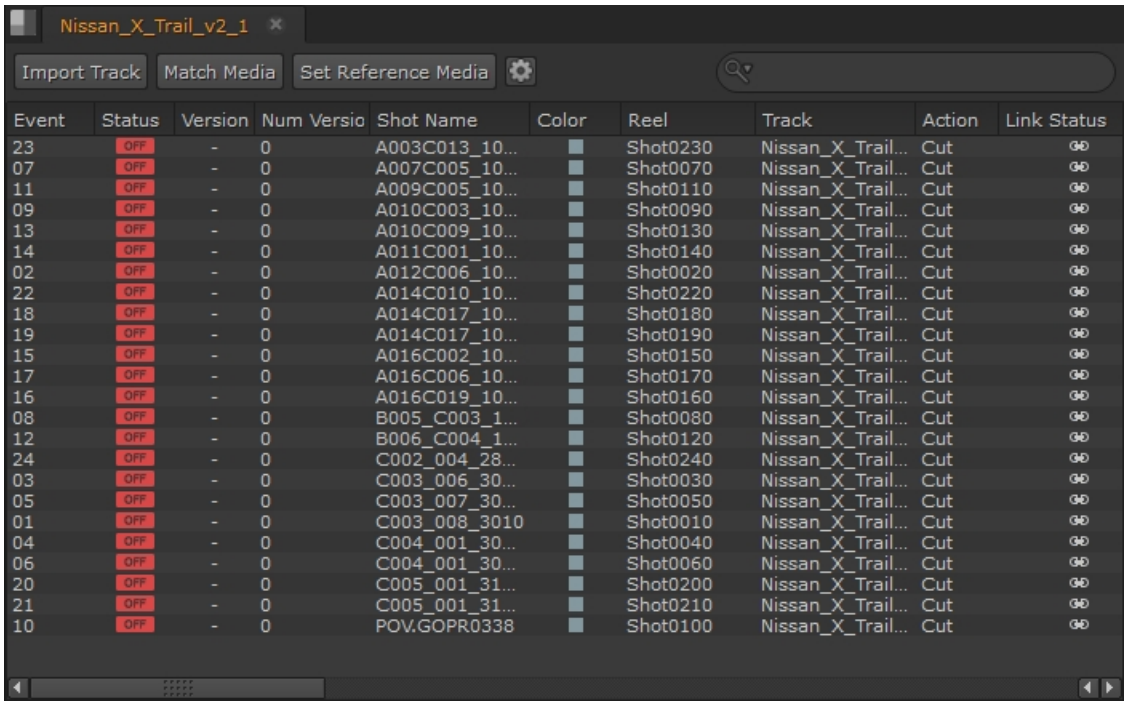
Note: Non-linear animation curves from **.otio** or **.xml** may not appear as expected when imported. As a result, you may need to adjust the handles on curves to match footage between keyframes in the Curve Editor or Dope Sheet.

See Animating Parameters for more information.

Additionally, Premiere Pro **.xml** exports only support constant, linear retimes. As a result, retimed shots on the Nuke Studio timeline may not match those on the Premier Pro timeline, because certain non-linear retime data is not written into the exported **.xml** file.

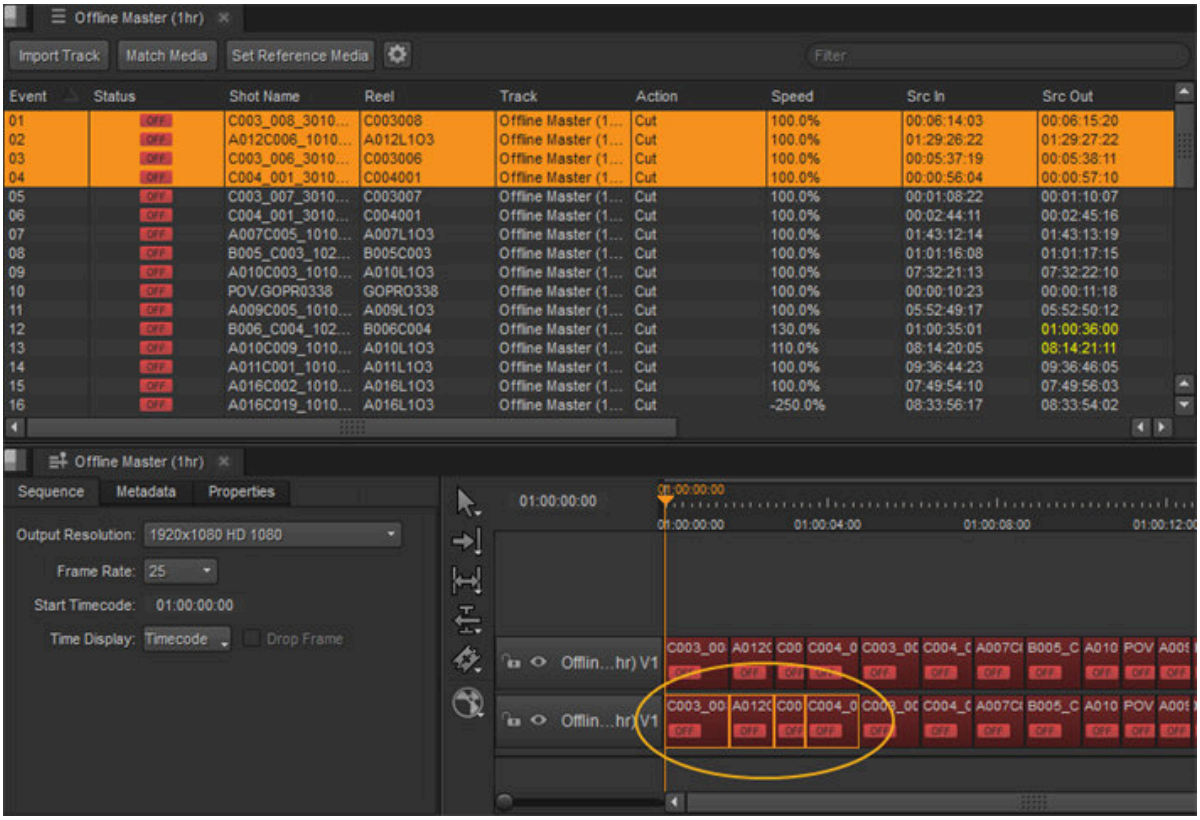
After importing the EDL, OTIO, AAF, or XML the **Conforming** workspace displays and the spreadsheet and timeline are populated with offline clips - media with an unknown location.

Note: The **Event** column represents the clip's position on the timeline, not its event number from the edit.



Event	Status	Version	Num Versio	Shot Name	Color	Reel	Track	Action	Link Status
23	OFF	-	0	A003C013_10...		Shot0230	Nissan_X_Trail...	Cut	
07	OFF	-	0	A007C005_10...		Shot0070	Nissan_X_Trail...	Cut	
11	OFF	-	0	A009C005_10...		Shot0110	Nissan_X_Trail...	Cut	
09	OFF	-	0	A010C003_10...		Shot0090	Nissan_X_Trail...	Cut	
13	OFF	-	0	A010C009_10...		Shot0130	Nissan_X_Trail...	Cut	
14	OFF	-	0	A011C001_10...		Shot0140	Nissan_X_Trail...	Cut	
02	OFF	-	0	A012C006_10...		Shot0020	Nissan_X_Trail...	Cut	
22	OFF	-	0	A014C010_10...		Shot0220	Nissan_X_Trail...	Cut	
18	OFF	-	0	A014C017_10...		Shot0180	Nissan_X_Trail...	Cut	
19	OFF	-	0	A014C017_10...		Shot0190	Nissan_X_Trail...	Cut	
15	OFF	-	0	A016C002_10...		Shot0150	Nissan_X_Trail...	Cut	
17	OFF	-	0	A016C006_10...		Shot0170	Nissan_X_Trail...	Cut	
16	OFF	-	0	A016C019_10...		Shot0160	Nissan_X_Trail...	Cut	
08	OFF	-	0	B005_C003_1...		Shot0080	Nissan_X_Trail...	Cut	
12	OFF	-	0	B006_C004_1...		Shot0120	Nissan_X_Trail...	Cut	
24	OFF	-	0	C002_004_28...		Shot0240	Nissan_X_Trail...	Cut	
03	OFF	-	0	C003_006_30...		Shot0030	Nissan_X_Trail...	Cut	
05	OFF	-	0	C003_007_30...		Shot0050	Nissan_X_Trail...	Cut	
01	OFF	-	0	C003_008_3010		Shot0010	Nissan_X_Trail...	Cut	
04	OFF	-	0	C004_001_30...		Shot0040	Nissan_X_Trail...	Cut	
06	OFF	-	0	C004_001_30...		Shot0060	Nissan_X_Trail...	Cut	
20	OFF	-	0	C005_001_31...		Shot0200	Nissan_X_Trail...	Cut	
21	OFF	-	0	C005_001_31...		Shot0210	Nissan_X_Trail...	Cut	
10	OFF	-	0	POV.GOPR0338		Shot0100	Nissan_X_Trail...	Cut	

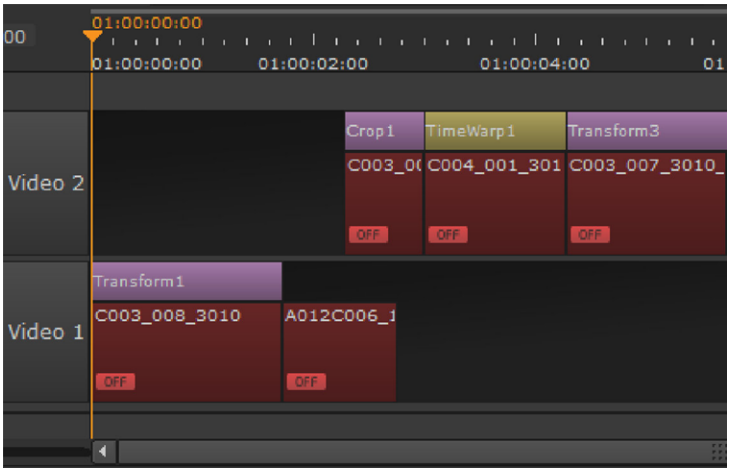
Notice that clicking entries in the spreadsheet highlights the corresponding shots on the timeline?



The spreadsheet, timeline, and Viewer are linked together when viewing sequences. If suitable screen real estate exists within the current workspace, double-clicking a sequence forces the associated panel to open automatically. If you want to close a single panel in a linked group, hold the **Alt** modifier while closing the linked panel, otherwise all panels in the group are closed.

Note: If you imported an XML sequence, you may find that Nuke Studio has automatically matched media for you.

Any transform, crop, or retime edit decisions from third-party software **.otio**, **.xml** or **.aaf** files are represented using soft effects. These effects are imported along with the shot to which they're associated.



Notes on AAF Sequences

Avid Media Composer supports retimes using curves that map **frame to frame** or **frame to speed**. Nuke Studio handles the import differently depending on the retime method.

- **frame to frame** - describes the retiming in relative terms, such as 'at frame 100 in the output clip, display frame 50 of the source clip'.
- **frame to speed** - describes the retiming in terms of overall output duration. For example, half speed doubles the duration of the clip.

Nuke Studio's TimeWarp effect only supports **frame to frame** mapping, which means that **frame to speed** retimes from **.aaf** files requires some curve-fitting to describe the required retime. As a result, the keyframes generated in Nuke Studio don't match those in Avid, but the resulting curve should match the original very closely.

Note: Nuke Studio currently only supports Fixed Keyframes from Avid Media Composer.

Tip: If you need to adjust the handles on curves, see Animating Parameters for more information.

Nuke Studio's TimeWarp effect supports the following Spline types when importing **.aaf** files:

- Shelf
- Linear
- Spline
- Bezier

Conforming Sequences

Once your EDL, OTIO, AAF, or XML sequences are imported, it's time to begin the conform process to match the offline shots in your spreadsheet with the source clips on disk. You can conform sequences by searching on disk or by pre-ingesting the required clips into Nuke Studio.



Note: Projects containing large amounts of movie files (for example **.r3d** and **.mov**) may exceed the number of available file handles per process, causing problems opening new files or projects and exporting.
You can increase the default limit of 1024 by entering the following command from the terminal, then running the application from the same session:

```
ulimit -Sn 2048
```

Conforming Using a Browser

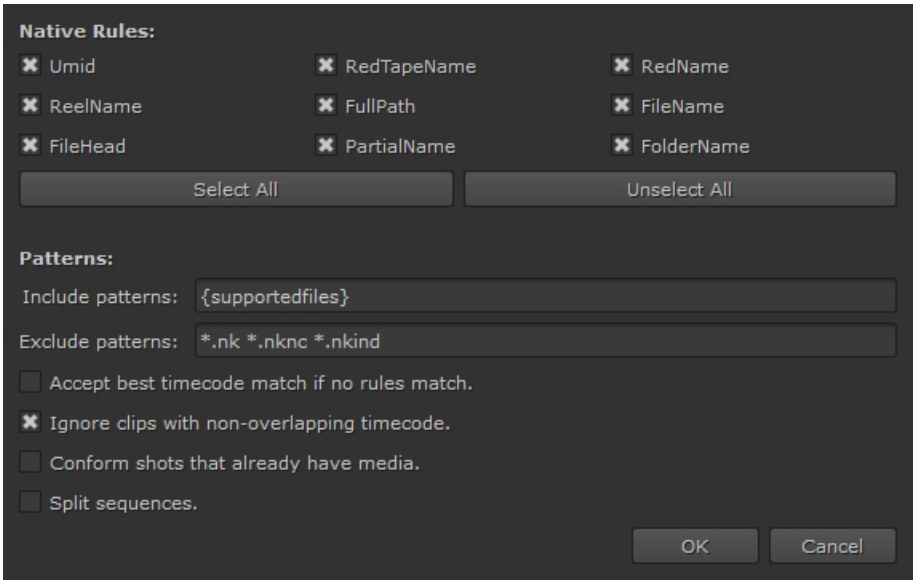
To conform a sequence using a browser:

1. After importing a sequence, click **Match Media** on the spreadsheet and use the browser to locate the source folder containing the correct media.



Note: **Match Media** can also be used on selected events in the Spreadsheet view.

2. Click **Open** to display the **Conform Options** dialog.



Nuke Studio uses a set of conform **Rules** and file name **Patterns** to match candidate media files on disk to the events, or shots, in a sequence:

- **Rules** - sets the offline media properties to match to the corresponding spreadsheet entry during conform.

  Rules that rely on information that doesn't exist in the event or candidate clip are ignored, and some rules compound others to identify a better match.

Rule	Description
Umid	Match a file's unique material ID (UMID) – that is written into the file's metadata on creation – to the candidate media's UMID. If either, or both, lack a UMID this rule is ignored.
RedTapeName	Match a RED-style camera reel name from the event to the candidate media name.
RedName	Look for a RED-style camera file name in the event that matches the candidate media name.
ReelName	Look for the event's reel name in the candidate's media name.
FullPath	Match the event's entire filepath to the candidate media's entire filepath. The **Event** is the first field in the Spreadsheet view, the order in which shots appear on the timeline. **Candidate** media is the media that **Nuke Studio** is testing the conform rules against.

Rule	Description
FileName	Match only the event's file name (no path) to the candidate media's file name.
FileHead	Match the event's file name head (no path, file extension, or padding) to the candidate media's file name.
PartialName	Look for the event's name in the candidate media's name and vice versa.
FolderName	Look for the event's name in the filepath of the candidate media.

All rules are enabled by default, but you may occasionally need to disable rules if they cause incorrect matches between a particular edit and set of source clips.



Tip: Use the **Select/Deselect All** buttons to quickly enable or disable rules.

- **Patterns** - sets the inclusion and exclusion parameters during the conform, separated by spaces. For example, ***mov *dpx** would only include or exclude **.mov** and **.dpx** files.

  You could also conform by name, such as **BR_Shot***, which would only include or exclude source clip names starting with that string.

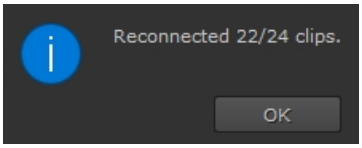
Tip: It's always a good idea to be as specific as possible with search locations during conforms, but if the need arises, conform **Rules** and **Patterns** can save time.

3. Enable **Accept best timecode match...** to use the nearest source timecode match to conform the event, if no rules are matched.
4. When **Ignore clips with non-overlapping timecodes** is enabled, any potentially matching source clip whose timecode doesn't overlap the shot in question at all is ignored.

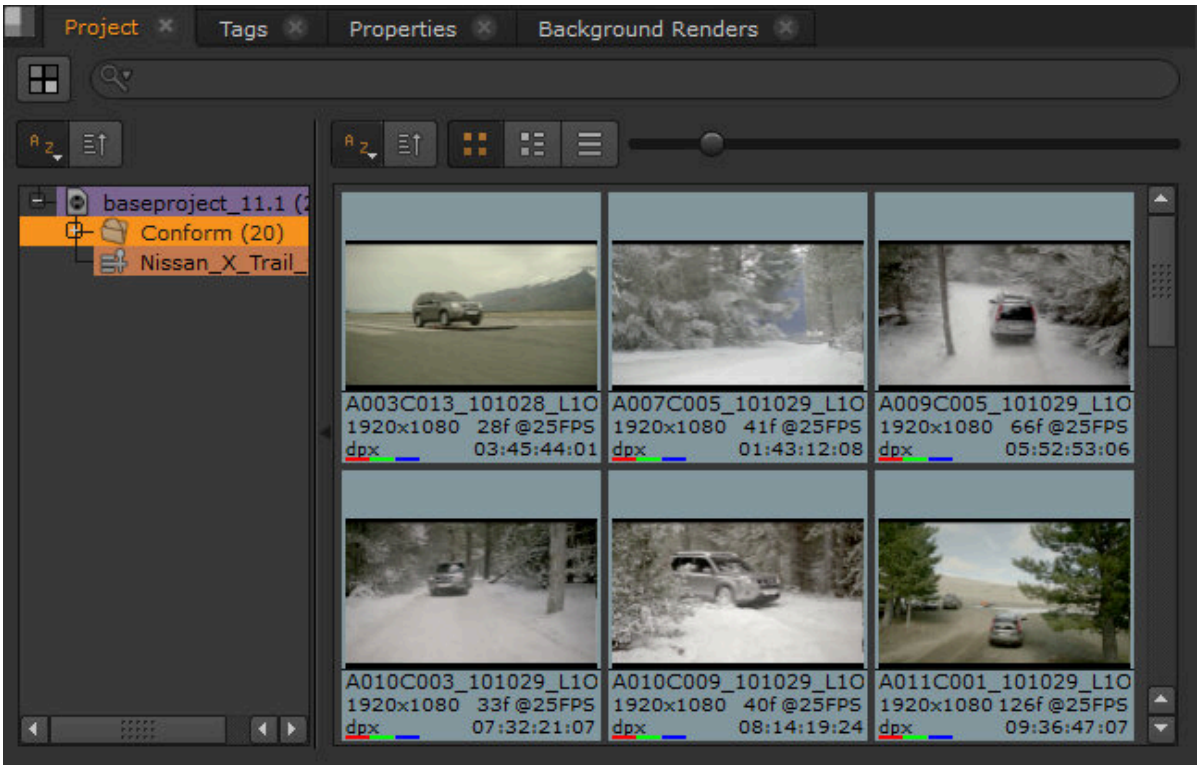
   Disabling **Ignore clips with non-overlapping timecodes** causes Nuke Studio to fall back to the other selected conform rules, even if the timecodes don't overlap.
5. Check **Conform shots that already have media** if you want to update all timeline shots. By default, the application doesn't try to conform events that are not offline.
6. When **Split sequences** is enabled, any non-contiguous file sequences found by the conform are split into separate clips, in the same way as when the **split seq** option is enabled in the file browser.
7. Click **OK** to begin the conform process.

   Nuke Studio attempts to conform the edits with the selected media.

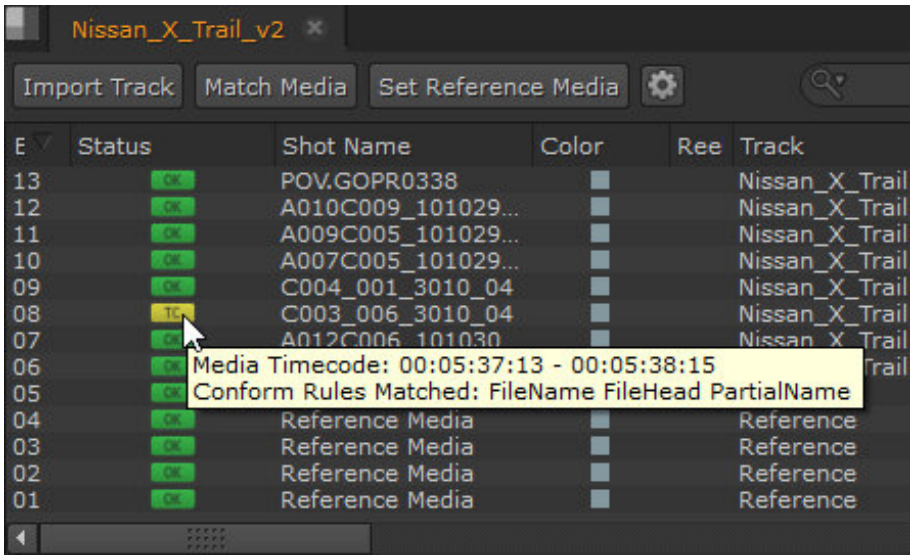
   A dialog box informs you of the success rate once the conform is complete.



Successfully matched media is placed in a new **Conform** bin in the project.



Note: You can display the conform Rules matched for each spreadsheet object by hovering the cursor over the required entry.

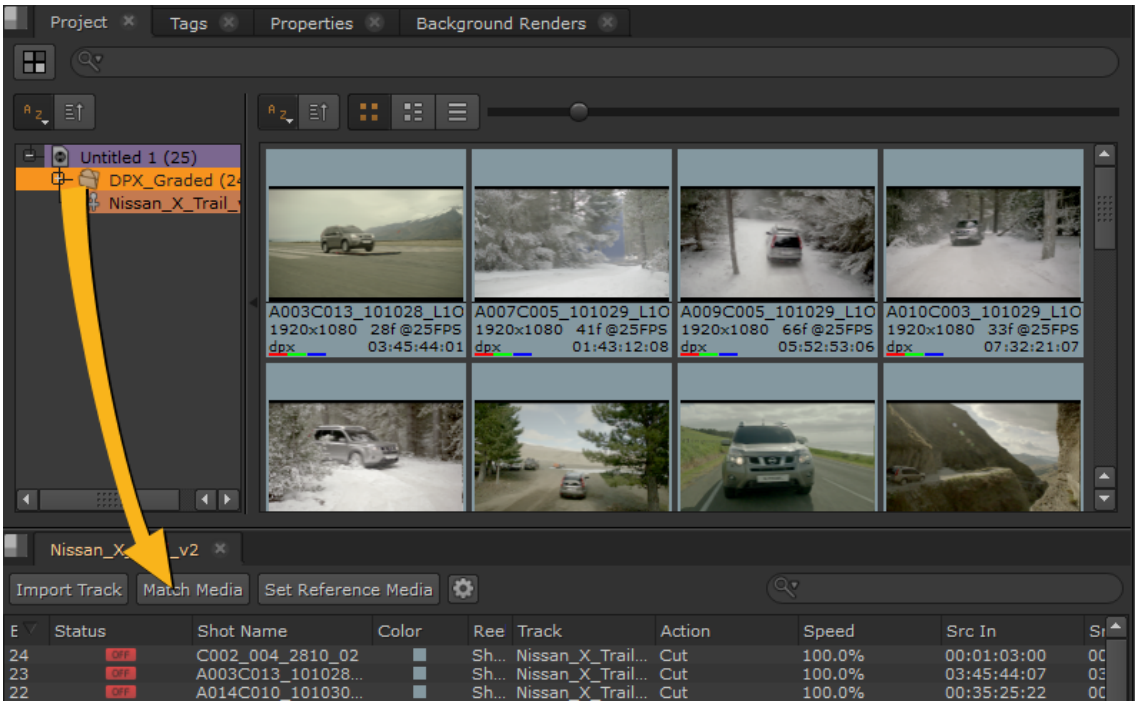


Conforming with Pre-ingested Media

To conform with pre-ingested media:

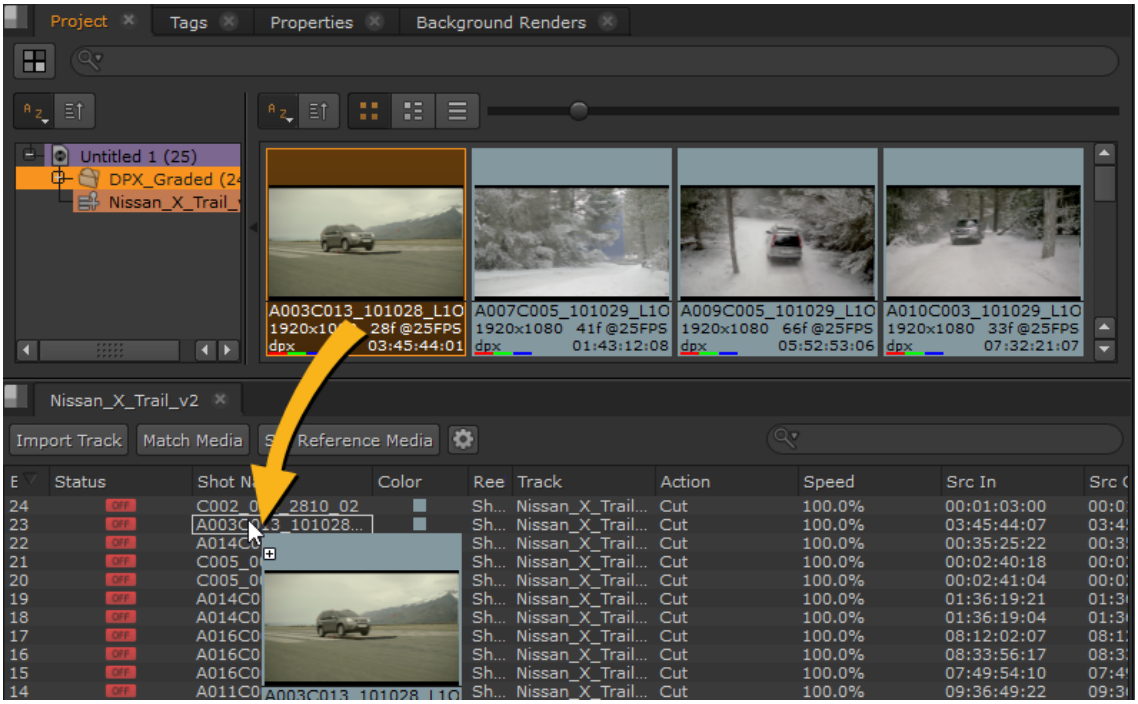
1. If your source media has been ingested, you can drag-and-drop media from the bin view onto the **Match Media** button.

   See Ingesting Media for information on getting media into Nuke Studio.



2. Follow the **Conform Options** instructions described previously to complete the conform process.

   If you want to conform a single entry in the spreadsheet, drag-and-drop the media from the bin view onto the required entry in the spreadsheet.



Conforming individual, pre-ingested media doesn't require **Conform Options** because Nuke Studio already knows the exact location of the media and trusts your decision to replace a shot.

About the Media Spreadsheet

All events in a sequence are displayed in an easy to read format in the spreadsheet including status, the track it resides on, length, and the source file location.

After conforming, you can use the spreadsheet to locate source clips or replace shots in the timeline, as well as massage timecodes if they are invalid.

The media spreadsheet displays each entry's current media state:

-  - the media was successfully conformed and its timecode is correct.
-  - the media was successfully conformed, but the timecode is currently incorrect.
-  - the media could not be conformed.



Note: Any source or destination field highlighted in yellow indicates that the entry has been rounded down for display purposes.

See Managing Timelines for more information on importing tracks and reference media.

Sorting and Custom Columns

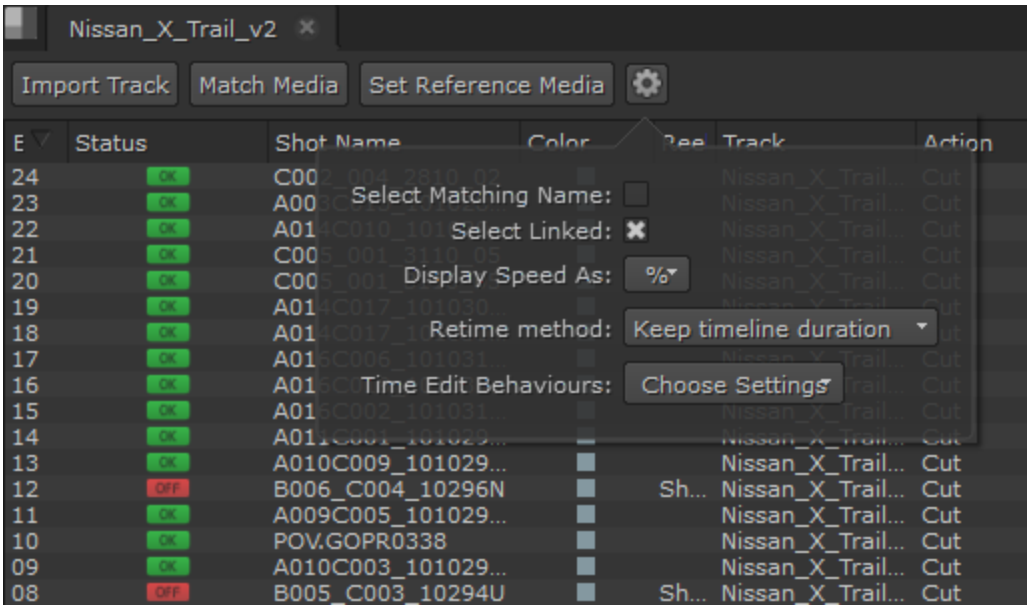
The spreadsheet can be organized in much the same way as accounting spreadsheets:

- Right-click the column headers to display the list of default columns available. Enable or disable each column using the checkboxes.
- Click the required column to sort the spreadsheet in ascending or descending order, as indicated by the arrow in the column header.
- Drag-and-drop column headers to reorder the spreadsheet as required.
- Add custom columns, such as Tags, using the Python API. See **Help** > **Documentation** for more information on the Python API.

Tip: Nuke Studio's **Project** panel search functionality extends to the spreadsheet, allowing you to enter strings and apply searches on all or partial matches with the option to include metadata searches. Nuke Studio searches for items that match any of the input string and displays only those items by default. See Sorting and Searching Media for more information.

Spreadsheet Controls

There are also a number of controls, accessed by clicking the cog icon, that determine the spreadsheet's appearance and behavior.

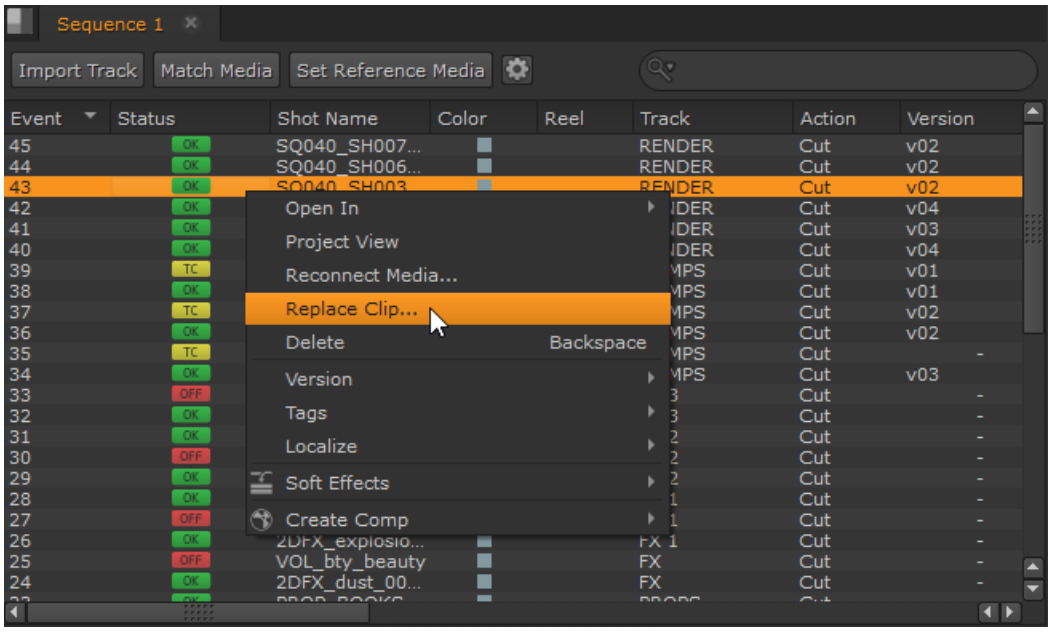


- **Select Matching Name** - when enabled, selecting an item in the spreadsheet highlights all items with the same name.
- **Select Linked** - when enabled, selecting an item in the spreadsheet highlights other items linked to it, such as audio tracks ingested with video tracks.
- **Display Speed As** - sets the unit used in the Speed column of the spreadsheet. Select either **fps** (frames per second) or **%** (the percentage of the media frame rate).
- **Retime method** - sets the type of **Speed** retime applied on the timeline.
- **Time Edit Behaviors** - sets how source and destination **In**, **Out**, and **Duration** are calculated.

  SeeRetiming Shots for more information on retime methods and Timeline Editing Tools for source/destination calculations.

You can locate, display, reconnect, or rename shots directly from the spreadsheet.

- Hold **Alt** and click an entry to move the playhead to the shot's **In** point on the timeline.
- Hold **Alt** and double-click an entry to move the playhead to the shot's **In** point on the timeline and zoom to fit the timeline view.
- Right-click a spreadsheet entry and select the required option:



- **Open In** - the associated source clip opens in the selected location, such as a Viewer.
- **Project View** - the associated source clip is highlighted in the bin view.
- **Reconnect Media** - attempt to reconnect the media from a specified location on disk, such as when the source was originally on a drive that is no longer connected.
- **Replace Clip** - replaces the selected entry with a specified source clip. Nuke Studio assumes that any source clip you choose is acceptable, regardless of timecode.
- **Delete** - deletes the selected entries from the spreadsheet and timeline.
- **Version** - access the powerful versioning system to control which version of the selected shot is visible in the sequence. See Using Versions for more information.
- **Tags** - add tags to your selection directly from the spreadsheet view. See Using Quick Tags for more information.
- **Localize** - control the localization of clips, tracks, and sequences from the spreadsheet. See Localizing Media for more information.
- **Create Comp** - create comps of shots directly from the spreadsheet. See Create Comp for more information.
- **Soft Effects** - add soft effects to shots directly from the spreadsheet, see steps below.

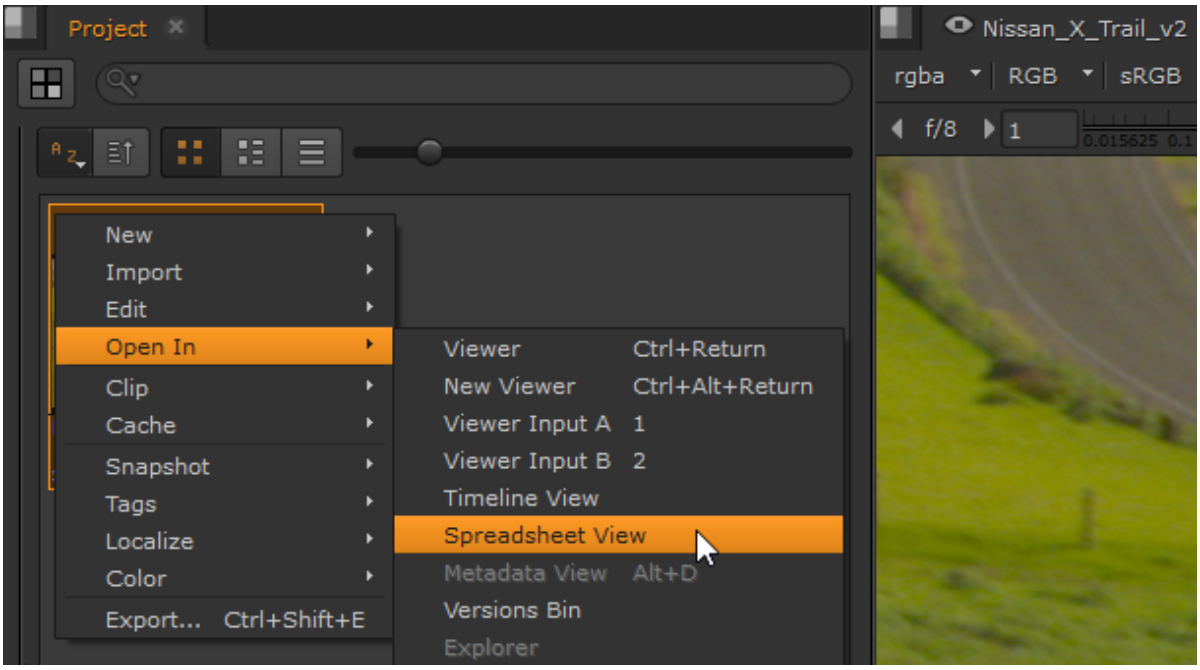
Using the Spreadsheet View

You can add soft effects using the spreadsheet view in any workspace by doing the following:

1. To open the spreadsheet view in any workspace, select **Window** > **New Spreadsheet View**.
2. Right-click a sequence in the bin view.

   This opens a context menu.

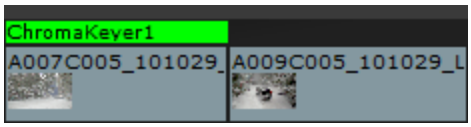
3. If the spreadsheet view is not already populated in the context menu, select **Open In** > **Spreadsheet View**.



This loads the sequence in the spreadsheet view that you previously opened.

4. Right-click an event from the list in the spreadsheet view and select **Soft Effects** to open a list of all available soft effects. See Adding Effects on the Timeline for more information.
5. Select the required soft effect from the **Soft Effects** list.

   The sequence-level soft effect is then displayed above the shot and is color coordinated. For example, if the effect appears in green on the timeline, the corresponding effect properties are highlighted in the same color in the **Properties** pane.



When you insert a soft effect, its **Properties** open automatically. If you have it open, the effect properties panel displays in the **Properties** pane. If the **Properties** pane is not open, the effect's properties panel appears as a floating dialog.

Adjusting Timecodes

You can easily adjust single or multiple event timecodes:

1. Select the invalid entry or entries in the spreadsheet.

2. Double-click in the **Src In** column.
3. Adjust the timecode as required. You can enter **absolute** or **relative** timecode values:
   - **Absolute** - absolute timecodes contain eight digits and specify the new timecode for the event, regardless of the current timecode.

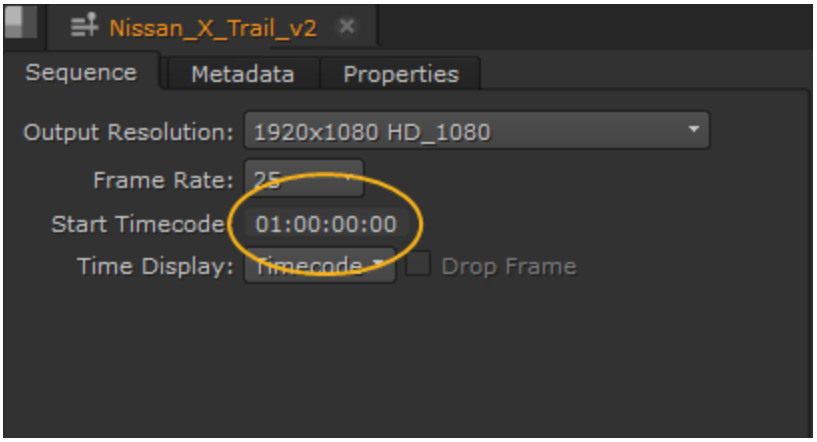
Example	Result
01:05:43:21	Sets the timecode at 1 hour, 05 minutes, 43 seconds, and 21 frames
01054321	

   - **Relative** - uses **+** and **-** values to alter the timecode relative to its current value. You can also use **h**, **m**, and **s** to denote hours, minutes, and seconds.

Current Position	Example	Result
01:05:43:21	+1h	02:05:43:21
	-110	01:05:42:11
	+10000	01:06:43:21
	-6m	00:59:43:21

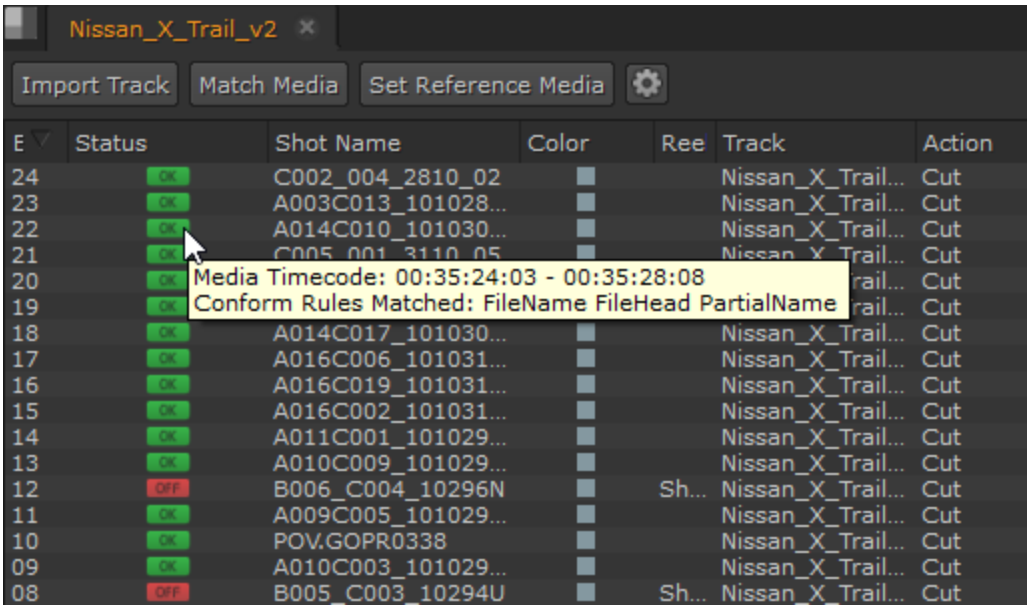
The media changes state to .

Note: Timelines start at 01:00:00:00 by default, but you can change this to any value using the **Sequence** panel.



If you're not sure what the timecode should be, you can:

• Hover the mouse over the target entry in the spreadsheet to view a timecode tooltip.

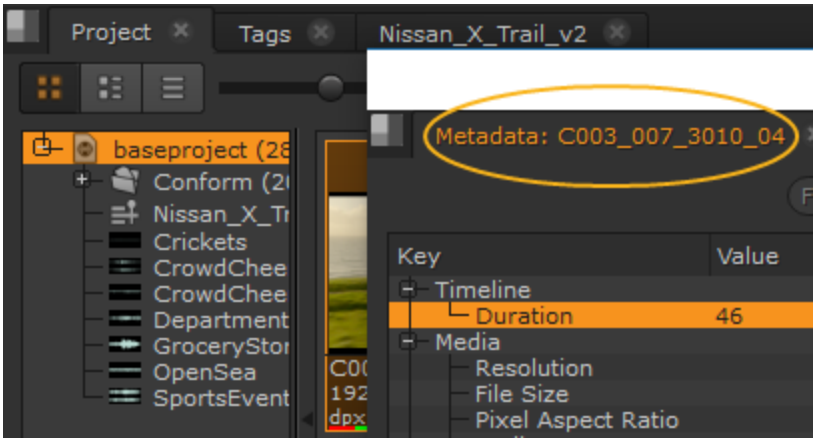


OR

• Examine the source clip's metadata and calculate the correct **Src In**:

1. Right-click the required entry and select **Project View**.
2. Right-click the clip in the bin and select **Open In** > **Metadata View**, or press **Alt**+**D**.

   The selected clip metadata is displayed in a floating pane.



Renaming Shots on the Timeline

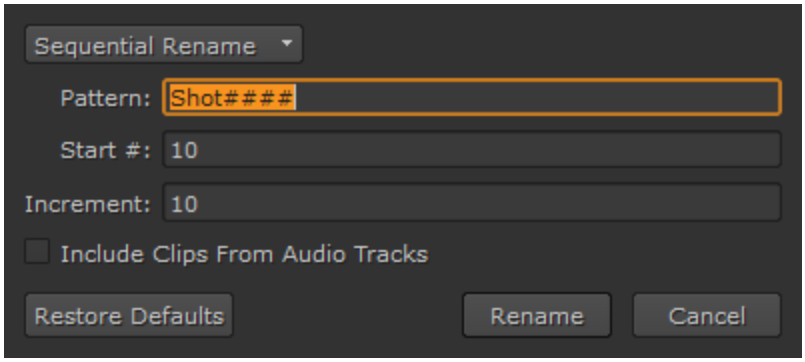
Once you've conformed your edit, you may want to rename shots on the timeline sequentially for clarity.

To rename shots:

1. Select the shots to rename using the timeline or spreadsheet view.
2. Right-click on the timeline and select **Editorial** > **Rename Shots**.

Tip: You can also navigate to **Timeline** > **Rename Shots** or use the **Alt**+**Shift**+**/** keyboard shortcut.

The **Rename Shots** dialog displays.



3. Select the rename type from the dropdown:
   - **Simple Rename** - all shots are replaced by the **New Name** specified.
   - **Find and Replace** - a simple find and replace shot name. All selected shots containing the specified **Find** pattern are substituted with the **Replace** pattern.
   - **Sequential Rename** - rename shots sequentially using the **Pattern**, **Start #**, and **Increment** fields.
   - **Match Sequence** - allows you to select a sequence to copy shot names from, providing that they use the same shots. For example, renaming shots on a 30 second timeline to mirror the shot names from a 60 second timeline.



Note: You can only use sequences that reside in the same project and shots that have overlapping frame ranges.

   - **Clip Name** - all shot names are replaced by the name of the source clip they reference. This option can be used to revert previous rename operations.
   - **Change Case** - the case of all shot names is changed, as specified by the **Case** dropdown. For example, selecting **Title Case** capitalizes the first character of each word.
4. Rename operations also accept token substitutions. The following tokens are recognized:

Token	Resolves to
{clip}	The name of the source clip referenced by the target shot.
{event}	The EDL event number associated with the target shot.
{filename}	The file name of the shot's source.
{fps}	The frame rate of the sequence containing the rename target.
{sequence}	The name of the sequence containing the rename target.
{shot}	The name of the shot.
{track}	The name of the track containing the rename target.

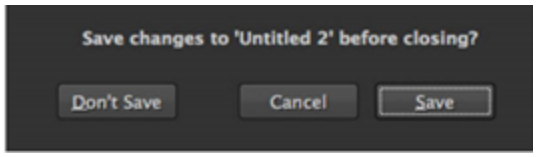
5. Enable **Include Clips From Audio Tracks** to rename audio shots as well as video shots.
6. Click **Rename**.
   The selected shots are renamed as specified.

Saving and Loading Projects

You can save your work in a project using the **.hrox** file extension. Projects can contain **.nk** scripts in the form of shots added by using **Create Comp**. See Create Comp for more information.

Warning: Any **.hrox** files generated using Nuke Studio and Hiero 16.1 or 17.0 are not backward compatible and are not able to be opened in previous versions of the software (16.0 and before).

If you quit the application without saving, you'll be prompted to save or discard your changes:



Click the required button or press **D** for **Don't Save** or **S** to **Save**.



Note: If you have a **.nk** script open in the same session, a second prompt is displayed so you can save your script as well.

To save a project:

1. Navigate to **File** > **Save Project** or **Save Project As...**

   OR

   Use the **Ctrl**/**Cmd**+**S** or **Shift**+**Ctrl**/**Cmd**+**S** keyboard shortcuts respectively.

   The **Save Project** dialog box displays.
2. Browse to the save location and enter a name for the project.
3. Click **Save**.

   Your project is saved to the location specified and appends the **.hrox** file extension automatically.

To load a project:

1. Navigate to **File** > **Open Project**.

Tip: If you need to open a project that you worked on recently, you can select **Open Recent Project** to quickly locate your work.

   OR

   Use the **Ctrl**/**Cmd**+**O** keyboard shortcut.

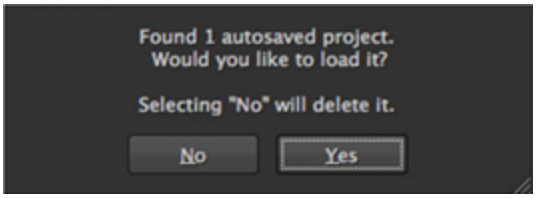
   The **Open Project** dialog box displays.
2. Browse to the file location and click **Open**.

   Your project opens and populates the necessary panel automatically.

Autosaved Projects

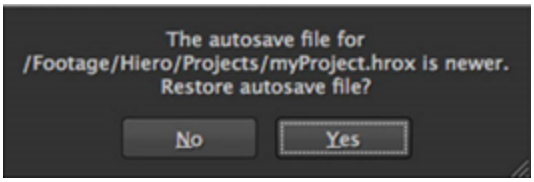
The autosave function creates a temporary project save file at 5 minute intervals, but you can adjust **force project autosave after** in the **Preferences** > **General** dialog. See Appendix A: Preferences for more information.

At startup, the application scans for autosaved projects and displays a prompt if autosaves exist.



Click **Yes** to load the autosave or **No** to ignore and delete it.

Opening a project also uses the autosave functionality. If the autosave is more recent than the saved project file, a prompt displays:



Click **Yes** to load the autosave file or **No** to load the original project file.

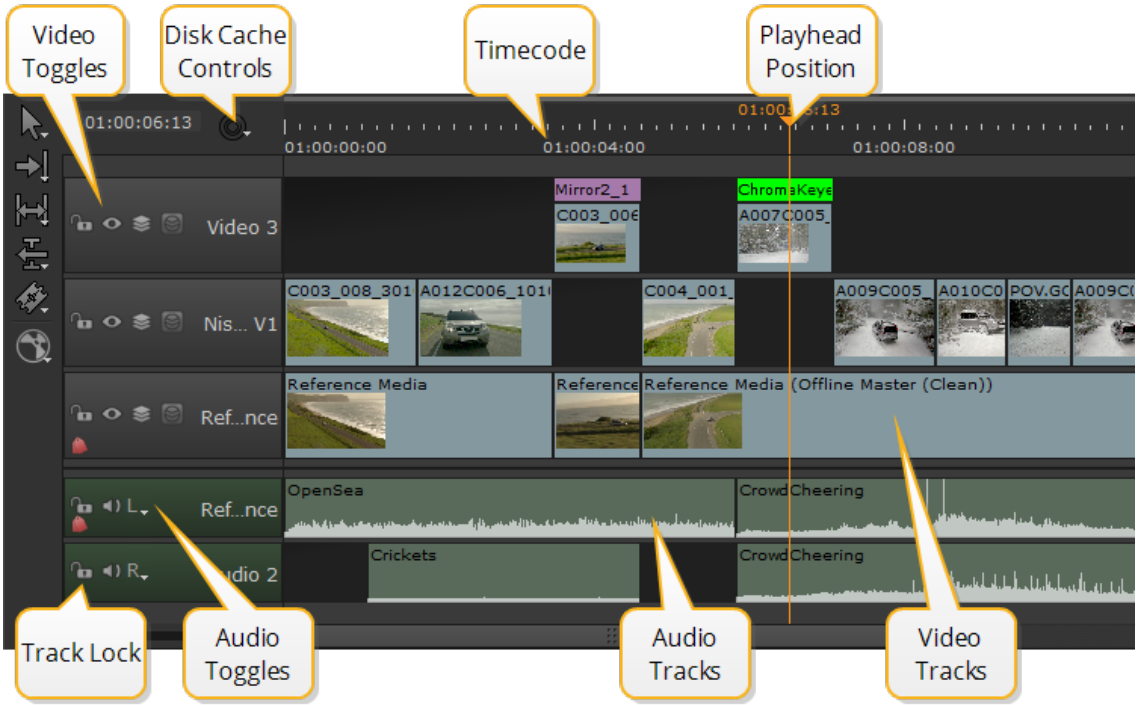
Note: Clicking **No** does not delete the autosaved project in this case.

Managing Timelines

Timelines contain video and audio shots that reference the source clips in your project. Once the conform process is complete, the timeline displays your clips in context and enables you make finer edits. Timelines can contain any number of video sequences and audio tracks with each track containing shots that reference the source clips in your project - making changes to shots in the timeline does not affect the original source clip.

Nuke Studio also features real-time soft effects on the timeline and the ability to add shots containing **.nk** scripts. See Adding Effects on the Timeline and Create Comp for more information.

Note: Conformed EDLs only support one video sequence. If you've created multiple EDLs from the same edit, you can add each one into the timeline using the right-click **New Track** > **New Track(s) from EDL/XML** option or the **Import Track** button in the spreadsheet tab. See Adding Tracks to the Timeline.



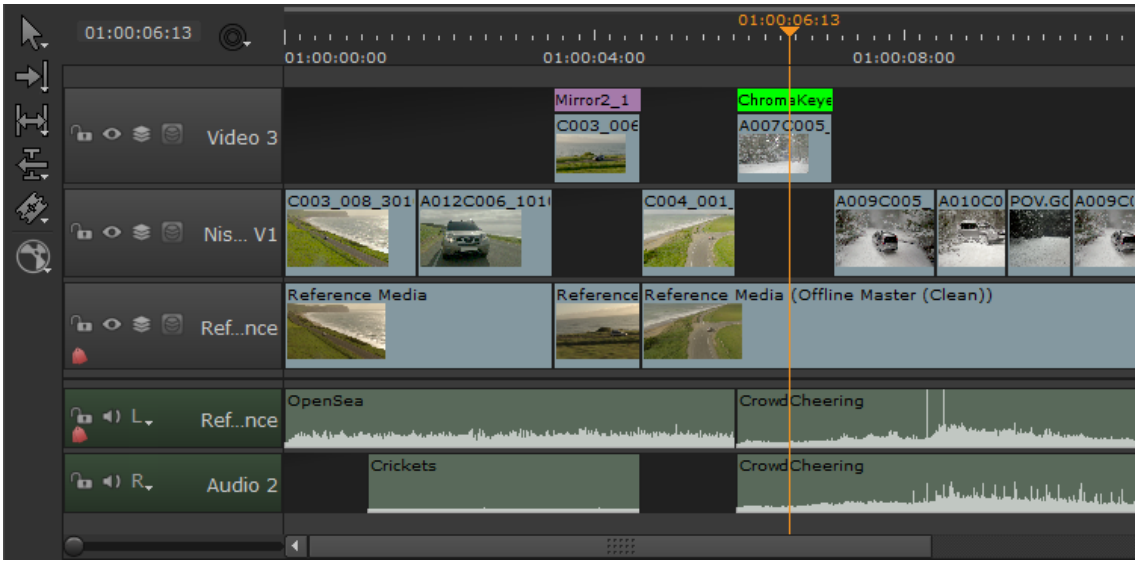
- **Video Toggles** - quickly turn off and on video tracks during playback. Hold **Alt** and click to solo the selected track. You can also enable and disable track blending and masking. See Blending Tracks on the Timeline for more information.
- **Disk Caching Controls** - click and hold to display the disk caching options for the current timeline. See Caching Frames in the Disk Cache for more information.
- **Timecode** - displays the timecode or frame number depending on the **Time Display** mode selected. You can adjust the scale using the Scale Slider or by using the mouse wheel.
- **Playhead Position** - displays the playhead location synchronized with the contents of the Viewer.
- **Video Tracks** - contain all video sequences for the current timeline.
- **Audio Tracks** - contain all the audio clips for the current timeline.
- **Audio Toggles** - quickly mute audio or set the track output during playback to left, right, or mono.
- **Track Lock** - secure the selected track to disable all editing tools.

Tip: Selecting tracks while holding **Ctrl**/**Cmd** allows tools to affect multiple tracks at once, such as locking, disabling, and resizing tracks.

Video tracks in multi-track timelines are read from the highest number track downward, for example Video 3, Video 2, Video 1. As a result, if video is present on track 3, video on track 2 in the same time slice is obscured.

Tip: Nuke Studio automatically selects clips as the playhead passes over them on the timeline, so you can make changes to shots without needing to manually select them first. Automatic shot selection is also reflected in the spreadsheet.

In this example, although the playhead crosses clips on two video tracks, only the clip in Video 3 is displayed in the Viewer.

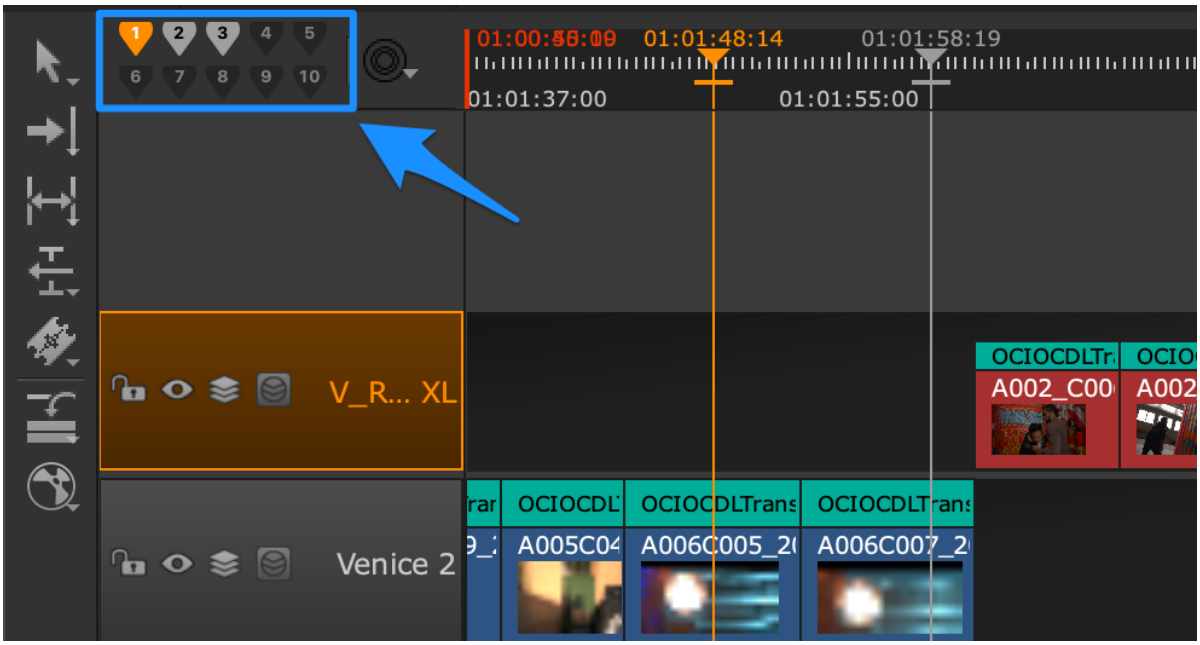


Audio tracks, on the other hand, play back simultaneously - all the audio tracks crossed by the playhead in the example play back together, creating a complete audio backing for the video.

Tip: Enabling **Preferences** > **Panels** > **Timeline** > **show frame end marker** draws an extra line on the timeline to the right of the playhead, indicating the end of the current frame.

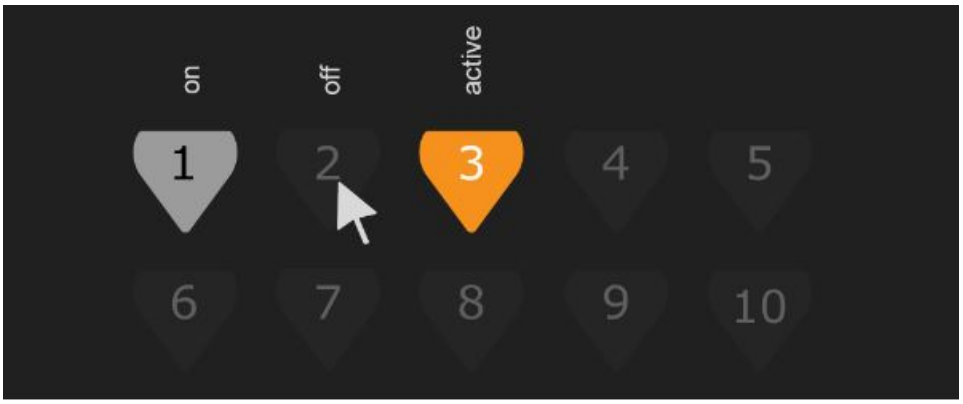
Multi-playhead

Nuke Studio allows you to work with up to 10 playheads on the timeline, each with its own dedicated In and Out points. Once a playhead has been added to the timeline, you can switch between them with the hotkeys 1 - 0, or via the panel to the top left of the timeline. Each playhead can also provide input to the A/B Compare buffers independently.

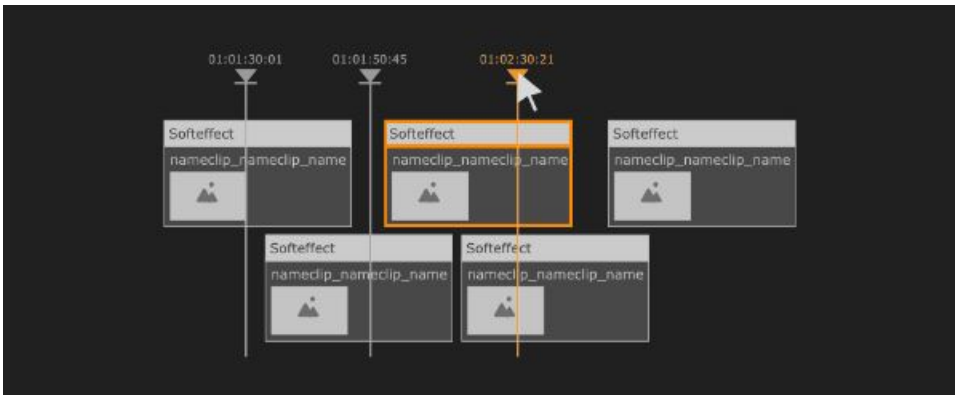


The playheads have 3 states:

- **Active** - Single-click on the playhead and it changes color to orange. The clip under the active playhead is rendered in the viewer.
- **On** - The playhead is visible on the timeline and is shown in gray color.
- **Off** - Double-click to turn a playhead off and remove it from the timeline.



Note: When you activate a new playhead for the first time, it will appear on the timeline at the same point as the current playhead. Drag the new playhead to focus on the next area of the timeline that you wish to work on.



Adding Tracks to the Timeline

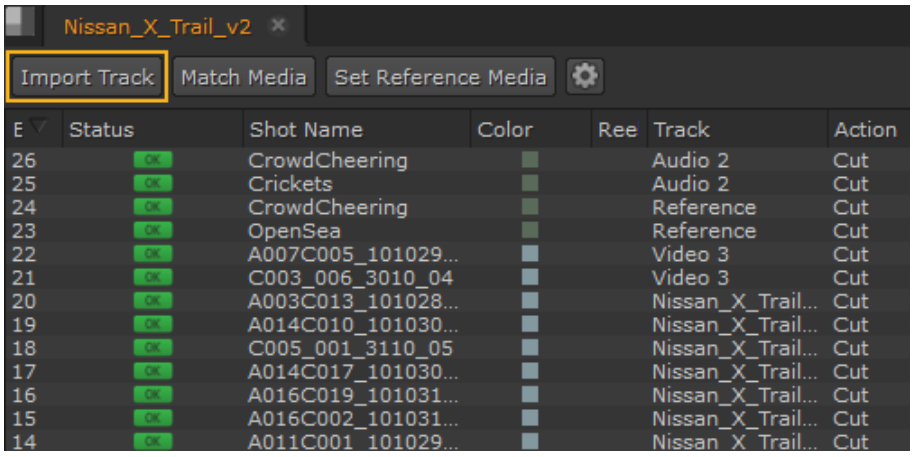
You can add empty tracks to existing timelines or import other EDL, OTIO, AAF, or XML edits - effectively another sequence within the timeline.

To import EDL, OTIO, AAF, or XML edits:

1. Select the required sequence in the project bin, right-click, and select **Import** > **New Track(s) from EDL/XML** or **New Track(s) from OTIO**,

   OR

   Click **Import Track** in the spreadsheet tab.



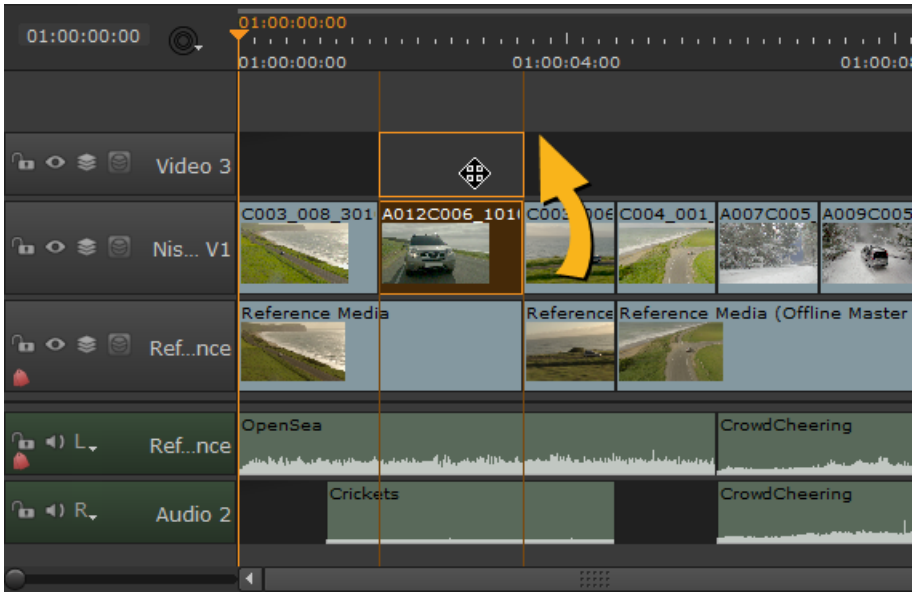
2. Use the browser to locate the EDL, OTIO, AAF, or XML files, select the file(s) and click **Open** to import the sequence.

Note: If you're importing EDLs, bear in mind that there is no guaranteed frame rate information included in the file. Select the correct frame rate then click **OK** in the dialog supplied.

3. Conform the new track as described in Conforming Sequences.

To add new tracks:

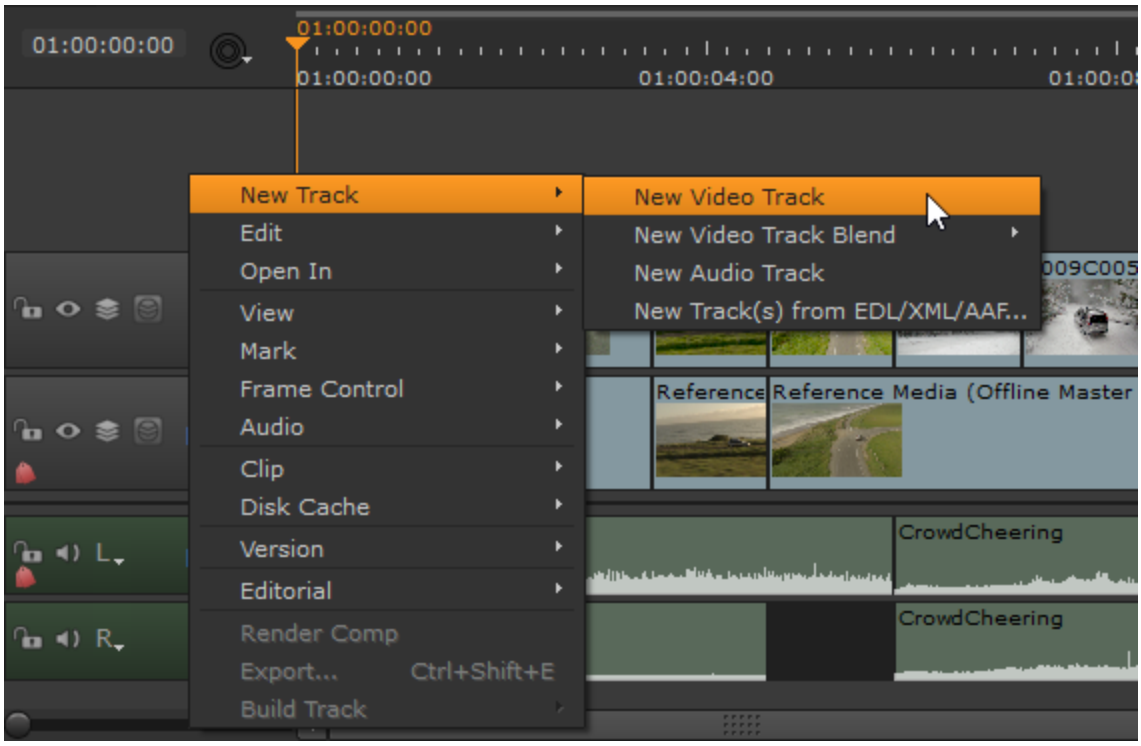
• Hold **Shift** and drag a shot to insert a new track above or between existing tracks,



Tip: This is the default behavior. If you disable **Preferences** > **Panels** > **Timeline** > **Hotkey required to insert new track**, the new track is created automatically.

OR

• Right-click in the timeline and select **New Track** > **New Video Track** or **New Audio Track**.



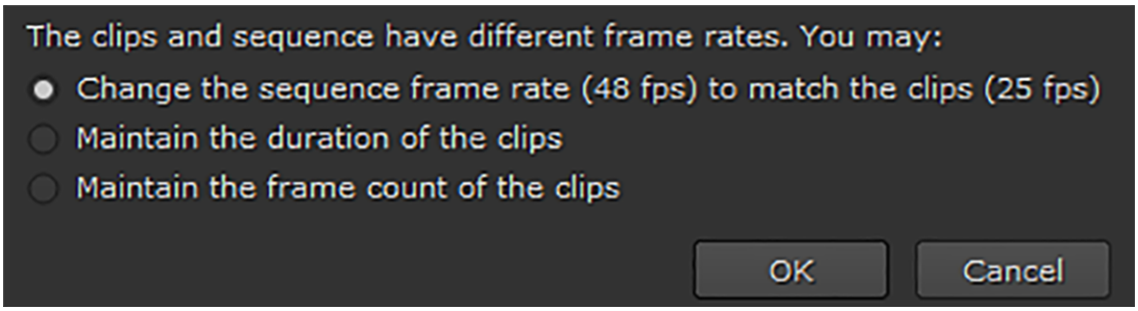
Note: You can also collapse and expand existing tracks using the right-click **Editorial** menu, and resize the track header to accommodate longer track names.

Adding Clips to the Timeline

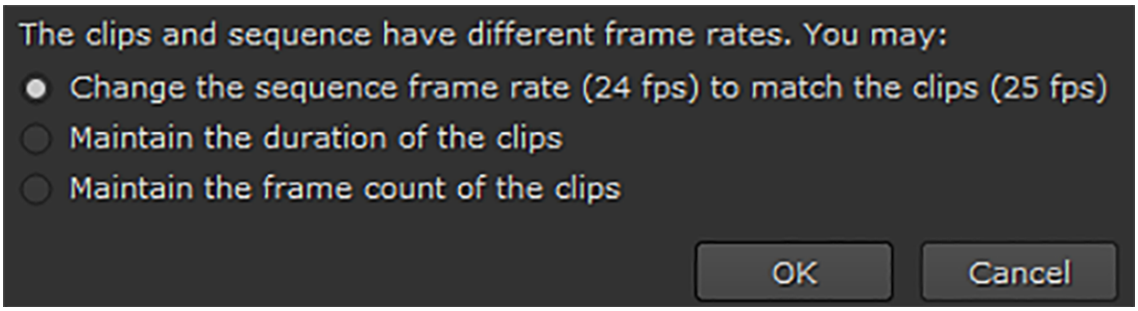
The timeline allows you to add clips by simple drag-and-drop from either the Viewer or bins. Using the Viewer restricts you to a single clip, the current clip, but you can drag as many clips as you like from bins.

Tip: You can create a new sequence by dragging a clip to an empty timeline pane.

New timelines pick up their frame rate from the **Project** > **Edit Settings** > **Sequence** sub-menu by default. Dropping a clip with a different frame rate on a new timeline displays a warning:



However, if the timeline is already populated and the clip you're adding doesn't have the same frame rate as the timeline, you're prompted to choose whether the clip's duration or frame count is retained.



Take care not to overwrite existing shots - the most recent clip overlays any existing shot. To avoid this, do one of the following:

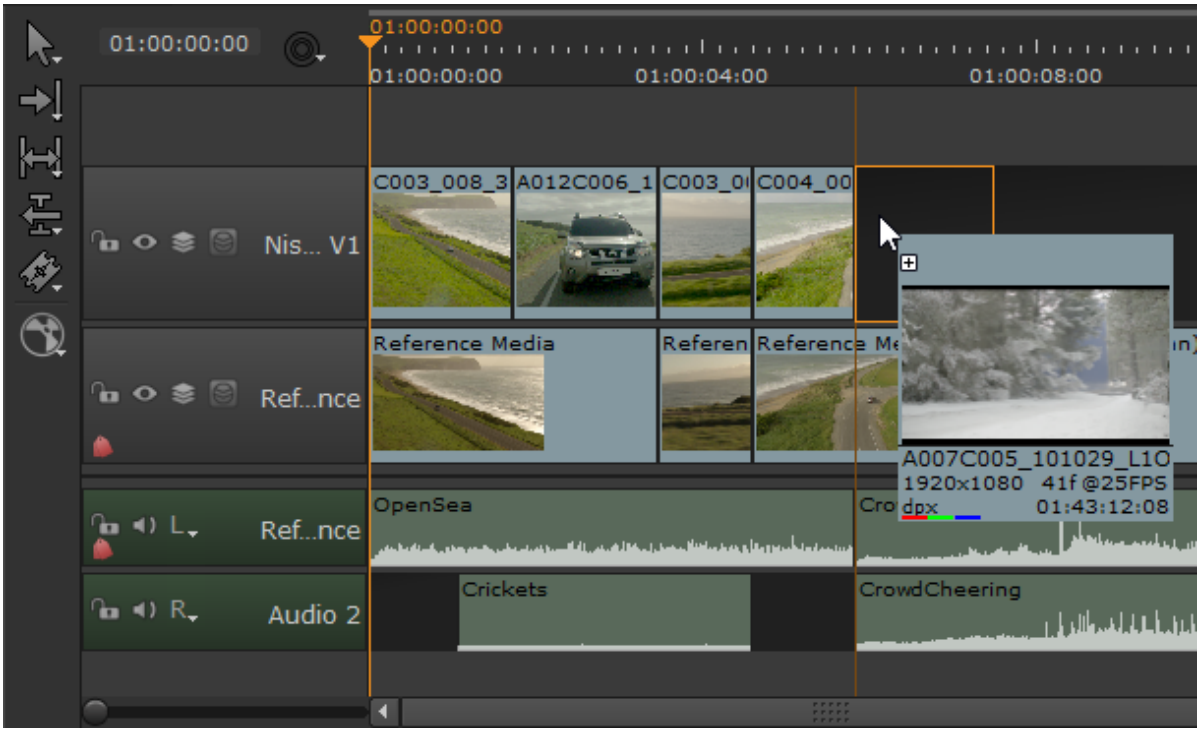
- Move the playhead to the target area of the timeline in the record Viewer, load the required clip in a source Viewer, and then use **Insert** (N) or **Overwrite** (M) to place the clip into the timeline at the playhead position on the lowest, unlocked track available.



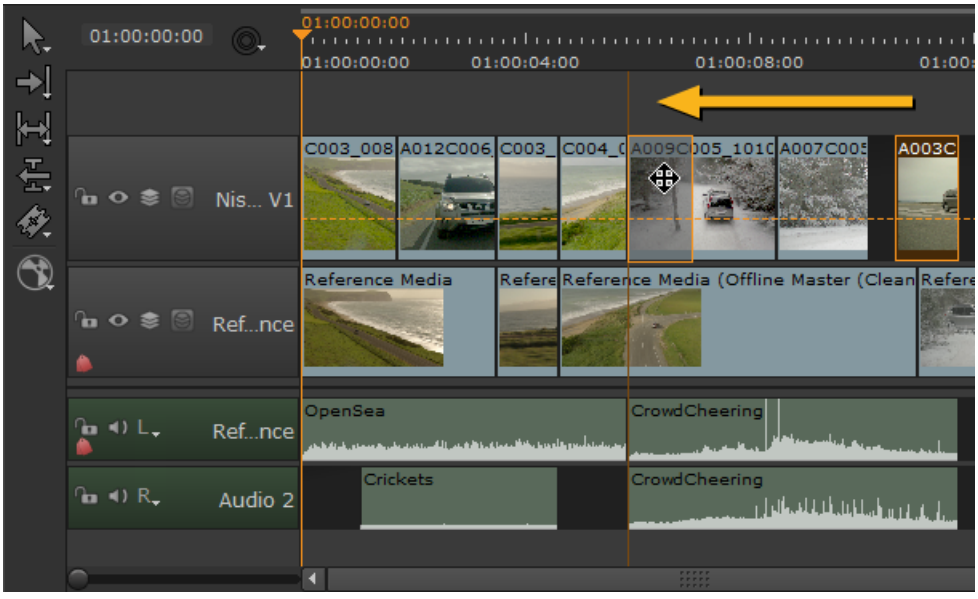
Note: You can only **Insert** or **Overwrite** using clips from the current project.

See Insert, Overwrite, and 3-Point Editing for more information on source/record editing.

- Use the **Multi** or **Move/Trim** tools to make space for the new clip and then drag-and-drop it in to the space (see Using the Move/Trim Tool for more information).



- Drag-and-drop the new clip at the end of the sequence, then using the **Multi** or **Move/Trim** tools, drag the new clip to an existing transition, hold down the **Alt** modifier, and drop the clip to **Ripple** all other shots down the timeline.



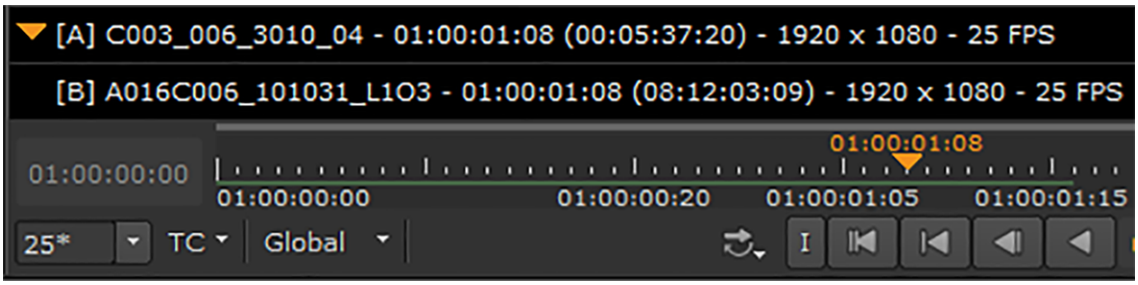
Tip: Enabling **Preferences** > **Panels** > **Timeline** > **show frame end marker** draws an extra line on the timeline to the right of the playhead, indicating the end of the current frame.

Audio and the Timeline

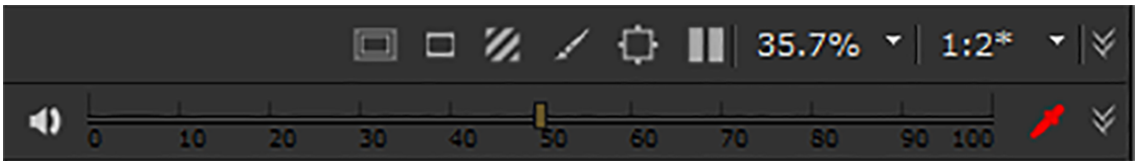
Audio tracks on the timeline are handled in much the same way as video tracks. By default, linked audio and video tracks are edited at the same time, but you can lock either track and move them independently or hold **Alt** to select a single track, if required.

Nuke Studio supports integrated audio from container files, such as **.mov**, and audio specific files, such as **.wav**, on the timeline. Nuke Studio assigns the channels in the audio file to separate tracks and matches the channels in the file to the available channels on the timeline.

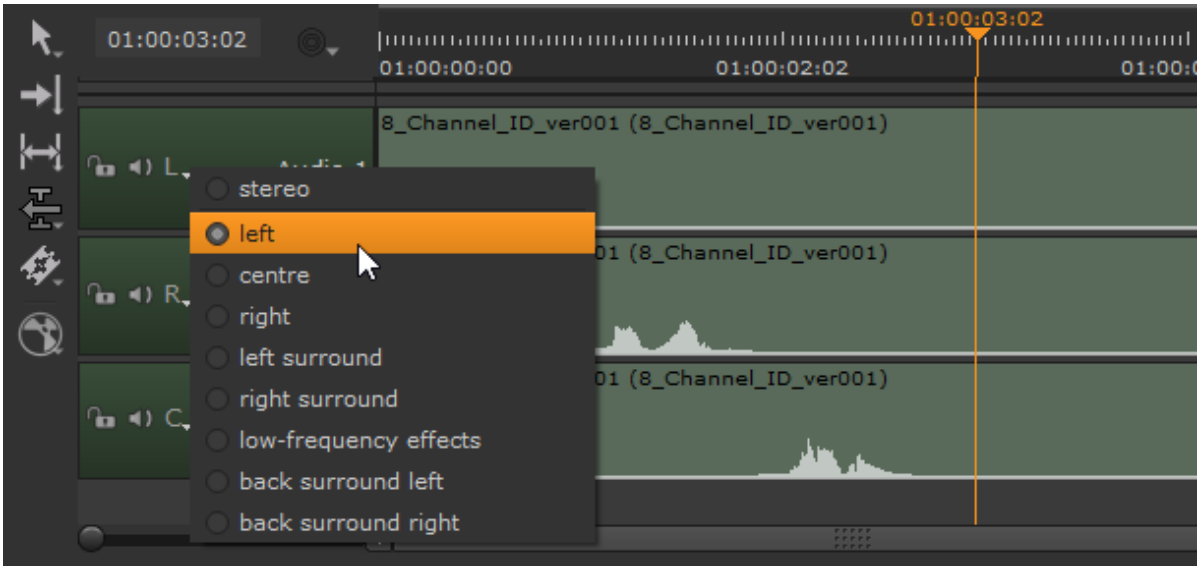
When both Viewer inputs contain clips, the audio output is set by the orange marker in the color picker information bar, displayed by clicking . In the following example, input A is providing the audio output:



The volume slider in the upper-right corner of the Viewer controls the output level for that Viewer only.

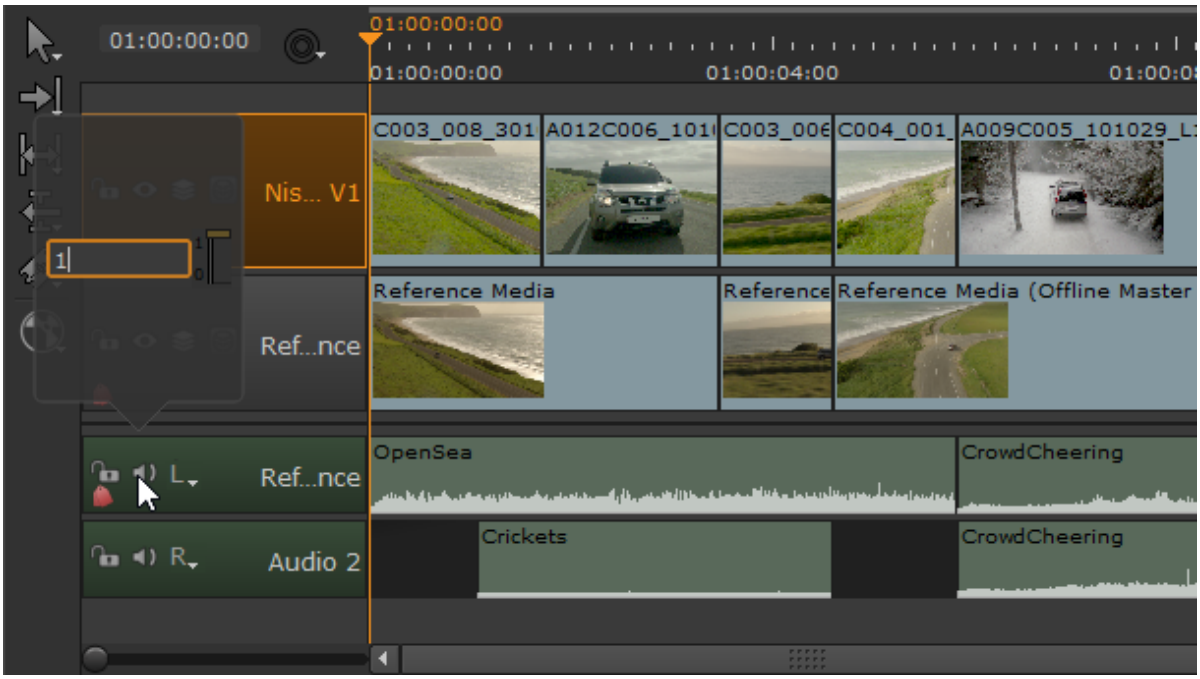


Audio output for shots can be toggled between eight channels, depending on the requirements of the clip or shot: left, center, right, left surround, right surround, low-frequency effects, back surround left, and back surround right. Click on the dropdown to cycle between outputs:



You can also control audio on a per track and per shot basis. Audio track headers and shots have independent volume controls in the timeline and **Properties** tab.

• **Track headers** - click and hold the mute icon on the header to display the volume slider.



• **Shots** - select an audio shot and click on the **Properties** tab to display the Volume control.

Tip: You can control the volume on multiple shots simultaneously by holding **Ctrl**/**Cmd** and selecting the required items before adjusting the volume slider.

The **Preferences** > **Panels** > **Viewer (Sequence)** sub-menu contains audio controls allowing you to control the volume level for all new Viewers. See Appendix A: Preferences for more information.

Note: If the frame rate drops too low, audio is automatically muted and the speaker changes to the no audio playback icon.

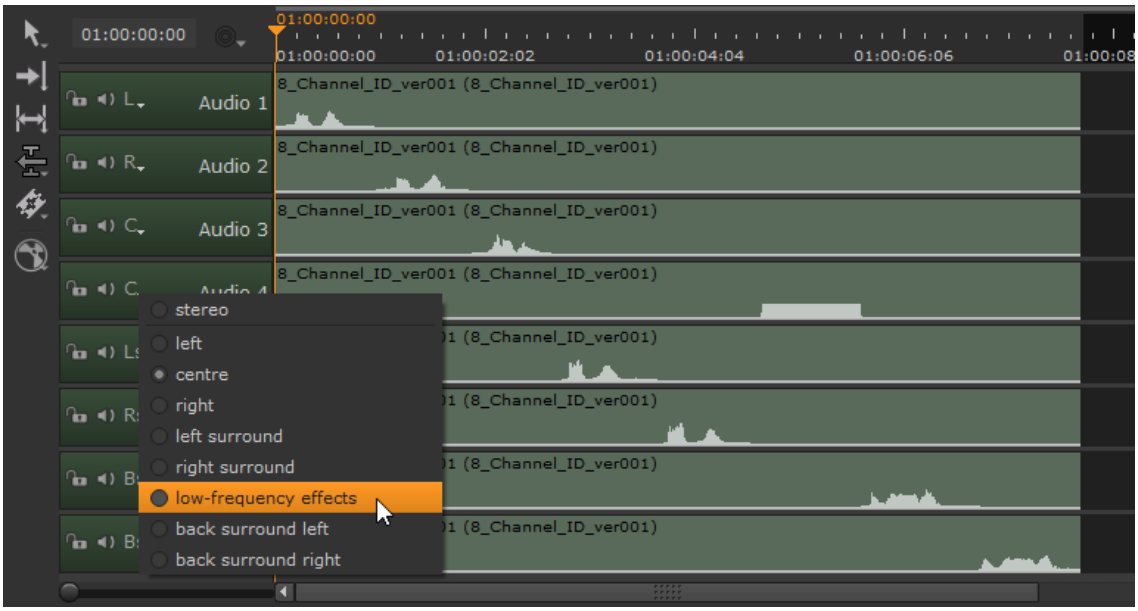
WAV Shots

Audio can be recorded at the same time as shooting the video or it can be unrelated to the shoot, for example sound effects or music. You can add **.wav** clips to the timeline in two ways:

- **Drag-and-drop** - drag your **.wav** clip to a timeline audio track and drop it in to place.
- Navigate to **File** > **Import File(s)** or **Import Folder(s)**.

Tip: You can also use the **Ctrl/Cmd**+**I** and **Ctrl/Cmd**+**Shift**+**I** keyboard shortcuts.

Nuke Studio assigns the channels in the audio file to separate tracks and matches the channels in the file to the available channels on the timeline. If a channel is not matched correctly, you can select the channel manually using the dropdown in the track header. For example, **LFE** (low frequency effects) may be assigned to **C** (center) incorrectly.



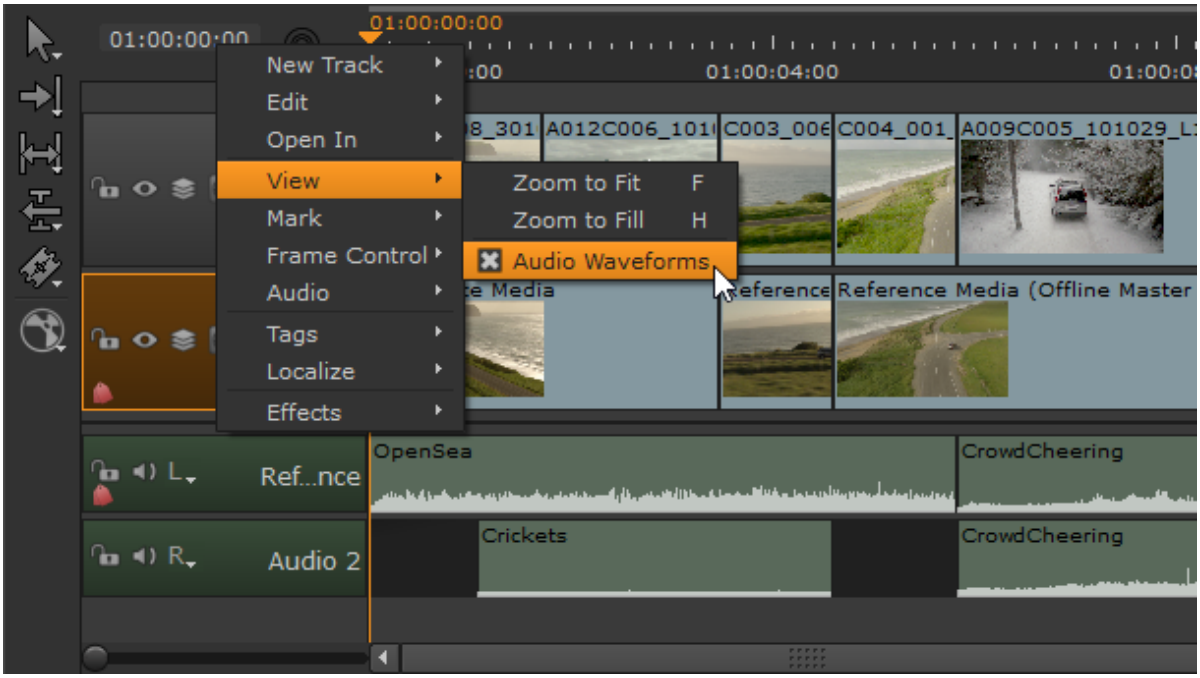
Use theTimeline Editing Tools to move the clip into place and set its output.

Displaying Audio Waveforms

Visualizing an audio waveform helps synchronization with video events, and Nuke Studio displays waveforms in the timeline by default.

Audio shots are manipulated in the same way as video shots, so using waveforms in conjunction with the Timeline Editing Tools enables you to quickly synchronize audio and video events. Audio shots also support **Fade In**, **Fade Out**, and **Dissolve** transitions in the same way as video. See Adding Transitions Between Shots and Audio Clips for more information.

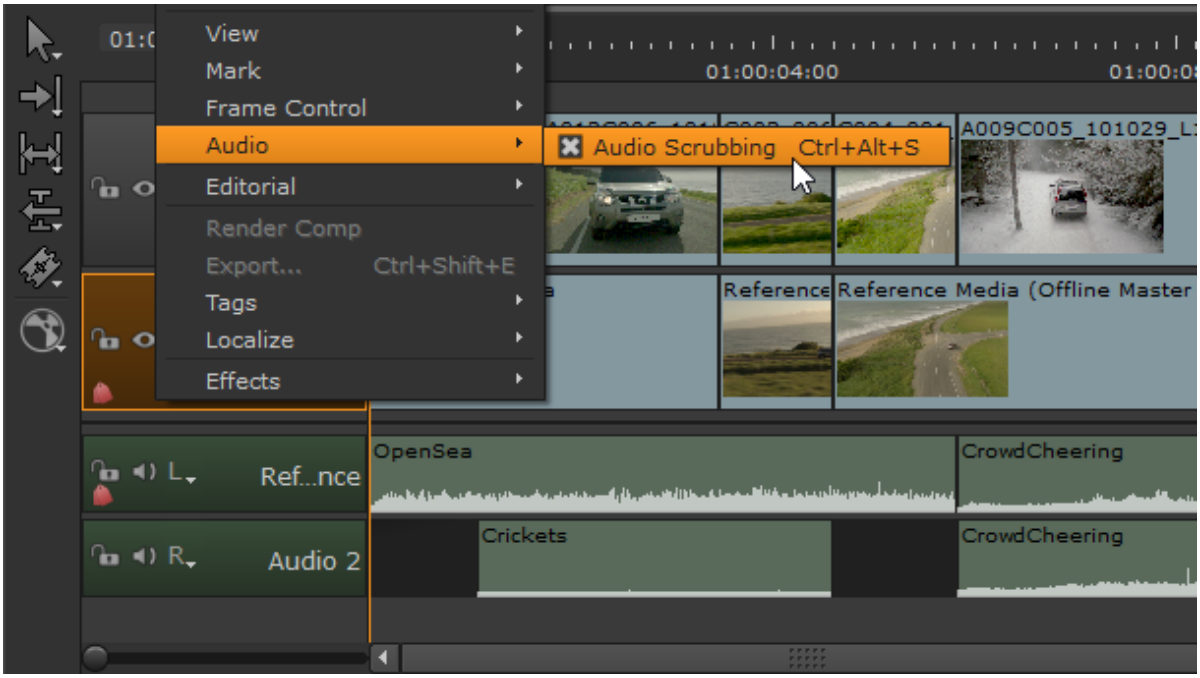
You can toggle the waveform display on and off by right-clicking in the timeline and selecting **View** > **Audio Waveforms**. You can also control how the waveform appears, when enabled. Open the **Preferences** and navigate to **Panels** > **Timeline** to toggle between full and **half waveforms**.



Displaying waveforms in audio-heavy projects can cause significant slow down on the timeline, so Nuke Studio includes a preference to limit how much system memory is available for waveform display. In the Preferences, navigate to **Performance** > **Caching** > **Audio Waveforms** and set the **waveform memory** control to the required amount.

Audio Scrubbing

Nuke Studio's timeline supports audio scrubbing, allowing you synchronize audio and video more easily while scrubbing the playhead. Audio scrubbing is disabled by default, but you can enable it by right-clicking in the timeline tab and clicking **Audio** > **Audio Scrubbing** or by pressing **Ctrl/Cmd**+**Alt**+**S**.



Audio shots cache temporarily to increase responsiveness during scrubbing. If you need to clear the audio cache, navigate to **Cache** > **RAM Cache** > **Clear Audio Cache**.

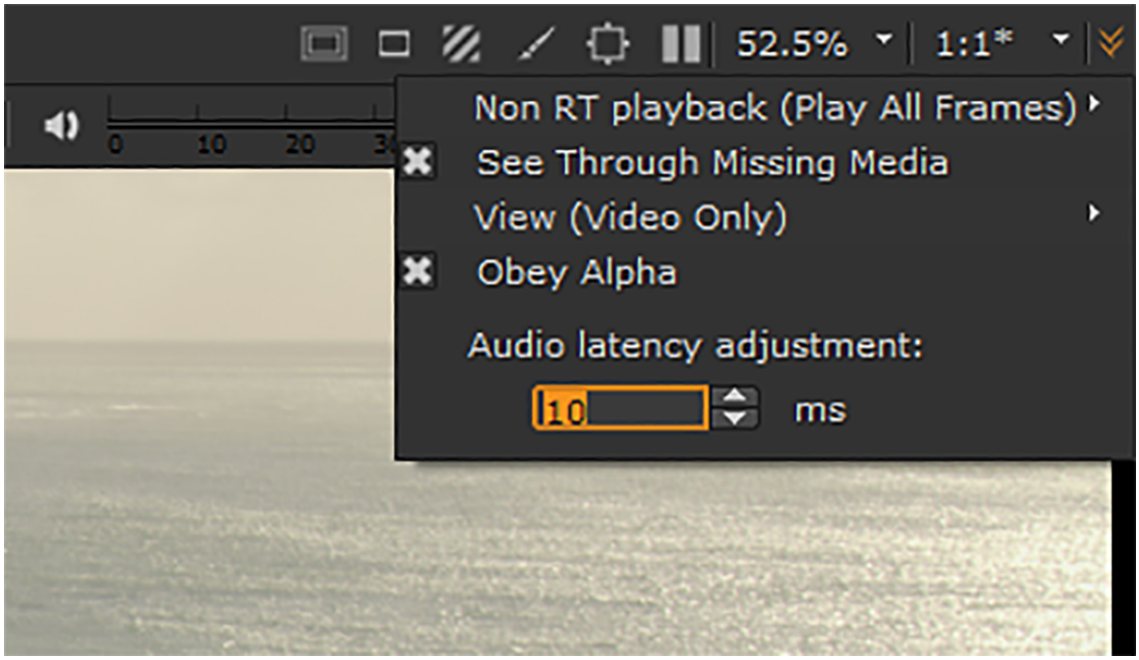
Note: Audio scrubbing is not currently available through monitor output cards. Audio scrubbing is only supported through internal audio output devices.

Synchronizing Audio and Video

Nuke Studio allows you to massage the synchronization between audio and video tracks using audio latency adjustment during playback in the Viewer, or by a default amount in the **Preferences** > **Panels** > **Viewer (Sequence)** sub-menu.

Note: Latency adjustments can take a few seconds to affect the audio track.

1. Mark a portion of the timeline containing the target audio and video shots using In and Out markers.
2. Press  or use the **L** keyboard shortcut to begin playback.
3. Click the Viewer settings icon and increment the latency using the controls in the popup.



4. Adjust the latency until the tracks are in sync.

PulseAudio on Linux

PulseAudio on Linux distributions has been linked with fluctuating frame rates due to the latency when retrieving audio samples. If detects that your setup is running PulseAudio alongside the application, a warning message displays.



Stopping PulseAudio

You can disable PulseAudio for the current user or all users on a machine. To stop the daemon, do the following:

1. Open the **~/.pulse/client.conf** file to disable PulseAudio for the current user,

   OR

   Open the **/etc/pulse/client.conf** file to disable PulseAudio for all users.
2. Set the following attribute and ensure the line is not commented out:
   `autospawn = no`
3. Call **pulseaudio --kill** to end the PulseAudio process.
4. Call **ps -e | grep pulse** to check the process stopped correctly.



Note: Ending PulseAudio while other applications are running may disable audio output. Stop and start the application to re-enable audio output. Additionally, the desktop audio slider may be removed.

Restarting PulseAudio

To start the PulseAudio daemon, do the following:

1. Open the **~/.pulse/client.conf** file to enable PulseAudio for the current user,

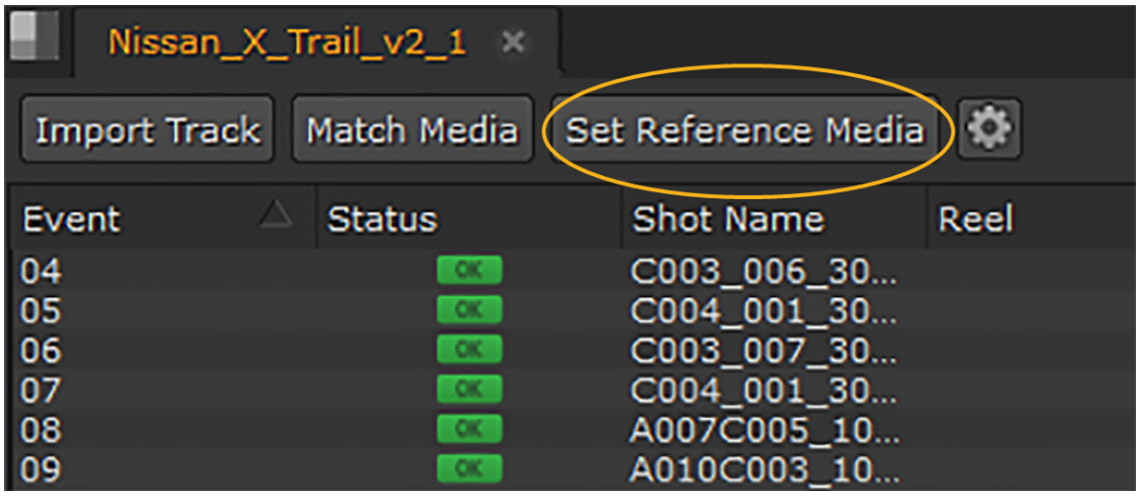
   OR

   Open the **/etc/pulse/client.conf** file to enable PulseAudio for all users.
2. Set the following attribute and ensure the line is not commented out:
   `autospawn = yes`
3. Call **pulseaudio --start** to start the PulseAudio daemon.
4. Call **ps -e | grep pulse** to check the process started correctly.

Using Reference Media

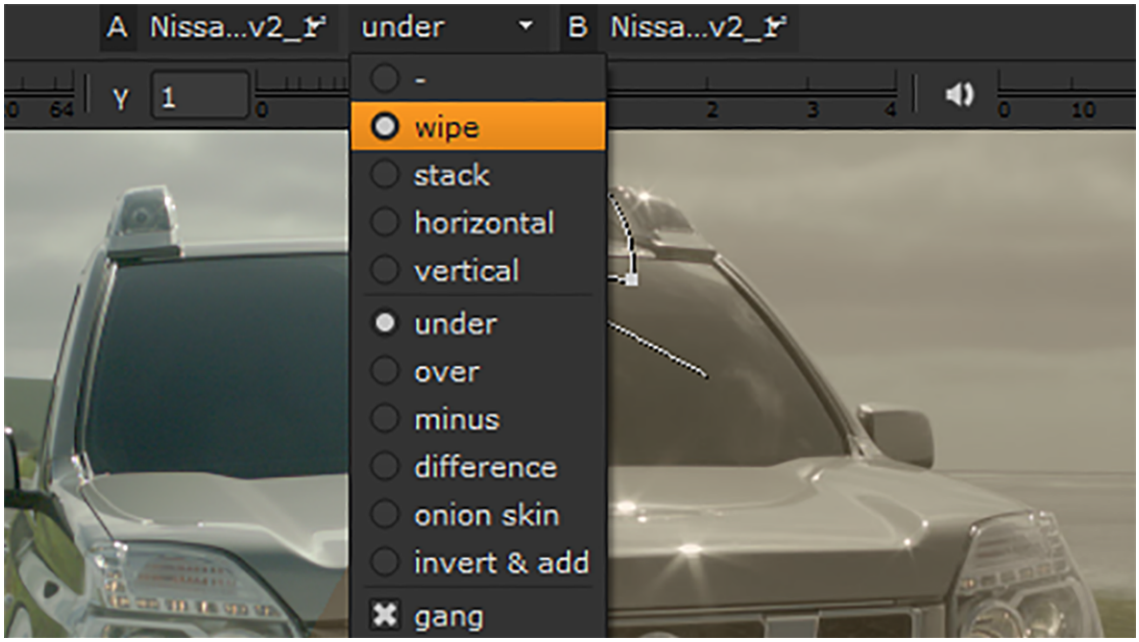
Importing a reference version of your timeline enables you to compare your current timeline against the reference media to avoid issues with continuity, missed frames, and so on.

To import reference media, click **Set Reference Media** and use the browser to locate the required file.



The reference media is automatically imported into **Reference** tracks, pushing existing tracks outward, and marked with the **Reference Media** tag.

After importing the reference media, use the show/hide icon or A/B input tools to compare the current timeline against the reference clip. SeeComparing Media for more information.

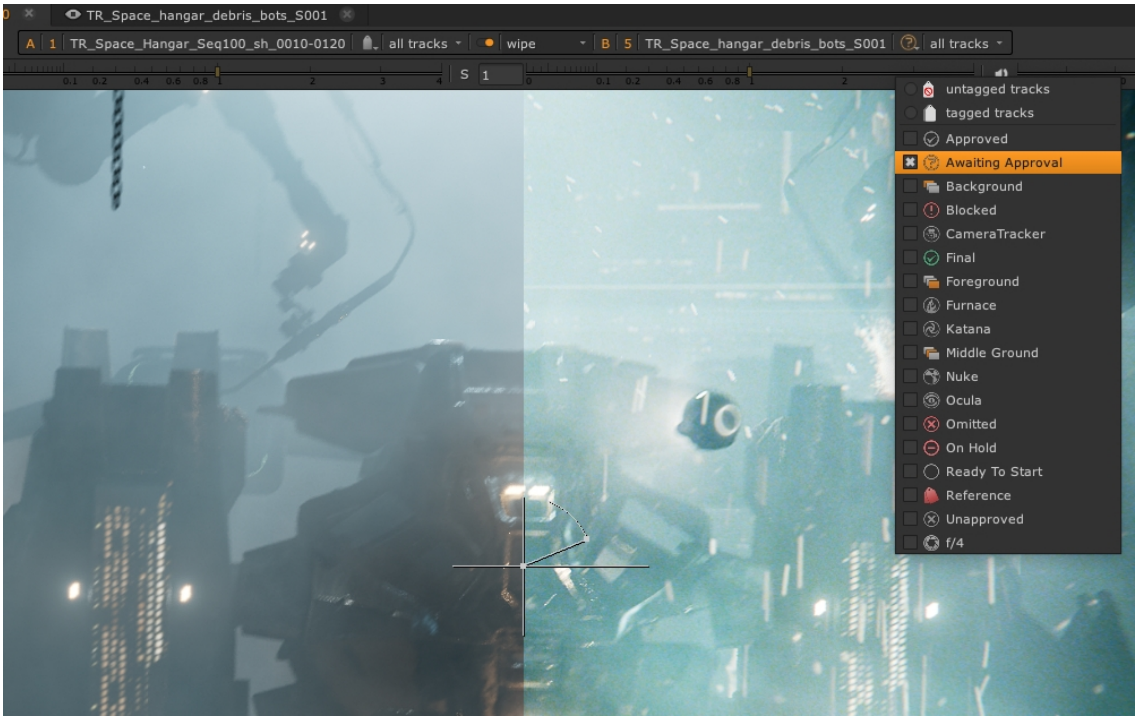


Comparing Media

The Viewer A/B tools allow you to quickly compare media using the two Viewer input buffers. Select a clip, sequence, shot or track and press **A** or **B** to place your selection in the Viewer input buffers. You can also drag-and-drop items into the input buffers using the Viewer hotspots.

Note: The Viewer currently treats all alpha channels as premultiplied, which can result in the Viewer background being "added" to the image. If you're working with un-premultiplied images, set the Viewer background to **Black**. See Appendix A: Preferences for more information.

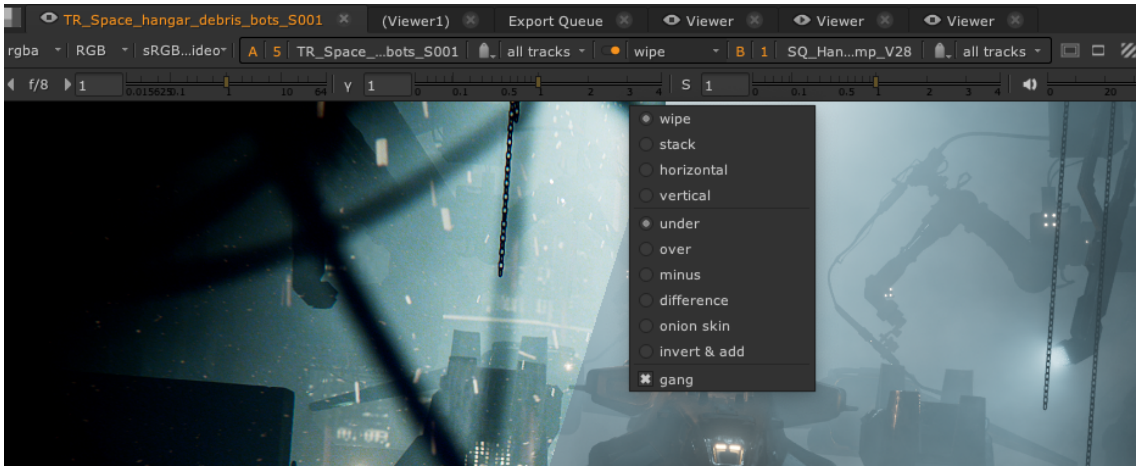
When the Viewer input buffers contain sequences, the A and B dropdowns control what is displayed in the Viewer using track names and tags. Selecting a track or tag from the dropdown displays the selected media in the Viewer.



Use the **wipe**, **stack**, **horizontal**, and **vertical** modes to control how the buffers are displayed in the Viewer.

Note: If you're working in a multi-view project, using stereo footage for example, you can set which view is output in the A and B buffers using the Views buttons over the Viewer. See Displaying Views in the Viewer for more information.

The **wipe** and **stack** modes also allow you to blend the two buffers together, and in the case of **wipe** mode, provides a handle in the Viewer to quickly wipe between the two inputs.



The color picker overlay displays a description of the contents of the A and B inputs, or **No Clip** when there is no clip at the playhead, for instance, when there is a gap in a timeline or if a track is disabled.

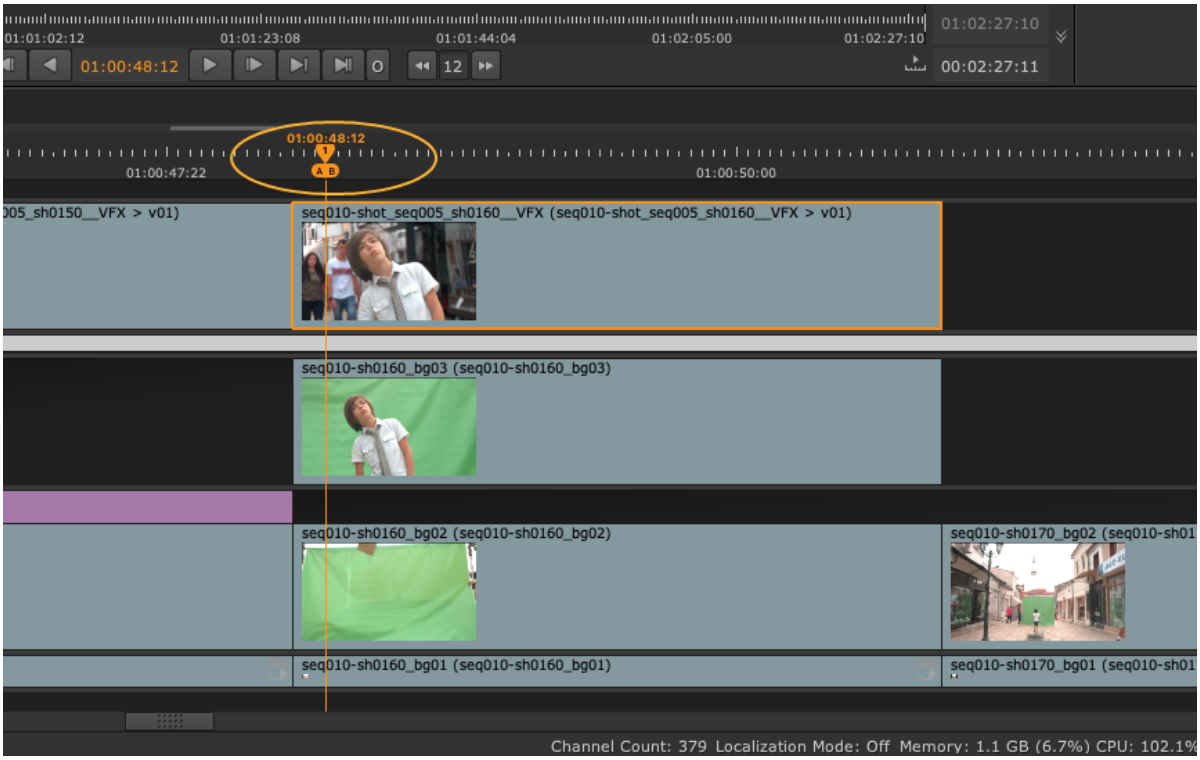


The orange triangle in the overlay denotes the clip currently supplying audio and timecode information in the Viewer.

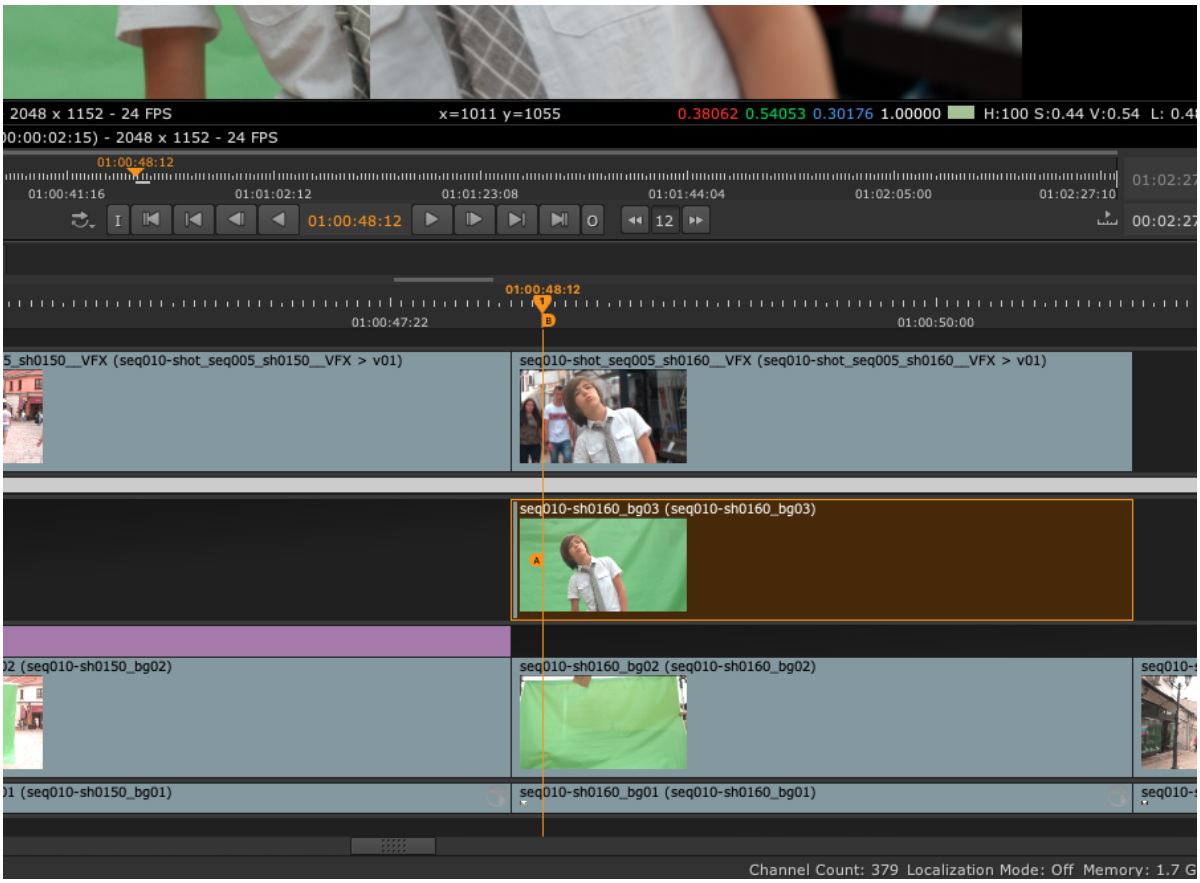
Playhead A/B Indicators

The timeline playhead indicates what is currently displayed in the Viewer using a number of different display markers. Swapping tracks between buffers automatically updates the playhead so that you always know which tracks you are looking at in the Viewer.

- Selecting **all tracks** in the current Viewer buffer displays an 'AB' at the top of the playhead to indicate that you're viewing all tracks in the sequence from top to bottom.

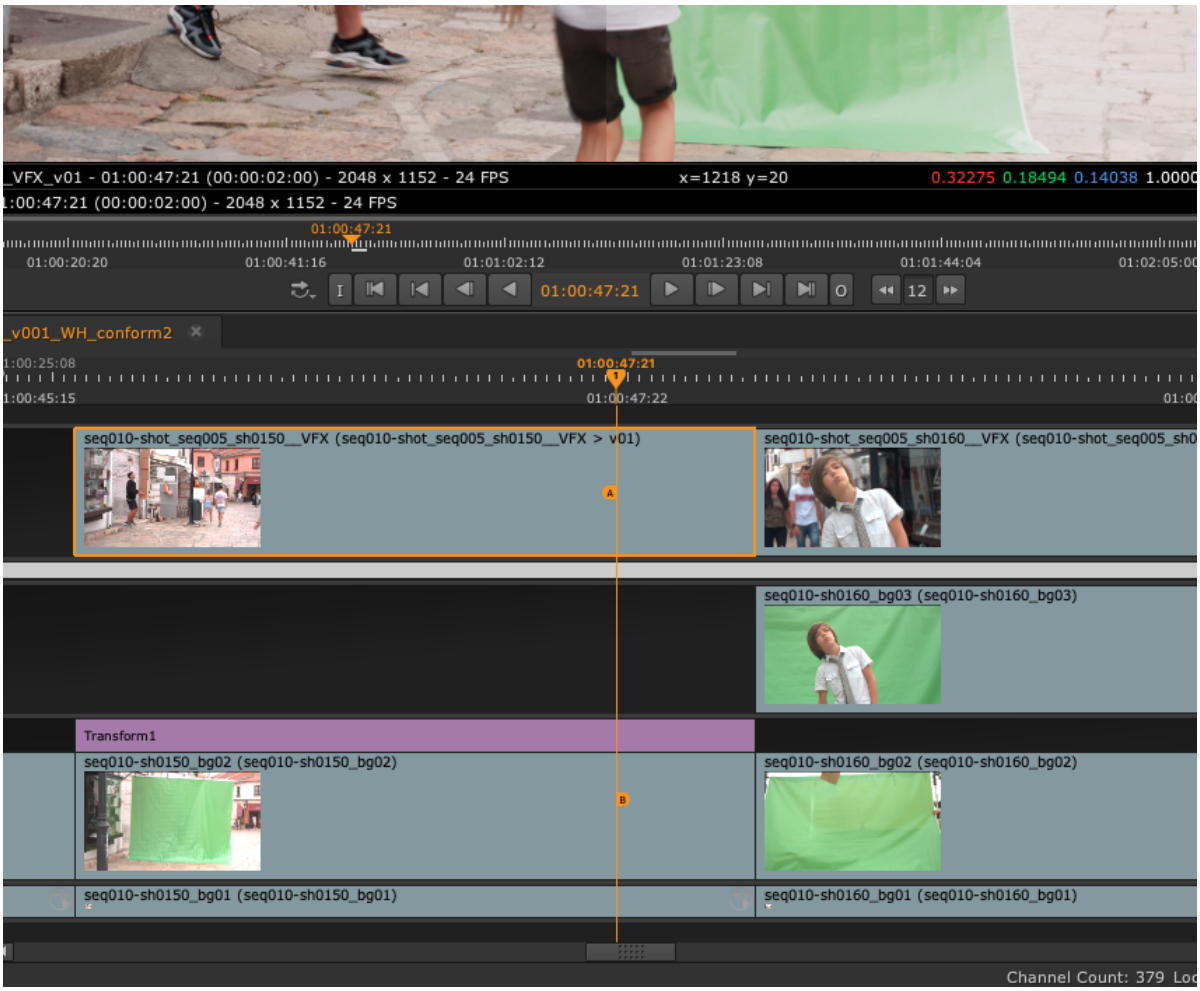


- Selecting a track for the current compare Viewer buffer displays your A or B tab at the playhead in the selected track.



- Selecting different tracks to the Viewer's A/B buffers draws A and B tabs on the playhead in the selected tracks.

Tip: Use **A** or **B** hotkeys whilst selecting shots to quickly assign Compare viewers.



Caching Frames in the Playback Cache

The playback cache places frames in RAM for rapid retrieval during playback, rather than creating files locally as with Caching Frames in the Disk Cache orLocalizing Media.

The white bar under the Viewer represents the contents of the playback cache, a full bar indicating that the entire clip or timeline is currently accessible from RAM, optimizing playback. You can:

• Temporarily disable caching using the pause button above the Viewer, or use the **P** keyboard shortcut.

Clicking pause again, resumes caching from the playhead position.

- Flush the cache completely by navigating to **Cache** > **Clear Playback Cache**. Caching is automatically paused after flushing, but clicking the pause button resumes caching from the playhead position.

There are also a number of **Preferences** that affect how much RAM is available and when caching should occur. To set the caching behavior:

1. Navigate to **Nuke** > **Preferences** (OS X) or **Edit** > **Preferences** (Linux and Windows),

   OR

   Use the Preferences keyboard shortcut **Shift**+**S**.
2. Select **Performance** > **Caching** and set the total RAM cache available using the **playback cache size** field.



Note: You can't set this to a value higher than 80% of the memory available (rounded down to the nearest half-GB). For example, if you have 6 GB of memory available, the maximum cache size available 4.5 GB.

3. You can enable **free timeline playback RAM** to discard any frames cached to RAM (the white bar in the timeline Viewer) when you switch to the Node Graph within Nuke Studio, freeing the RAM for use in Nuke.



Note: When you switch back to the timeline, the cached files are re-cached, which can be time consuming.

4. Enable **pause caching when the application goes to the background** to pause playback caching when the application loses focus.

   When you click back into Nuke, caching picks up from where it stopped.
5. Enable **clear cache when the application goes to the background** to flush the playback cache when the application loses focus.

   When you click back into Nuke, caching starts again from the position of the playhead.

Caching Frames in the Disk Cache

Timeline Disk Caching provides reliable playback for more complex timelines by rendering frames to disk using the GPU. The cache provides persistent frames per edit in the timeline that only needs updating for full changes on the edit, such as adding a soft effect. For editorial changes, only the new frames need to be cached.

Note: Frames are always cached at the sequence resolution, regardless of source clip format and Viewer proxy settings.

You can cache whole sequences, selections of clip ranges, and frame ranges specified using In and Out points. Files in the disk cache are frames identical to what you see rendered in the timeline Viewer, written into **.exr** sequences, and saved in **NUKE_TEMP_DIR/TimelineCache** by default.

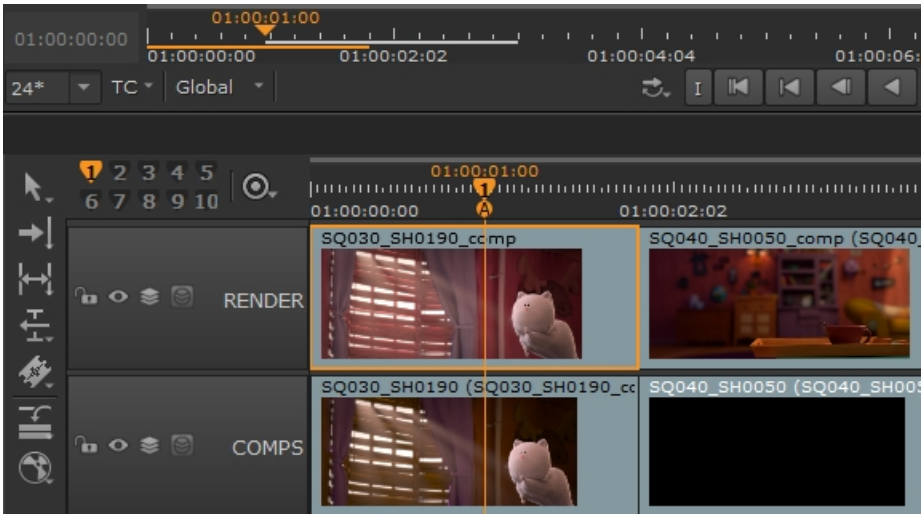
Tip: You can find the location of Nuke's general cache directory from within Nuke by hitting **X** on your keyboard, when the focus is on the Node Graph, and then running the following Tcl command:

```
value preferences.TimelineDiskCachePath
```

Cached frames are represented in the timeline by the state of the timeline cache icon:

- Automatic caching is active. See Update Changes to Cached Frames Automatically for more details.
- None of the frames in the current timeline are cached.
- The current timeline is partially cached.
- The current timeline is fully cached.

Cached frames are represented in the Viewer with an orange bar, under the RAM cache bar, which is white by default.



See Caching Sequence Ranges, Caching Selected Shot Ranges, and Caching In/Out Ranges for more information.

You can set the cache directory location, size of the timeline cache, and type of EXR compression used:

- Click the timeline cache icon and select **Cache Settings**, or
- Open the **Preferences** and navigating to **Performance** > **Caching**.

Note: You can override the caching mode in the **Preferences** using the **Sequence** panel's **Disk Caching Mode** dropdown.

Update Changes to Cached Frames Automatically

Timeline disk caching has two modes, **Automatic** and **Manual**, providing greater flexibility when used in conjunction with soft effects to ensure the best possible performance when playing back on the timeline.

- **Manual** - the default mode. When frames are cached, any changes made to soft effects on the cached frames are not automatically re-cached.
- **Automatic** - when frames are cached, any changes to soft effects made after the initial cache operation automatically triggers a re-cache of the affected region of your cached timeline. This means that your timeline is always cached and up to date in the selected regions, regardless of changes made.

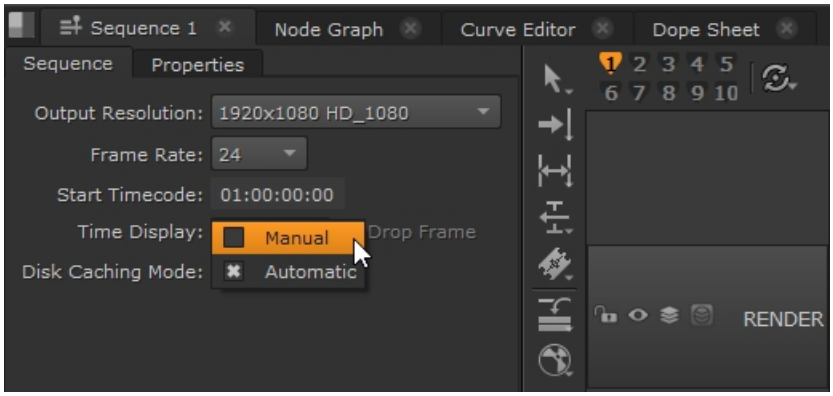
To set the caching mode for all new sequences:

1. Navigate to **Nuke** > **Preferences** (macOS) or **Edit** > **Preferences** (Linux and Windows),

   OR

   Use the **Preferences** keyboard shortcut **Shift**+**S**.
2. Select **Performance** > **Caching**.
3. Click the **default disk caching mode** dropdown and select **Manual** or **Automatic**.

   Any new sequences you create default to the option selected in the **Preferences**.
4. You can override this preference in the **Sequence** panel using the **Disk Caching Mode** dropdown.



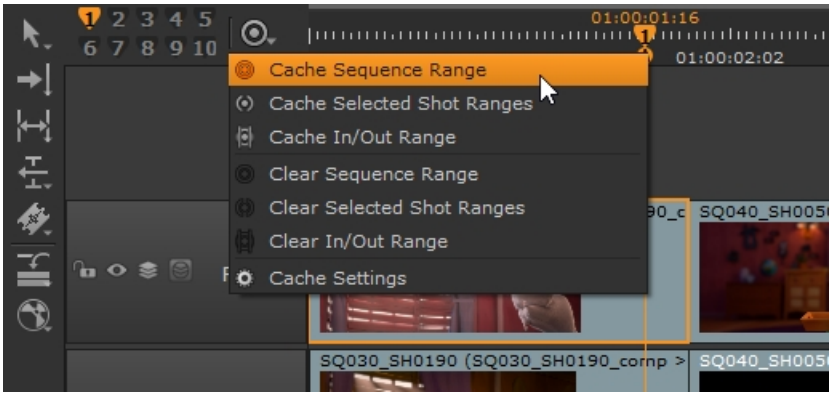
Caching Sequence Ranges

Caching a sequence caches exactly what you see when play through the entire timeline. As the playhead walks the timeline, all tracks are read from the highest track downwards, so what you see in the Viewer is what you cache to disk.

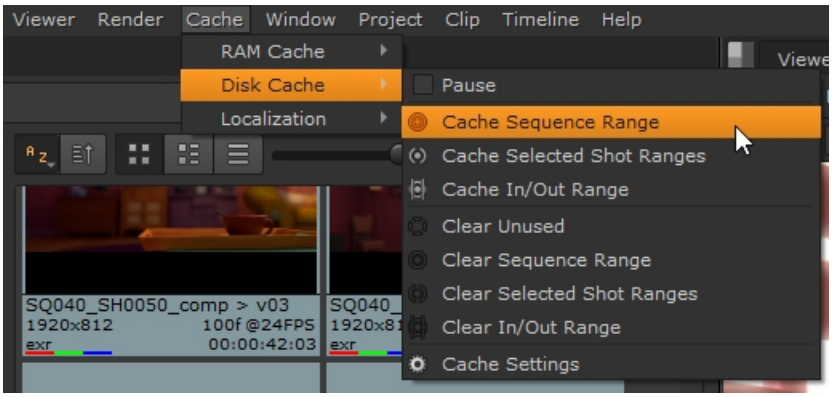
You can only cache one sequence at a time, either from the timeline itself, the **Cache** menu, or the **Project** bin:

Tip: You can pause caching at any time by navigating to **Cache** > **Disk Cache** > **Pause** in the menu bar.

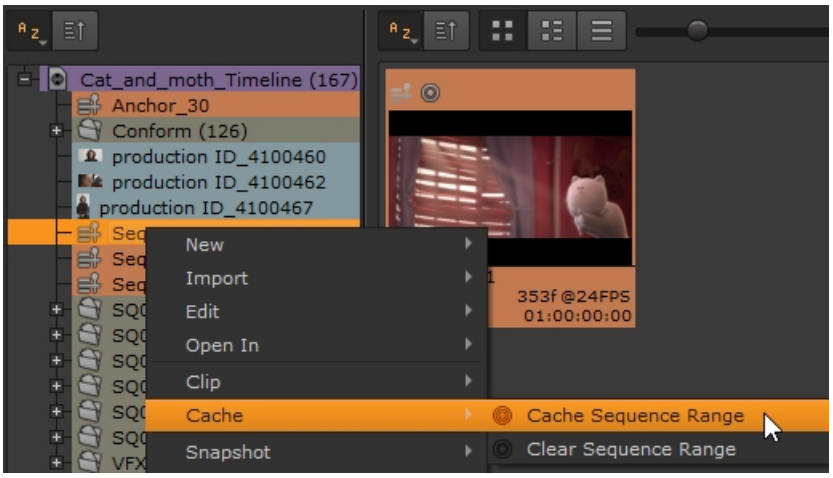
- Select the required sequence in the **Sequence** tab, click the icon, and then select **Cache Sequence Range**.



- Select the required sequence in the **Sequence** tab and then select **Cache** > **Disk Cache** > **Cache Sequence Range** from the menu bar.



- In the **Project** bin, right-click the sequence you want to cache, and then select **Cache** > **Cache Sequence Range**.



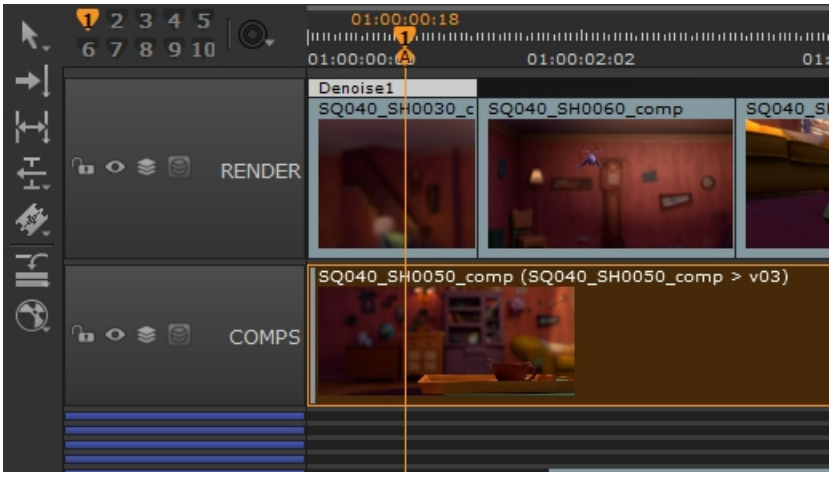
See Clearing Cached Frames for information on how to clear frames from the disk cache.

Caching Selected Shot Ranges

Caching shot ranges allows you to be more selective than caching an entire sequence, though it works in the same way. All tracks are read from the highest track downwards, so what you see in the Viewer for the shot selection is what Hiero caches to disk.

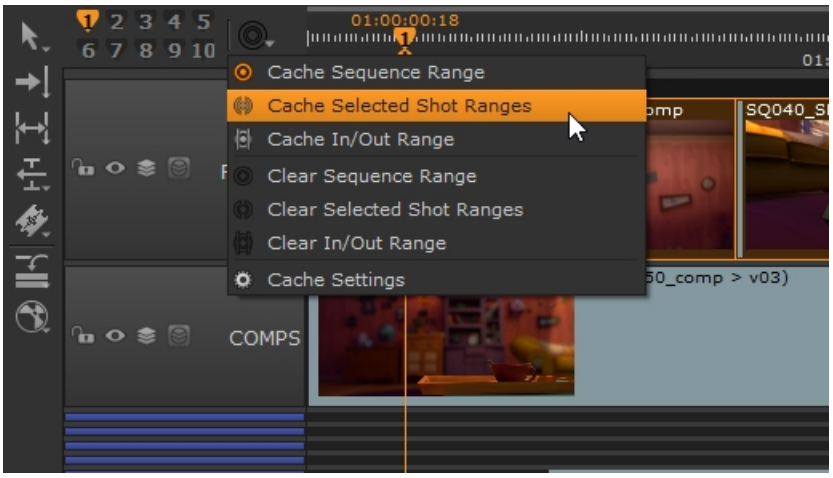
Tip: You can pause caching at any time by navigating to **Cache** > **Disk Cache** > **Pause** in the menu bar.

As a result of this, you may not get the frames you expect. For example, caching the shot in the image produces frames from the three shots on the highest track, including the soft effect, not the selected shot on the COMP track.

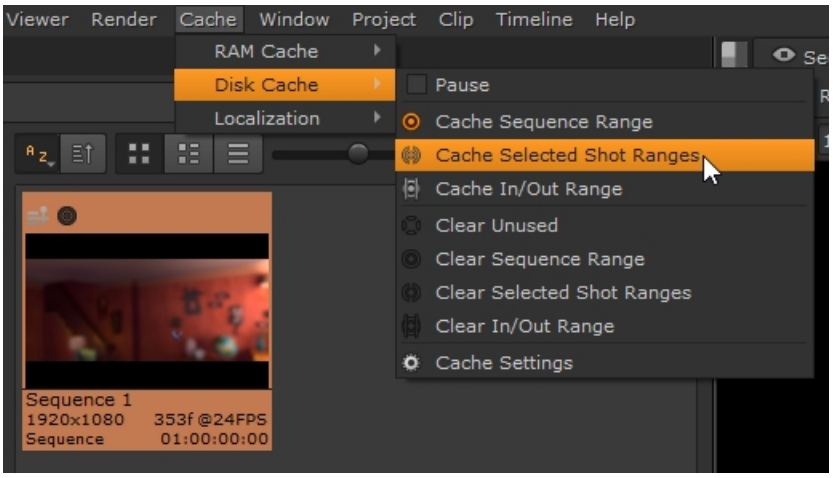


You can cache clip ranges from the timeline itself or from the **Cache** menu:

- Select the required clip range in the **Sequence** tab, click the  icon, and then select **Cache Selected Shot Ranges**.



- Select the required clip range in the **Sequence** tab and then select **Cache** > **Disk Cache** > **Cache Selected Shot Ranges** from the menu bar.



See Clearing Cached Frames for information on how to clear frames from the disk cache.

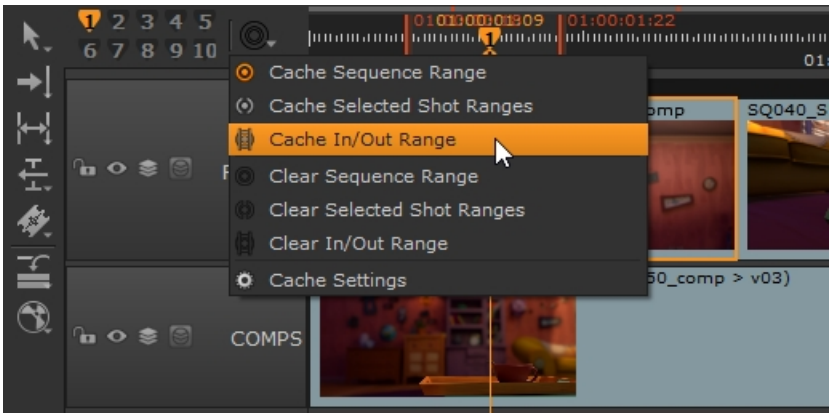
Caching In/Out Ranges

Caching ranges uses In and Out points set on the timeline to determine what you cache to disk. All tracks are read from the highest track downwards, so what you see in the Viewer between the In and Out markers is what you cache to disk. See Using In and Out Markers for more information.

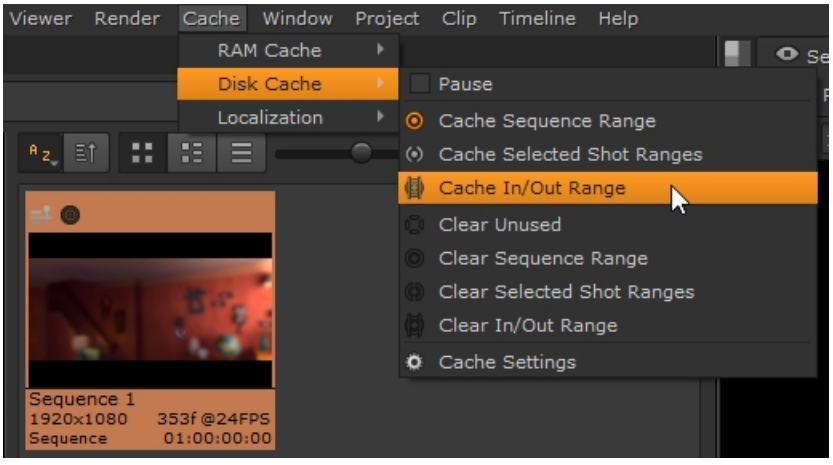
Tip: You can pause caching at any time by navigating to **Cache** > **Disk Cache** > **Pause** in the menu bar.

After setting In and Out points, you can cache that range from the timeline itself or from the **Cache** menu:

• In the **Sequence** tab, click the icon, and then select **Cache In/Out Range**.



• Select **Cache** > **Disk Cache** > **Cache In/Out Range** from the menu bar.



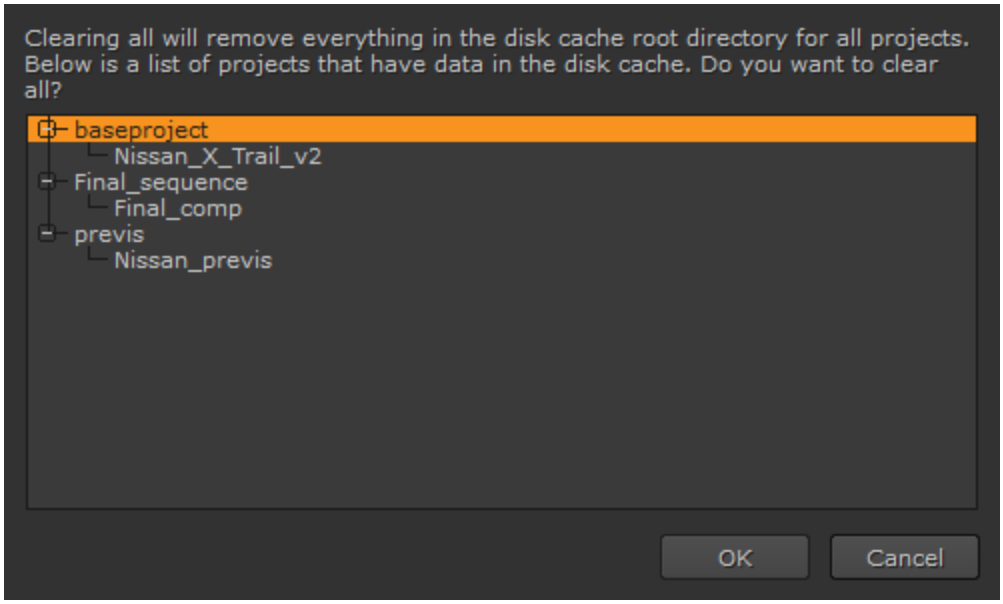
See Clearing Cached Frames for information on how to clear frames from the disk cache.

Clearing Cached Frames

Frames in the timeline cache can be cleared in a number of ways, including **Clear All** frames across all projects and the more selective **Clear Range** option.

Clearing All Cached Frames

The **Clear All** option is only available from the **Preferences** dialog under **Performance** > **Caching**, because it clears all frames from all projects and should be used with care. Clicking **Clear All** displays a warning dialog containing all the projects that are affected.

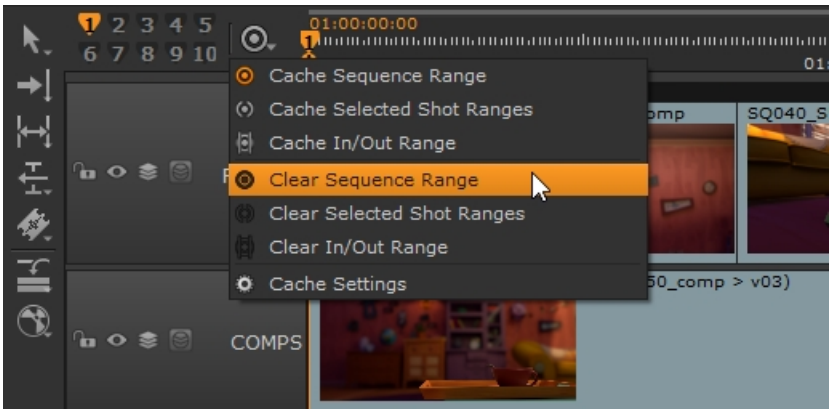


Click **OK** to proceed or **Cancel** to retain all the frames in the cache.

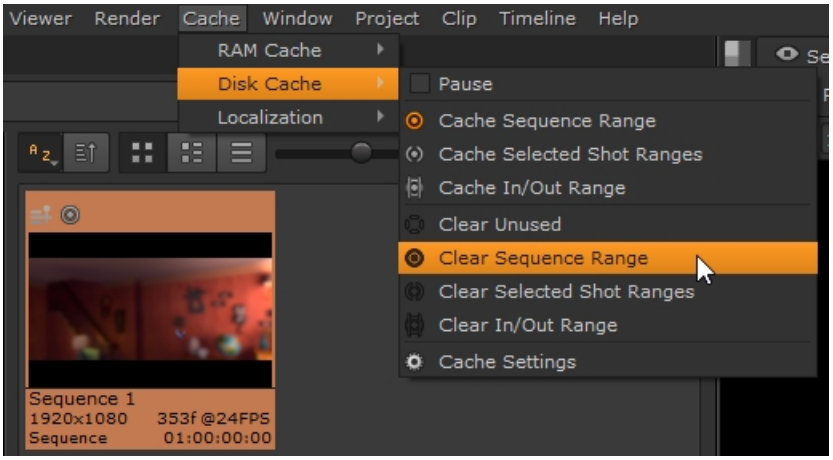
Clearing Sequences

Instead of using the **Clear All** option, you can clear certain sequences within a project. You can clear a sequence from the timeline itself, the **Cache** menu, or the **Project** bin:

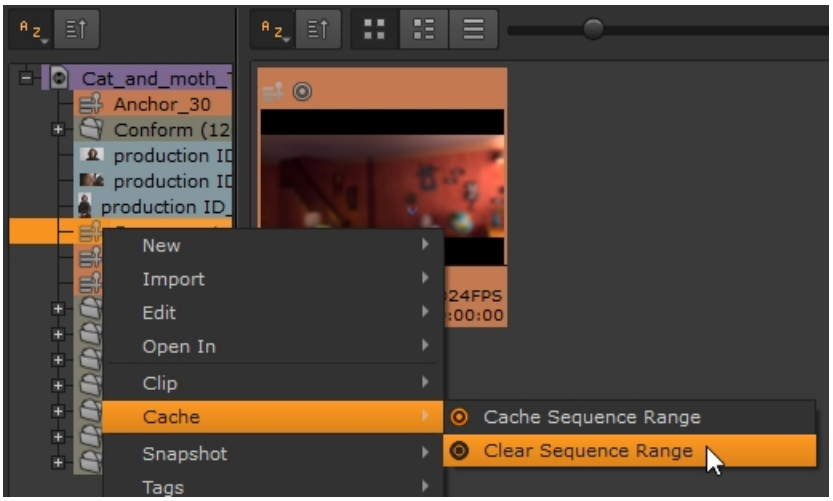
- Select the required sequence in the **Sequence** tab, click the icon, and then select **Clear Sequence Range**.



- Select the required sequence in the **Sequence** tab and then select **Cache** > **Disk Cache** > **Clear Sequence Range** from the menu bar.



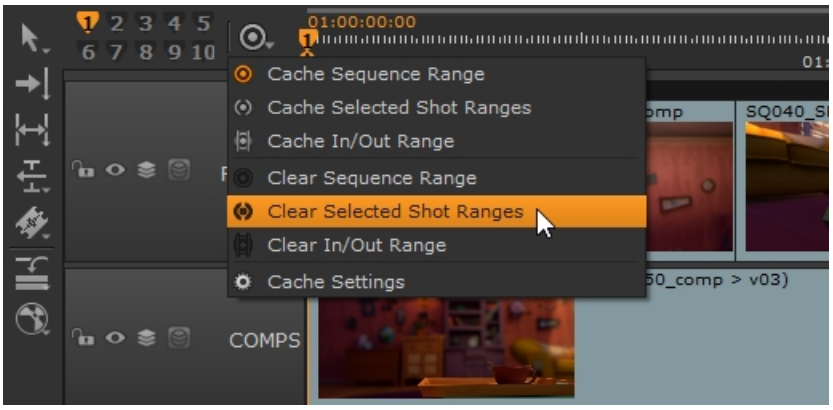
- In the **Project** bin, right-click the sequence you want to clear, and then select **Cache** > **Clear Sequence Range**.



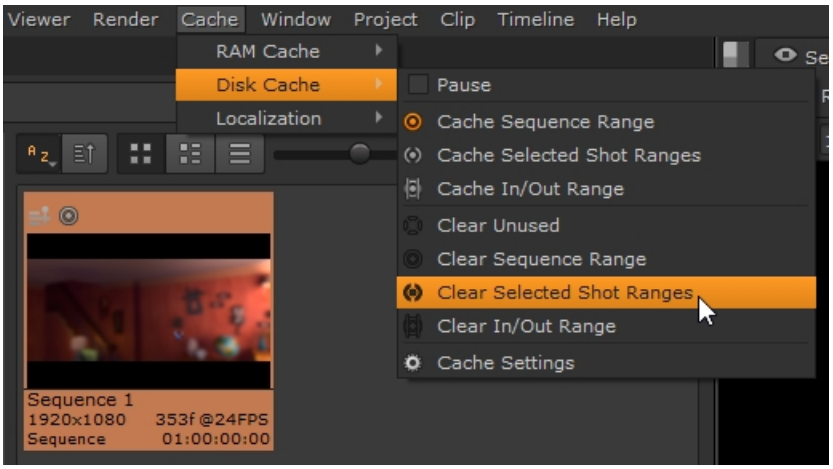
Clearing Selected Shot Ranges

You can also clear selected ranges of shots from a sequence, rather than the entire sequence, from the timeline itself or from the **Cache** menu:

- Select the required clip range in the **Sequence** tab, click the icon, and then select **Clear Selected Shot Ranges**.



- Select the required clip range in the timeline tab and then select **Cache** > **Disk Cache** > **Clear Selected Shot Ranges** from the menu bar.

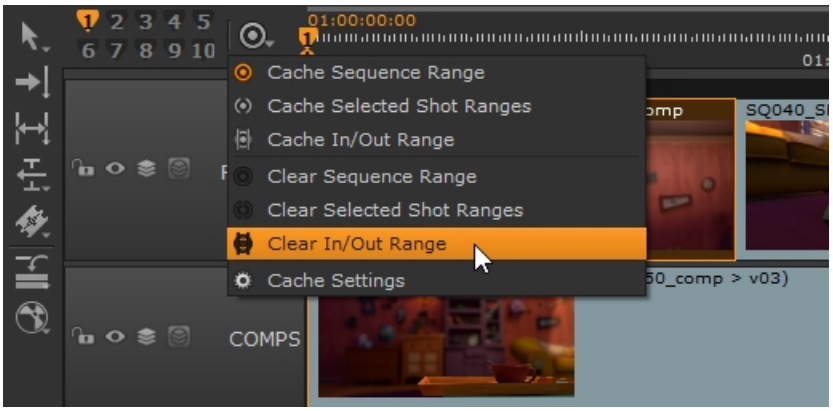


Clearing In/Out Ranges

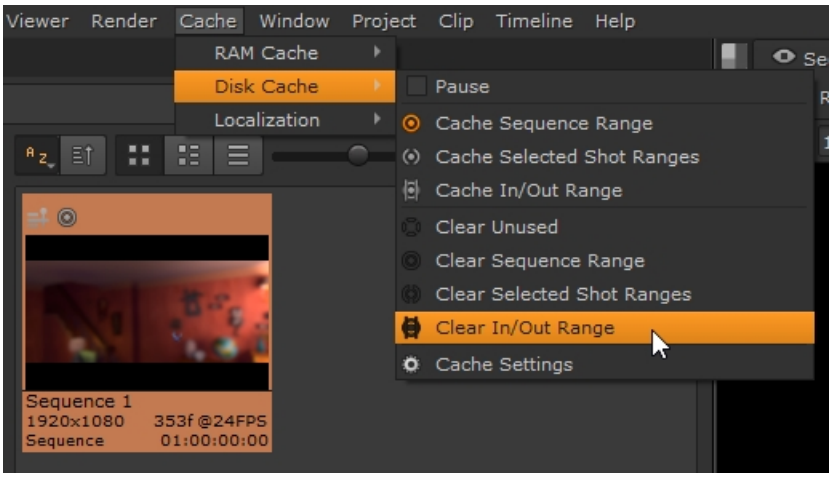
Clearing frames using In and Out markers is the most fine-grained method for removing frames from the timeline cache. Only the frames bracketed by the markers are removed, regardless of how the frames were cached originally. See Using In and Out Markers for more information.

After setting In and Out points, you can clear that range from the timeline itself or from the **Cache** menu:

• In the **Sequence** tab, click the icon, and then select **Clear In/Out Range**.

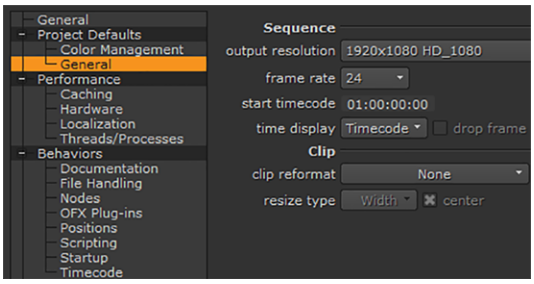


• Select **Cache** > **Disk Cache** > **Clear In/Out Range** from the menu bar.

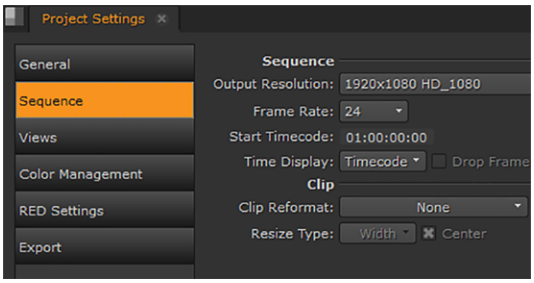


Viewing Multi-Format Timelines

All sequences and shots added to them adopt the **output resolution** and **clip reformat** settings from the **Preferences** dialog under **Project Defaults** > **General**. You can override these settings on a per **Project** or per **Sequence** basis.

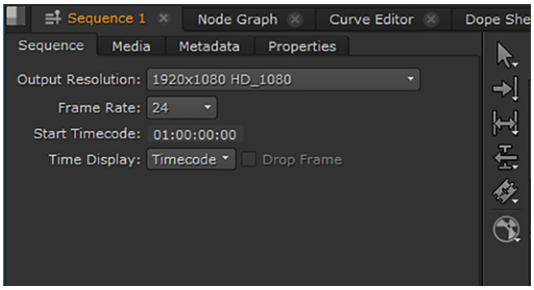


The **Preferences** dialog controls the default settings for new projects. Existing projects are unaffected by changes to the preferences.



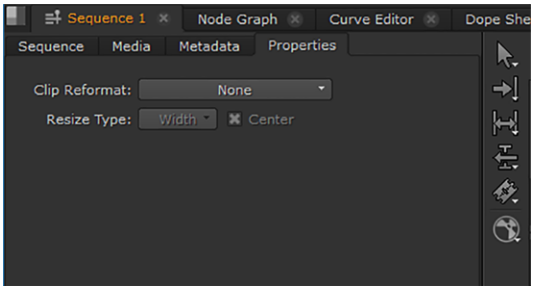
The **Project Settings** control the default settings for new sequences and shots in the current project. Existing sequences and shots are unaffected.

Project Settings override **Preference** settings.



The **Sequence** settings control the current sequence.

Sequence settings override **Project Settings** and **Preference** settings for the current timeline.



The **Properties** settings control shots in the current sequence.

Properties settings override **Sequence** settings, **Project Settings**, and **Preference** settings for shots on the current timeline.

Reformatting applied to shots on a timeline carry over into any export, including **Create Comp**. The reformat options in the **Export** dialog are applied after the transforms applied here. See Exporting from Nuke Studio for more information.

Setting the Sequence Format

The current sequence format is controlled by the **Output Resolution** dropdown in the timeline **Sequence** tab. The selected resolution is applied to all shots on the timeline, but by default, shots retain their native resolution. For example, setting the **Output Resolution** to **2k_DCP** on a timeline containing **HD_1080** shots results in the output resolution being larger than the shot.

Tip: Enabling the **Custom Sequence Format** guide at the top of the Viewer makes it easier to see differences between output resolution and shot format.



A 1920x1080 shot at native resolution.



The same shot with **Output Resolution** set to 2048x1080.

Similarly, setting the **Output Resolution** to a format smaller than the shot's native resolution results in the clip resolution being larger than the sequence resolution.



A 1920x1080 shot at native resolution.



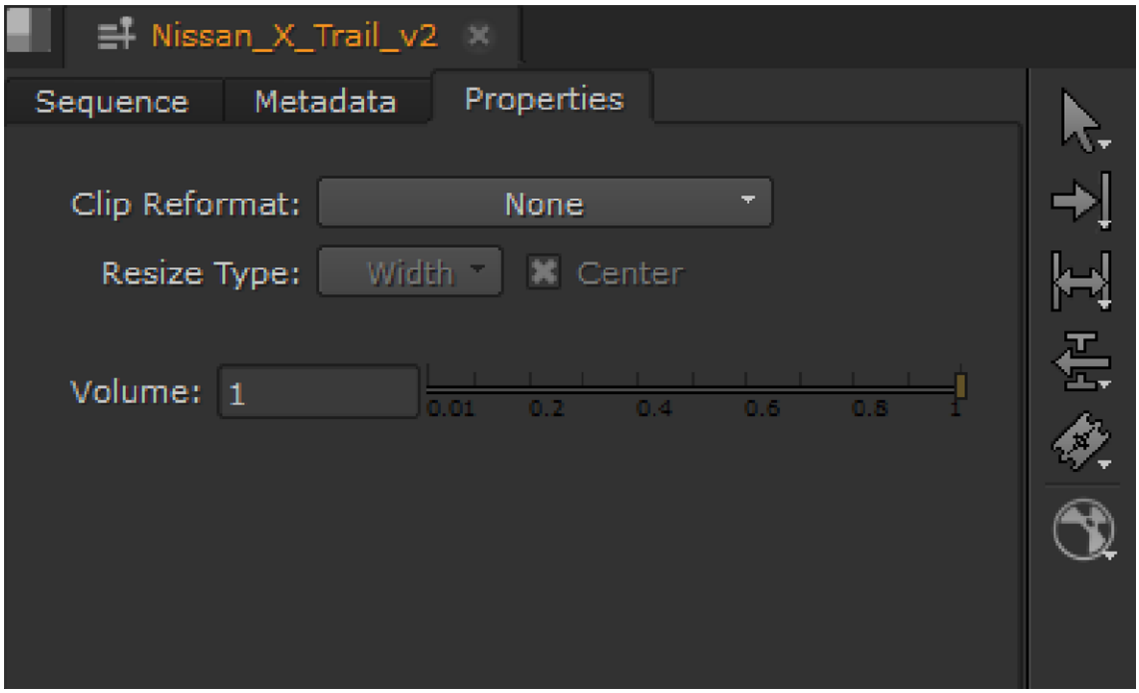
The same shot with **Output Resolution** set to 720x486.

Shots with a higher resolution than the sequence format enable you to apply soft effects, such as Transform, more easily. See Adding Effects on the Timeline for more information.

Reformatting Shots

Shots on the timeline retain the source clip's resolution by default. You can force shots to the sequence format using the timeline's **Properties** tab:

1. Select the target shots on the timeline.
2. Click the **Properties** tab to display the shot options.



Note: The **Volume** control is only available when you have an audio shot selected.

3. Click the **Clip Reformat** dropdown and selected **To Sequence Resolution**.
4. The shot is reformatted to the **Output Resolution** specified on the **Sequence** tab.
5. Depending on the source clip, you may need to change the **Resize Type**:
   - **Width** - scales the original until its width matches the format's width. Height is then scaled to preserve the original aspect ratio.
   - **Height** - scales the original until its height matches the format's height. Width is then scaled to preserve the original aspect ratio.
   - **Fit** - scales the original until its smallest side matches the format's smallest side. The original's longer side is then scaled to preserve original aspect ratio.
   - **Fill** - scales the original until its longest side matches the format's longest side. The input's shorter side is then scaled to preserve original aspect ratio.
   - **Distort** - scales the original until all its sides match the lengths specified by the format. This option does not preserve the original aspect ratio, so distortions may occur.

Tip: When **Clip Reformat** is **To Sequence Resolution**, disabling **center** places the clip at the bottom-left of the **Output Resolution**, rather than the center.

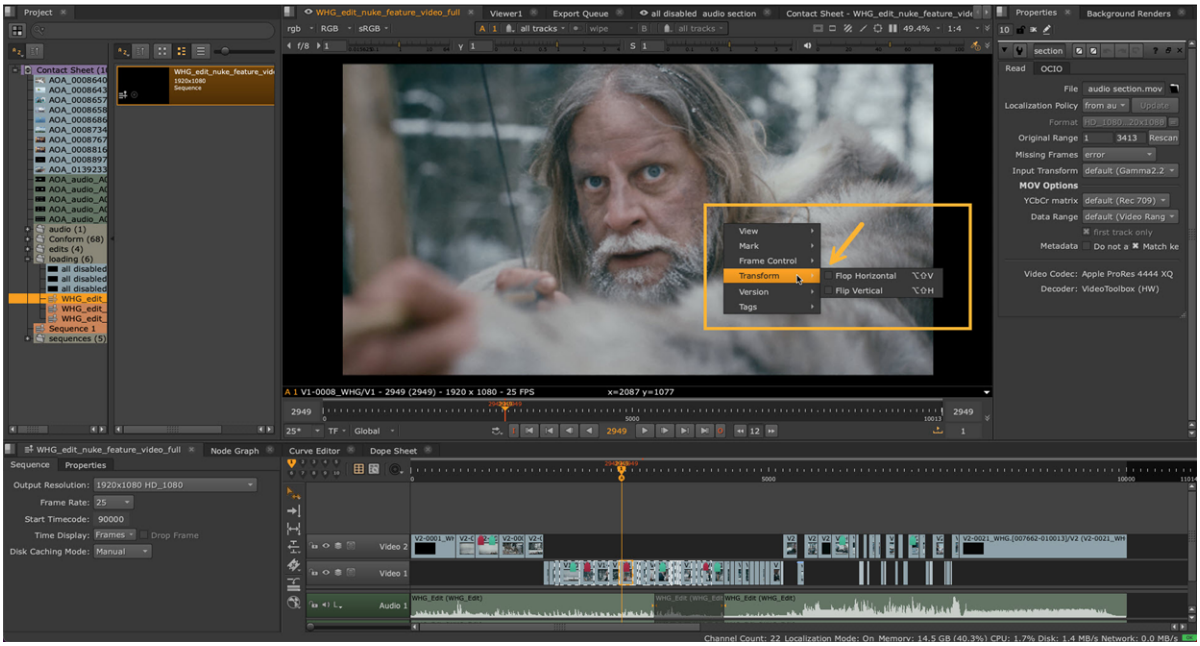
Transforming Shots

If you right-click on a shot in the timeline Viewer, a contextual menu will appear with various options.

By selecting **Transform** from the menu, you can then choose to either **Flop Horizontal** or **Flip Vertical**.

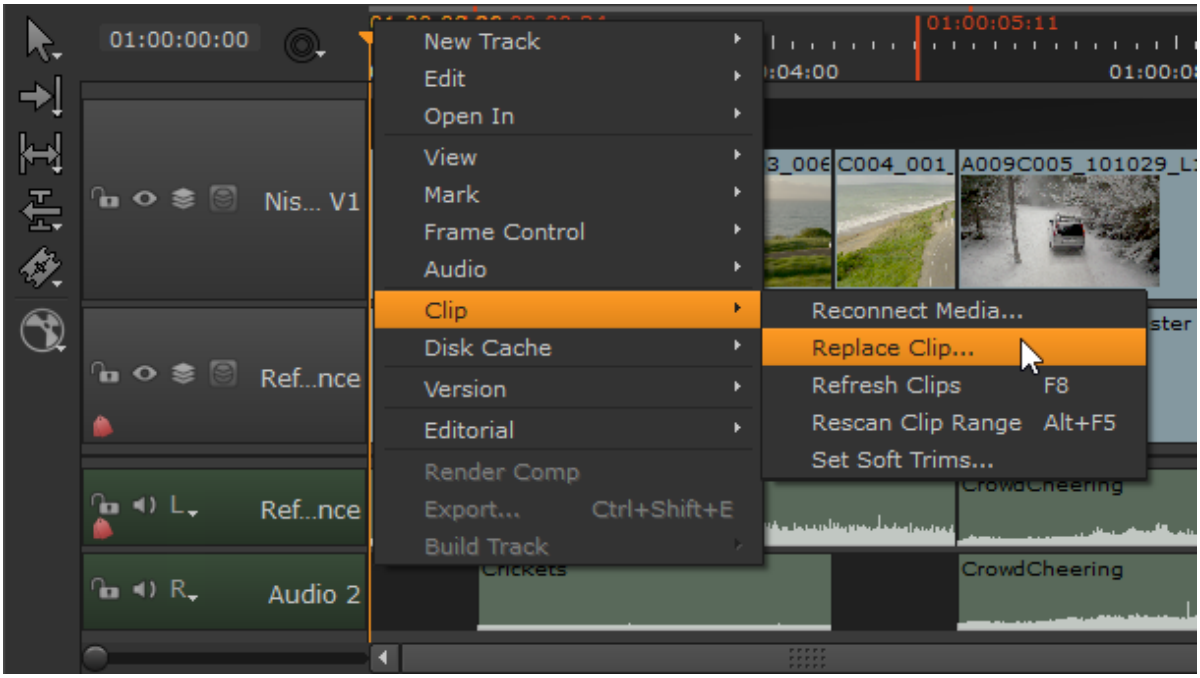
When you select the checkbox next to each option, the current shot will be transformed.

It is possible to perform both of these transformations at once. To revert back, you can simply select the checkbox again to disable the transformation.



Refreshing and Replacing Shots

During the post process, media inevitably changes location or form. Nuke Studio can reload or replace your media using the refresh, rescan, reconnect, and replace functions.



Though all four options deal with reloading shots, each has a particular use dependent on context:

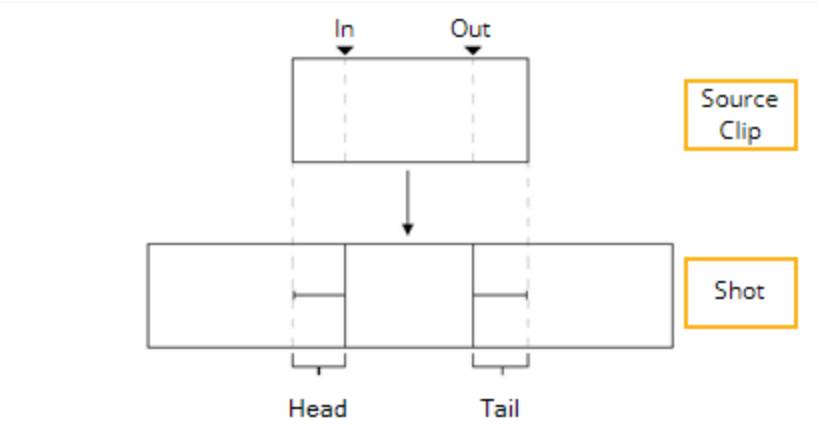
- **Reconnect Media** - allows you to redirect the file path when the source file location changes.
- **Replace Clip** - replaces the selected shot with a specified source clip. Nuke Studio assumes that any source clip you choose is acceptable, regardless of timecode.
- **Refresh Clips (F8)** - allows you to reload the shot when the source file location has not changed, such as when work has been done on the clip offline. Selecting refresh only refreshes the clip's current frame range.
- **Rescan Clip Range (Alt+F5)** - similar to **Refresh Clips**, above, but rescan also checks for additional frames that may have been added to the source file and adds them to the shot's frame range.
- **Set Soft Trims** - sets the files handles on the selected clip(s). SeeSetting Soft Trims for more information.

Setting Soft Trims

Soft Trims limit the handles on shots to a pre-defined amount, simulating In and Out points on the source clips, allowing you to use other timeline tools on the shots such as **Slip Clip** and **Slide Clip**.

To set Soft Trims on a shot(s):

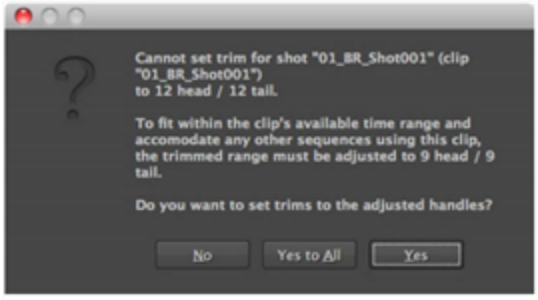
1. Select the shot(s) on the timeline.
2. Right-click and select **Clip** > **Set Soft Trims**.
   The **Set Soft Trims** dialog displays.
3. Set the number of frames to add to the head and tail of each shot:



- **Use full available range** - sets the handles to the full extent of the source clip frame range.
- **Use Frames** - adds the specified number of frames to the head and tail of the shot(s).

4. Click **OK** to add the specified number of handles.

   If the handles requested are not within the available frame range, a warning dialog displays with a suitable correction for each selected shot.



   Click **Yes** to accept, or **No** to abort the operation.

Note: With shots used in multiple sequences, click **Yes to All** to accept the correction in all instances.

Enabling and Disabling Shots

You can temporarily enable or disable tracks and shots on the timeline to selectively view your media without removing shot(s), for example if you wanted to view to lower level video tracks within a timeline.

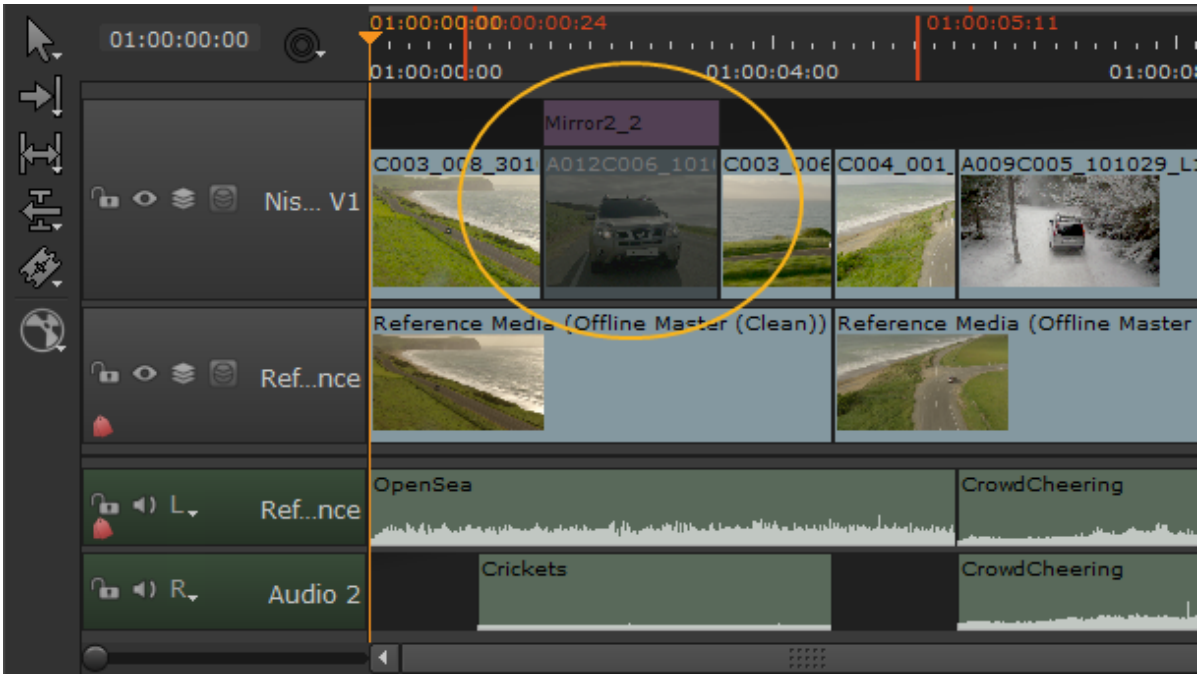
To enable or disable a track or shot(s):

1. Select the item(s) you want to enable or disable.

2. Right-click on a highlighted item and select **Editorial** > **Disable Track** or **Disable Items** to disable the selection.

Tip: You can also use the **D** keyboard shortcut to disable or enable your selection.

Disabled items appear gray, and are effectively removed from the timeline.



3. Right-click the item and select **Enable Clip**, or press **D** again, to re-enable the clip.

Adding Transitions Between Shots and Audio Clips

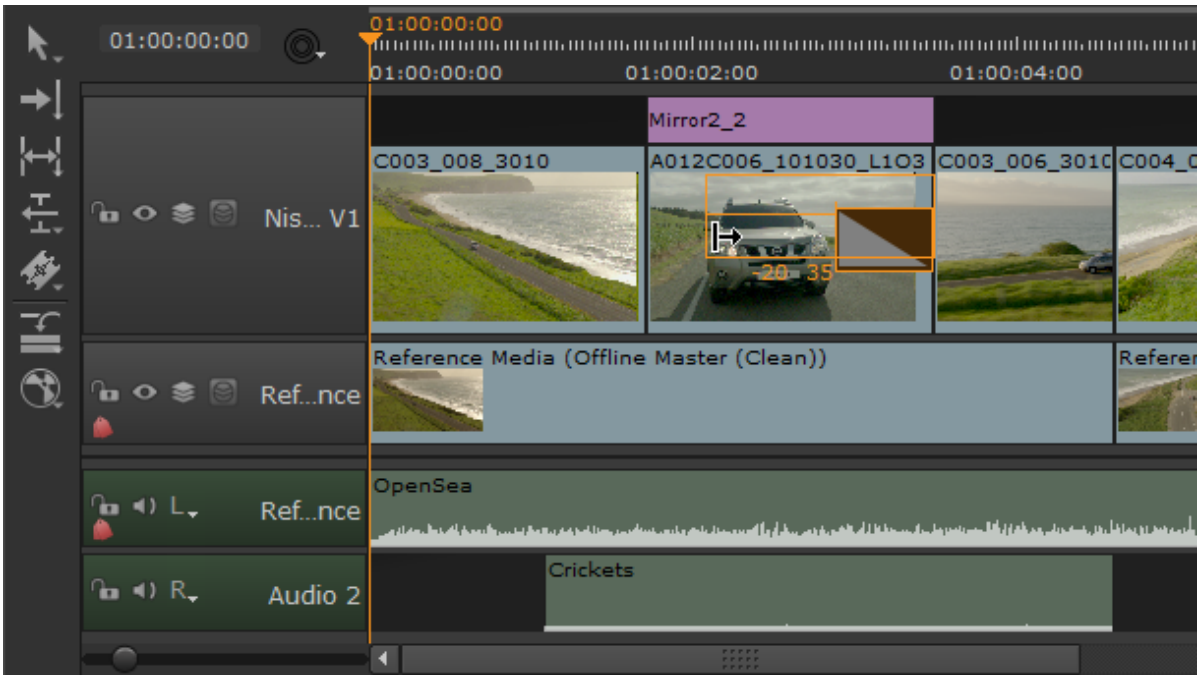
Nuke Studio supports video and audio fade transitions as well as dissolves between shots on the same track. Transitions come in three flavors:

- **Fade in** - fades in from black on a single shot or fades in audio.
- **Fade out** - fades out to black on a single shot or fades out audio.
- **Dissolve** / **Audio Crossfade** - fades out from one shot or audio clip and into the next, by merging frames or audio.

Tip: Once a transition is in place, it can be nudged in the same way as an edit using the **,** (comma) and **.** (period) keyboard shortcuts, providing the required handles exist.

To add a fade transition:

1. Right-click the target shot or audio clip and select **Editorial** > **Add Transition** > **Fade In**, **Fade Out**, **Audio Fade In** or **Audio Fade Out** to add the fade icon.
2. Adjust the fade by dragging the fade icon using the **Multi Tool** or **Move/Trim** tool.



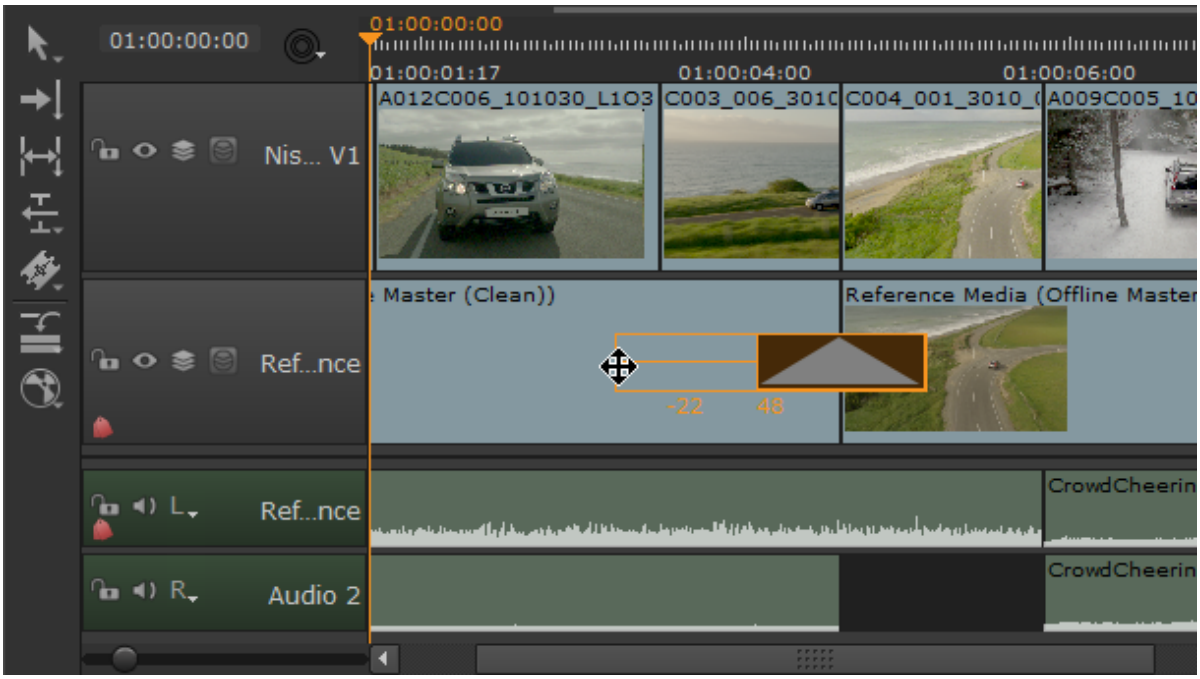
To add a dissolve transition:

Note: You can only add dissolves between shots when they're on the same track and have sufficient handles available on both sides of the transition.

1. Select the **Multi Tool** or **Roll Edit** tool and hover the mouse pointer over an edit between two shots or audio clips.
2. **Tip:** Clicking and holding the edit point displays available handles as a red overlay.

3. Right-click and select **Editorial** > **Add Transition** > **Dissolve** or **Audio Crossfade**, or use the **Ctrl**/**Cmd**+**T** keyboard shortcuts, to add the dissolve icon to the edit.

4. Adjust either side of the dissolve by dragging the icon, in a similar way to using the **Multi Tool** or **Move/Trim** tool.



Non-Linear Transitions

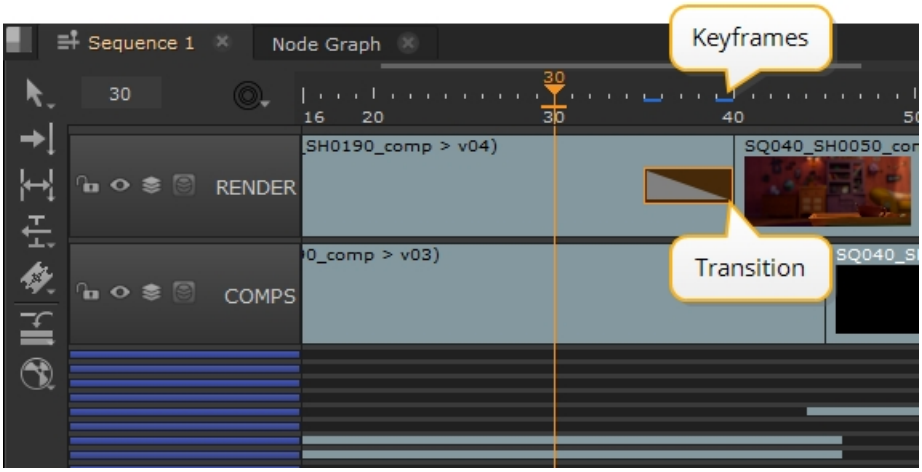
Transitions in Nuke Studio are linear by default, meaning that the fade or dissolve happens at the same rate throughout the transition. For example, a linear fade out on the last five frames of a shot would appear as follows:



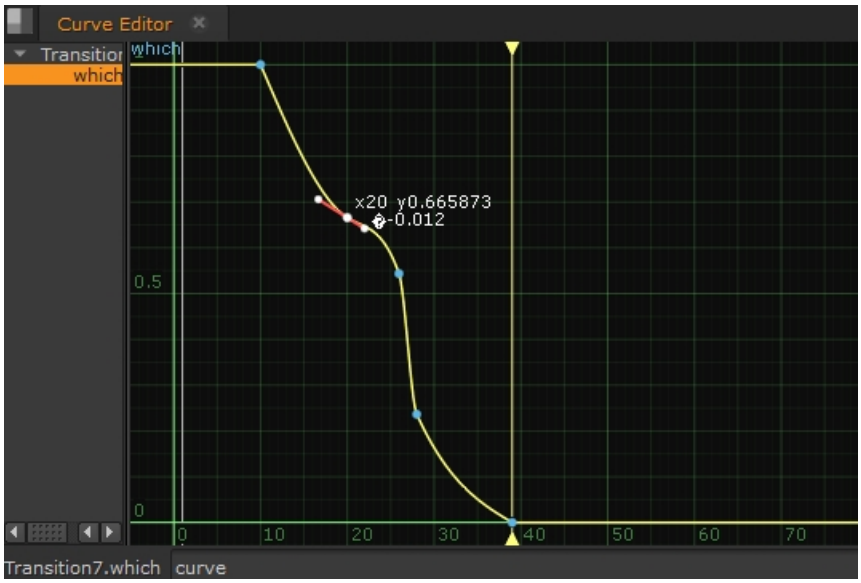
Non-linear transitions allow you to vary the speed at which the transition happens, giving you more creative control. For example, you might want a fade out to be slow and then speed up towards the end of the shot.



When you add a transition, keyframes are added to the timeline as blue chips, so you can see where the transition begins and ends.

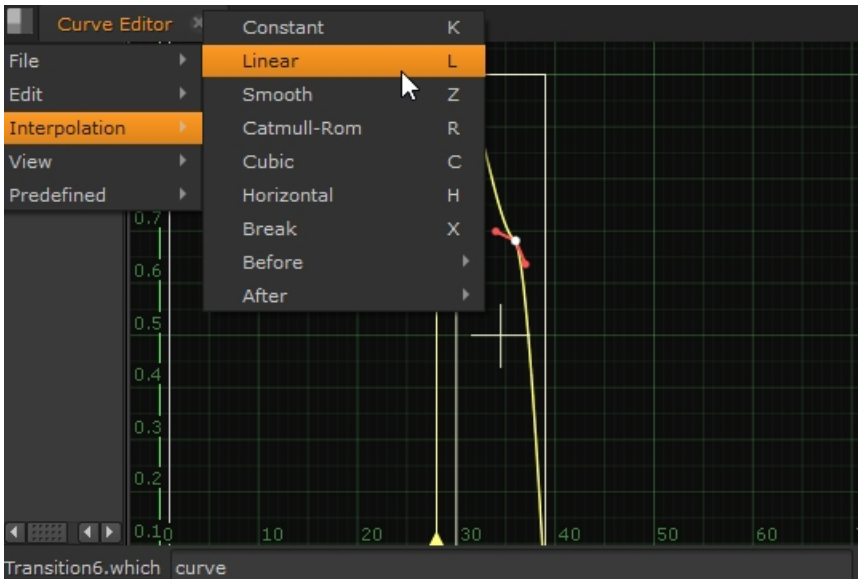


You can adjust the keyframes in the Curve Editor to achieve the required transition speed. You can adjust the keyframes manually by dragging them to different frames and adjusting the curve handles to change the speed of transition. You can also add keyframes to edit the curve with even more precision by pressing **Ctrl**/**Cmd**+**Alt** and clicking on the curve.



The Curve Editor also has some preset curves that you can apply to keyframes:

1. Select all the keyframes by pressing **Ctrl**/**Cmd**+**A**.
2. Right-click in the Curve Editor and select **Interpolation** to display the available presets.



See Using the Curve Editor for more information on the Curve Editor and keyframes.

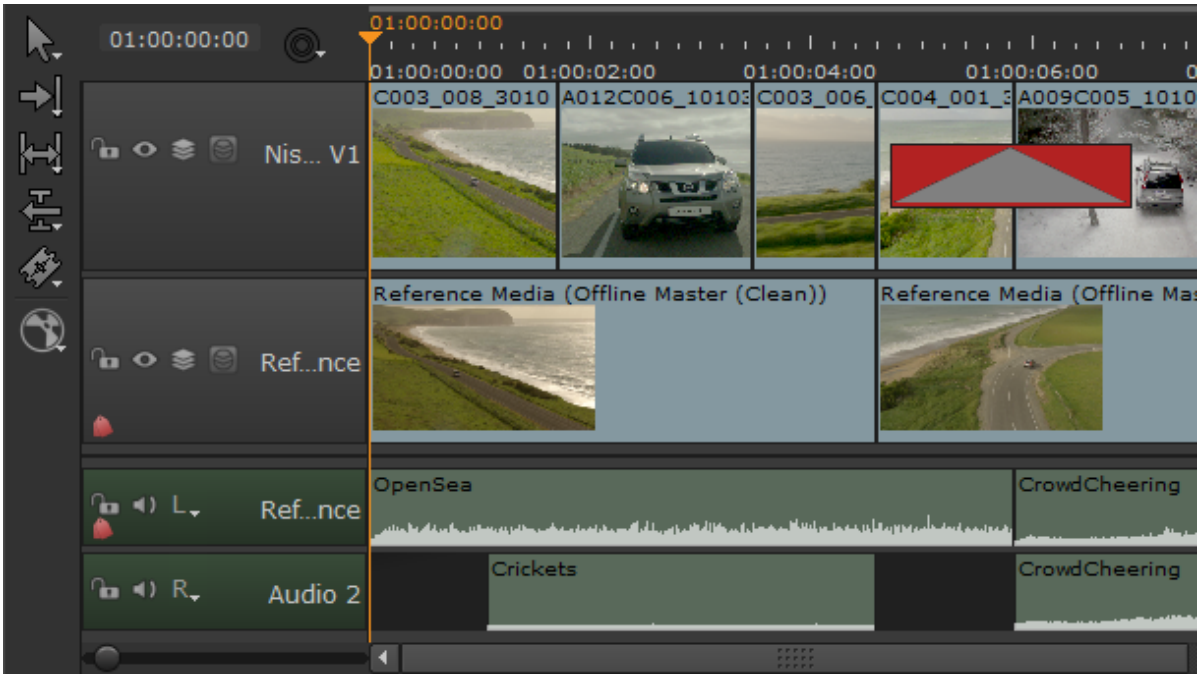
Invalid Transitions

Transitions are controlled in a similar way to shots, in that you can drag-and-drop them, but with the following restrictions:

- A fade can not be dragged past the ends of the shot it's attached to, and if the item is deleted, the fade is deleted with it.
- Dissolve ends can not be dragged past the ends of the shots they are attached to, and if both items are deleted, then the dissolve is also deleted.

  If only one of the shots linked by the dissolve is deleted, the transition remains so that another item can be placed on the other side.

Invalid transitions are colored red on the timeline. In most cases, adjusting the length of the transition should be enough to correct the error.



Retiming Shots

In addition to transitions, Nuke Studio supports constant retimes on shots. Decreasing the speed of a shot causes frames to play more than once, whereas increasing the speed skips frames.

Note: Audio is not currently supported for retimes and is automatically muted to avoid playback problems.

To retime shots using the **Speed** column in the spreadsheet:

1. Select the required event(s) in the spreadsheet view.
2. Click the cog icon and select the required **Retime method**:
   - **Keep source duration** - the shot length is altered on the timeline depending on the retime applied.

     For example, retiming a shot to 50% renders frames 1, 1, 2, 2, 3, 3, 4, 4, and so on in the Viewer, and as a result, the item's length is doubled on the timeline.

     Retiming a shot to 200% renders frames 1, 3, 5, 7, and so on in the Viewer, but the item's length is halved on the timeline.
   - **Keep timeline duration** - the shot length on the timeline is maintained regardless of the retime applied.

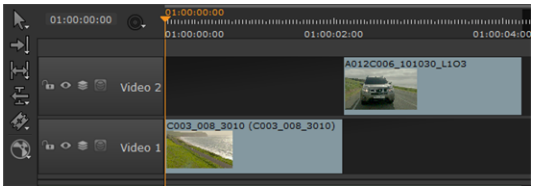
For example, retiming a shot to 50% renders frames 1, 1, 2, 2, 3, 3, 4, 4, and so on in the Viewer, but the item's length on the timeline remains the same, effectively removing the second half of the item.

Retiming a shot to 200% renders frames 1, 3, 5, 7, and so on in the Viewer, but the item's length on the timeline remains the same. If no extra frames are available from the source, the item is filled with black frames.

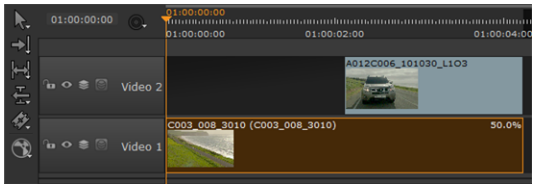
3. Double-click the **Speed** column and enter the retime value.

The following example shows a shot and the results of 50% and 200% retimes with the **Keep source duration** and **Keep timeline duration** methods selected.

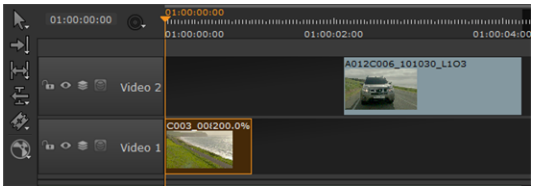
Notice that the **Keep timeline duration** method doesn't change the length of the shot on the timeline and inserts blank filler frames on the 200% retime?



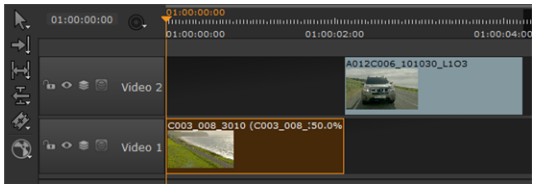
Original clip



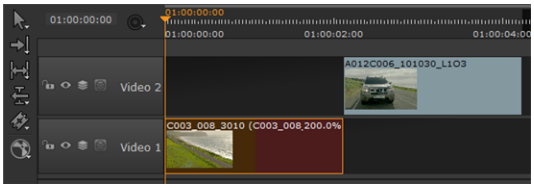
50% **Source** retime



200% **Source** retime



50% **Destination** retime

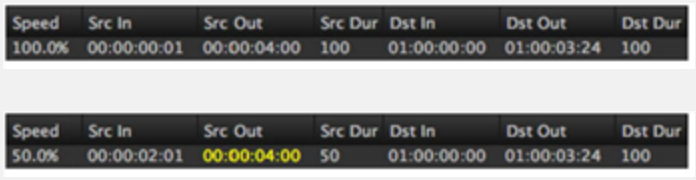


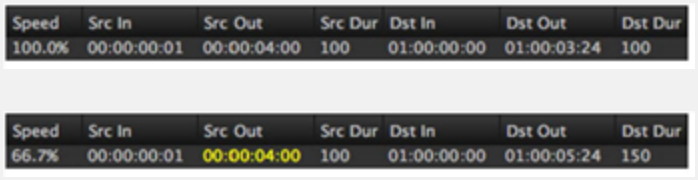
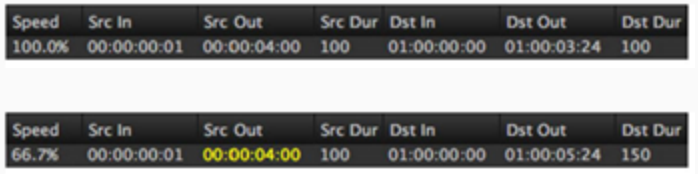
200% **Destination** retime

You can also retime shots using the **Src**, **Dst**, and **Duration** columns of the spreadsheet, though the calculation method depends on the **Time Edit Behaviors** applied.

1. Select the event(s) in the spreadsheet view.
2. Click the cog icon and select the required **Time Edit Behaviors**:

Modify	Using	Result
Src In	Retime	Adjusts the event's **Src In** and retimes the remaining frames to maintain **Dst**

Modify	Using	Result
		Duration. Before and after a 2 second **Src In** increase: 
Src Out	Retime	Adjusts the event's **Src Out** and applies a retime to maintain **Dst Duration**. Before and after a 2 second **Src Out** increase: 
Src Dur	Retime	Adjusts the event's **Src Dur** and **Src Out**, and applies a retime to maintain **Dst Duration**. Before and after a 50 frame **Src Dur** increase: 
Dst In	Retime	Adjusts the event's **Dst In** and retimes the remaining frames to maintain the relationship between **Dst In** and **Out**. Before and after a 2 second **Dst In** increase: 
Dst Out	Retime	Adjusts the event's **Dst Out** and retimes the remaining frames to maintain

Modify	Using	Result
		the relationship between **Dst Out** and **In**. Before and after a 2 second **Dst Out** increase: 
Dst Dur	Retime	Adjusts the event's **Dst Dur** and **Dst Out**, and applies a retime to accommodate the new **Dst Duration**. Before and after a 50 frame **Dst Dur** increase: 

3. Adjust the values as required to retime the shot(s) by the specified amount.



Note: Any source or destination field highlighted in yellow indicates that the entry has been rounded down for display purposes.

To retime a shot using the **Timeline** menu:

1. Select the required shot(s) on the timeline.
2. Navigate to **Timeline** > **Retime**.
   The **Clip Speed** dialog displays.
3. Enter the required retime value as a percentage.
4. Select the required retime method using the dropdown:
   - **Keep timeline duration** - the shot length on the timeline is maintained regardless of the retime applied. When increasing speed, if no extra frames are available from the source, the shot is filled with black frames.
   - **Keep source duration** - the shot length is altered on the timeline depending on the retime applied. For example, a 200% retime halves the length of the item.

- **Anchor current frame** - the shot length on the timeline and the current frame's position are maintained after the retime. When increasing speed, if no extra frames are available from the source, the shot is filled with black frames.

5. Click **OK** to retime the shot(s).

Using Freeze Frames

The freeze frame feature enables you to create shots of any length using a single frame. To achieve this, the application takes the first frame of the shot and applies a 0% retime, which is reversible by applying a 100% retime.

To freeze frame shots:

1. Select the target item(s) on the timeline.
2. Right-click the item and select **Editorial** > **Make Freeze Frame**,

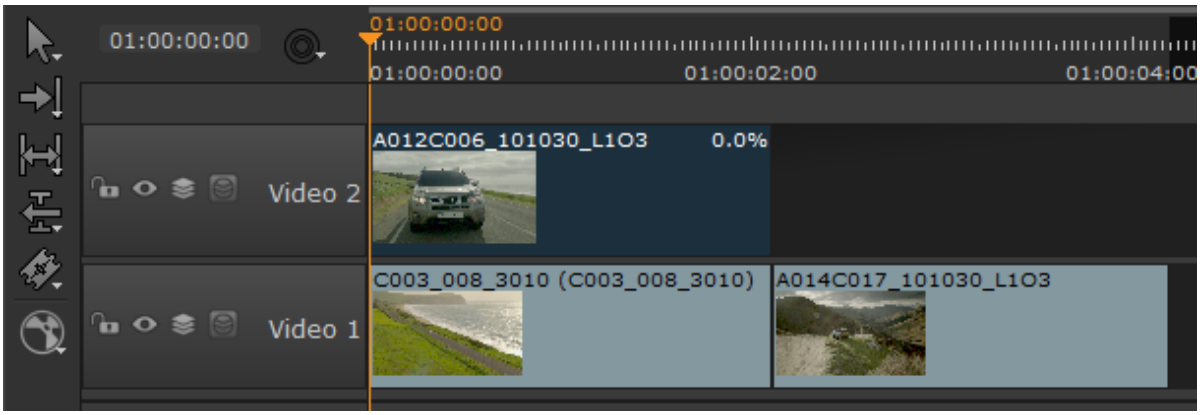
   OR

   Navigate to **Timeline** > **Make Freeze Frame**.



Note: Freeze Frames can also be created using the spreadsheet retime modes to modify **Src Dur** to 0, or make **Src In** and **Src Out** equal.

3. The selection is retimed to 0% and colored blue on the timeline for easy identification.



4. Use the **Multi Tool** or **Slip Clip** to set the freeze frame from the available range.
5. Drag the item's edit points, using the **Multi Tool** or **Move/Trim** as required, to set the length of the shot. There's no upper limit for the length of a freeze frame shot.

Blending Tracks on the Timeline

Nuke Studio allows you to perform merges between tracks in the timeline, for example overlaying a logo on a shot without heading into the Compositing environment. Tracks that are designated as blend tracks have a blue header in the timeline for convenience and are blended using a sub-set of Nuke's Merge node operations.

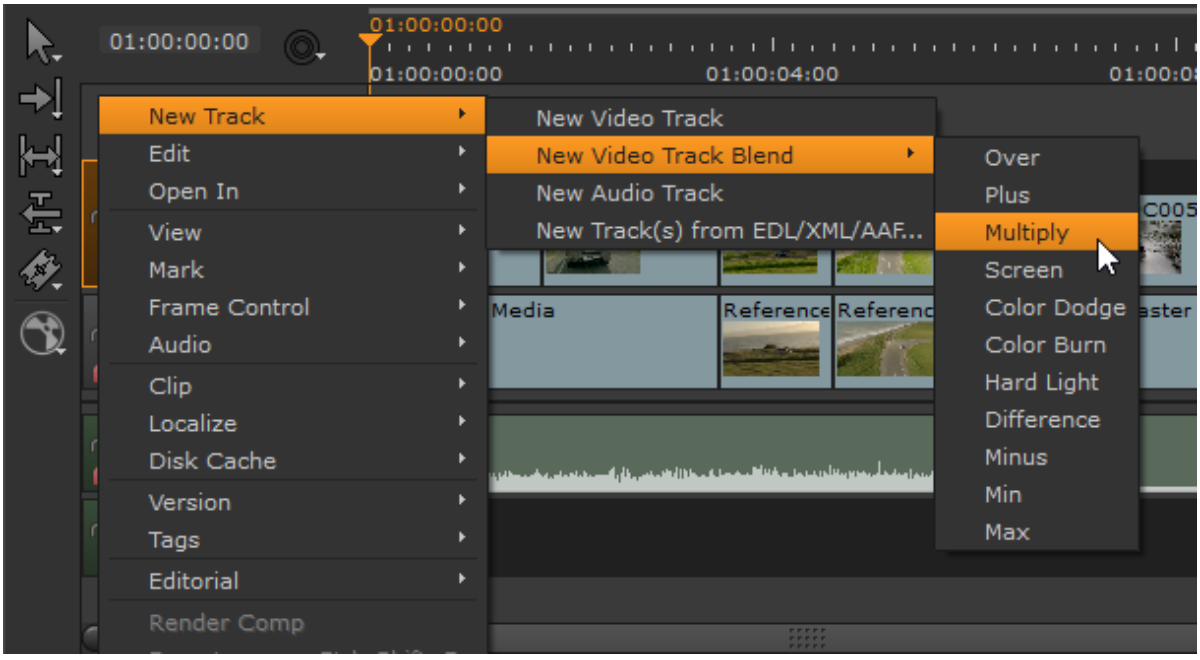
See the Nuke Online HelpMerge Operations for a full description of the available blend modes.

Note: The Viewer currently treats all alpha channels as premultiplied, which can result in the Viewer background being "added" to the image.

You can add soft effects to blended tracks as normal (see Adding Effects on the Timeline for more information) and blended tracks are included along with the shot in Export operations. See Exporting from Nuke Studio and Create Comp for more information.

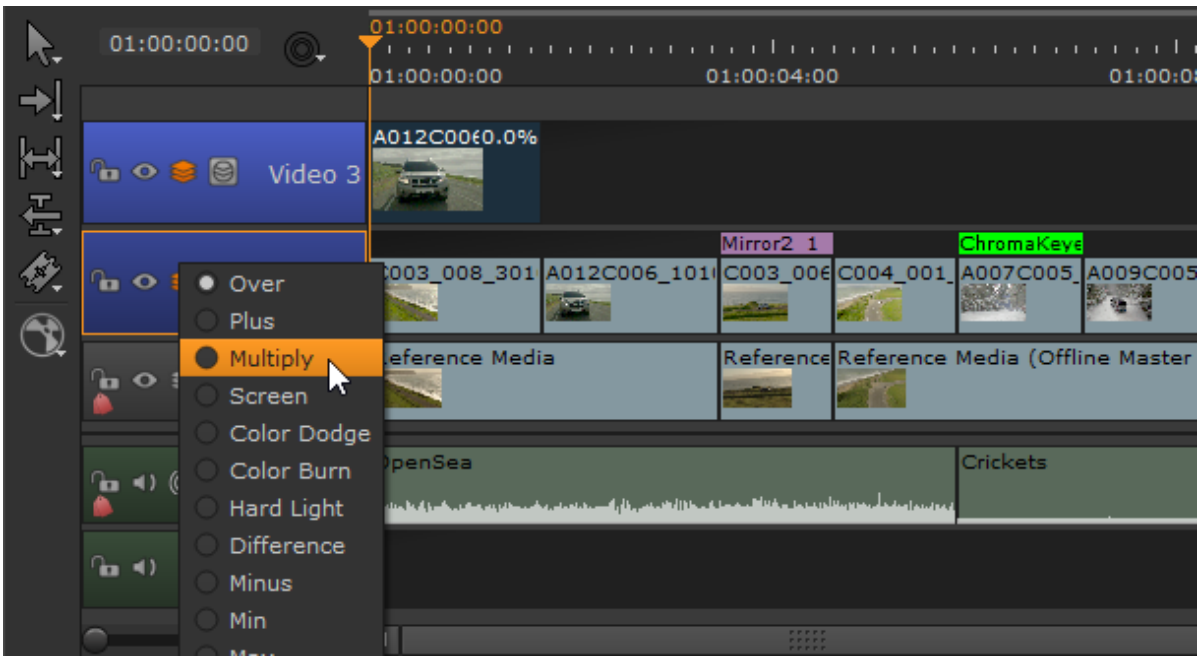
Adding New Blend Tracks

1. Right-click in the timeline, select **New Track** > **New Video Track Blend**, and then choose the blend mode to apply.



A new track is added at the top of the track stack, colored blue to indicate that it's going to be blended over the track below.

2. Add the required shot to the blend track as you would any other shot. See Adding Clips to the Timeline for more information.
3. Click and hold the **Blend** icon to select the blend mode.



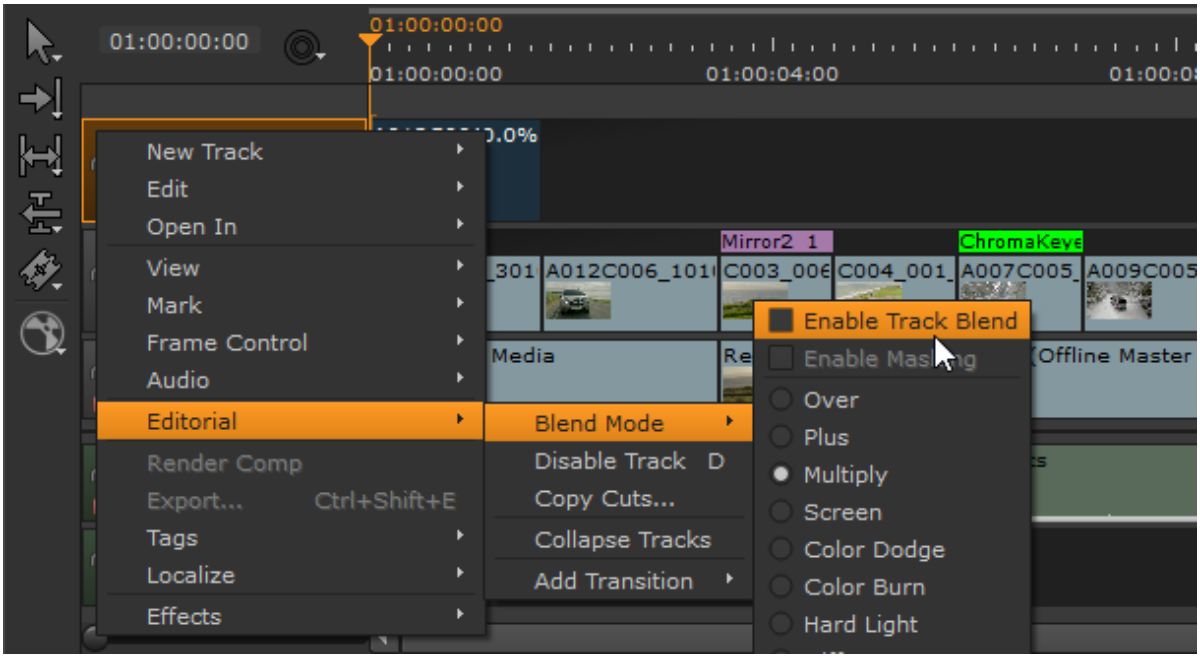
4. Click the **Blend** icon to toggle blending on and off.

Converting Tracks to Blend Tracks

1. Click the **Blend** icon to toggle blending on,

   OR

   Right-click in the header of the target track and select **Editorial** > **Blend Mode** > **Enable Track Blend**.



The selected track is converted into a blend track, colored blue to indicate that it's going to be blended over the track below.

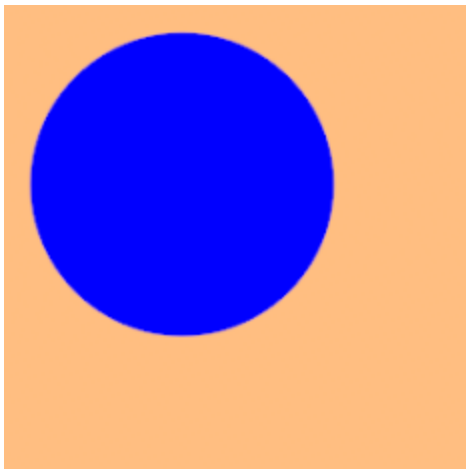
2. You can add shots to the blend track as you would any other shot. See Adding Clips to the Timeline for more information.

   The Viewer displays the higher track blended with the track below.

3. Click and hold the **Blend** icon to select the blend mode.

Masking Blended Tracks

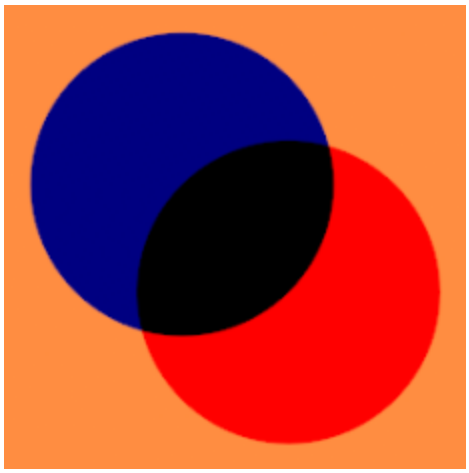
Masking limits the effect of the blend track to just those areas covered by the alpha channel in the blend image. For example, using the **Multiply** blend mode with masking disabled multiplies the background plate in non-alpha areas, which may not be the result you require.



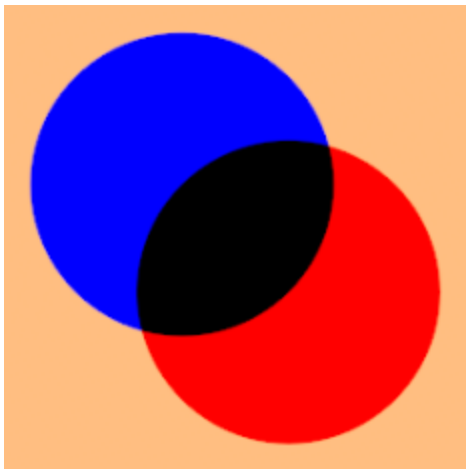
Background track **B**.



Blend track **A**.



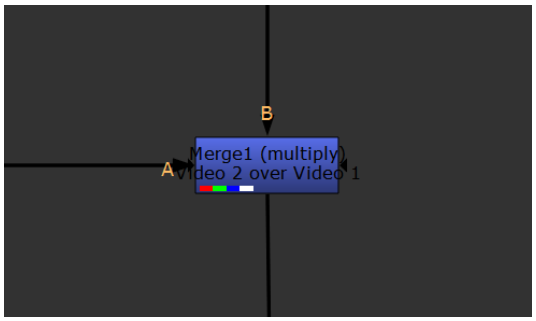
A multiplied by **B** with Masking disabled.



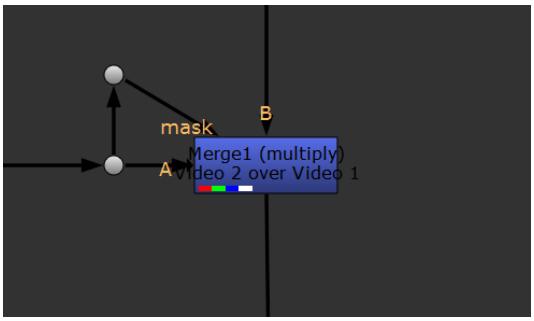
A multiplied by **B** with Masking enabled.

See the Nuke Online HelpMerge Operations for a full description of the available blend modes.

Click the Mask icon to toggle alpha masking on and off. The mask option also carries over into the Node Graph when you Create Comp for a masked blend operation:



Comp of A over B using the **multiply** merge operation with masking disabled.

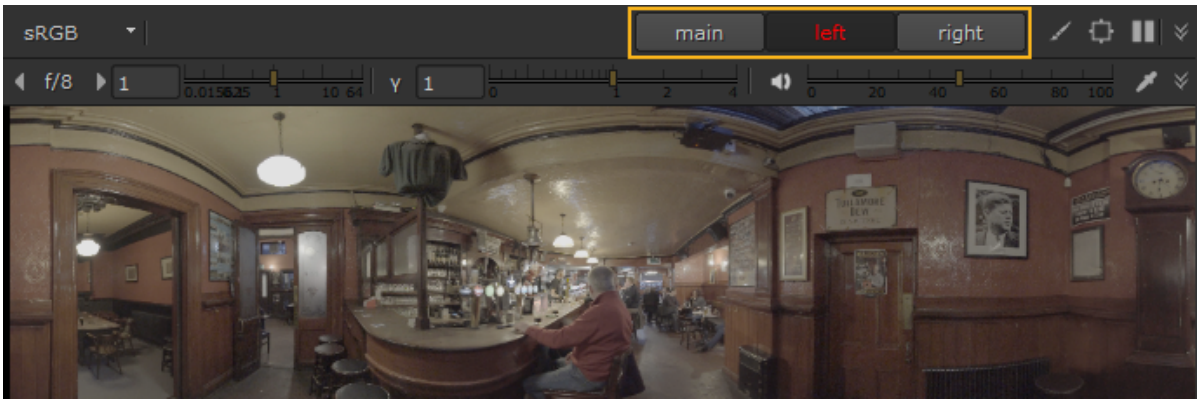


The same comp using the **multiply** merge operation, but with masking enabled.

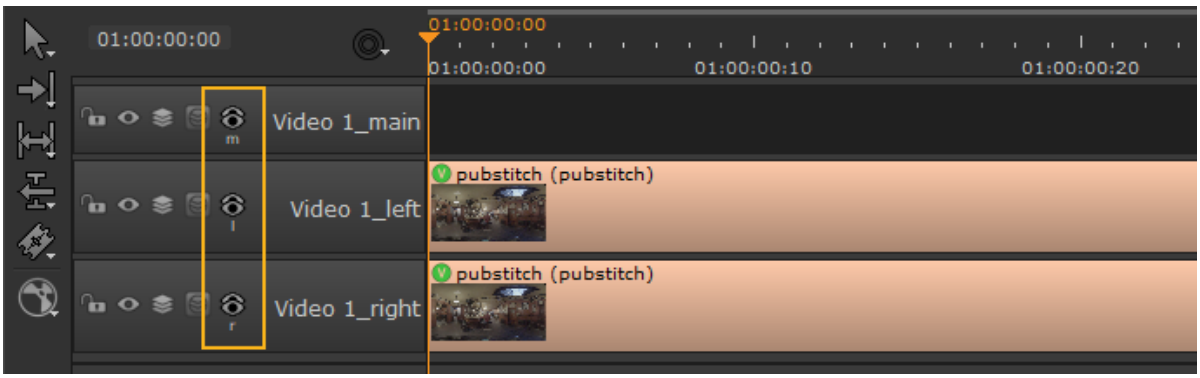
Stereoscopic and Multi-View Projects

Nuke Studio provides multi-view support for as many views as you need. The views do not have to be stereo pairs, but since that is the most obvious application, these pages mainly deal with stereoscopic projects. See Creating Views in a Project and Importing Source Clips for more information.

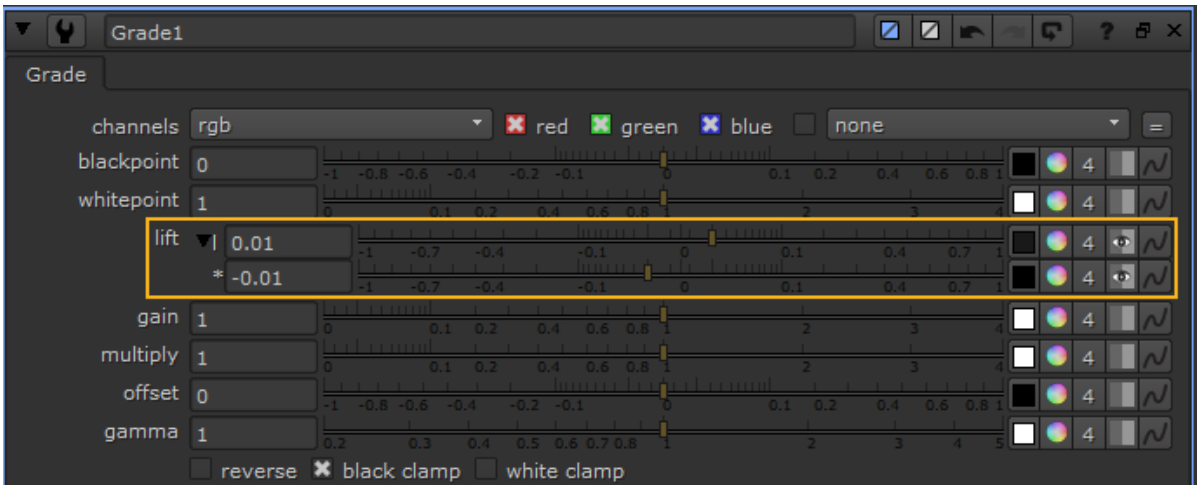
Existing views inside a project are managed in the Viewer, timeline, and in the **Properties** panel of most soft effects. In the Viewer, all views in the current project are represented by buttons that allow you to switch between views with a single click. See Displaying Views in the Viewer for more information.



The timeline employs a views button that allows you to switch between views per track. See Displaying Views in the Timeline for more information.



The **Properties** panel includes a split button for controls that support multiple views. Split controls only affect individual views. See Applying Changes to Selected Views for more information.



You can create comps or export multi-view shots and effects in a similar way to regular shots. See Exporting Multi-View Source Clips for more information.

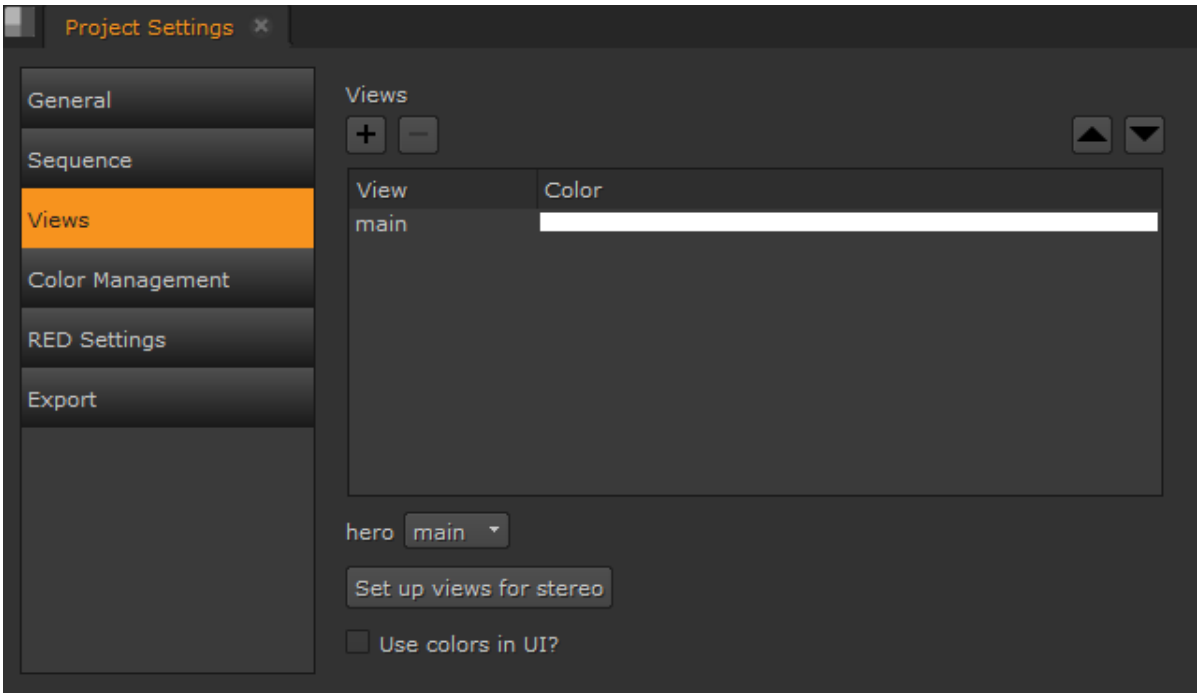
Creating Views in a Project

Views in Nuke Studio are managed in the **Project Settings**. Views can be processed separately or together and you can see the effect of your changes on each view. If you're working with multi-view source clips, such as **.exr** and **.sxr**, Nuke Studio offers to create views for you on import. See Importing Multi-View Clips for more information.

Creating and Managing Views

If you're working with single-view clips, it's a good idea to create the project views before you import your footage so that the views are assigned correctly. If you're working with multi-view clips, the views are set up automatically on import.

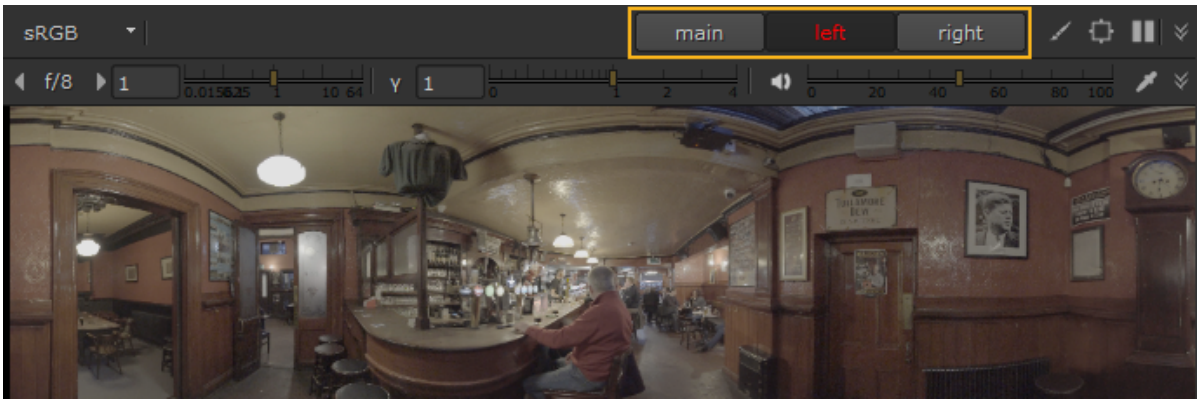
1. Select **Project** > **Edit Settings**.
2. Go to the **Views** sub-menu. The available views are listed in the **Views** panel.



3. If you want to remove the view called **main** and add views called **left** and **right**, click the **Set up views for stereo** button. The two views are assigned colors. To change the colors, double-click on the color field and select another color from the color picker that opens.
4. Enable **Use colors in UI** to apply the selected color for each view to the associated button above the Viewer.

   You can add and remove views using the **+** and **-** buttons or move views using the up and down arrows above the views panel.

   Each view has its own button above the Viewer controls.



Tip: If you decide that you only need the **main** view in the project, click **Set up Views for Mono** in the **Project Settings**.

See Importing Multi-View Clips for information on reading source clips in to Nuke Studio.

Importing Source Clips

Nuke Studio supports multi-view clips in two formats:

- **single-view** - formats such as **.dpx** and **.jpg**, with file names that Nuke Studio can interpret as multi-view using specific naming conventions. For example:

  `myStereo_comp1_left.####.dpx` and `myStereo_comp1_right.####.dpx`

  See Importing Single-View Clips for more information.
- **multi-view** - formats such as **.exr**, **.sxr**, and **.mov**, that support multiple views per file. For example:

  `myStereo_comp1.####.exr` containing two layers called **left** and **right**.



Note: Nuke Studio can assign any shot on the timeline to any view in the project, but stereo is the most common use case.

See Importing Multi-View Clips for more information.

Importing Single-View Clips

After setting up views in the **Project Settings**, Nuke Studio can interpret single-view clips as multi-view using specific naming conventions. These conventions can be applied to the file names themselves or to the directories in which they reside.

File Name Variables

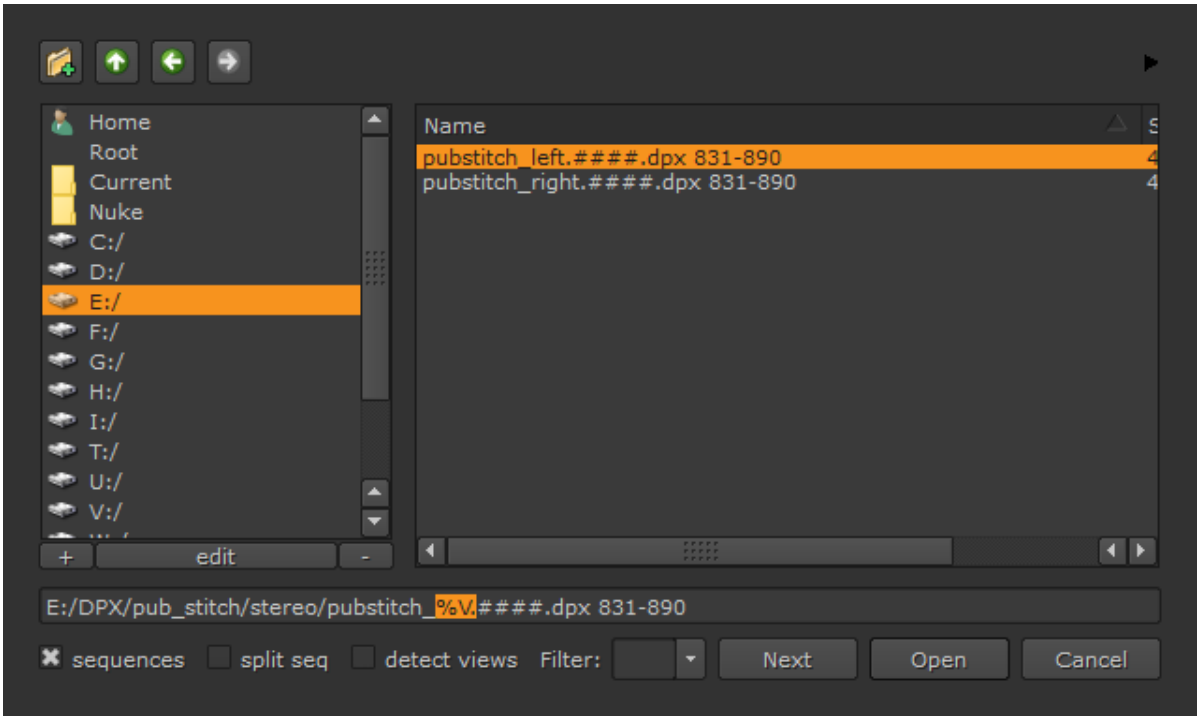
If the images you want to read in contain a view name or the initial letter of one (for example, **left**, **right**, **l** or **r**) in their file names, replace this with the variable **%V** or **%v** in the file name. Use **%V** to replace an entire view name (for example, **left** or **right**), and **%v** to replace an initial letter (for example, **l** or **r**). When a variable is used, Nuke Studio reads in the missing inputs and combines all inputs into a single output.



Note: You can enable **detect views** in the file browser to automatically substitute the **%V** or **%v** variable where possible.

To use the **%V** and **%v** variables manually:

1. Click **File** > **Import File(s)** or press **Ctrl**/**Cmd**+**I** to display the file browser.
2. Locate a single-view file, for example **pubstitch.left.####.dpx**
3. Replace the view name with the **%V** variable, continuing the example **pubstitch.%V.####.dpx**

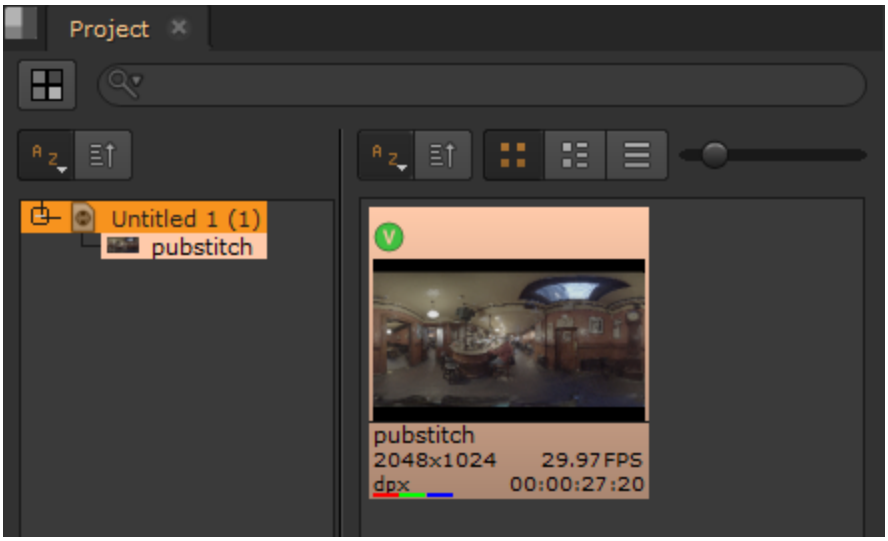


4. Click **Open**.

   Nuke Studio reads in both **pubstitch.left.####.dpx** and **pubstitch.right.####.dpx** with the same Read node, provided that views called **left** and **right** exist in the **Project Settings**.

Note: Mac and Linux operating systems can be case-sensitive or case-insensitive. If your OS is case-sensitive, you'll need to make sure you use the correct case when naming your left and right views, as the **%v** variable can only retrieve the case used in the view name.

Both input images are combined into a single source clip, marked with a icon, which can display any of the combined views.



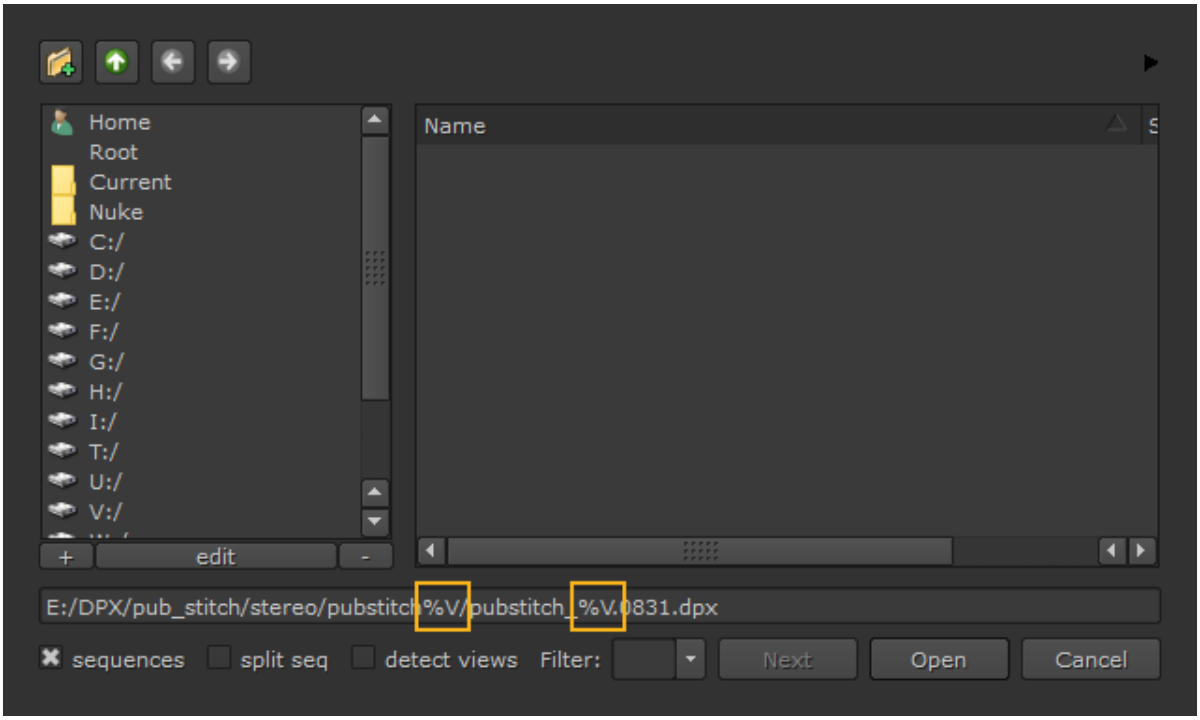
See Displaying Views in the Viewer and Displaying Views in the Timeline for more information.

Directory Name Variables

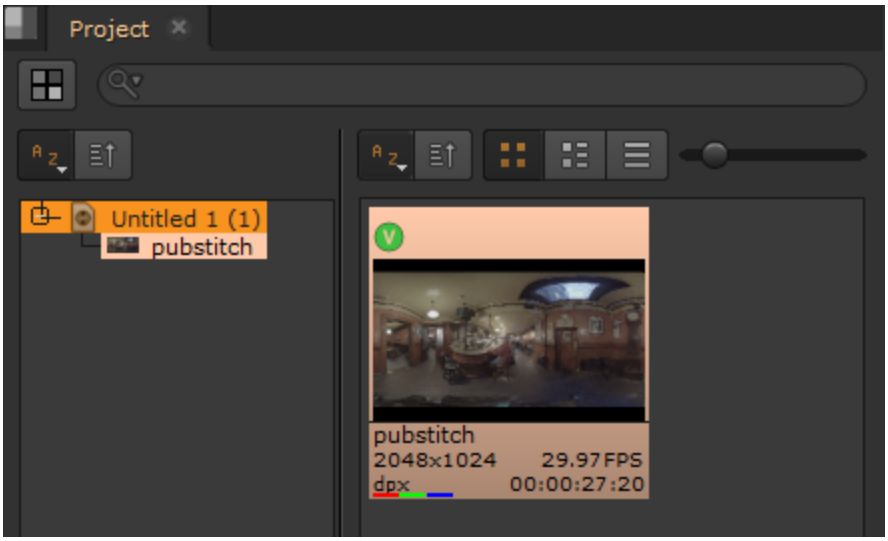
You can also use the **%V** and **%v** variables at a directory level. For example, if you have set up views called **cam3**, **left**, and **right**, and you have the following directories and files:

```
stereo/pubstitchcam3/pubstitch_cam3.####.dpx
stereo/pubstitchleft/pubstitch_left.####.dpx
stereo/pubstitchright/pubstitch_right.####.dpx
```

If you read in **pubstitch_cam3.####.dpx** and change the file path to **stereo/pubstitch%V/pubstitch_%V.####.dpx**, all three inputs are read in with the same Read node, providing that the **cam3**, **left**, and **right** views exist in the **Project Settings**.



All three input images are combined into a single source clip, marked with a V icon, which can display any of the combined views.



See Displaying Views in the Viewer and Displaying Views in the Timeline for more information.

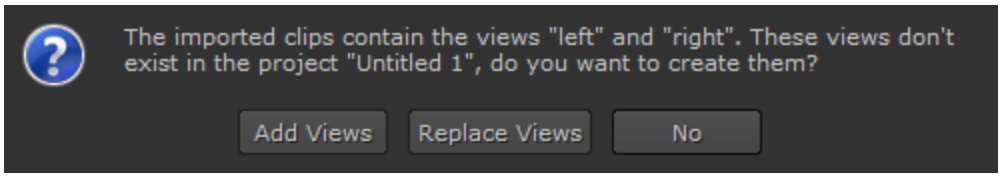
Importing Multi-View Clips

Nuke offers to automatically create views when you import single files containing multiple views, such as **.exr** and **.sxr**, unless they already exist. For multiple files, views must be created manually. See Importing

Source Clips for more information.

1. Click **File** > **Import File(s)** or press **Ctrl/Cmd+I** to display the file browser.
2. Locate a multi-view file, for example **pubstitch_stereo.####.exr**
3. Click **Open**.

   The **Create missing views?** dialog displays.



4. Click **Add Views**, **Replace Views**, or **No**:
   - **Add Views** - add the views in the incoming clip to those that exist in the project.
   - **Replace Views** - replace all existing project views with those in the incoming clip.
   - **No** - import the clip and display only the first view in the file, retaining any existing views in the project.

   You can now access the views in your project. See Displaying Views in the Viewer and Displaying Views in the Timeline for more information.

Multi-View QuickTime Files

Multi-view **.mov** files only display one view by default. To enable all views in a multi-view **.mov** file:

1. Double-click the **.mov** to display its **Properties** panel.
2. Disable **First track only**.

   You'll notice that the **.mov** in the bin is now marked with V to denote multiple views.

   You can now access the views in your project. See Displaying Views in the Viewer and Displaying Views in the Timeline for more information.

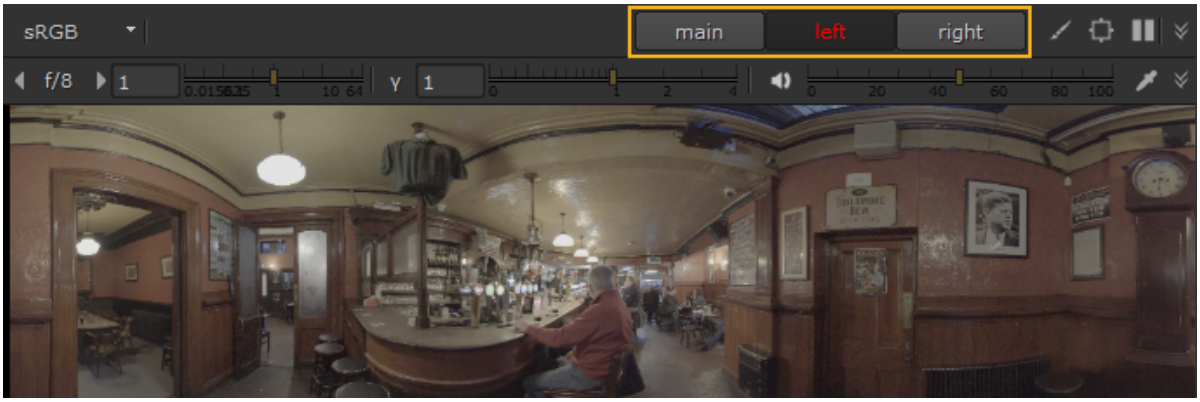
Displaying Views in the Viewer

You can only display the views that exist in your **Project Settings**. To see a list of these views, or to add or delete views, select **Project** > **Edit Settings** and go to the **Views** tab. For more information, see Creating Views in a Project.

Displaying a Particular View

1. Double-click the clip or sequence to load it in the timeline Viewer.
2. On top of the Viewer controls, click the view to display. In the example, **main**, **left** or **right**.

Tip: You can also press the **;** (semicolon) and **'** (forward single quote) keys to move between different views in the Viewer.



Note: Nuke Studio lists the views in **.exr** files in the order they appear in the clip's header, so a view named '**left**' may not always be the first view displayed above the Viewer.
If your views do not appear in the correct order, you can rearrange them in the **Project** > **Edit Settings** > **Views** tab. See Creating Views in a Project for more information.

Displaying Two Views Next to Each Other

1. Right-click in the Viewer and select the **Stereo Modes** menu.
2. Select one of the following options:
   - **Side by Side** - displays the views side by side at the correct aspect ratio, and adds selection controls above the Viewer.
   - **Squeezed Side by Side** - displays the views side by side and squeezed to fit the format horizontally, and adds selection controls above the Viewer.
   - **Squeezed Above by Below** - displays the views above and below each other and squeezed to fit the format vertically, and adds selection controls above the Viewer.

Displaying a Blend Between Two Views

1. Right-click in the Viewer and select the **Stereo Modes** menu.
2. Select one of the following options:
   - **Interlace H** - displays the views interlaced horizontally, and adds selection controls above the Viewer.
   - **Interlace V** - displays the views interlaced vertically, and adds selection controls above the Viewer.
   - **Checkerboard** - displays the views using an alternating checkerboard pattern (one pixel from left and one pixel from right), and adds selection controls above the Viewer.
   - **Anaglyph** - displays the views simultaneously using a red hue for left and green hue for right, and adds selection controls above the Viewer.
   - **Flicker** - displays both views alternately, and adds selection controls above the Viewer.

Displaying OpenGL Stereo in Timeline Viewers

The Viewer **OpenGL Stereo** mode allows you to see both views at once on a 3D monitor for review purposes. OpenGL Stereo is only supported on NVIDIA Quadro series GPUs and AMD Radeon Pro series GPUs onWindowsand Linux OS.



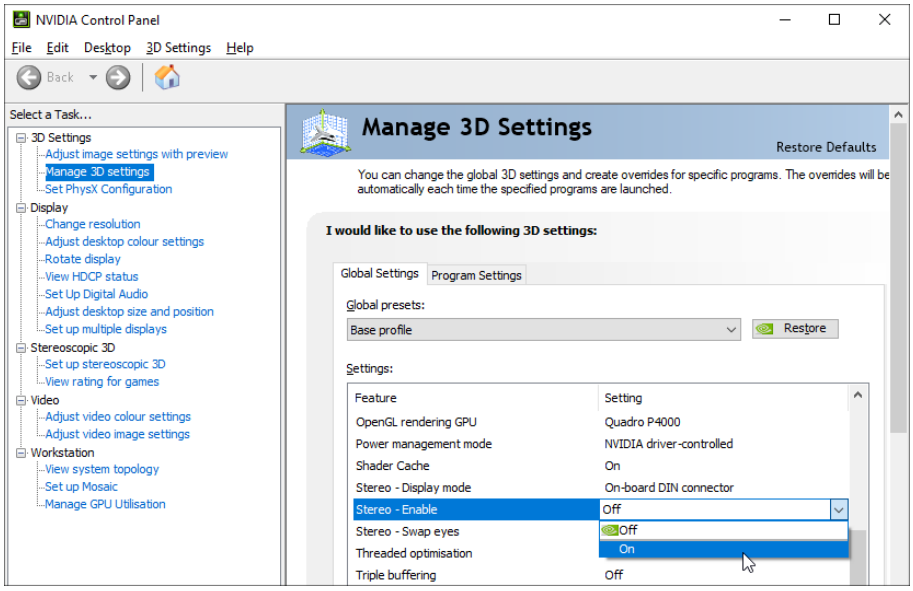
Note: OpenGL Stereo mode is not supported on Mac due to limitations in macOS and not supported with AMD GPUs on Linux.

Enabling OpenGL Stereo Output

Windows

To enable NVIDIA GPU stereo output:

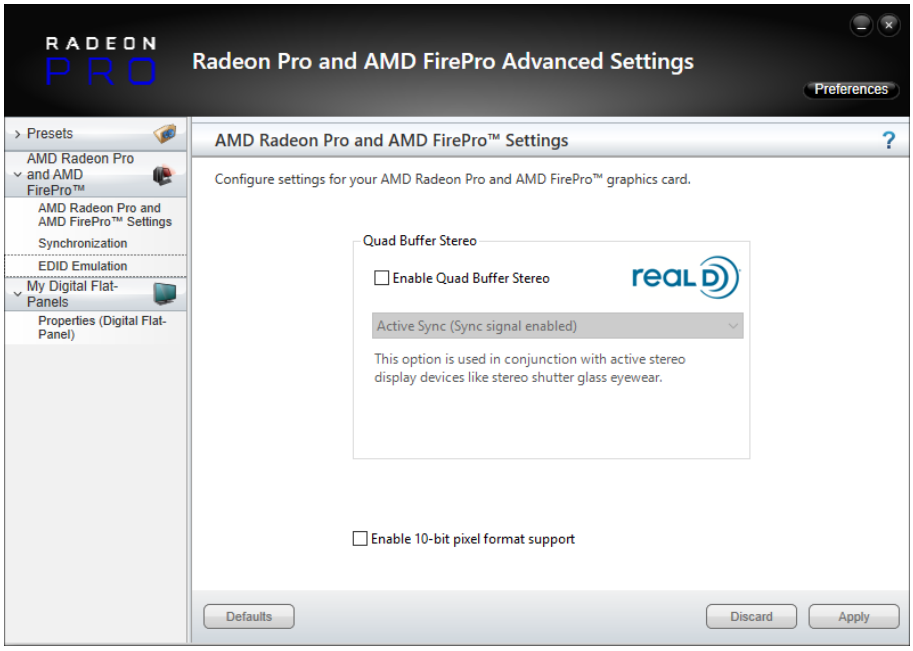
1. Right-click on the desktop and select NVIDIA Control Panel.
2. Navigate to **3D Settings** > **Manage 3D Settings** > **Stereo - Enable**.



3. Proceed to Switching to OpenGL Stereo Output.

To enable AMD GPU stereo output:

1. Double-click the AMD taskbar icon and select **Advanced Settings**.
2. Navigate to the AMD Pro Settings and check **Enable Quad Buffer Stereo**.



3. Select either **Auto-Stereo (Horizontal Interleaved)** or **Auto-Stereo (Vertical Interleaved)** and click **Apply**.
4. Proceed to Switching to OpenGL Stereo Output.

Linux

To enable NVIDIA GPU stereo output:

1. Open a command prompt and enter:
   `nvidia-xconfig --stereo=3`

Tip: For more information on the **nvidia-xconfig** utility, please see the **man** page: `man nvidia-xconfig`

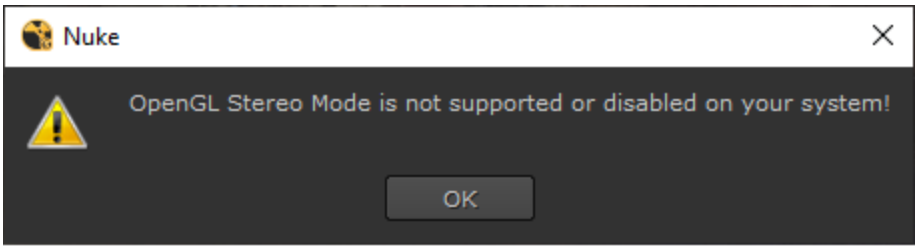
2. Proceed to Switching to OpenGL Stereo Output.



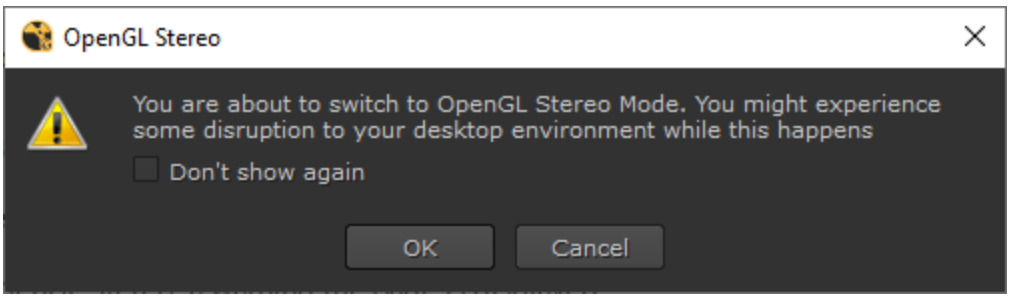
Note: **OpenGL Stereo** mode is not supported with AMD GPUs on Linux.

Switching to OpenGL Stereo Output

1. Right-click in the Viewer to display the context-sensitive menu.
2. Navigate to **Stereo Modes** and select **OpenGL Stereo**.
   If you select **OpenGL Stereo** mode before enabling your GPU settings, a warning is displayed.



The first time you select OpenGL Stereo, a warning message is displayed.





Tip: You can disable the warning by enabling **Don't show again** and clicking **OK**.

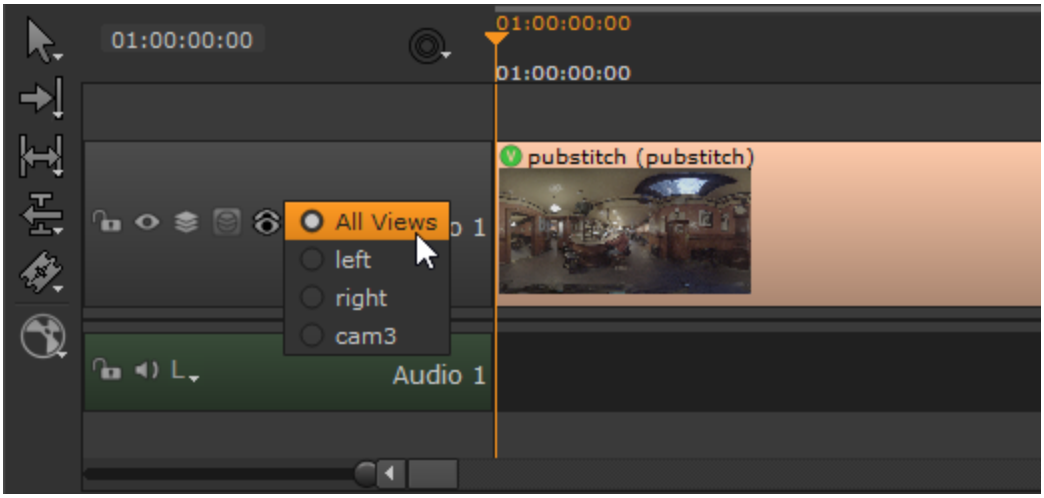
OpenGL Stereo is displayed in the Timeline Viewer.

Note: Switching to and from OpenGL Stereo mode causes playback to pause. Press play to resume playback.

Displaying Views in the Timeline

You can only display the views that exist in your **Project Settings**. To see a list of these views, or to add or delete views, select **Project** > **Edit Settings** and go to the **Views** tab. For more information, see Creating Views in a Project.

Adding a multi-view clip to the timeline groups all views into a single track. All the views in the clip are assigned a Viewer button. Click the assigned view icon to display the views available in the shot. **All Views** are visible by default.



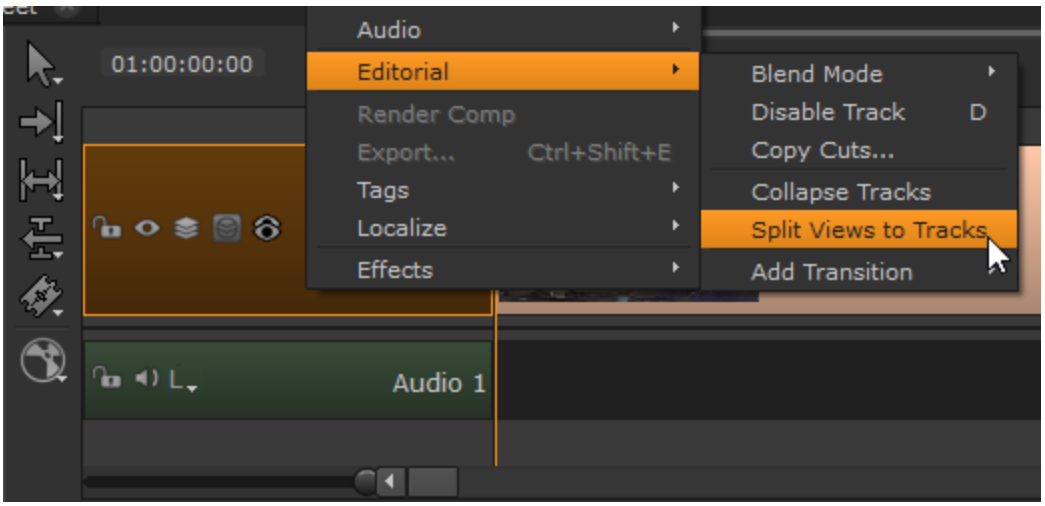
Selecting a particular view for a track in the timeline, such as **left**, means that only the left view is visible in the Viewer for that track.

Note: You can also import single-view files manually and then assign them views in the timeline individually, providing that the views exist in the **Project Settings**.

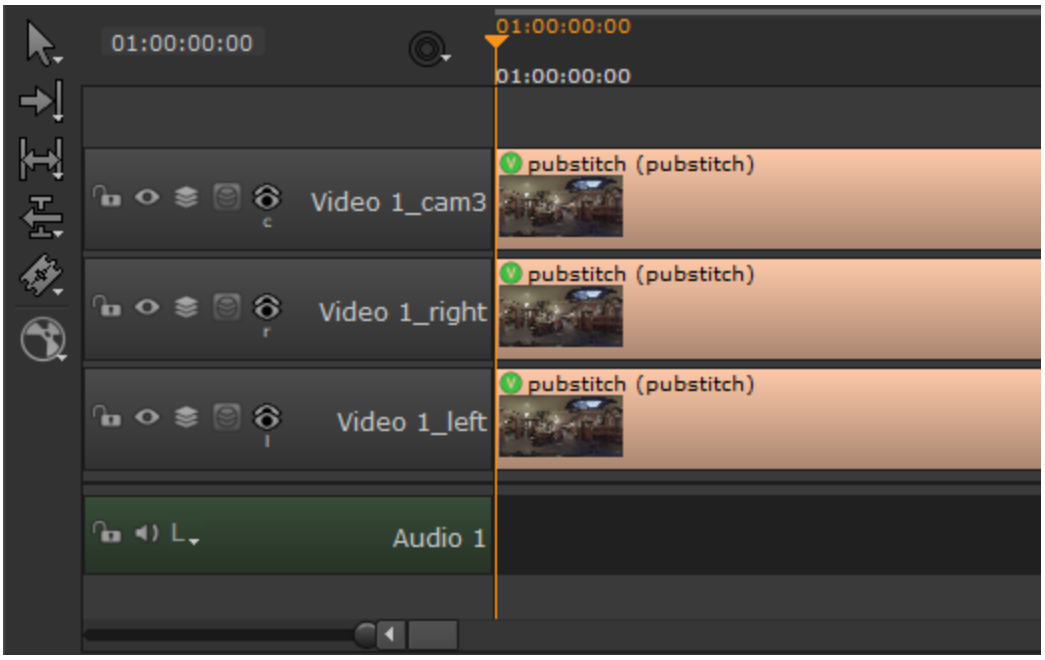
Splitting Views to Separate Tracks

Working with all views in a single track is not always convenient, such as when you need to apply different color corrections to stereo cameras. Nuke Studio includes a **Split Views to Tracks** option that quickly separates views into individual tracks. You can use **Split Views to Tracks** with grouped single- and multi-view shots.

1. Right-click a multi-view shot on the timeline and select **Editorial** > **Split Views to Tracks**.



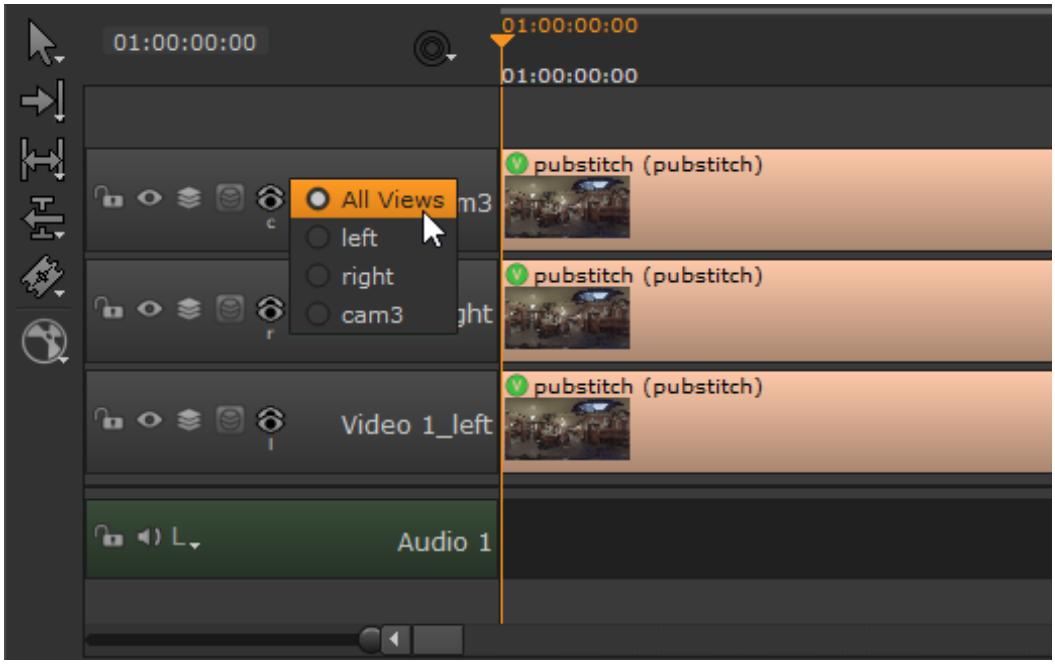
A separate track is created for each view in the group or file.



The new track names are suffixed with the view name, for example **Video1_left**, and the views are assigned appropriately.

Note: If a view exists in the Project Settings, but there's no corresponding view in the source files, empty placeholder tracks are added.

2. You can change the view assigned to a track by clicking the icon and selecting from the list of available views.



See Applying Changes to Selected Views for information on adding soft effects to different views.

Displaying Two Views Next to Each Other

1. Right-click in the Viewer and select the **Stereo Modes** menu.
2. Select on of the following options:
   - **Side by Side** - displays the views side by side at the correct aspect ratio, and adds selection controls above the Viewer.
   - **Squeezed Side by Side** - displays the views side by side and squeezed to fit the format horizontally, and adds selection controls above the Viewer.
   - **Squeezed Above by Below** - displays the views above and below each other and squeezed to fit the format vertically, and adds selection controls above the Viewer.

Displaying a Blend Between Two Views

1. Right-click in the Viewer and select the **Stereo Modes** menu.
2. Select on of the following options:
   - **Interlace H** - displays the views interlaced horizontally, and adds selection controls above the Viewer.

- **Interlace V** - displays the views interlaced vertically, and adds selection controls above the Viewer.
- **Checkerboard** - displays the views using an alternating checkerboard pattern (one pixel from left and one pixel from right), and adds selection controls above the Viewer.
- **Anaglyph** - displays the views simultaneously using a red hue for left and green hue for right, and adds selection controls above the Viewer.
- **Flicker** - displays both views alternately, and adds selection controls above the Viewer.

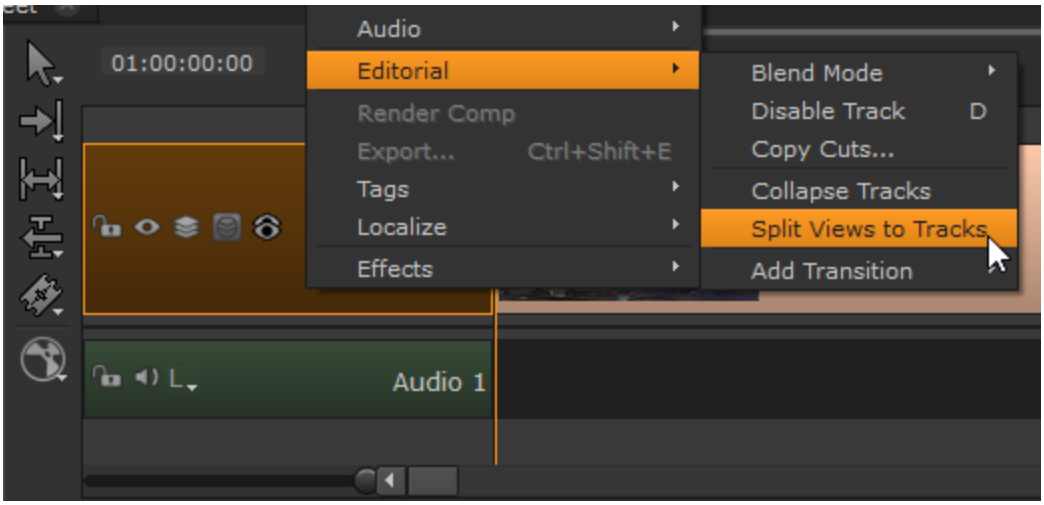
Applying Changes to Selected Views

By default, Nuke Studio applies any changes you make to all views. To apply changes to a particular view only (for example, the **left** view but not the **right**), you can:

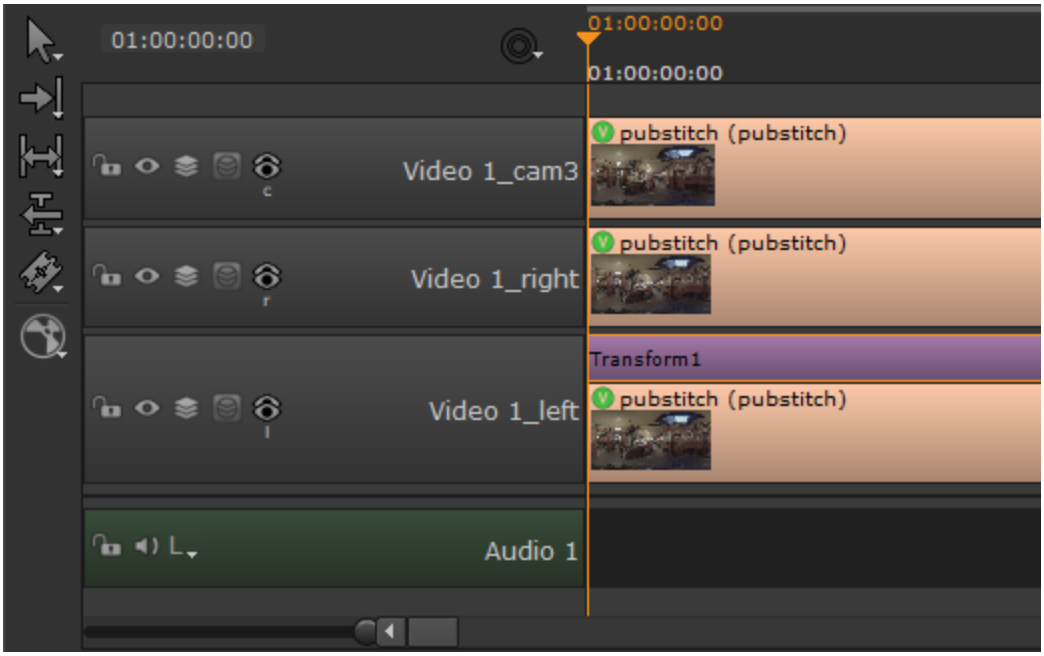
- Split the views into separate tracks. Separating views allows you to apply effects, such as a Transform or Timewarp, to individual views without affecting the other views. See Splitting Views to Tracks for more information.
- Split the view off in the soft effect's **Properties**. Splitting properties allows you to perform the same operation on multiple views, but with different values for each, such as a Grade or ColorCorrect. See Splitting Views in the Properties Panel for more information.

Splitting Views to Tracks

1. Right-click a multi-view shot on the timeline and select **Editorial** > **Split Views to Tracks**.



2. Right-click the track to which you want to add the soft effect, select **Effects** and then the required soft effect. For example, adding a Transform to the left view applies the effect to only the left view track. See Adding Effects on the Timeline for more details on adding effects to shots.

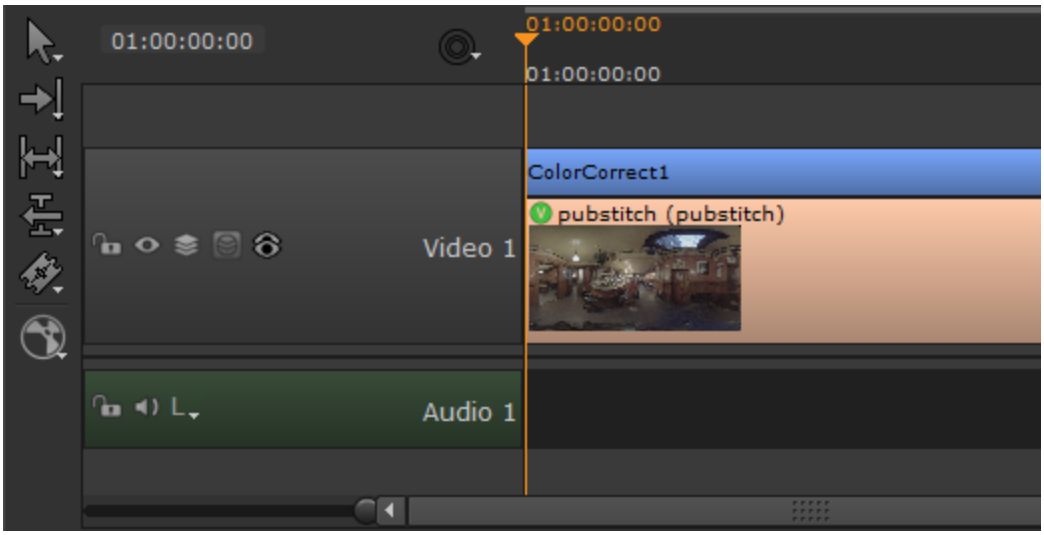


You can create comps or export multi-view shots and effects in a similar way to regular shots. See Exporting Multi-View Source Clips for more information.

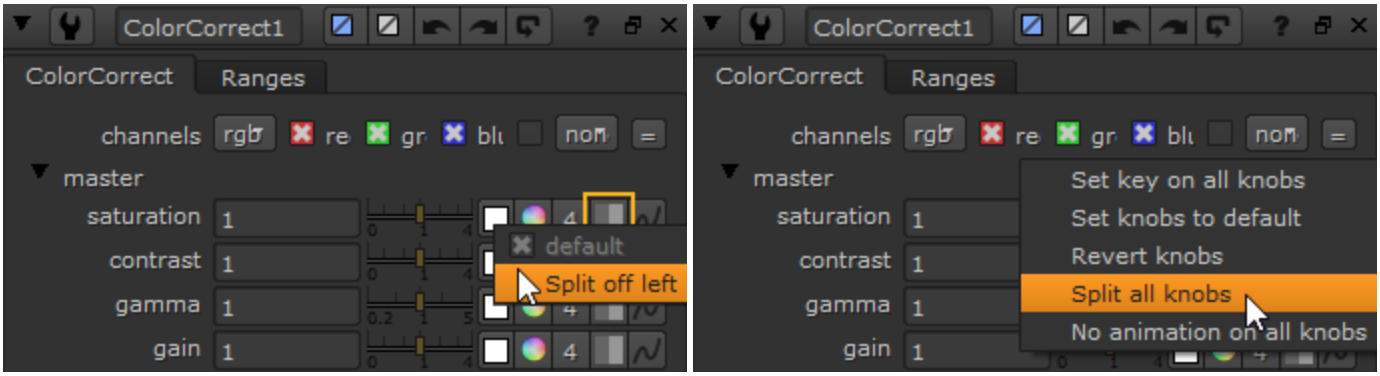
Splitting Views in the Properties Panel

1. Right-click the track to which you want to add the soft effect, select **Effects** and then the required soft effect. For example, adding a ColorCorrect a multi-view shot.

   See Adding Effects on the Timeline for more details on adding effects to shots.



2. Select the view you want to make changes to using the buttons above the timeline Viewer.
3. Open the effect's **Properties** and click the view button on the right, next to the control you want to adjust.
4. Select **Split off [view name]**. For example, to apply changes to a view called **left**, select **Split off left**. You can also split all the effect's controls by selecting **Split all knobs** from the right-click menu.

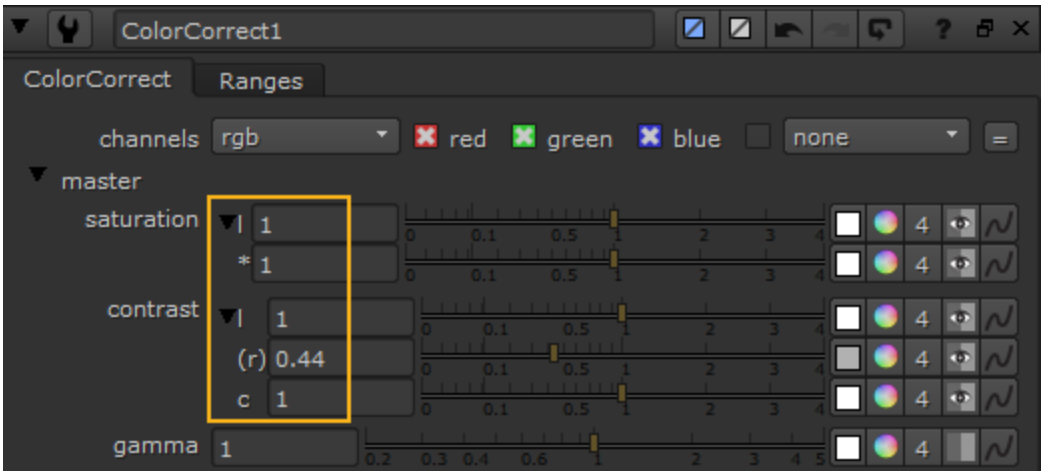


An eye appears on the split view button . Any changes you make using the control in question are only applied to the view you chose to split off. Changes to controls that have not been split off are still applied to all views.

You can create comps or export multi-view shots and effects in a similar way to regular shots. See Exporting Multi-View Source Clips for more information.

Showing Separate Values for Each View

Once you have split off a view, you can apply changes to the existing views separately. Click on the arrow on the left side of a split control to divide the control into values for each view.



In the example, the **left** view is split for the **saturation** control and **left** and **cam3** views are split for the **contrast** control.

Note: The ***** (asterisk) denotes there is more than one unsplit view remaining for the **saturation** control and **( )** denotes that the **right** view is the only unsplit view for the **contrast** control.

Unsplitting Views

1. In the effect's controls, click the **view** button .
2. Select **Unsplit [view]**. For example, to unsplit a view called **left**, you'd select **Unsplit left**.
   The view is unsplit, and all changes you made to individual views are lost.

Adding Effects on the Timeline

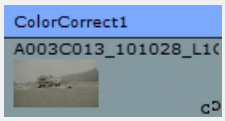
A soft effect is a real-time effect, such as Transform or TimeWarp, processed on GPU instead of CPU. You can add soft effects to your timeline in any of the workspaces.

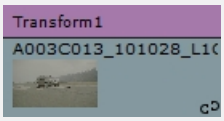
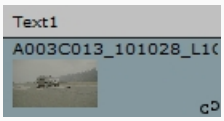
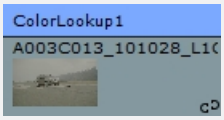
You can add custom plug-in or gizmo soft effects to the **Add Effect** menu using Python. Valid custom soft effects must have a GPUEngine implementation using **DD::Image::Iop::gpuEngine** type functions. For more information see Nuke's NDK Reference Guide.

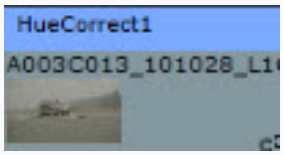
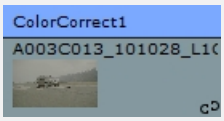
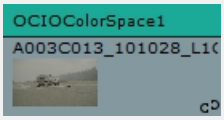
Soft Effects must also be registered after creation. An example of how to register a plug-in or gizmo as a custom soft effect is located in:
`<install_directory>/pythonextensions/site-packages/hiero/examples/custom_soft_effect.py`

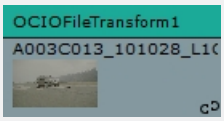
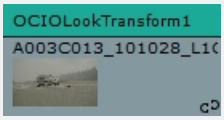
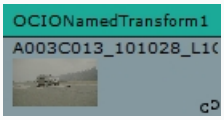
Available Soft Effects

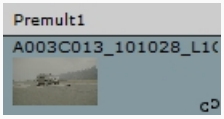
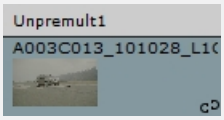
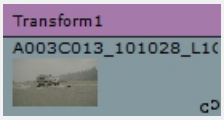
Below is a brief summary of the available soft effects. These are similar to the tools in Nuke's Node Graph. See the Nuke Online Help for more information about them.

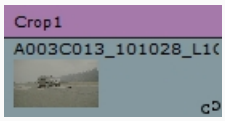
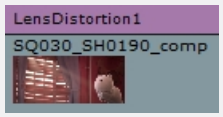
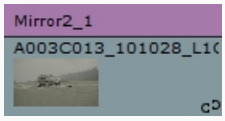
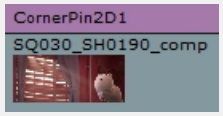
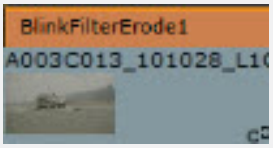
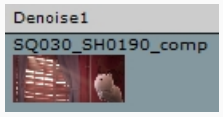
Soft Effect	Color Code	Summary
ColorCorrect		Make quick adjustments to saturation, contrast, gamma, gain, and offset. You can apply these to a clip's entire tonal range, shadows, midtones, or highlights. You can control the range of the image that is considered to be in the shadows, midtones, and highlights using the lookup curves on the **Ranges** tab. However, do not adjust the midtone curve - midtones are always equal to 1 minus the other two curves. **Alt**+**C** inserts a ColorCorrect soft effect on any shot selections or at the playhead

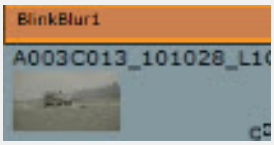
Soft Effect	Color Code	Summary
Grade		Define white and black points by sampling pixels from the Viewer. For example, you can use this for matching foreground plates to background plates. **Alt**+**G** inserts a Grade soft effect on any shot selections or at the playhead
Transform		Translate, rotate, scale, and skew shots from a single control panel. **Alt**+**T** inserts a Transform soft effect on any shot selections or at the playhead
Text		Add text overlays on your images. You can simply type in the text you want to have displayed or use Tcl expressions (such as **[metadata values]**) or Tcl variables to create a text overlay. Text overlays can also be animated using animation layers in the Groups tab, so that their properties (such as position, size, and color) change over time.
TimeWarp		Slow down, speed up, or even reverse selected frames in a clip without necessarily altering its overall length. Sequences imported from **.xml** and **.aff** files also support non-linear retimes. **Note:** TimeWarp effects are only allowed on tracks with clips as they are linked to the clips.
Color Submenu (OCIO soft effects are also available in HieroPlayer)		
ColorLookup		Make color adjustments using lookup tables (LUTs). LUTs refer to line graphs of a given color channel's brightness. The horizontal axis represents the original, or input, values and the vertical axis represents the new, or output values.

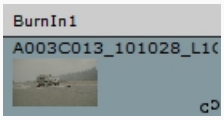
Soft Effect	Color Code	Summary
HueCorrect		Adjusts the hue of the images.
ColorCorrect		Make quick adjustments to saturation, contrast, gamma, gain, and offset. You can apply these to a clip's entire tonal range, shadows, midtones, or highlights. You can control the range of the image that is considered to be in the shadows, midtones, and highlights using the lookup curves on the **Ranges** tab. However, do not adjust the midtone curve - midtones are always equal to 1 minus the other two curves. **Alt**+**C** inserts a ColorCorrect soft effect on any shot selections or at the playhead
Grade		Define white and black points by sampling pixels from the Viewer. For example, you can use this for matching foreground plates to background plates. **Alt**+**G** inserts a Grade soft effect on any shot selections or at the playhead
OCIO Submenu		
OCIO CDL Transform		Applies an ASC CDL (American Society of Cinematographers Color Decision List) grade based on the OpenColorIO Library. See OCIOCDLTransform in the Nuke Online Help for more information on the **Properties** panel for this effect.
OCIO ColorSpace		Converts images from the input colorspace of the shot on the timeline to an output colorspace more appropriate to a given process or intended display device.
OCIO Display		Applies a colorspace conversion to an image sequence so that it can be accurately represented on a specific display device.

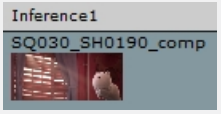
Soft Effect	Color Code	Summary
		See OCIODisplay in the Nuke Online Help for more information on the **Properties** panel for this effect.
OCIO FileTransform (LUT)		Loads a colorspace conversion from a file (usually a 1D or 3D LUT) and applies it. You can also load other file-based transformations, for example an ASC ColorCorrection XML. File transforms are applied without any input or output colorspace handling. If the file expects log-encoded pixels, but you apply the node to a linear image, you will get incorrect results. See OCIOFileTransform in the Nuke Online Help for more information on the **Properties** panel for this effect.
OCIO LogConvert		Converts from **compositing_log** to **scene_linear**, or the reverse. Sometimes log data is required for certain processing operations, like plate resizing or pulling keys. Usually a colorspace is provided specifically for these operations.
OCIO LookTransform		Applies a creative colorspace transform using a predefined look in the OCIO config file. A look is a named color transform intended to modify the 'look' of an image in a creative way, rather than a technically or mathematically defined colorspace conversion. See OCIOLookTransform in the Nuke Online Help for more information on the **Properties** panel for this effect.
OCIO NamedTransform		OCIONamedTransform uses **NamedTransforms** in the OCIO configs shipped with Nuke to emulate older methods of color management that ignored the RGB primaries and simply applied one-dimensional transformations. These NamedTransforms have been added to the config as sometimes it is helpful to include one or more transforms in a config that are essentially stand-alone transforms that do not have a fixed relationship to a reference space or a process space. An example would be a utility curve transform where the

Soft Effect	Color Code	Summary
		intent is to simply apply a LUT1D without any conversion to a reference space. See OCIONamedTransform in the Nuke Online Help for more information on the **Properties** panel for this effect.
 **Tip:** For more information on the OpenColorIO (OCIO) library, see https://opencolorio.org/		
ChromaKeyer		Pull a quick chroma key from green or bluescreen areas of your footage. Use the **screen color** selector to choose a color from the **Source** input to use as the blue/green screen color. To remove blue/green spill from the foreground object, use the **despill** controls to pick skin tones from the source. Use the **matte** parameters to improve the matte.
Merge Submenu		
Premult		Just like the Premult and Unpremult node in Nuke, these effects multiply or divide the shot's rgb channels by its alpha channel. If the alpha channel doesn't exist or covers the whole frame equally, premultiply and unpremultiply have no effect. See Premult and Unpremult in the Nuke Online Help for more information on the **Properties** panel for this effect.
Unpremult		
Transform Submenu		
Transform		Translate, rotate, scale, and skew shots from a single control panel. **Alt**+**T** inserts a Transform soft effect on any shot selections or at the playhead

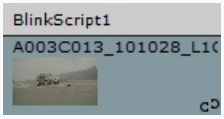
Soft Effect	Color Code	Summary
Crop		Cut out the unwanted portions of the image area. You can fill the cropped portion with black or adjust the image output format to match the cropped image.
LensDistortion		The LensDistortion effect estimates the lens distortion in a given image, either through Grid Detection or manual Line Detection. The warp can then be used to add or remove distortion or produce an STMap in the motion channel for use elsewhere. See Correct Shots Using LensDistortion for more information.
Mirror		Flip the input image around the center of the format area. A flip on the x axis mirrors the image vertically. A flop on the y axis mirrors the image horizontally.
CornerPin		The CornerPin2D effect is designed to map the four corners of an image sequence to or from positions derived from tracking data. In practice, this effect lets you replace any four-cornered feature with another image sequence. You can use it to place an image in an on-screen television, for example. See Using the CornerPin2D Node for more information.
Filter Submenu		
FilterErode		Filters input pixel's relation to the size control. Negative values cause brighter areas to expand into darker areas and positive values cause darker areas to expand into lighter areas.
Denoise		The Denoise effect is an efficient tool for removing noise or grain from your footage. It uses spatial or temporal filtering to remove noise without losing image quality. See Removing Noise with Denoise for more information.

Soft Effect	Color Code	Summary
Blur		Adds blur to the images.
(Continued)		
ModifyMetaData		Inject metadata key pairs into the shot during export in a similar way to the ModifyMetaData node in Nuke.

Burn-In		Quickly add standard burn-in elements on the timeline. You can control the color, opacity, font, scale, and so on, as well as use the dropdowns to determine what element is added from the file or sequence metadata.
		
		You can also reference custom metadata from shots. For example:
		`hiero/shot/tags/Approved`
		
		Extracts the **Approved** tag from the shot. You can also append **note** to include any notes associated with the tag:
		`hiero/shot/tags/Approved/note`
		
		Note: You must precede spaces and slashes in the tag name with \\ (backslashes) to enable Hiero to process the tag name correctly. For example:
		`hiero/shot/tags/Blue\\ Screen/note`
		
		You can also add burn-in through the **Export** dialog, see Adding Burn-in Text to Exports for more information.

Inference (Machine Learning)		The Inference effect applies a pre-trained **.cat** file to create the effect modeled by a neural network across the input image. You can use pre-trained **.cat** files from Nuke's CopyCat node, models from the Cattery, or third-

Soft Effect	Color Code	Summary
		party converted files from CatFileCreator can be used in the Inference effect. See Apply Pre-Trained Models to Shots Using Inference for more information.

BlinkScript		Run Foundry's Blink framework on the timeline, enabling you to write code once and run it on any supported device.
		
		Warning: BlinkScript is very flexible, as there are no restrictions on the code you can write within a kernel. As a result, code complied from the Kernel Source can cause Nuke to crash, so please use caution!
		
		The BlinkScript soft effect supports a subset of the functionality available in the full BlinkScript node:
		• You can't publish your kernels to Groups or Gizmos.
		• Due to the way stacks of soft effects are processed in Hiero, the BlinkScript soft effect only contains one input source and produces only one output.
		• The following functions are not supported by the BlinkScript effect if all the images have eAccessPoint data access:
		  • log10
		  • round
		  • rsqrt
		  • abs for integer types
		  • modf(a, *b)
		  • sign
		  • rcp
		  • max, min, and clamp for integer types
		  • median, atomicInc and atomicAdd.

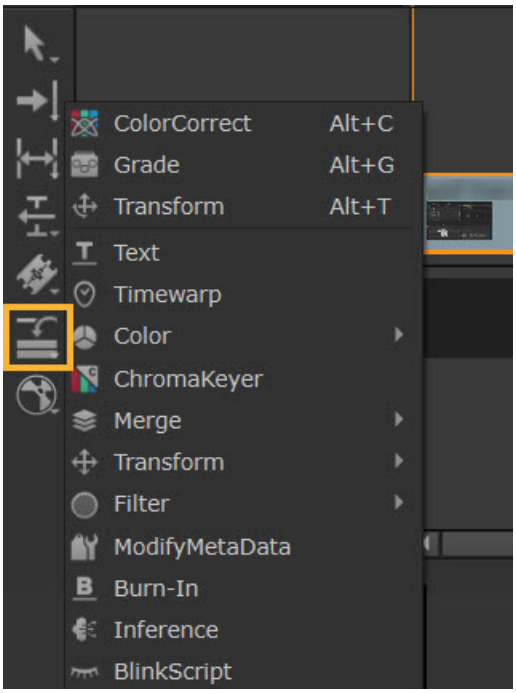
Soft Effect	Color Code	Summary
		• The only data types supported by the BlinkScript effect are **int**, **float**, and **bool**. **Warning:** Soft effects using RIP will not work if there is a mismatch between your OpenCL GPU (blink device) and the OpenGL GPU. **Note:** The BlinkScript soft effect supports both *pixel-wise* and *component-wise* kernels, but the former is preferred for performance reasons. For more information about this and a more detailed description of the language, see https://learn.foundry.com/nuke/developers/17.0/BlinkKernels/

Adding Sequence-Level Soft Effects

Warning: Please keep in mind the following:
• Sequence-level soft effects are only permitted on the same track as clips if they're trimmed to exactly match the in and out points of individual clips. In this case, each effect is linked to a specific clip.
• Soft effects can be trimmed arbitrarily if they're on tracks with no clips.

The timeline provides some additional ways to add soft effects. You can either:

- Right-click a shot on the timeline, select **Soft Effects** and then select the soft effect you want to apply, or
- Select a shot on the timeline, click the **Add Effect** button to the left of the timeline, and then select the soft effect you want to apply.



Note: You can add a soft effect to multiple shots by selecting the required shots first and then right-clicking on one of them and selecting the soft effect you want to add. A soft effect is added to each of the selected shots. You can also add a single soft effect for the whole track by right-clicking on the track header and then selecting the soft effect you want to add.

Note: TimeWarp effects are only allowed on tracks with clips (and therefore linked to clips).

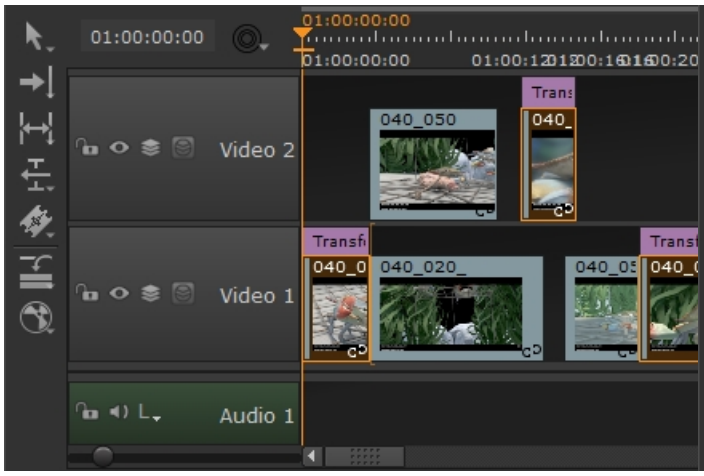
Using the Spreadsheet View

You can also add soft effects using the spreadsheet view in any workspace - see Using the Spreadsheet View.

Soft Effect Keyboard Shortcuts

You can add Transform, Grade, and ColorCorrect effects using the keyboard shortcuts **Alt**+**T**, **Alt**+**G**, and **Alt**+**C**, just like the Node Graph equivalents in the compositing environment. You can add effects to multiple shot selections or at the playhead if no shots are selected.

Note: If no shots are selected, the effect is added to the top-most shot under the playhead.



Adding effects to multiple selections.



Adding effects at the playhead.

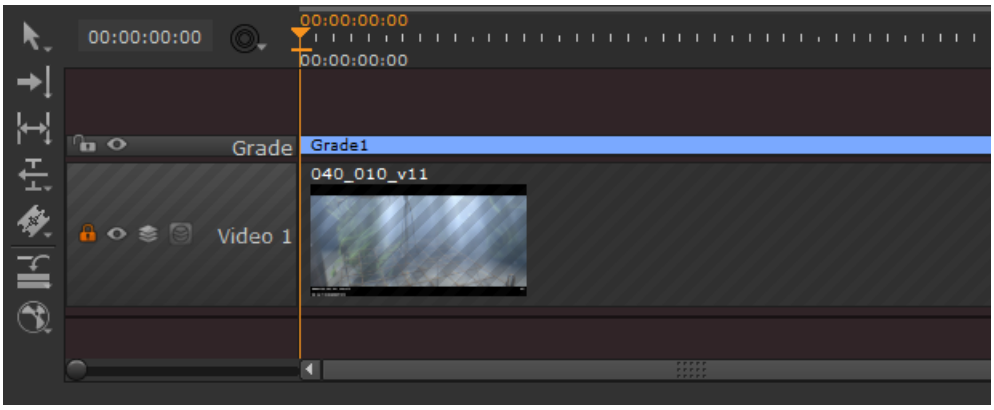
Adding Shot-Level Soft Effects

Warning: Soft effects added at shot-level, must match the length of the shot on the locked track. Any soft effect that is trimmed beyond the end of a shot, or a different length from the shot is marked with red hashing to show that it is invalid.

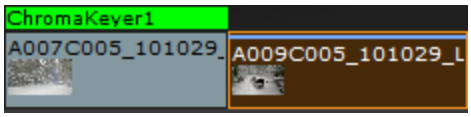
You can add shot-level soft effects on the timeline, by doing the following:

1. Right-click the shot that you want to add a soft effect to.
2. Select **Open In** > **Timeline View**.
3. Click the **Soft Effects** menu icon and select the soft effect you want to apply. For example, you can select **Grade**.

   The soft effect is then displayed above the shot as a colored box.

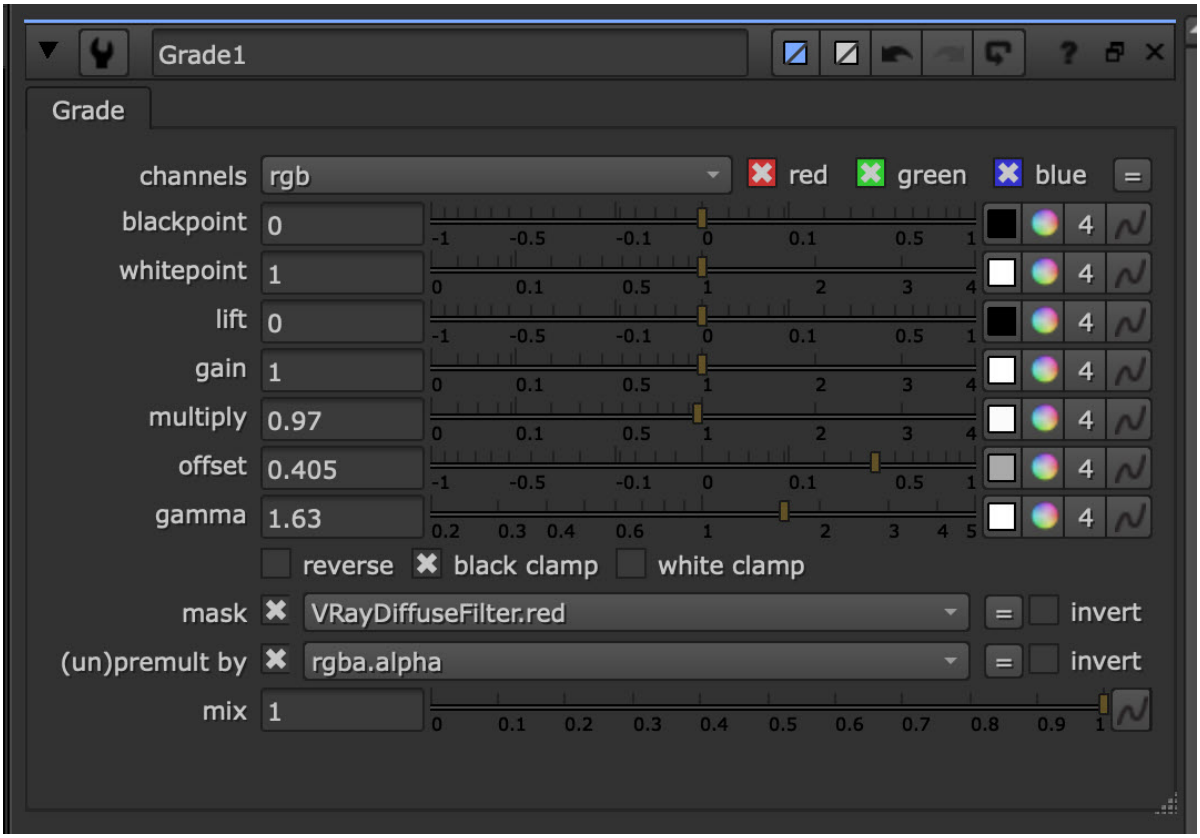


When you close the timeline view of the shot – as it is a shot-level soft effect – the soft effect is displayed as a colored line within the top of the shot. The color of the line displayed reflects the highest soft effect added to the shot.



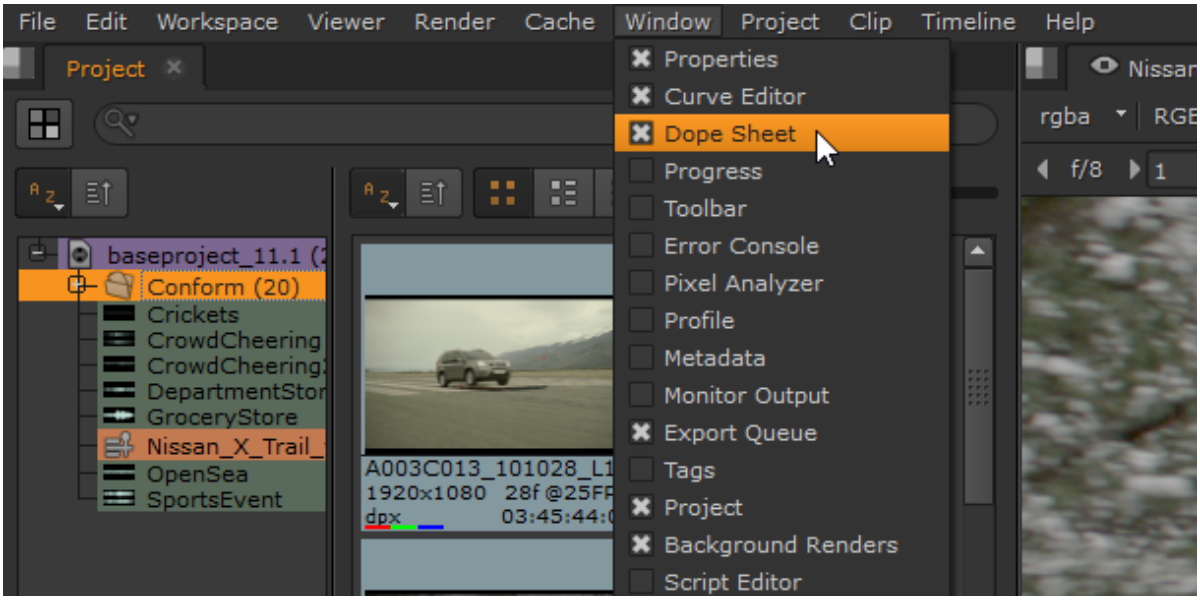
Soft Effect Controls

Adding an effect displays the associated controls in the **Properties** panel, similar to Nuke nodes. Adjusting the controls affects the shots underneath the effect in real-time. For example, adding a Grade effect at sequence level displays the Grade controls in the **Properties** panel.



See Properties Panels for more information on node controls.

If you intend to animate soft effect controls using keyframes, you can use the Curve Editor and Dope Sheet to fine-tune the output. To add the Curve Editor or Dope Sheet to the interface, navigate to **Window** and select the required panel.



See Animating Parameters for more information.

Editing Sequence-Level Soft Effects

You can copy, move, and cut soft effects just like you can with shots in the timeline. You can perform these actions by either accessing them from **Edit** in the right-click menu, or by using the keyboard shortcuts. You can copy soft effects to different tracks, different sequences, and different projects.

Moving

You can move a sequence-level soft effect by simply clicking and dragging the soft effect to a different shot, or even onto a different video track.

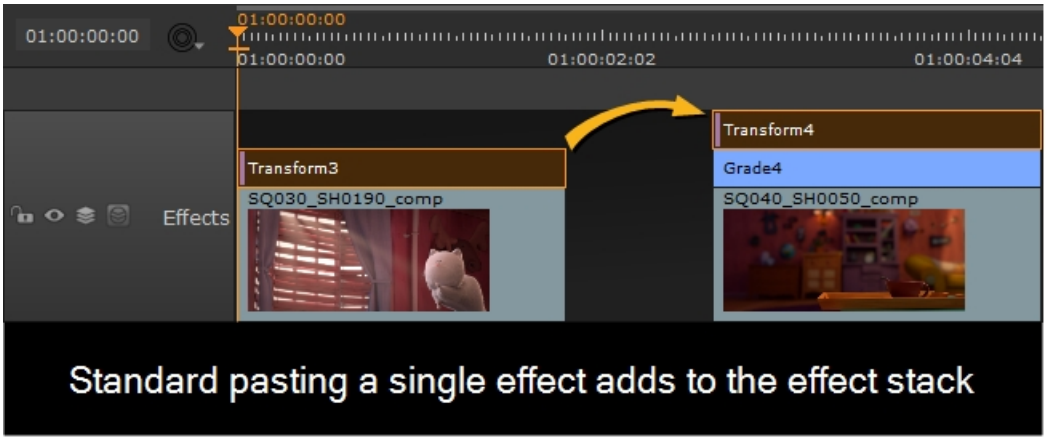
Copying

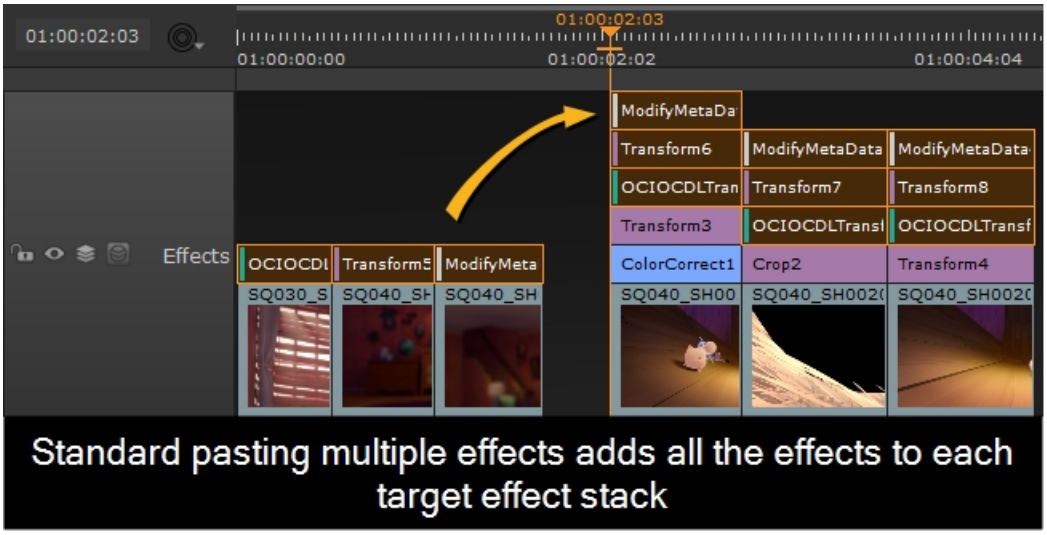
Nuke Studio and Hiero allow you to copy a sequence-level soft effect above the original to create a stack, to a different track, or to a different sequence. You can also copy a sequence-level soft effect to a shot open in the timeline view, therefore pasting it as a shot-level soft effect. There are several copy operations available for soft effects, standard copy/paste, replace paste, sequential paste, and sequential replace paste.

Standard Copy and Paste

Standard copy and paste adds the copied effect(s) to the stack on the target shot(s). Effects are read from the top down, so pay attention to the order of the stack.

1. Select the soft effect you want to copy by clicking it.
2. Press **Ctrl**/**Cmd**+**C** or right-click and select **Edit** > **Copy**.
3. Select the target shot or move the playhead to where you want to paste the copy.
4. Press **Ctrl**/**Cmd**+**V** or right-click and select **Edit** > **Paste**.

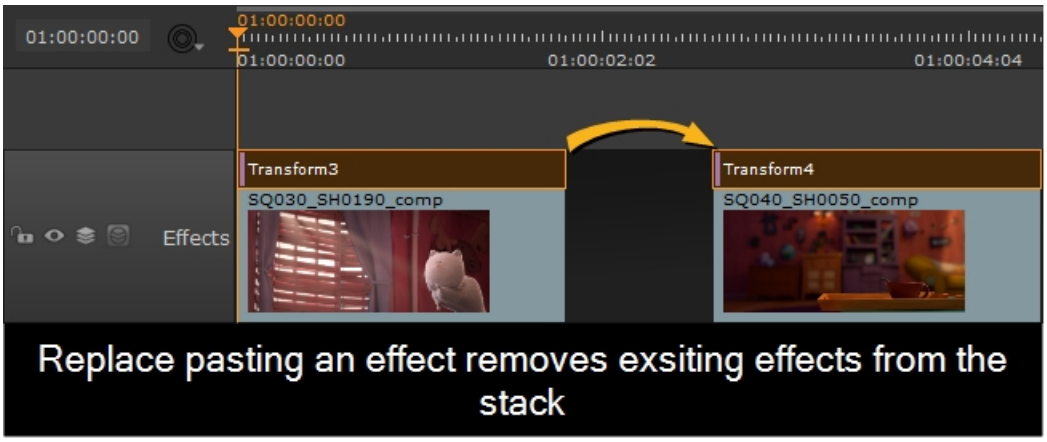




Replace Paste

Replace paste copies the source effect(s) to the target shot and removes any existing effects on the shot.

Press **Shift**+**V** or right-click and select **Edit** > **Replace Paste**.



Sequential Paste

Unlike standard paste, sequential paste only pastes each effect once, in the order that they appear on the source shots.

Press **Ctrl/Cmd**+**Alt**+**V** or right-click and select **Edit** > **Sequential Paste**.

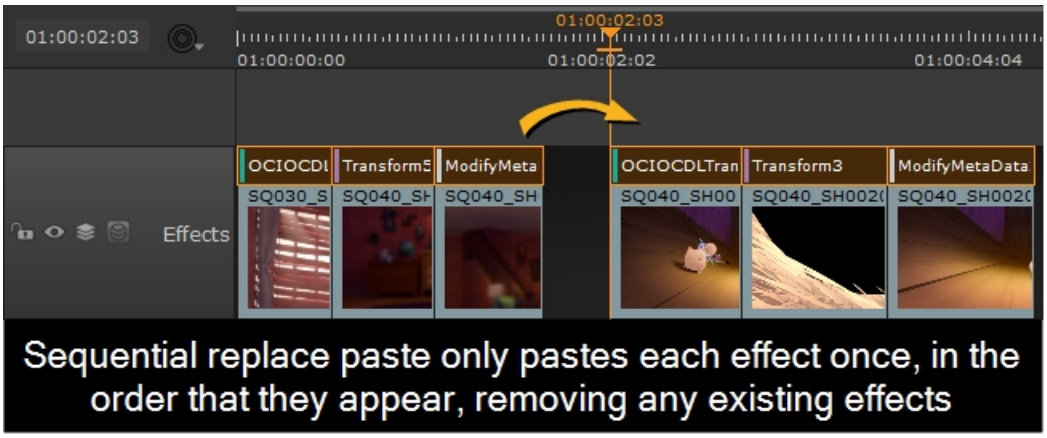


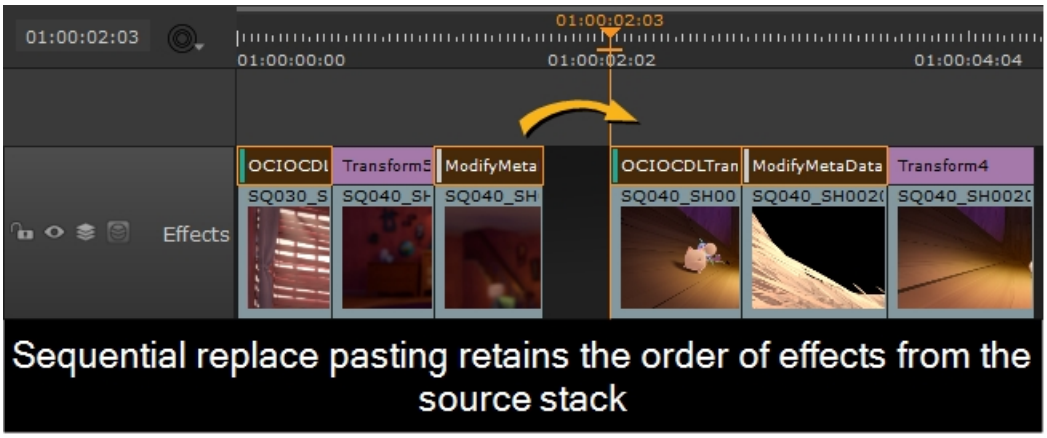


Sequential Replace Paste

Sequential replace paste only pastes each effect once, in the order that they appear on the source shots, removing any existing effects from the target shots.

Press **Alt**+**Shift**+**V** or right-click and select **Edit** > **Sequential Replace Paste**.





Cloning

You can clone a sequence-level soft effect. This copies the soft effect and links it to the original, which means when one of these is edited, the changes are reflected in the other one. You can clone a soft effect to a different track or even a different sequence. You cannot clone a soft effect in different projects.

To clone a soft effect:

1. Select the soft effect you want to clone by clicking on it.
2. Select **Edit** > **Copy** (or press **Ctrl**/**Cmd**+**C**).
3. Move the playhead on the timeline to where you want to place the new clone.

4. Select **Edit** > **Clone** (or press **Alt**+**K**).

   The new clone is placed at the current playhead position on the timeline. You can repeat steps 3 and 4 to create multiple clones that are all linked, at different places on the timeline.

   Clones are indicated by a **C** highlighted in red in the left of the soft effect.

Note: Cloning animation in soft effects is not supported.

Copying as Clone

You can also copy a sequence-level soft effect as a clone. This means, when you paste a new copy of the soft effect above a selected shot, it is automatically linked to the original soft effect as a clone. Therefore, any changes made to either of the cloned soft effects, are reflected in the other.

To copy a soft effect as a clone, do the following:

1. Select the soft effect you want to copy as a clone by clicking on it.
2. Select **Edit** > **Copy as Clones** (or press **Ctrl/Cmd**+**K**).
3. Click on the shot that you want to clone the soft effect to.
4. Select **Edit** > **Paste** (or press **Ctrl/Cmd**+**V**).

   The soft effect is copied as a clone on your selected shot. You can repeat steps 3 and 4 to create multiple clones that are all linked above different selected shots.

   Clones are indicated by a **C** highlighted in red in the left of the soft effect.

Note: Cloning animation in soft effects is not supported.

Decloning

To declone a soft effect, simply click on the clone you want to declone and select **Edit** > **Declone** (or press **Alt**+**Shift**+**K**).

Note: For more information about copying, moving, and cutting soft effects, see Timeline Editing Tools .

Deleting

To delete a soft effect, simply right-click on it and select **Edit** > **Delete** (or press **Backspace**).

Editing Shot-Level Soft Effects

To edit a shot-level soft effect, you first need to open the shot with the applied soft effect, in the timeline view. You can do this by right-clicking the shot and selecting **Open In** > **Timeline View**. You can then copy, delete or move the shot-level soft effect in exactly the same way as sequence-level soft effects.

You can copy and paste the shot-level soft effect on top of the original, creating a stack in the timeline view. When you close the timeline view, stacked shot-level soft effects are displayed as a single line within the top of the shot. Nuke Studio allows you to copy a shot-level soft effect, return to the full sequence, and then paste it as a sequence-level soft effect. You can also paste a shot-level soft effect to another shot open in the timeline view.



Note: You cannot clone shot-level soft effects.

Enabling and Disabling Soft Effects

You can choose to disable and re-enable soft effects from the output. To enable or disable a sequence-level soft effect, select the soft effect by clicking on the colored box and then pressing **D**. To enable or disable a shot-level soft effect, you first need to open the shot – that includes the soft effect – in the timeline view. Then you can select the soft effect and press **D**.

Using Soft Effects across multiple channels

You can edit Multichannel EXRs directly in the timeline. This allows you to use effects and make precise adjustments to specific layers, and combine them to get fast shot tweaks, which you can copy and paste across shots in the timeline.

See Using Multichannel Soft Effects to learn more.

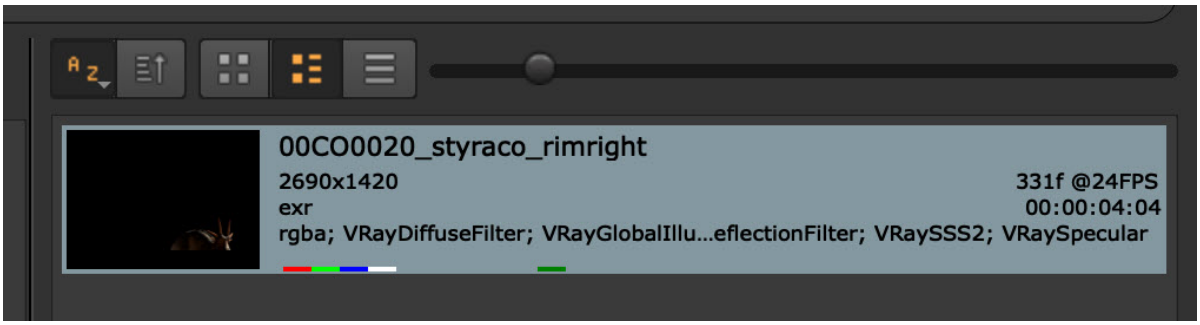
Using Multichannel Soft Effects

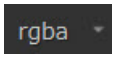
You can utilize Multichannel EXRs directly in the timeline. This allows you to use effects and make precise adjustments choosing specific layers, and combine them to get fast shot tweaks, which you can copy and paste across shots in the timeline.

Import EXR into the timeline

Import your EXR files into the timeline using **File** > **Import Files(s**), and selecting your **.exr** file.

Your file will now be visible in the project panel.

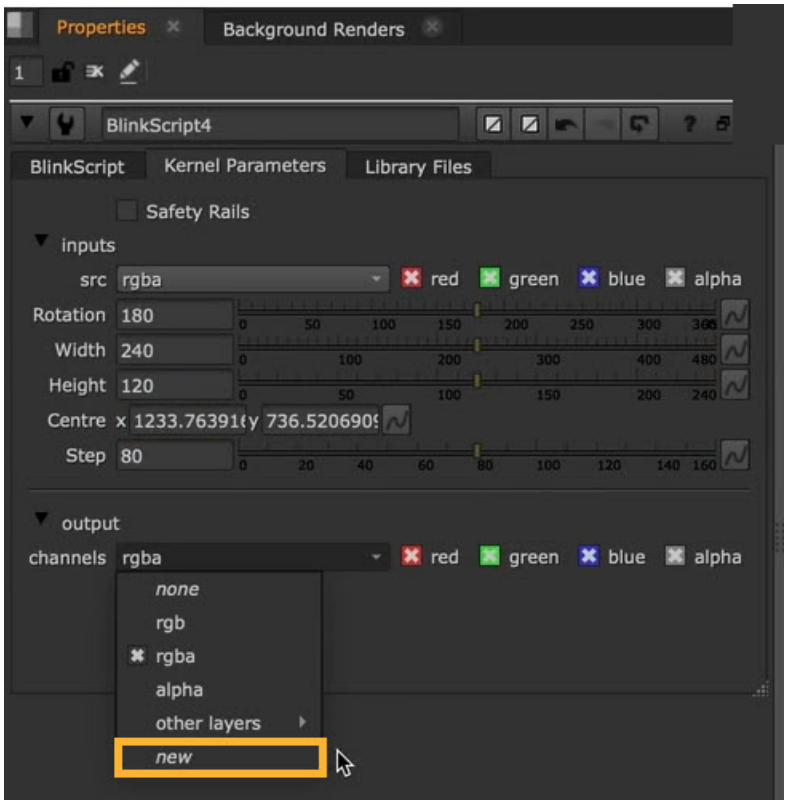


The layers available in the current frame are visible on the first level of the Layer dropdown at the top of the Viewer. 

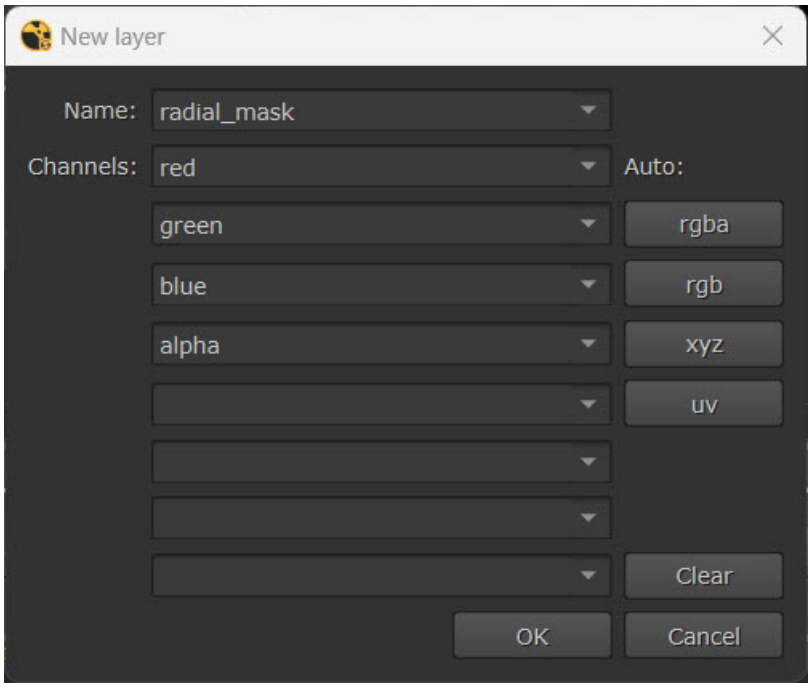
Add new channels

You can create new channels from any Soft Effect.

To do this, go into a soft effect's properties, select the **output** > **channels** dropdown, and select **new**.



This opens a window where you can choose what type of channel to create using the buttons in the **auto** menu.

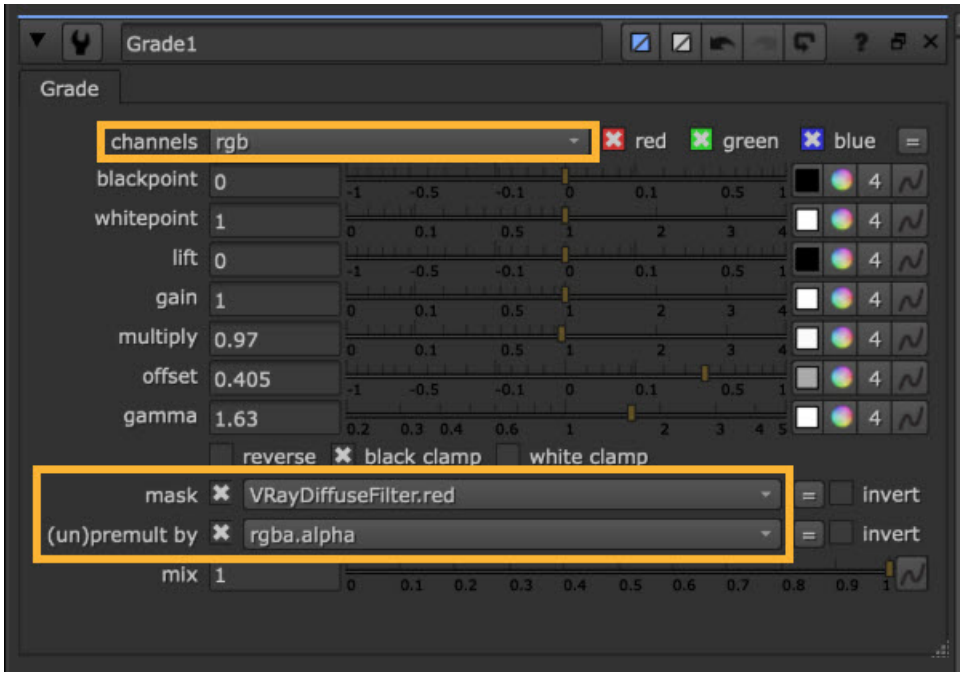


Apply soft effects to specific layers and mask through any available channel

You can select, filter, and isolate specific layers for detailed editing. For example:

- Use soft effects such as HueCorrect, Grade, and Blur, and apply them to specific channels.
  To do this, open each effect's **Properties**, and use the **channel** dropdown.
- Mask a color soft effect via a non-rgba layer such as depth or mattes.
  To do this, you can open each effect's **Properties**, and use the **mask** dropdown.
- Premult and unpremult using any layer.
  To do this, you can open each effect's node **Properties**, and use the **(un)premult** by dropdown.
- Use BlinkScript with multiple layers as inputs.
  To do this, you can open the **BlinkScript** effect's **Properties**, and use the **src** dropdown.

For example, you can see specific layers have been selected for a **Grade** effect here:

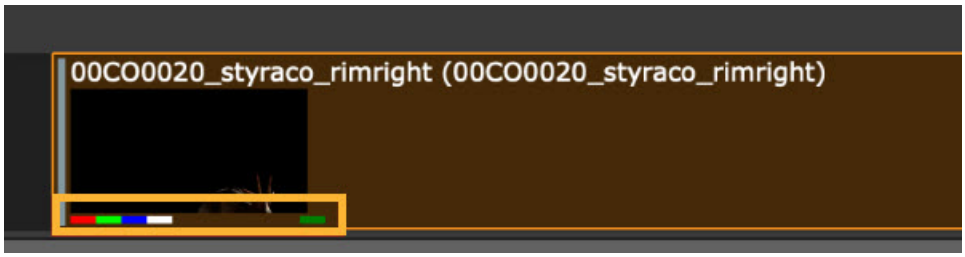


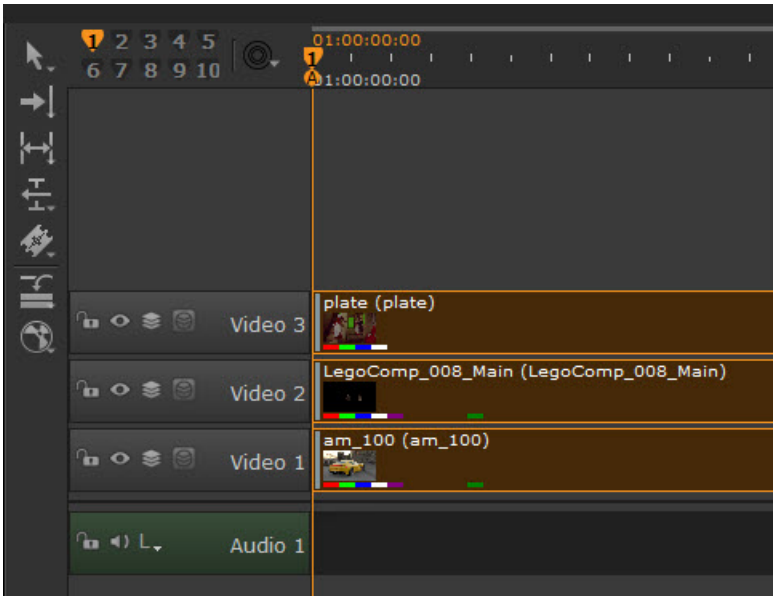
Tip: See Adding Effects on the Timeline to learn more about each soft effect.

Channel indicator chips

You'll be able to see which channels are being used by a soft effect, when a soft effect is using a mask, and which channels are being used by a track item.

Colored indicator chips will appear on track items and soft effects when they are selected.





- red color layer.

- green color layer.

- blue color layer.

- alpha layer.

- depth layer.

- forward motion vector layers.

- backward motion vector layers.

- all other custom layers, such as disparity.

When a layer is being used as a mask it will show the icon - 

Share your edits to the node graph

All of your work can be easily translated into Nuke nodes to convey creative intent to collaborators in their complex composites.

Simply copy and paste from your timeline into the node graph to carry your changes forward.

Note: Multi-part EXRs are required for optimal performance. If you haven't already, we recommend you switch to this type of files in order to get the best of this feature.

Note: While the multi-channel aware timeline is a powerful tool, it's important to note that extremely large or complex EXR files may still impact performance. We recommend optimizing your files where possible for the best experience.

Tip: The **BlinkScript** soft effect allows multiple layers as an input, which means a great expansion of the potential of this versatile tool. For example, using secondary inputs to process, such as **DepthBlur** Kernel, Shuffle(BlinkScript) and others.

Tip: You can use **.cat** files requiring multiple inputs with the Inference soft effect. See Apply Pre-Trained Models to Shots Using Inference.

Correct Shots Using LensDistortion

Nuke Studio and Hiero's soft effect collection also includes a LensDistortion effect in the timeline. Using LensDistortion in the timeline enables new workflows for shot prep and distribution and you don't have to go back and forth between the timeline and comp environments. The results can be applied to multiple shots that use the same lens metadata or at sequence-level across an entire show.

Just like Nuke's LensDistortion node, the timeline effect estimates the lens distortion in a given shot, either through Grid Detection or manual Line Detection. The warp can then be used to add or remove distortion or produce an STMap in the motion channel for use elsewhere.

See Working with LensDistortion for more information.

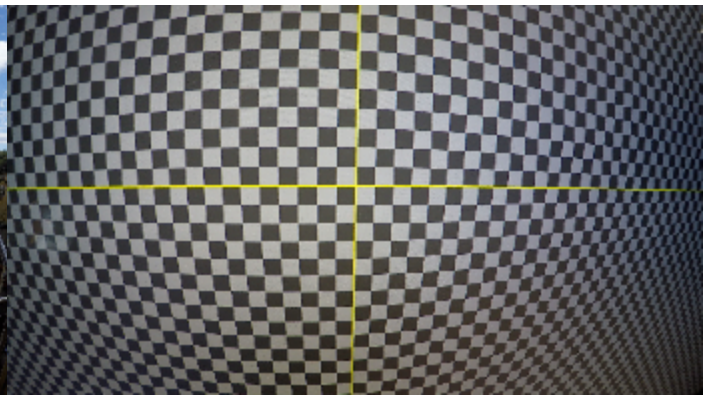
Remove Lens Distortion from a Shot

Here's how to remove distortion from a shot using the LensDistortion effect on the timeline.

1. Either select the shot on the timeline manually or use the handy auto-select feature to choose a shot for you depending on the current playhead's position.
2. Click the soft effect menu icon on the left-hand side of the timeline.
3. Select **Transformations > LensDistortion** to add the effect to the shot.
4. If you have a lens distortion grid shot with the same camera used for filming, you can use that to estimate the lens distortion applied to your shots,



The original distorted shot



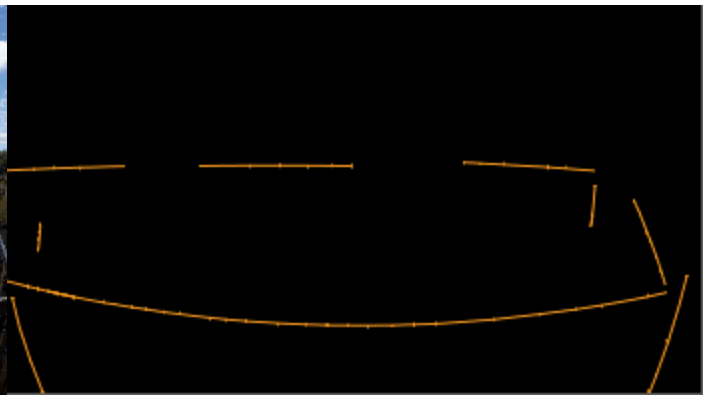
Associated grid shot

OR

If no estimation grid is available, you can use the LensDistortion effect's manual line drawing workflow to estimate lens distortion.



The original distorted shot

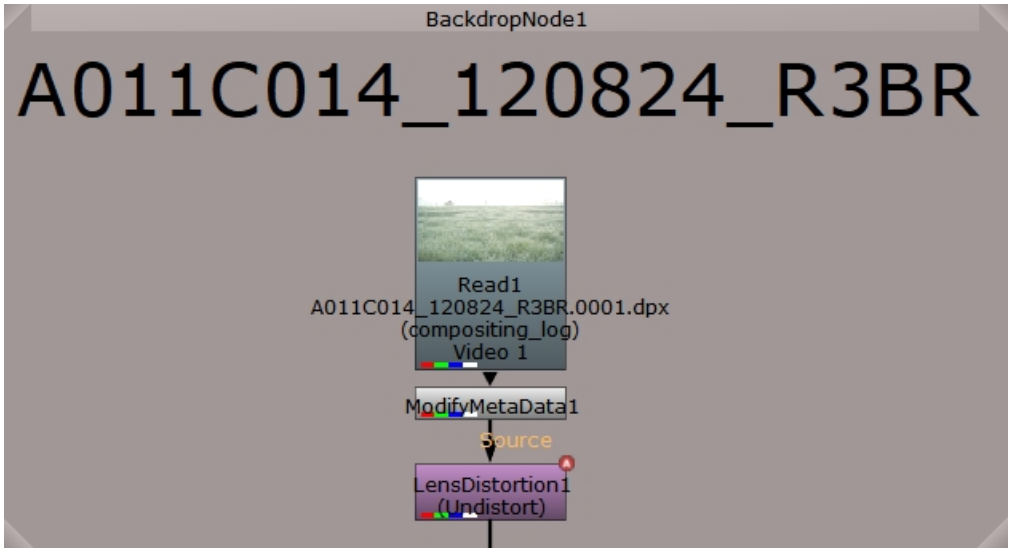


Example lines in the alpha channel

See Estimating Lens Distortion Using a Grid or Estimating and Removing Lens Distortion Using Lines for more information.

5. When you've calculated the required lens distortion, make sure that the LensDistortion effect's **Mode** is set to **Undistort** in the **Properties** panel.

   This step removes the calculated distortion from the shot so that you can send the undistorted plate to your comp department using **Create Comp**. Nuke Studio or Hiero automatically adds a LensDistortion node to the comp, as you can see in the example.



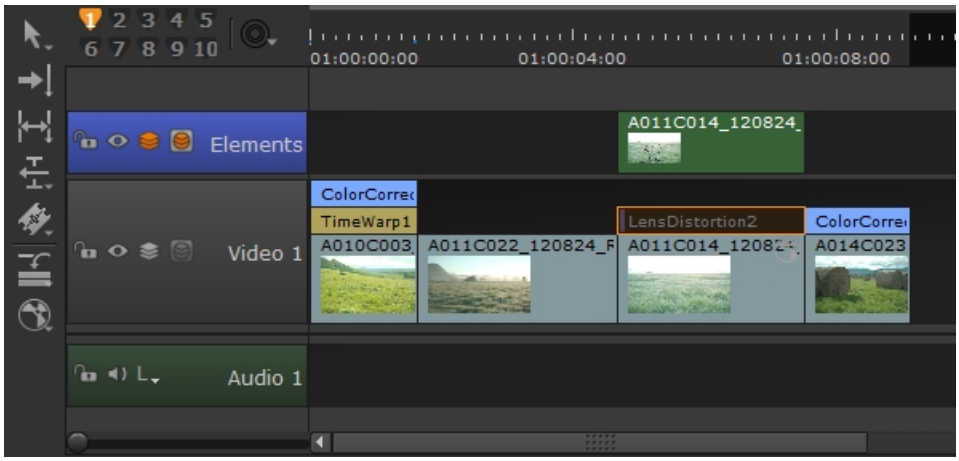
   Your comp department can now add their work to the undistorted plate.

Add Lens Distortion to a Shot

You can quickly add lens distortion to shots or whole sequences on the timeline by copying a LensDistortion effect set to **Undistort** and changing the **Mode** to **Redistort**. You might need to do this if your comp

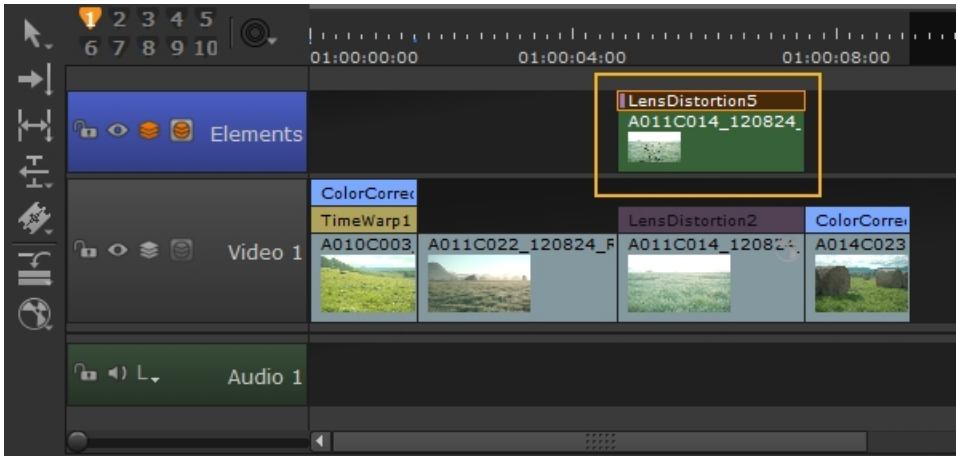
department didn't redistort the comp for you before sending it back from Nuke. In this example, we'll assume that our comp has come back from Nuke, but the CG elements are undistorted.

The shot on the **Elements** track is merged down onto the **Video 1** track, but because the LensDistortion effect is disabled, the original plate is no longer undistorted.



To make the CG elements match the original plate's lens distortion:

1. Copy the LensDistortion effect from the **Video 1** track and paste it on to the CG shot on the **Elements** track.



Note: The copy of the LensDistortion effect is not a clone, so the effect name is incremented as normal. In this case, to **LensDistortion5**.

2. Double-click the pasted LensDistortion effect to display its **Properties** panel.

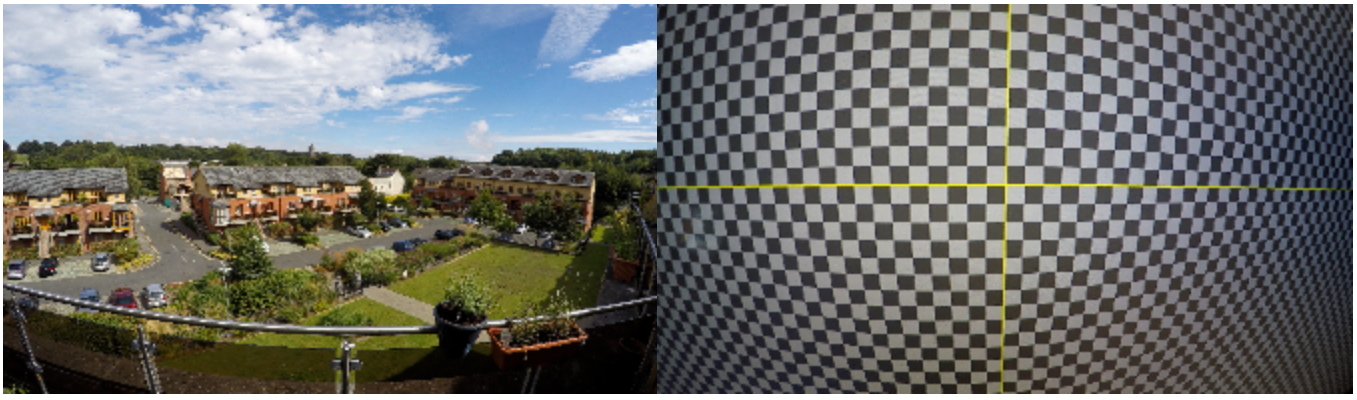
3. Click the **Mode** dropdown and select **Redistort** to invert the distortion applied by the original effect.



Remove Lens Distortion from Multiple Shots

If you intend to undistort multiple shots, it's a good idea to apply LensDistortion to one shot first and then clone the effect to the other shots in case you update the lens solve later on.

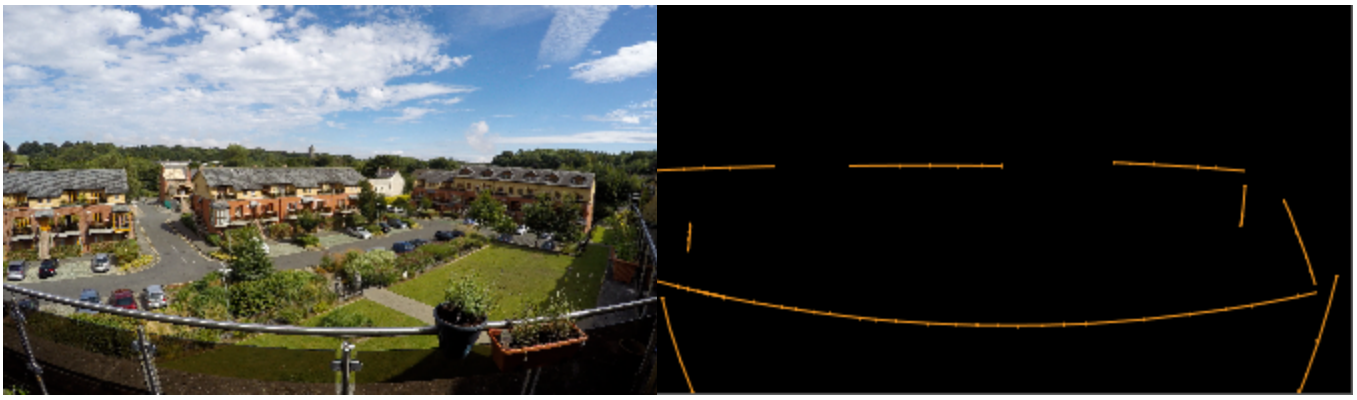
1. Either select the shot on the timeline manually or use the handy auto-select feature to choose a shot for you depending on the current playhead's position.
2. Click the soft effect menu icon on the left-hand side of the timeline.
3. Select **Transformations > LensDistortion** to add the effect to the shot.
4. If you have a lens distortion grid shot with the same camera used for filming, you can use that to estimate the lens distortion applied to your shots,



The original distorted shot | Associated grid shot

OR

If no estimation grid is available, you can use the LensDistortion effect's manual line drawing workflow to estimate lens distortion.



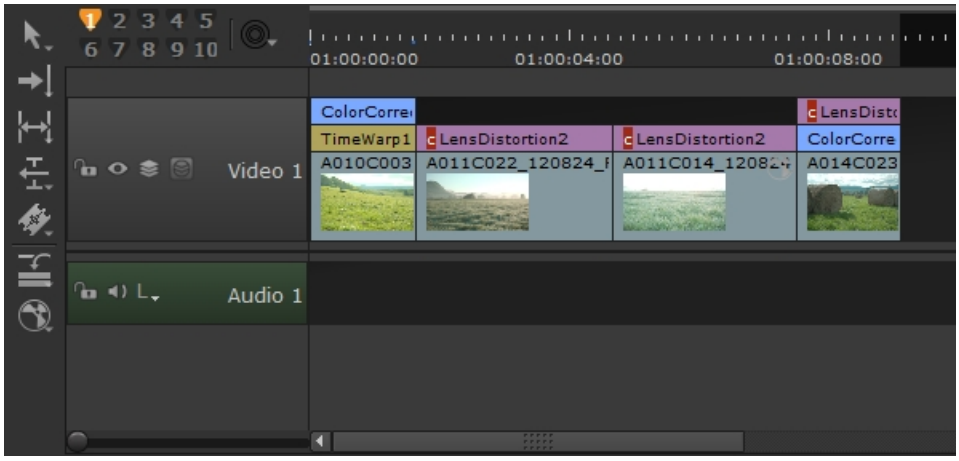
The original distorted shot

Example lines in the alpha channel

See Estimating Lens Distortion Using a Grid or Estimating and Removing Lens Distortion Using Lines for more information.

5. When you've calculated the required lens distortion, make sure that the LensDistortion effect's **Mode** is set to **Undistort** in the **Properties** panel.
6. Select the LensDistortion effect, right-click and choose **Edit** > **Copy As Clones** or press the **Ctrl**/**Cmd**+**K** keyboard shortcut.
7. Select the other shots in the timeline to which you want to add the cloned effect.
8. Right-click the selection and choose **Edit** > **Paste** or press the **Ctrl**/**Cmd**+**V** keyboard shortcut.

The effect is cloned to the selected shots and the LensDistortion effect is applied. Cloned effects are clearly marked with a **C** and retain the same effect name, in this case **LensDistortion2**, so you can quickly locate them.



If you update the lens solve further to improve the results, simply update the clone source LensDistortion effect to apply it to all the clones on the timeline.

9. You can now use **Create Comp** to send all the affected shots to your comp department.

Apply Pre-Trained Models to Shots Using Inference

Nuke Studio and Hiero's soft effect collection also includes an Inference effect in the timeline. The Inference effect applies a pre-trained **.cat** file to create the effect modeled by a neural network across shots on the timeline. The Inference effect is designed to maximize artist efficiency by eliminating the need for round-tripping between the timeline and comp environments in order to see the results of CopyCat training on multiple shots. You can also convert your own models to .cat files using the CatFileCreator, or download open-source models from Cattery for use on the timeline

See Train Neural Networks to Replicate Effects Using Machine Learning and Import Pre-Trained PyTorch Models for more information on creating and sourcing **.cat** files for use in the Inference effect.



Note: You can use **.cat** files requiring multiple inputs with the **Inference** soft effect, using the Channels In and Channels Out options.

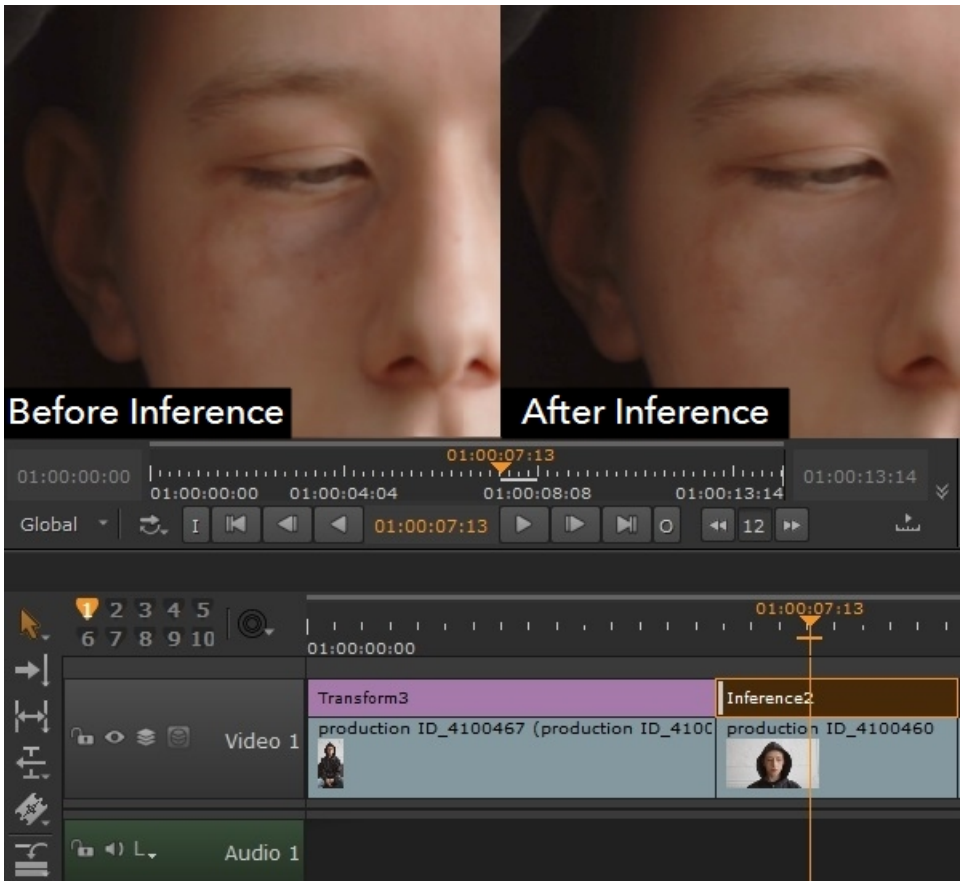
Add an Inference Effect to a Shot

So you've trained your model and it's ready to go. In this example, we've trained a model to remove a bruise from an actor's face. Here's how to apply the model to a shot on the timeline.

1. Either select the shot on the timeline manually or use the handy auto-select feature to choose a shot for you depending on the current playhead's position.
2. Click the soft effect menu icon  on the left-hand side of the timeline.
3. Select **Inference** to add the effect to the shot.

   An error is displayed in the Viewer because no **.cat** file is selected by default.
4. In the **Properties** panel, enter a **Model File** path to the **.cat** file you want to apply to the shot.

   The model is applied to the shot and you'll see the result in the Viewer.



Note: Depending on the model's size and complexity, there may be a delay while the effect is applied to the shot.

5. You can enable **Optimize for Speed and Memory** if you want a faster preview of the results. The **Properties** panel also displays the **Channels In** and **Channels Out** for reference.

Tip: You can also apply Inference soft effects to an entire sequence by creating a new track and adding the effect to the track, instead of individual shots. See Adding Sequence-Level Soft Effects for more information.

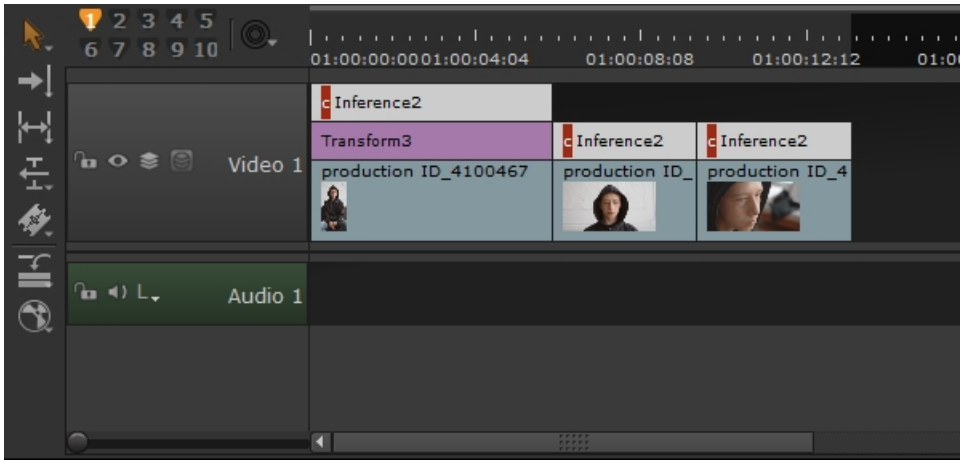
Apply an Inference Effect to Multiple Shots

You can easily test how well the Inference effect works on other shots by simply copying and pasting the effect onto those shots. However, once you intend to use an Inference effect on multiple shots, it's a good idea to apply it to one shot first and then clone the effect to the other shots in case you update the **.cat** file

later on with further training. Another useful method is to apply the soft effect to a track above the clip, then extend it to cover as many clips on the tracks below as needed. You can also cut the soft effect into pieces for more control over which clips it applies to.

1. Follow the steps above to add an Inference effect.
2. Clone the Inference effect onto other shots in the timeline. See Cloning to learn how to clone a soft effect.

   The effect is cloned to the selected shots and the Inference effect is applied. Cloned effects are clearly marked with a **C** and retain the same effect name, in this case **Inference2**, so you can quickly locate them.

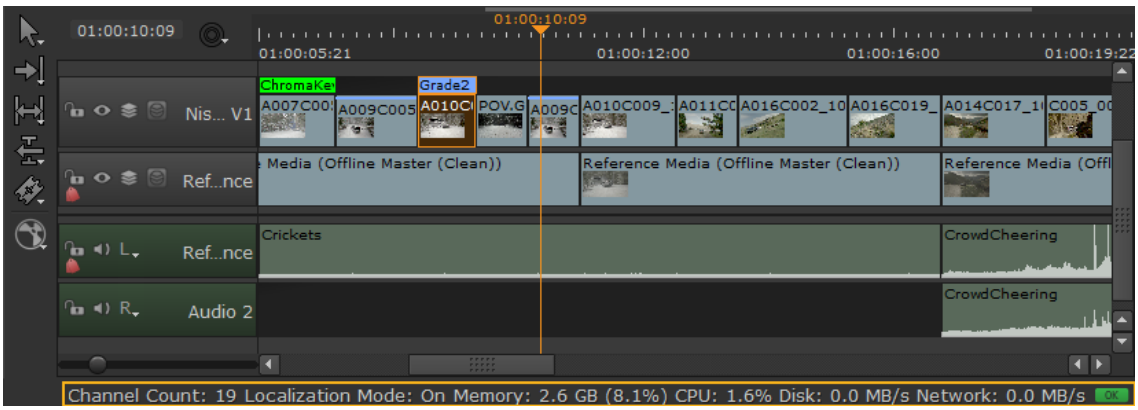


   If you train the model further to improve the results, simply update the clone source **Model File** control with the new **.cat** file to apply it to all the clones on the timeline.

Create Comp

In the Timeline environment, you can choose to create a Nuke Comp of a shot to be able to open it in the Compositing environment. You can then add any necessary compositing work and render out the script.

You can only create comps from the timeline when the Frame Server is running. You can check the status of the server in the status bar at the bottom-right of the interface.



Before you create a Nuke Comp, you can choose to change **Export** settings, and set the required **Shot Preset** setting in the **Project Settings** dialog to get the required result.

It is not necessary to change the **Export** and **Project Settings**. If you don't change these settings when creating a Nuke Comp, the default settings are used.

Note: You can use Nuke Studio's internal Frame Server to take advantage of external worker machines to process renders faster. See Using the Frame Server on Render Machines for more information.

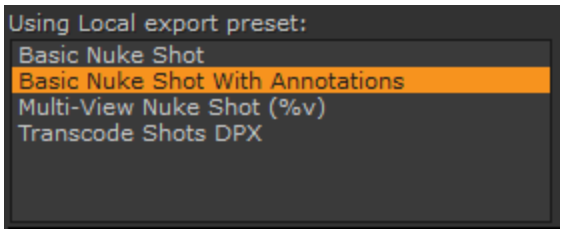
Edit Export Settings

If you do not change the export settings, the **Using Local export preset setting** control just uses the default setting of **Basic Nuke Shot With Annotations**.

To change this setting, do the following:

1. Select **File** > **Export...** or press **Ctrl+Shift+E**.

   This opens the **Export** dialog.
2. Go to the **Custom** tab. In the **Using Local export preset:** section, you can select the type of shot preset to export when you use **Create Comp**.



   For more information, see Introduction to the Export Dialog.
3. Select the required setting and then close the **Export** dialog.

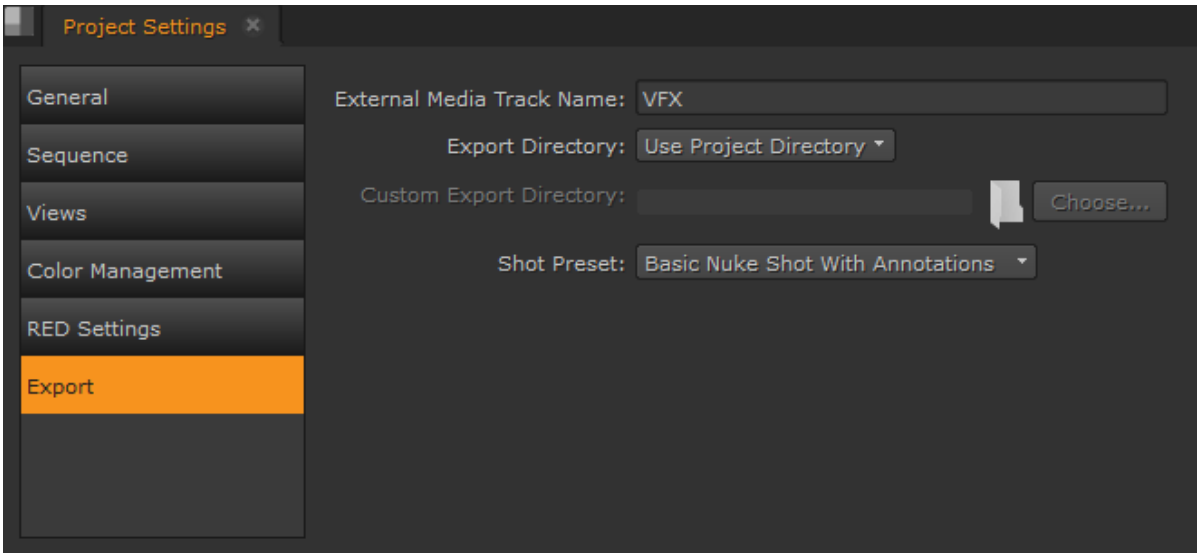
Edit Project Settings

If you do not change the **Project Settings**, the **Shot Preset** control just uses the default setting of **Basic Nuke Shot With Annotations**.

To change the **Shot Preset** setting in the **Project Settings** dialog, do the following:

1. Select **Project** > **Edit Settings**.
2. Open the **Export / Roundtrip** section by clicking on it.
3. Use the **Shot Preset** dropdown to select your required setting.

Note: You can use custom shot presets, but they must contain a **Nuke Project File** and **Nuke Write Node** Content preset to be valid for selection. See Using Local and Project Presets, Nuke Project File Settings, and Nuke Write Node Settings for more information.



4. After selecting the required setting, close the **Project Settings** dialog.

Creating and Editing a Comp

To create a Nuke Comp of a shot, do the following:

1. Right-click on the shot that you want use to create a Nuke Comp.
2. Select **Create Comp** > **Create Comp**.

   If you have not set a project root folder up in the **Project Setting** dialog, Nuke Studio asks you to set a project root folder.
3. Click **Choose**, select to the required location, and then click **OK**.

   A Nuke Comp is created, and placed directly above the original shot on the next available track.



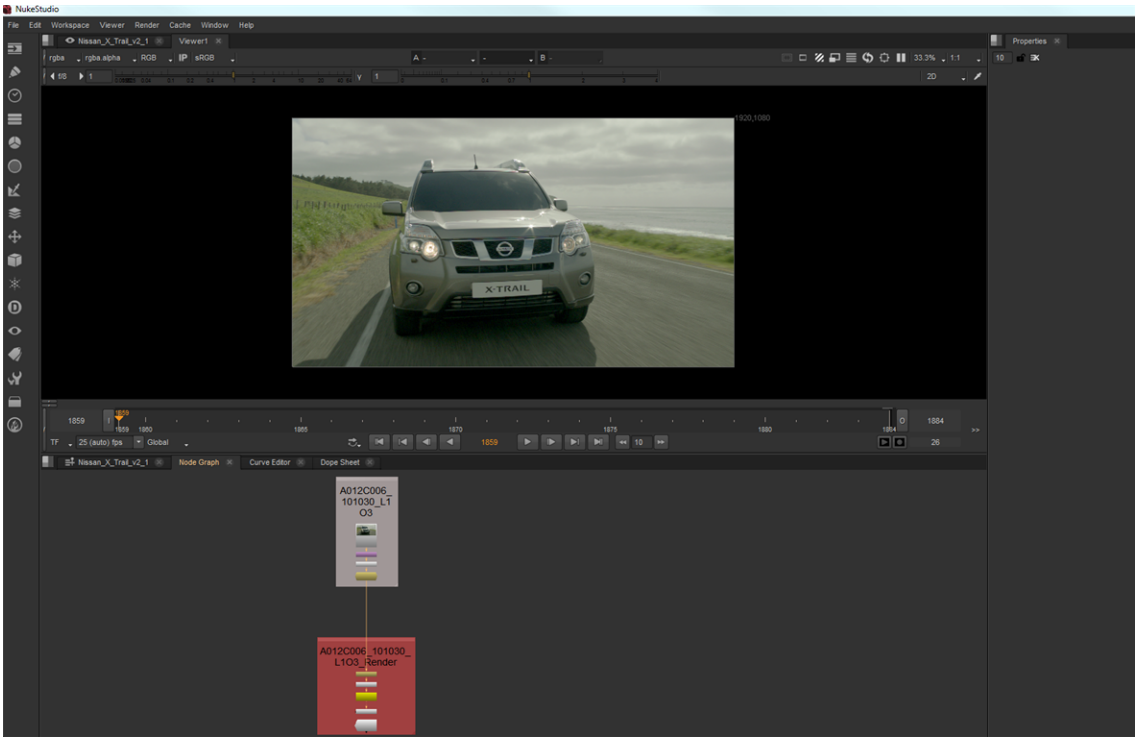
The Nuke Comp is displayed in light red, signifying that the Nuke Comp has not been rendered. You can choose to render it by either, right-clicking and selecting **Render Comp**, or by selecting **Render** > **Render Selected Comp Containers**.

Note: You can render all Nuke Comps by selecting **Render** > **Render All Comp Containers**.

4. Open the Nuke Comp in the Compositing environment in the same session by double-clicking the Nuke Comp. You can also open the Nuke Comp in the Compositing environment in the same session by either right-clicking on the Nuke Comp above the timeline, or in the Project bin and selecting **Open In** > **Node Graph**.

   To open the Nuke Comp in a new NukeX session you can either hold **Ctrl/Cmd** and double-click on the Nuke Comp, or you can right-click on the Nuke Comp and select **Open In** > **New Nuke Session**.

   If you have not set a project root folder up in the **Project Setting** dialog, Nuke Studio asks you if you want to save and set a project root folder.



You can now edit the script and add VFX using any of the tools available in the Compositing environment. See Using the Compositing Environment.

Nuke Comp Colors

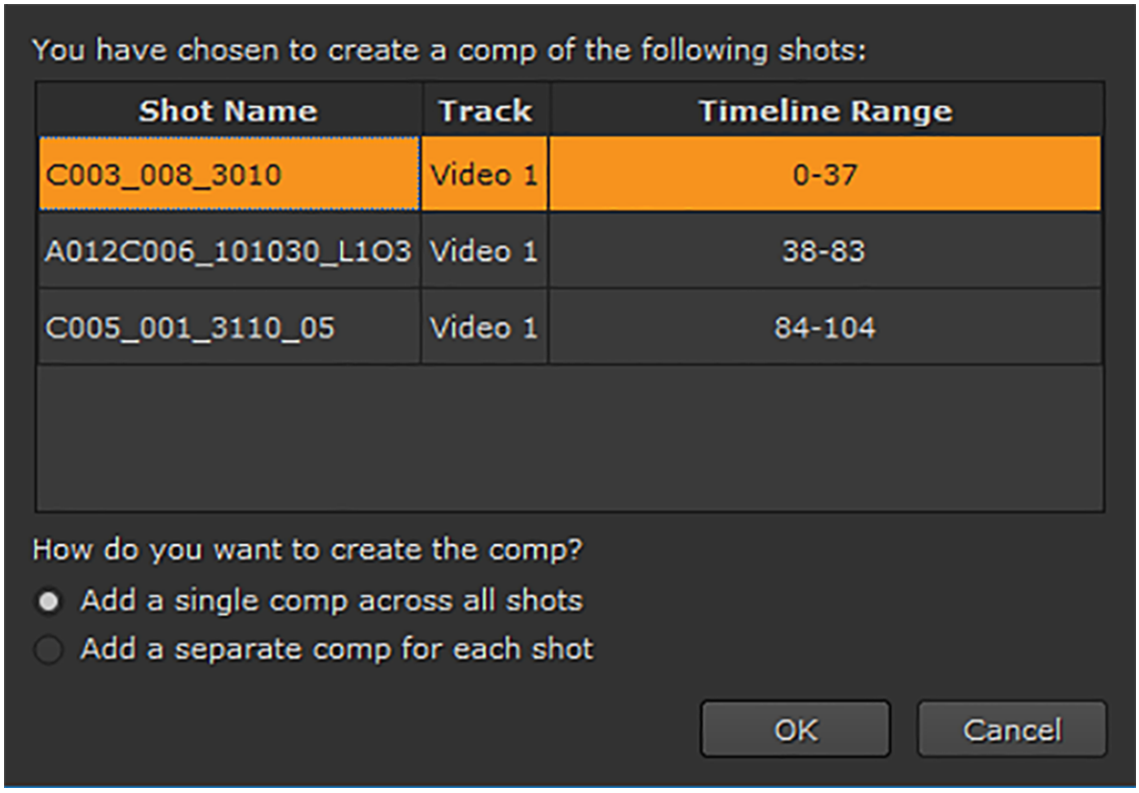
Color of Comp		Meaning
Light Red		This signifies that the comp has not been rendered.
Dark Red (with **OFF**)		This signifies that the comp is offline.
Light Green (with progress bar)		This signifies that the comp is being rendered.
Dark Green		This signifies that the comp has been rendered.

Creating Multiple Shot Comps

You can create Nuke Comps from multiple shots on a single track or across multiple tracks, by doing the following:

1. Select multiple shots on the timeline by using **Shift**+click to select a row of adjacent shots and/or **Ctrl**/**Cmd**+click to select multiple non-adjacent shots. These can be across multiple tracks.
2. Right-click on one of the selected shots and select **Create Comp** > **Create Comp**.

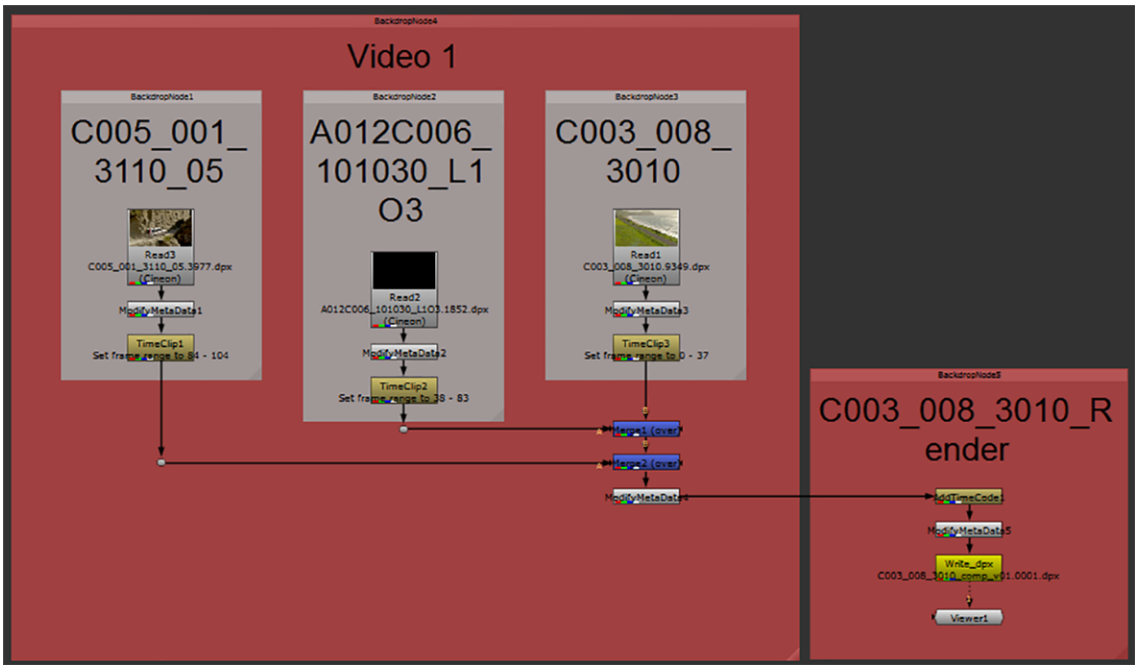
   This opens a **Create Comp** dialog.



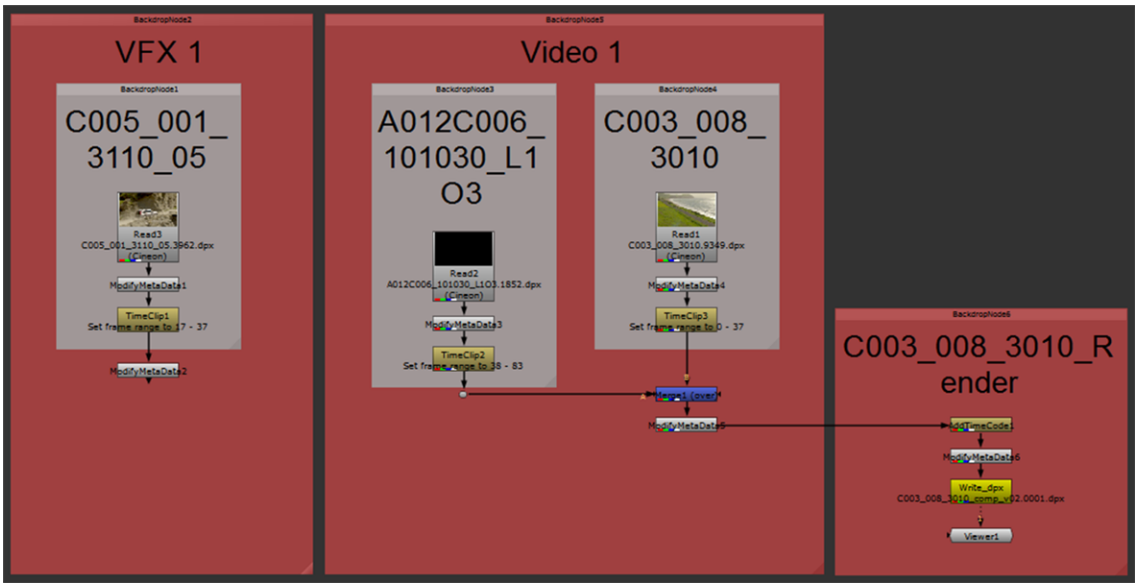
3. The **Create Comp** dialog displays all the selected clips, and allows you to select a main clip from the list simply by clicking on it.

   The main clip is highlighted yellow.
4. To create one Nuke Comp for all the shots, select the **Add a single comp across all shots** radio button. Or, if you want to create a Nuke Comp for each of the selected shots, select the **Add a separate comp for each shot** radio button.

   When you create a Nuke Comp on multiple shots on a single track, the Node trees are automatically connected and are displayed in reverse chronological order, as shown below.



When you create a Nuke Comp on multiple shots across different tracks, the Node trees from the same track are connected, but the Node trees on different tracks are not connected by default (see below). You can choose to have them connected using the **Create Comp Special** settings. See Create Comp Special for more information.



Create Comp Special

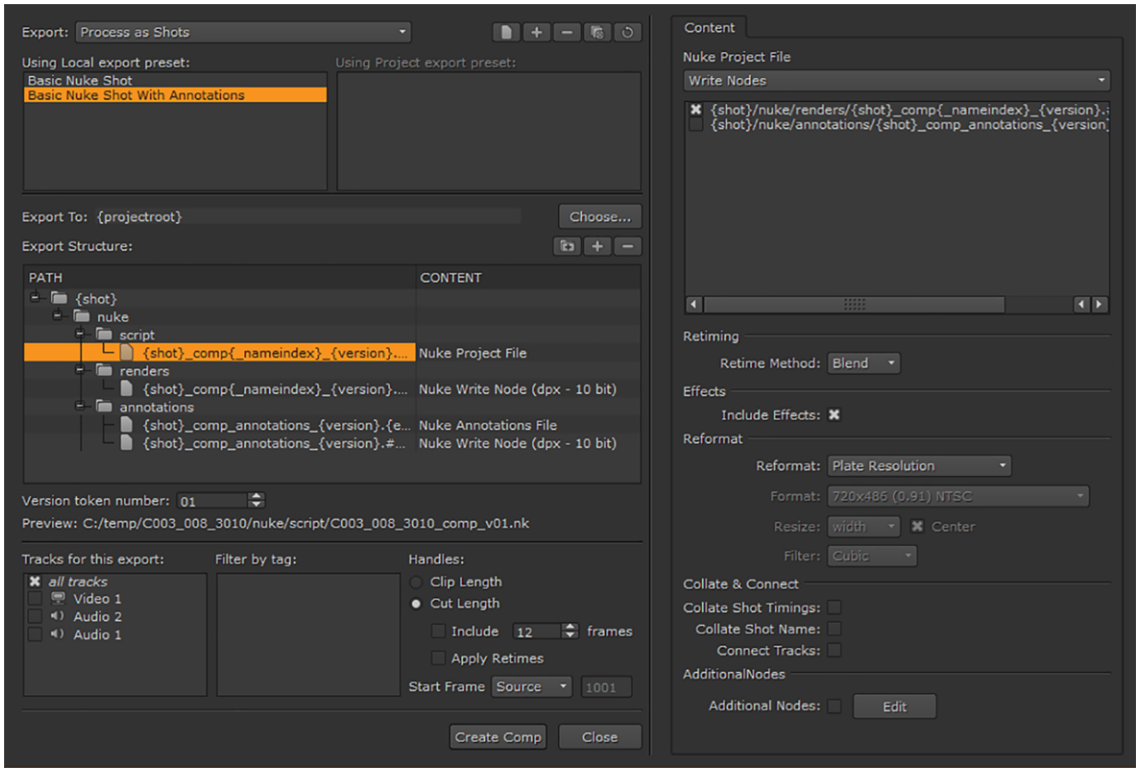
You can also choose to create a Nuke Comp using **Create Comp Special**. When you use the **Create Comp Special** option, a dialog containing export options is automatically opened, allowing you to set the export

settings for the selected Nuke Comp.

To use the **Create Comp Special** option:

1. Right-click on the shot that you want use to create a Nuke Comp.
2. Select **Create Comp** > **Create Comp Special...**

   This opens a **Create Comp** dialog that contains the available export settings.



   You can use this dialog to set the export location, set a version number, define the tracks to export, filter by tag, and define the range. See Introduction to the Export Dialog for more information.
3. Select the Nuke script in the **Path** section of the dialog, and then go to the **Content** section.

   A number of additional controls are displayed. You can use these to specify what your Nuke Comp includes and how it is created. You can select the **Connect Tracks** checkbox to connect all the Node trees – across different tracks – in the Node Graph. See Introduction to the Export Dialog for more information about these options.
4. After specifying your required settings, click **Create Comp**.

   A dialog appears stating you have changed the export templates. You can choose to keep the changes by selecting **Yes** or click **No** to discard the changes.

   The Nuke Comp is displayed in light red, signifying that the Nuke Comp has not been rendered. You can choose to render it by either, right-clicking and selecting **Render Comp**, or by selecting **Render** > **Render Selected Comp Containers**.
5. Double-click on the Nuke Comp to open it in the Compositing environment.

You can now add VFX using any of the toolsets available in the Compositing environment. See Compositing with Nuke.

Enabling and Disabling a Nuke Comp

You can disable and re-enable a Nuke Comp on the timeline by selecting the Nuke Comp you want to disable and pressing **D**.

Saving a New Version and Updating the Comp

When you are happy with the changes you've made, you can save a new version of the script by selecting **File** > **Save New Comp Version** (or pressing **Alt**+**Shift**+**S**).

After saving a new version of the Comp, you can update the original Nuke Comp you created. To version-up the Nuke Comp, do the following:

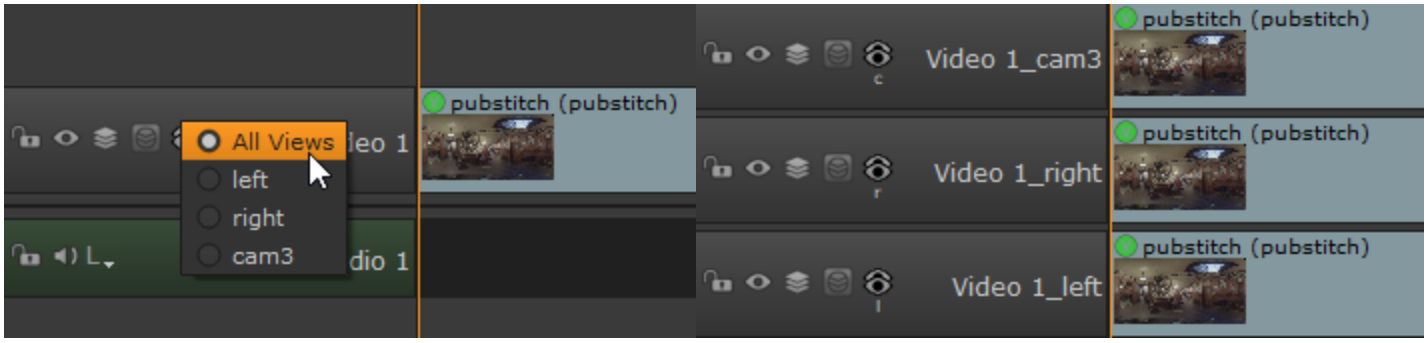
1. On the timeline, right-click the original Nuke Comp.
2. Select **Versions** > **Version Up**.

   The Nuke Comp is versioned-up. Depending on your **Preferences** > **Performance** > **Threads/Processes** > **Rendering** > **background renders** setting, the comp may need rendering manually.

Stereo and Multi-View Comps

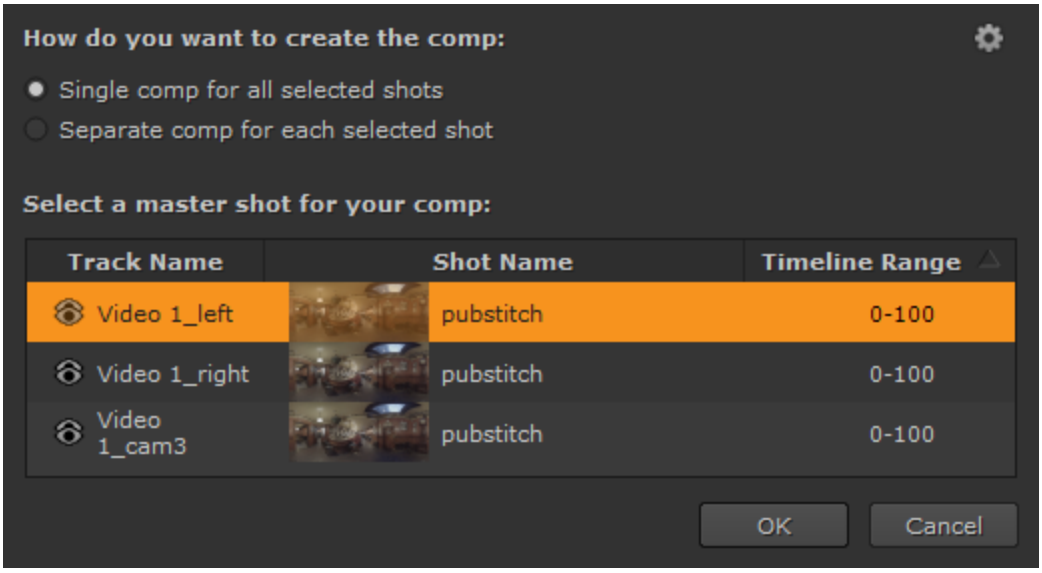
Multi-view comps are similar to regular comps, but you can create a single comp with all views or separate comps per view. The **Multi-View Nuke Shot** template is selected automatically when you create a comp from a multi-view project.

1. Right-click on the shot that you want use to create a multi-view Comp. The shot can be a multi-view track or single views split into separate tracks.

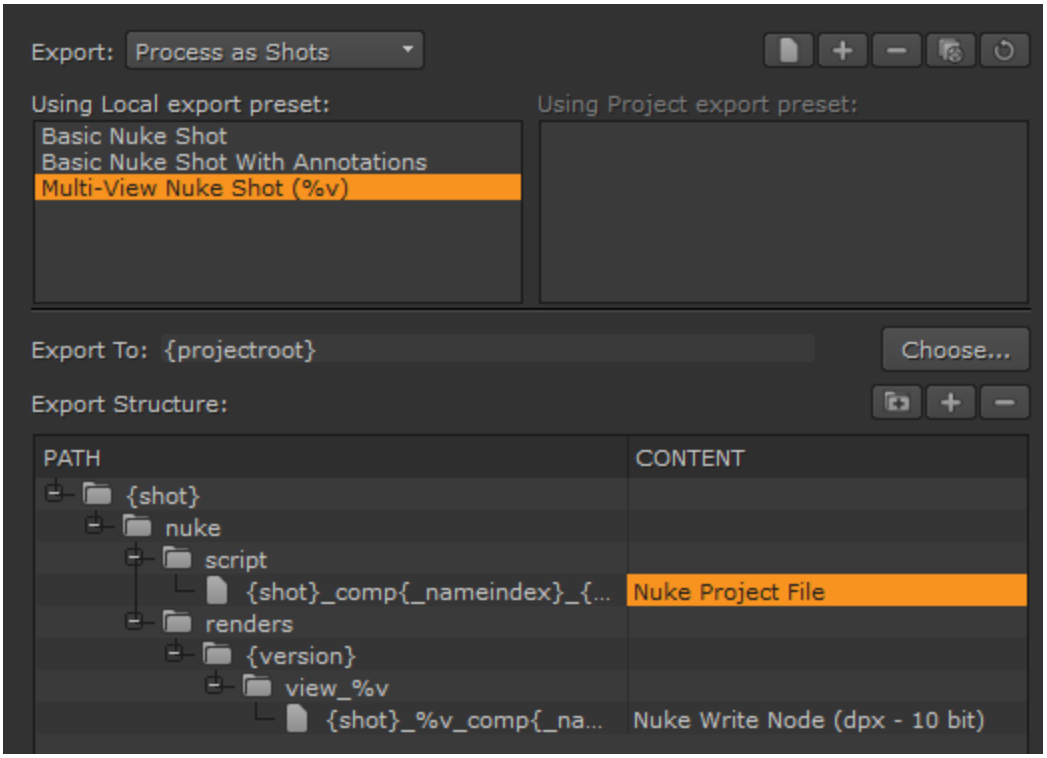


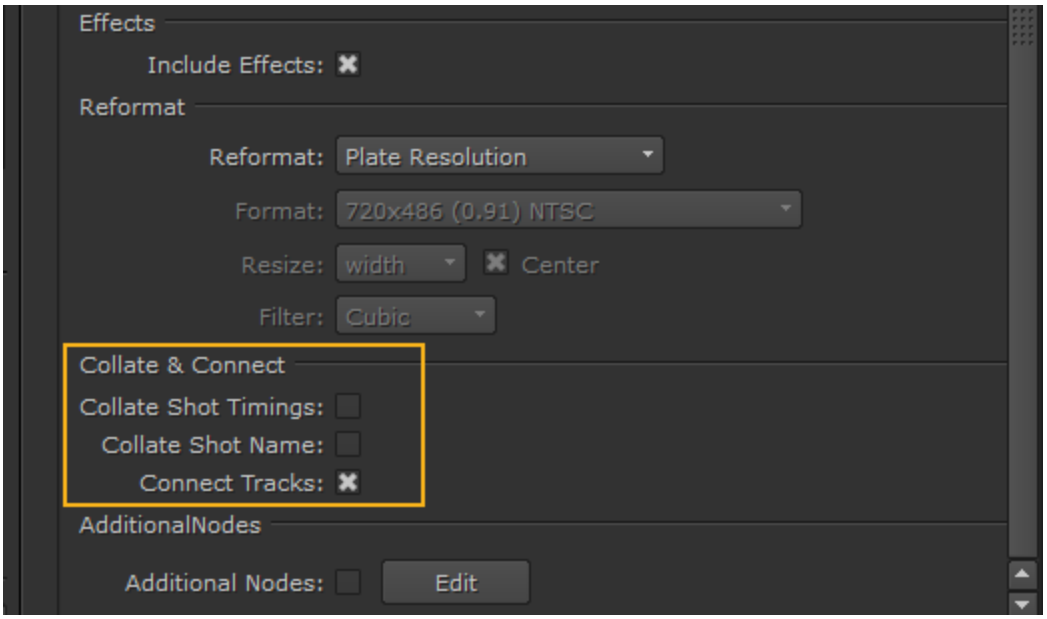
For separate tracks, you can right-click a single track to export all views or select all the per-track views and right-click.

2. Select **Create Comp** > **Create Comp**.
3. If you selected multiple per-track views to create the Comp, you can create a single Comp containing all views or a separate Comp per view and select the main shot.



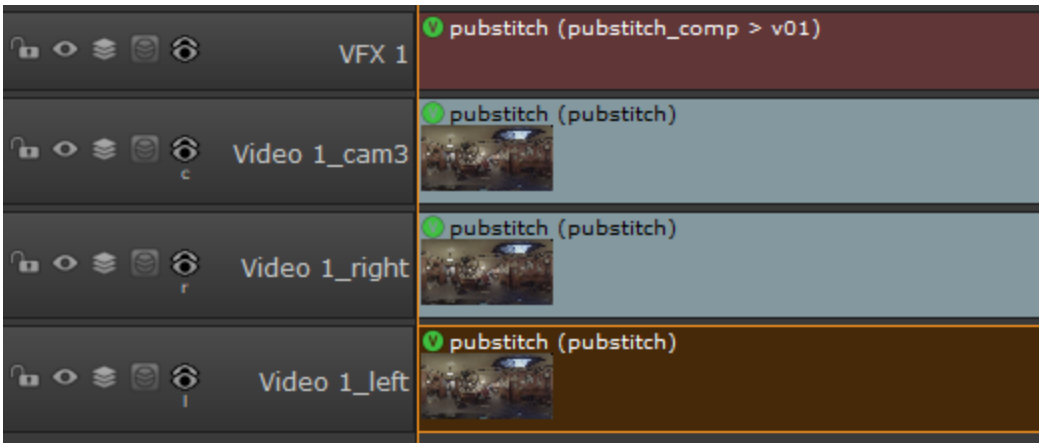
If you're using independent files per track, that is without importing multi-view files or using %V functionality, the separate tracks in the Comp script are not connected by default. If you want the tracks to be connected in the script, click the cog icon to display the **Create Comp** dialog and enable **Connect Tracks** in the **Nuke Project File** preset.





4. Click **OK**.

   A Nuke Comp is created, and placed directly above the original shot on the next available track.



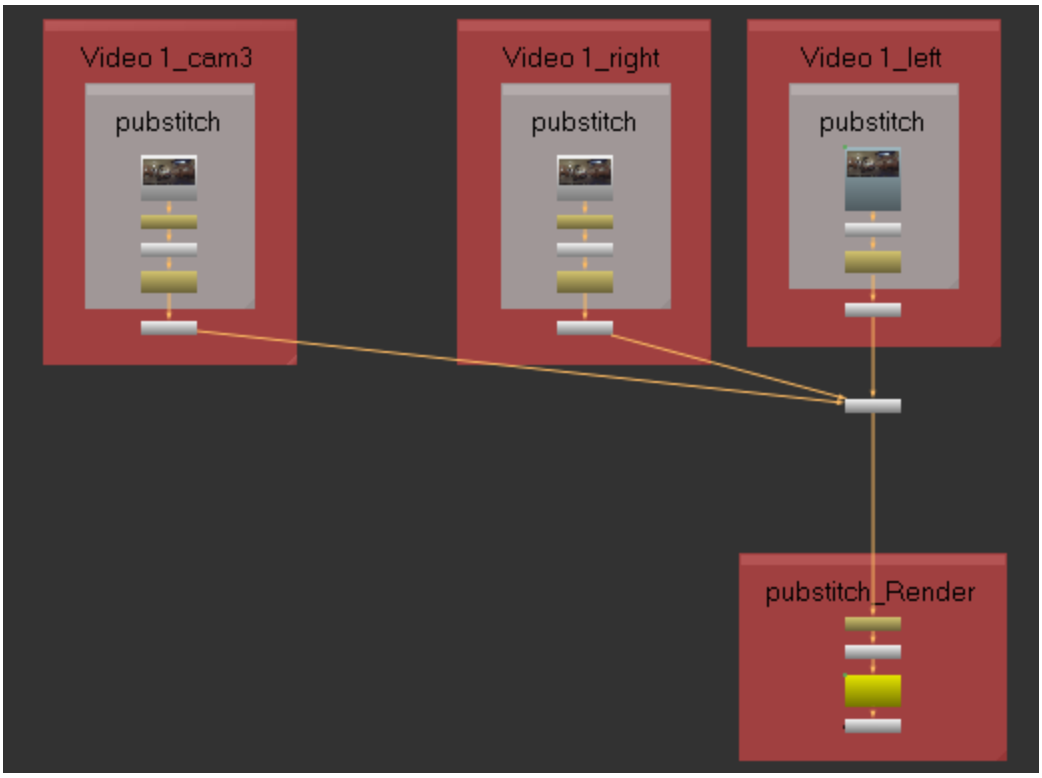
The Nuke Comp is displayed in light red, signifying that the Nuke Comp has not been rendered. You can choose to render it by either, right-clicking and selecting **Render Comp**, or by selecting **Render** > **Render Selected Comp Containers**.

Note: You can render all Nuke Comps by selecting **Render** > **Render All Comp Containers**.

5. Open the Nuke Comp in the Compositing environment in the same session by double-clicking the Nuke Comp. You can also open the Nuke Comp in the Compositing environment in the same session by either right-clicking on the Nuke Comp above the timeline, or in the Project bin and selecting **Open In** > **Node Graph**.

   To open the Nuke Comp in a new NukeX session you can either hold **Ctrl**/**Cmd** and double-click on the Nuke Comp, or you can right-click on the Nuke Comp and select **Open In** > **New Nuke Session**.

   If you have not set a project root folder up in the **Project Setting** dialog, Nuke Studio asks you if you want to save and set a project root folder.



You can now edit the script and add VFX using any of the tools available in the Compositing environment. See Using the Compositing Environment.

Annotations

In Nuke Studio and Hiero, you can use annotations to add instructions and feedback for compositors. You can add annotations to a single frame, a frame range or timecoded section marked with in and out points, a clip, or a whole sequence.

The **Annotations Panel** allows you to post comments and notes in reply. The **Annotations Toolbar** includes a variety of painting tools to help visually indicate your thoughts on the images in the Viewer.

Workflow

The following steps show an example of Nuke Studio workflow for annotations:

1. In the Viewer, open the Annotations Tools by selecting the annotations button  in the Viewer. See Annotations Tools.
2. Add an annotation to a Clip or a Sequence by using the Annotations Panel and Annotations Toolbar. See Adding Annotations.
3. You can choose to edit an annotation after it has been created. See Editing and Deleting Annotations.
4. You can respond to comments by leaving Notes. See Adding Notes.
5. You can burn annotations in via Quick Export. See Rendering Out Annotations.
6. You can export frames with annotations to .png or .jpg via Custom Export. See Export to PNG or JPEG.
7. You can use Create Comp to view your annotation comments in a Compositing environment. See Create Comp with Annotations.

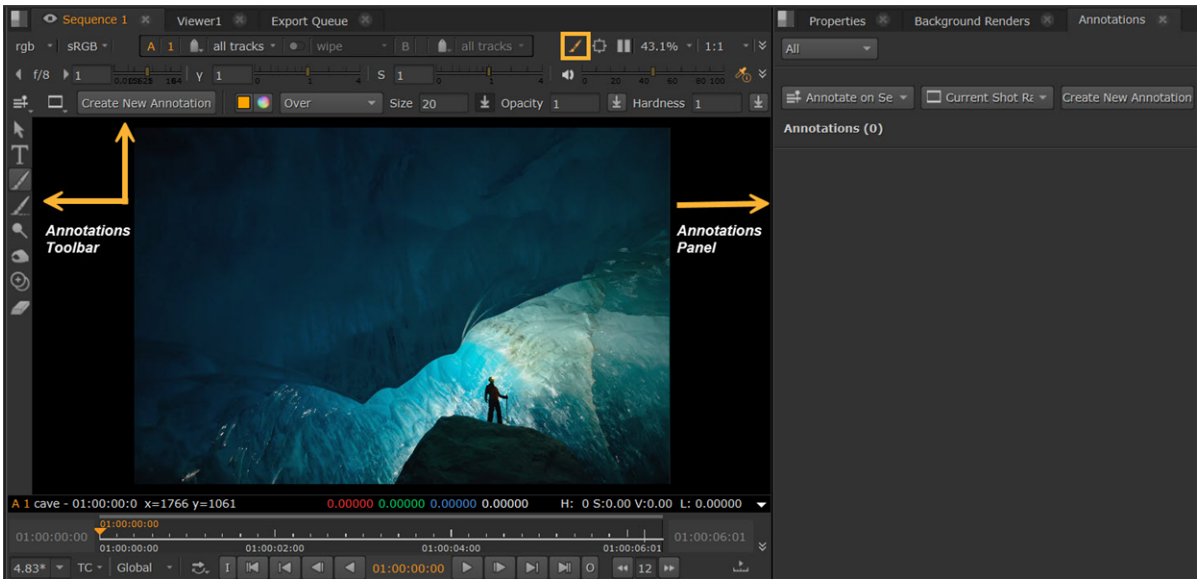


Note: This workflow is also supported in Sync Review sessions.

Annotations Tools

Open the Annotations Tools by selecting the annotations button  at the top-right of the Viewer.

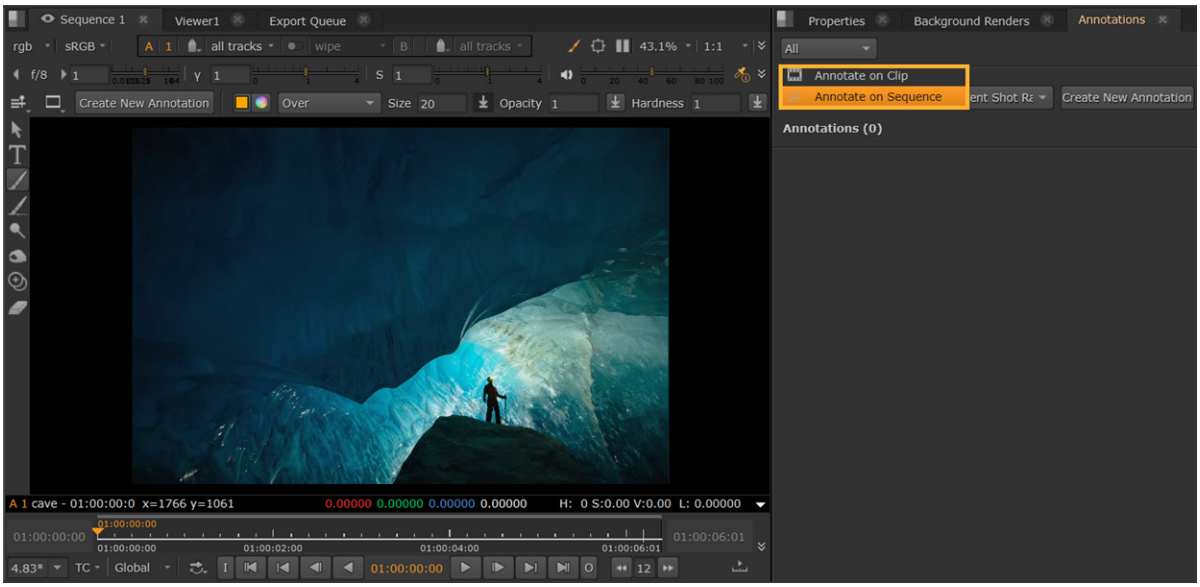
When in the default Nuke Studio workspace, this will open both the **Annotations Panel** on the right, and an **Annotations Toolbar** down the left side and horizontally above the Viewer.



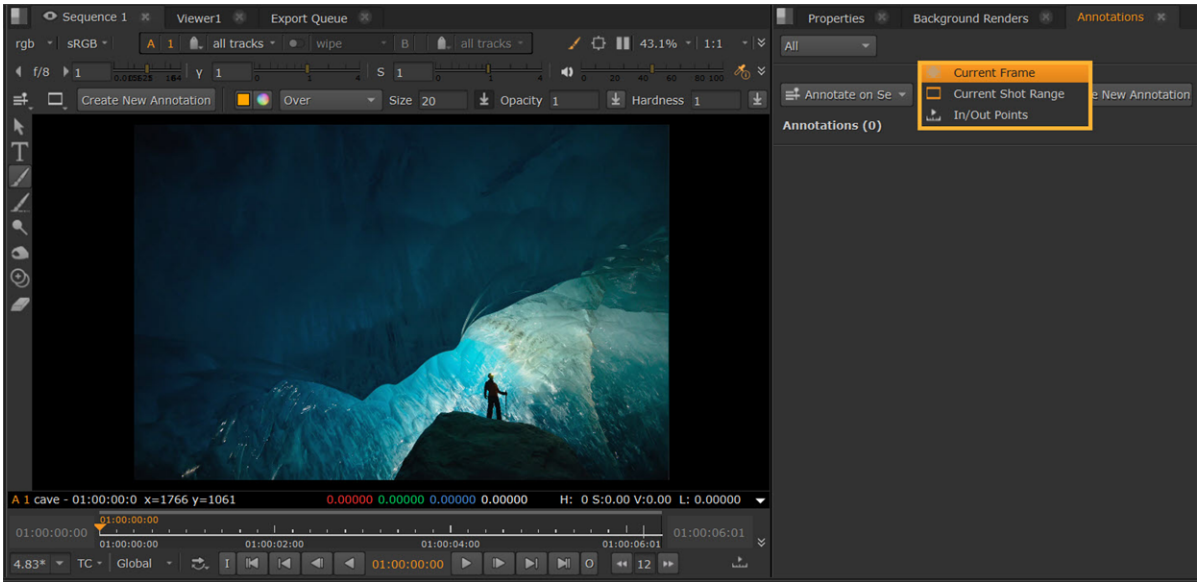
Adding Annotations

To add an annotation, do the following:

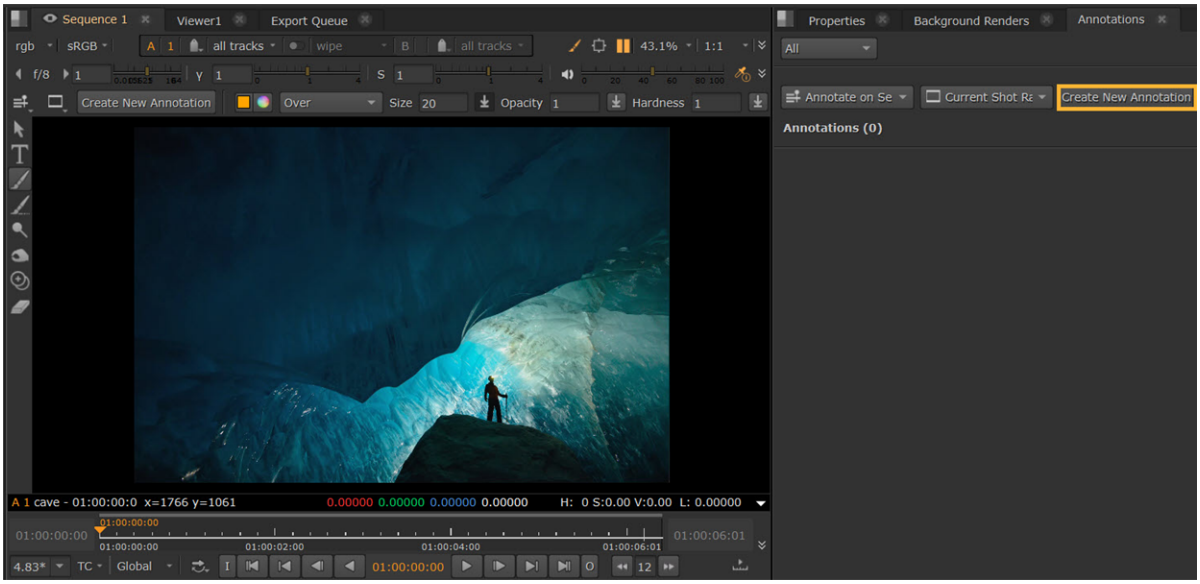
1. After you have opened the Annotations Panel, move the playhead on the timeline to where you want to add your annotation. Select either **Annotate on Clip** or **Annotate on Sequence** from the dropdown.



   Select whether to add the annotations to **Current Frame**, **Current Shot Range**, or **In/Out Points** on the dropdown.



Then select **Create New Annotation**.

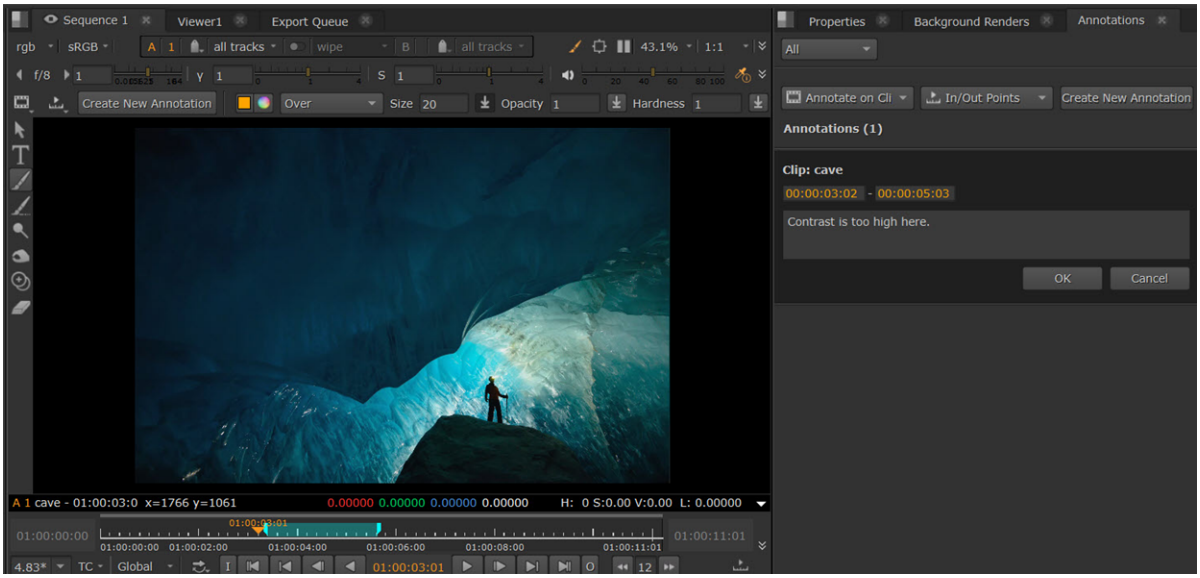


2. A text field to start typing a comment opens automatically. Select **OK** when you have finished typing.

Tip: To finish typing, you can also use **Ctrl**+**Enter**. To cancel you can press the **ESC** key.

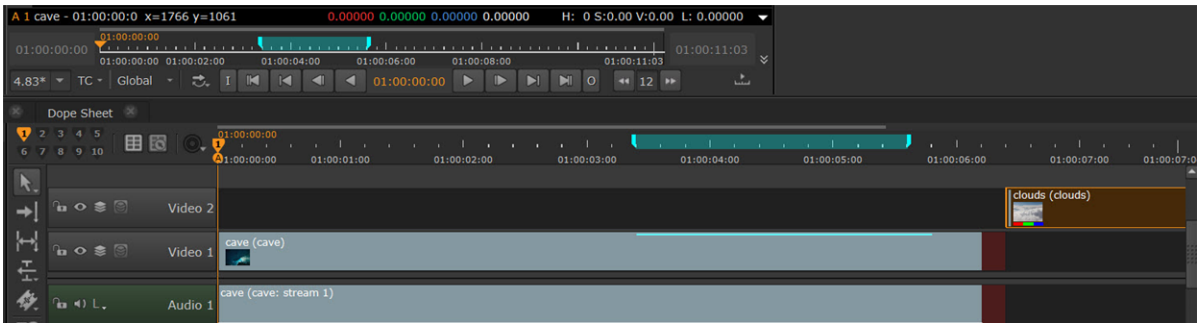
Tip: Use **Ctrl**+**N** while the panel is selected to create a new annotation. This functionality only works in the panel, not in the Viewer.

You can adjust the frames or timecode in and out points via the orange fields.



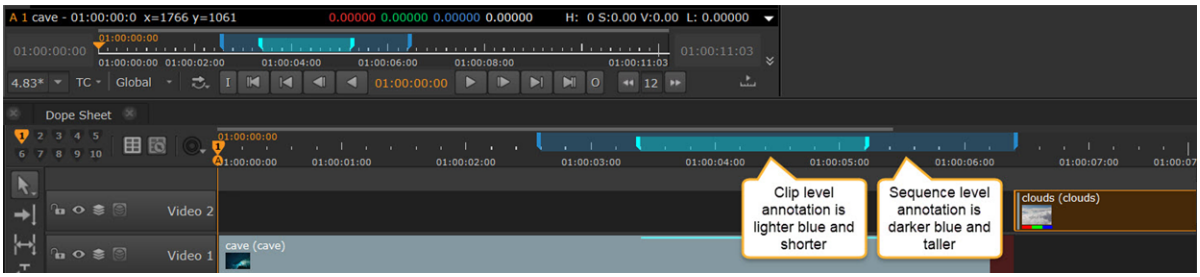
Tip: The Annotations Panel supports emojis and various language characters. It also supports some Markdown syntax, so you can easily make text bold (using ** **), add italics (using * *), bullet lists (using -), and more. You can read more about Markdown syntax on this Markdown Guide webpage, though note that not everything is supported currently.

3. Use the Annotation Toolbar to add any visual annotations to the image in the Viewer, such as using brushes. See Painting with the Annotations Toolbar. Any changes you make here are saved to the same timecode, and are linked to your comment.
4. You'll notice that blue markers are added to the timeslider and timeline to show where your annotation is located. These can be adjusted directly by dragging the ends of the markers. Holding **Shift** also moves either in or out.

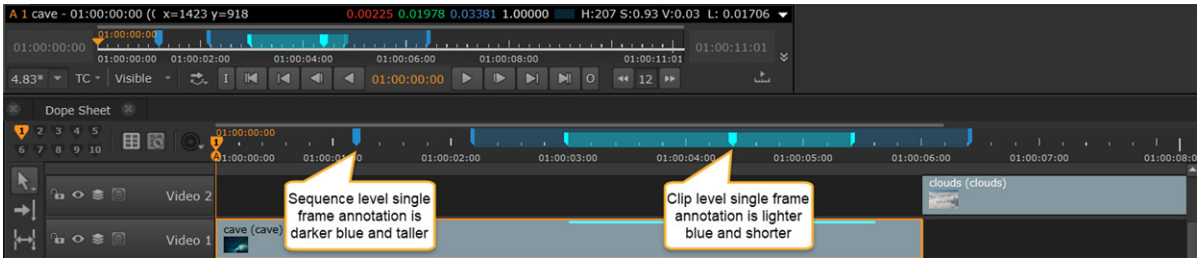


Note: There is no longer a separate Annotations track on the Timeline and your annotations are marked by a blue chip.

If there are multiple annotations in one area, you can see them both. Sequence level annotations are taller and darker blue, clip level annotations are shorter and lighter blue/cyan.

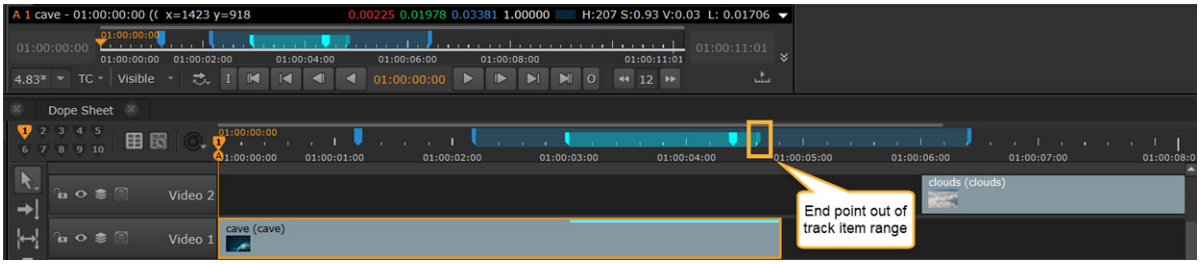


Single frame annotations are shown by a single marker on the timeline, with the same size and color differentiation for clip and sequence level.



Moving or trimming a clip moves its annotation with it. This is because clip level annotations are attached to the clip they are added to.

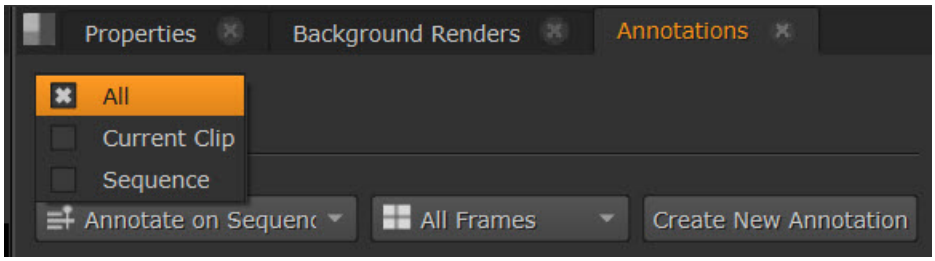
If, once moved or trimmed, the annotation remains inside the track item range, it continues to display markers. However, if you trim the clip in a way that removes the annotated frame, the marker(s) is no longer displayed.



However, it is still attached to the clip, so if you then re-open the clip from the project bins in Timeline view, you can see the annotation there once more.

5. Once annotations have been added, the Annotations Panel shows comments for the active Viewer, if any exist.

   To find specific annotations, you can also select the relevant blue marker on the timeslider to have it display on the Annotations Panel and in the Viewer. Similarly, clicking the comment in the Annotations Panel jumps to the relevant point on the Timeline.
   By default, in the Annotations Panel, the comments are ordered by where they are placed in the sequence. You can also use the Filter dropdown at the top of the Annotations Panel: **Current Clip**, **Sequence**, and **All** to only show comments from specific clips or just the sequence.



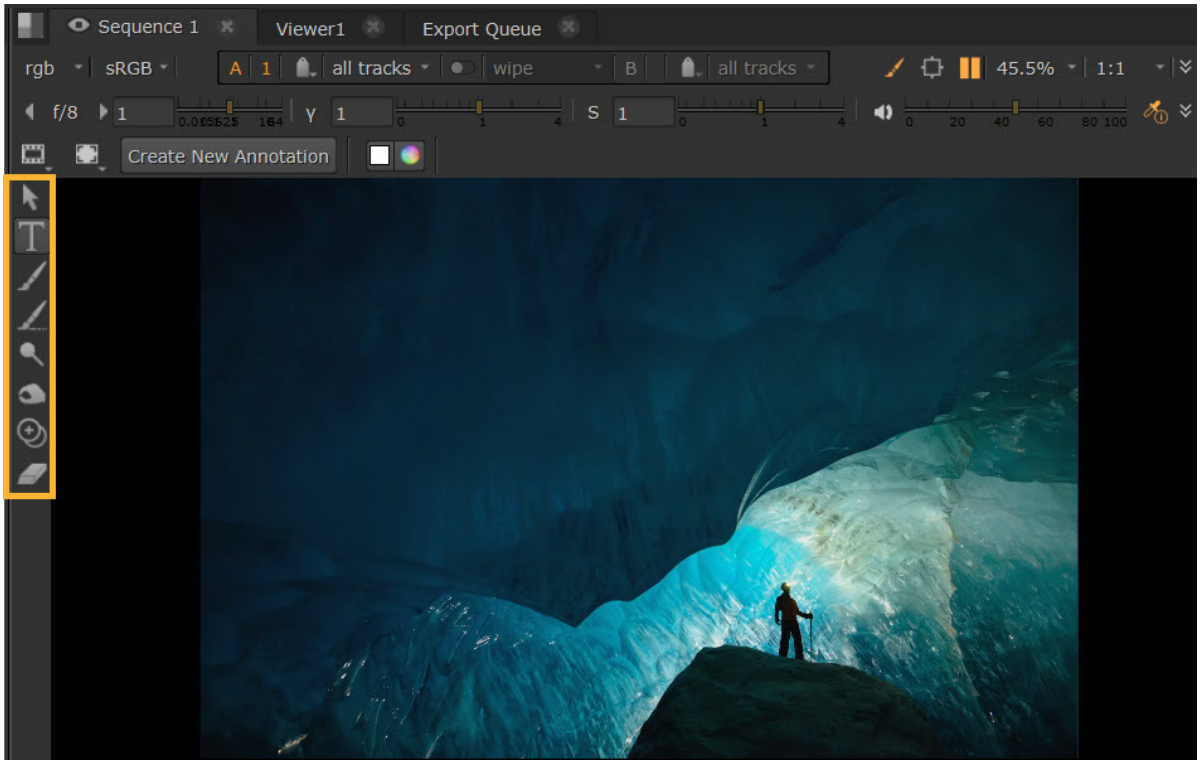
Note: You can use the Python API to set metadata on comments. Using this you can customize the Annotations Panel in a variety of ways, such as setting comment usernames, adding background colors for comments, and locking annotation editing. See the HieroPythonDevGuide to learn more.

Painting with the Annotations Toolbar

You need the annotations button  toggled on to start painting. If you already have an annotation created in the Annotations Panel at this location, the drawn annotation is now part of that. If you don't already have an annotation created in the Annotations Panel at this location, one is created when you start to draw. If you have an annotation at this location but want another annotation at this location, that's possible by clicking **Create New Annotation** again.

You can choose between using **Text**, **Brush**, **Vanishing Brush**, **Dodge**, **Burn**, **Clone**, and **Eraser** mode by selecting the brush selection buttons in the Annotations Toolbar. Selecting each one will bring up different options in the horizontal toolbar.

Apply any drawing directly onto images in the Viewer. Any drawing you do will be saved to the same timecode or frames as the annotation that you have open.



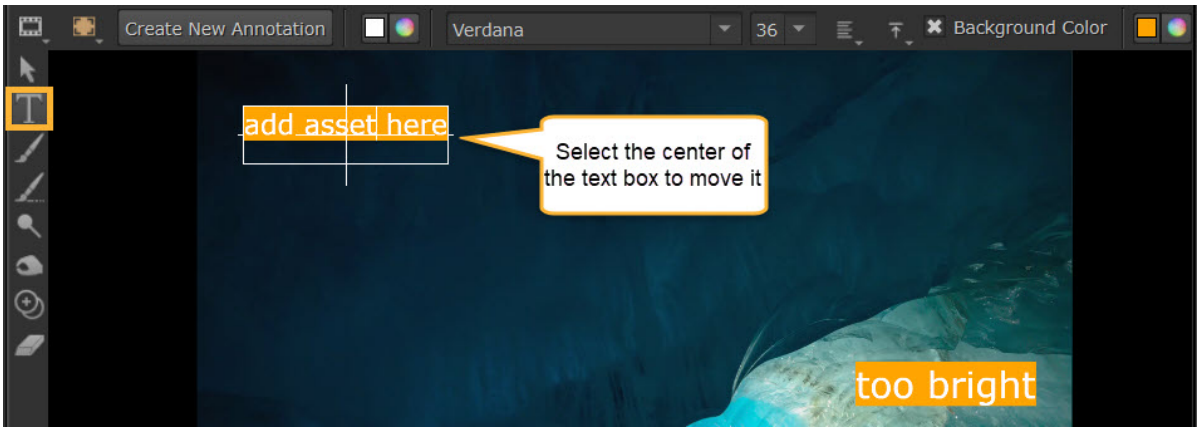
Text

To use the Text tool, simply click in the Viewer to create a text box and start typing into it.

The Text tool shows options to set the font color, style, and size. The font style dropdown will show all fonts on your system.

There are also alignment tools for horizontal and vertical alignment, and you can set a background color by toggling it on or off via the checkbox.

You can move a text box by clicking and dragging the center of the cross in the center of the text box. You can also rotate by using the longer right axis.



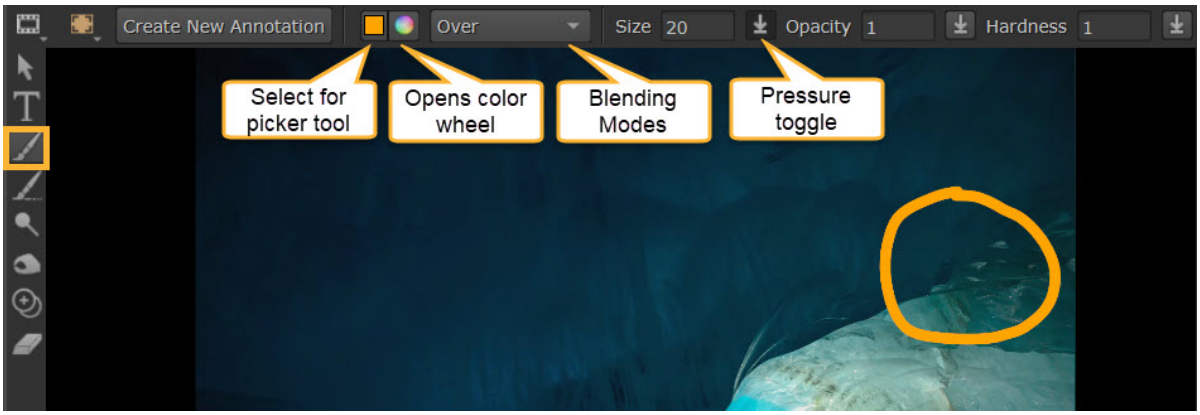
Tip: Use **T** keyboard shortcut to enable the Text tool. Use **Esc** to exit.

Tip: The text tool supports emojis and various language characters.

Note: The text written with the Text tool is available via Python, so that text can be uploaded to the production tracking software and be searchable if needed.

Brush

By enabling the Brush tool, you can paint directly onto the Viewer. The Brush tool will show options for color, blending modes, brush size, opacity and hardness.



- The color wheel icon opens a color wheel to pick from, or select the color square icon for a color picker tool to grab from source.

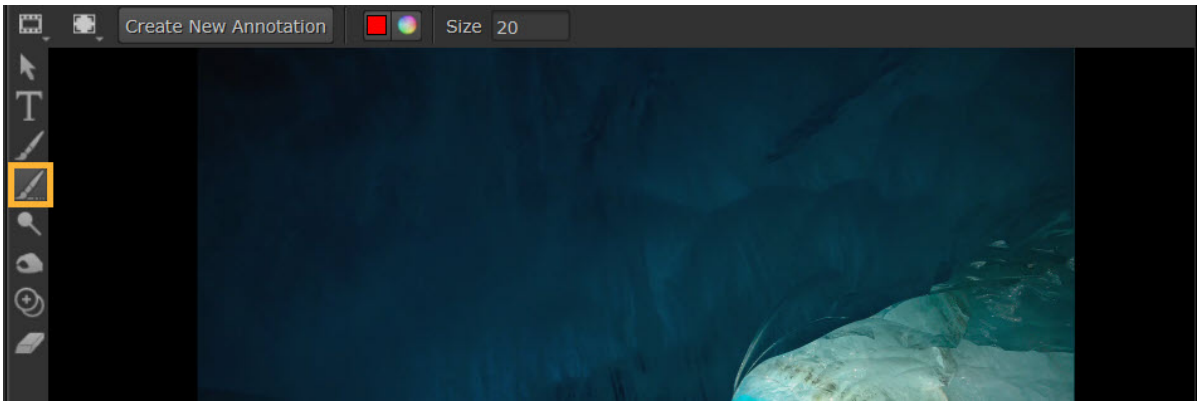
Tip: When using the color picker tool, use **Alt**+**Ctrl** to grab from behind annotations. **Alt**+**Ctrl**+**Shift** to grab from a selection.

- Adjust the size of the brush by setting a Size value in the toolbar, or by holding **Shift** and dragging the brush size inwards or outwards.
- Blending Modes: Choose from the dropdown to determine how the paint will be applied in relation to the rest of the image.
- Pressure: This toggle button  is for controlling pressure. Enabling it will mean that pressure affects the size, opacity, or hardness of your stroke. Opacity pressure is enabled by default.

Tip: Use **N** keyboard shortcut to switch between Brush and Eraser tools.

Vanishing Brush

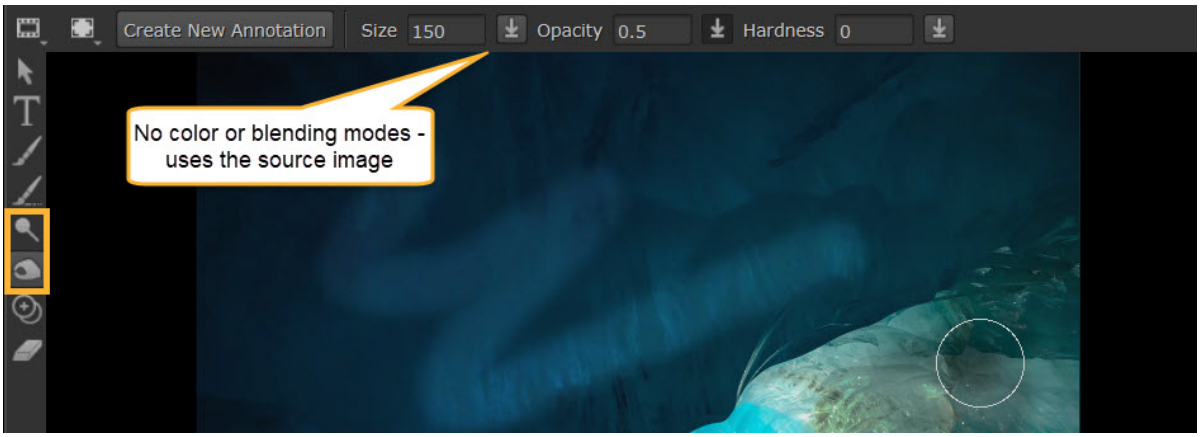
The Vanishing brush allows paint to vanish shortly after its application. For example this can be used for demonstration purposes where you don't want the brush strokes to remain. You can use the brush on the Viewer and it will automatically vanish after a short time.



Note: Use **M** keyboard shortcut to switch to using the Vanishing brush.

Dodge and Burn

The Dodge and Burn tools help to annotate shadows and light. Dodge lightens the image, and Burn darkens the image.

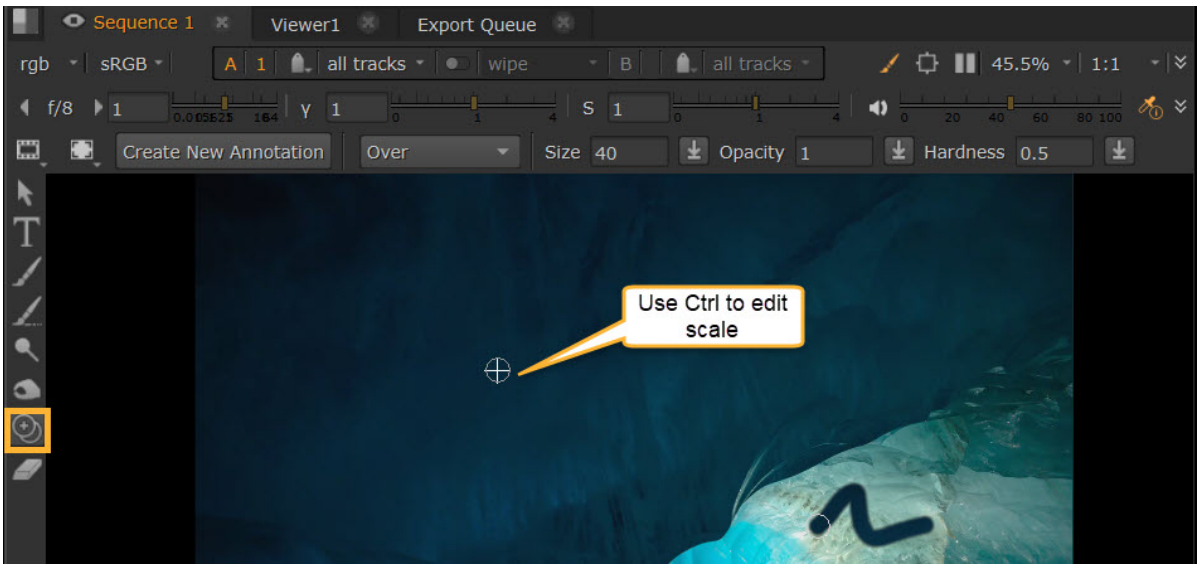


- There are no color picker or blending modes for Dodge and Burn - it will grab the color from the source image.
- Size, opacity, hardness, and pressure work the same way as the Brush tool.

Tip: Use **D** keyboard shortcut to switch between Dodge and Burn tools.

Clone

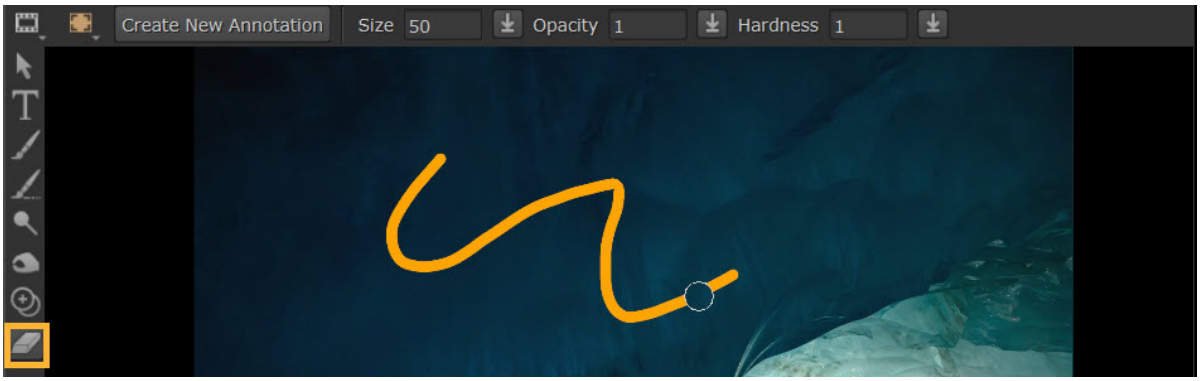
The Clone tool allows you to clone parts of your image. Select **Ctrl** to edit the clone under your cursor and stretch the scale of it.



Tip: Use **C** keyboard shortcut to switch to using the Clone tool.

Eraser

The Eraser tool allows you to remove any Brush, Dodge, Burn, or Clone you have already added.



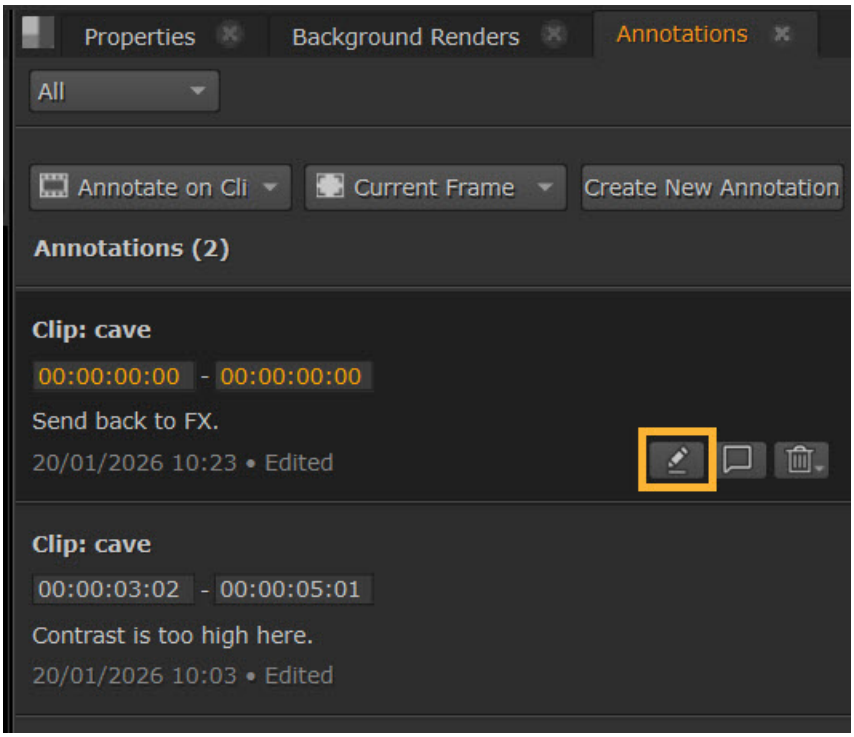
Tip: Use **N** keyboard shortcut to switch between Brush and Eraser tools.

Editing and Deleting Annotations

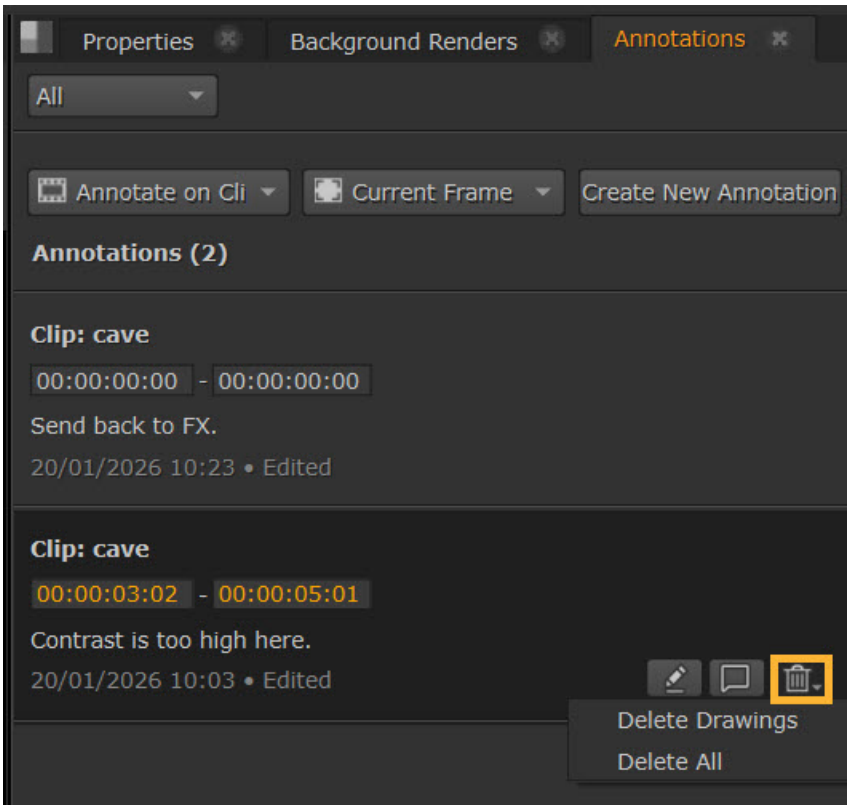
Annotations can be edited and deleted via the Annotations Panel. Any drawings are part of the annotations in the Annotations Panel.

Tip: Toggle the annotations button to hide or show drawings.

- Use the Edit button on any comment in the Annotations Panel. This allows you to edit the text in the Annotations Panel. Once edited, your comment will show with an 'Edited' state.



- You can delete comments via the trash can button. This brings up a Delete menu where you can select what to delete from **Delete Drawings** (removes drawings only) or **Delete All** (removes the comment and any drawings).

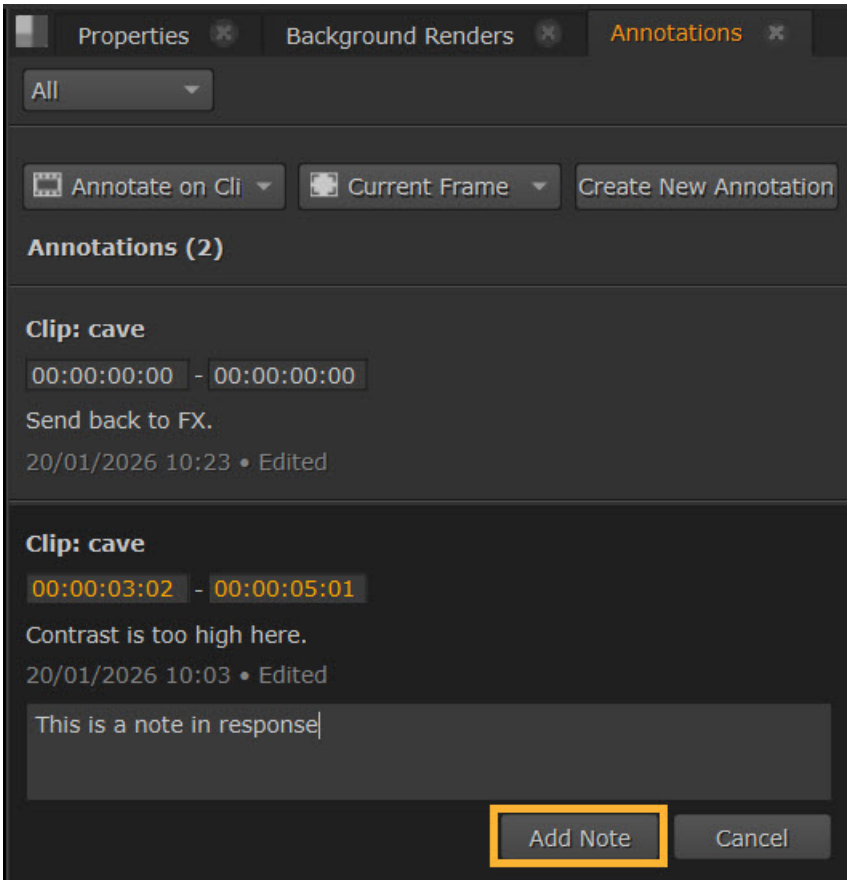


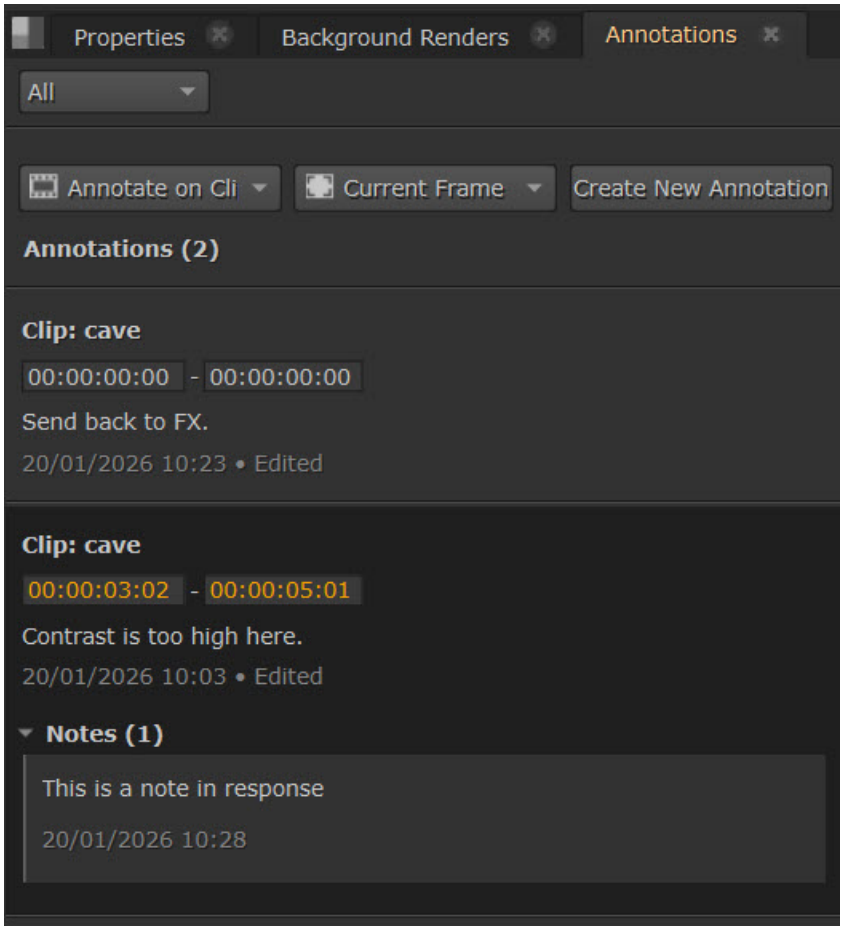
Tip: You can also edit the timestamp by typing in the orange fields in the Annotations Panel, or by dragging the blue markers to a new position on the timeslider.

Adding Notes

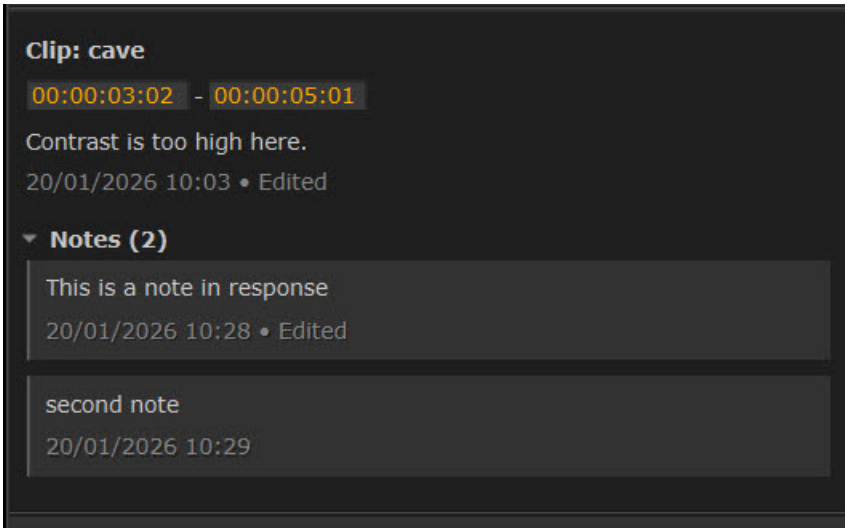
Select the notes icon to add notes to any comment to leave further thoughts or replies.

Write in the text field and select **Add Note** to save it. It will then appear beneath the original comment under **Notes**.

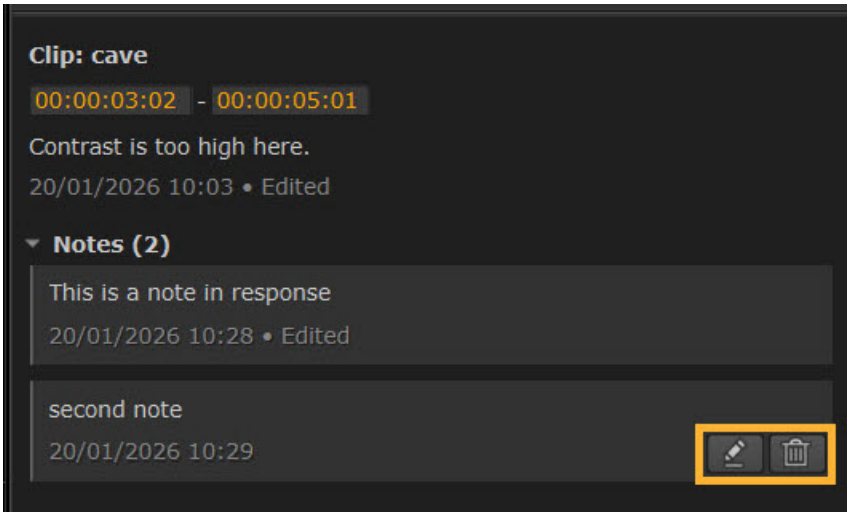




Follow the same process again to add further notes and replies.



Notes can be edited and deleted using the Edit and Delete buttons, which are found when hovering over your note.

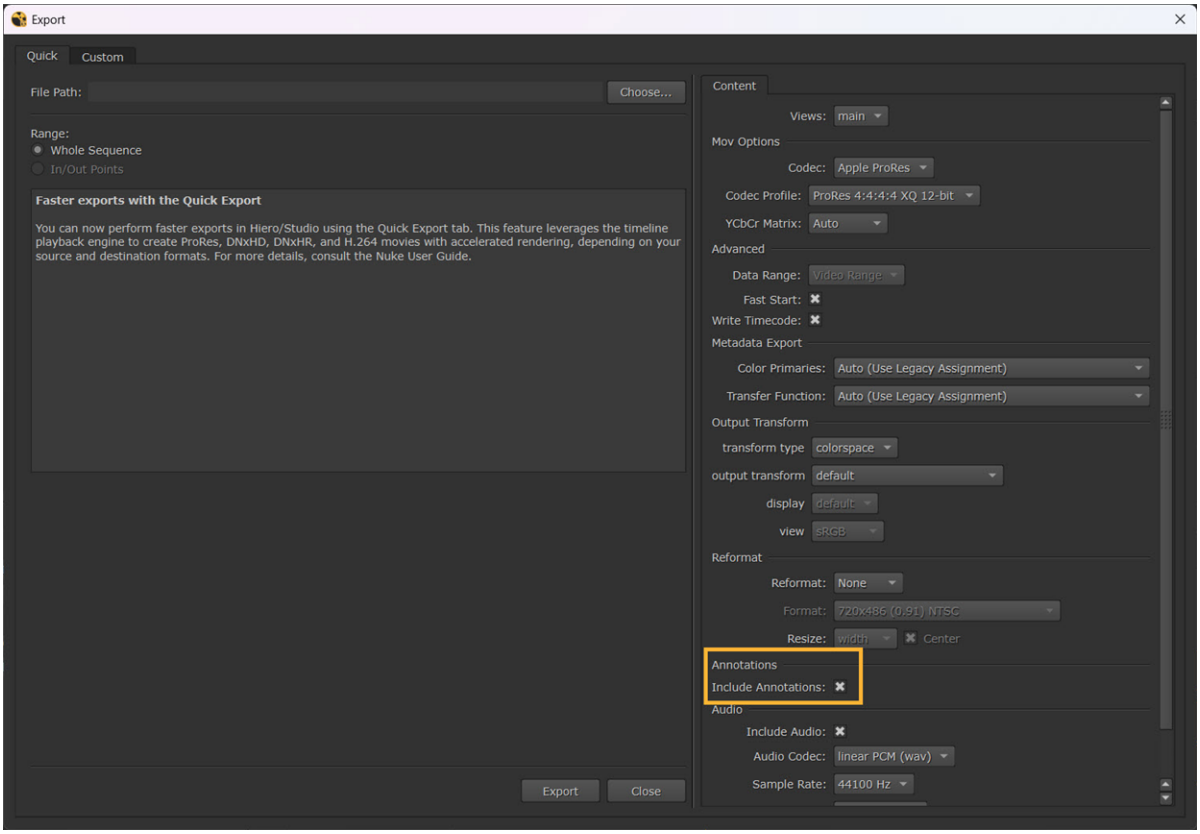


Rendering Out Annotations

You can burn the drawn annotations into your images or sequences via Quick Export.

To do this, go to **File** > **Export**, or use **Ctrl** + **Shift** + **E** to open the Export Dialog.

In the **Quick** tab, check to enable **Include Annotations**.



Note: See more about Quick Export at Exporting Sequences with Quick Export.

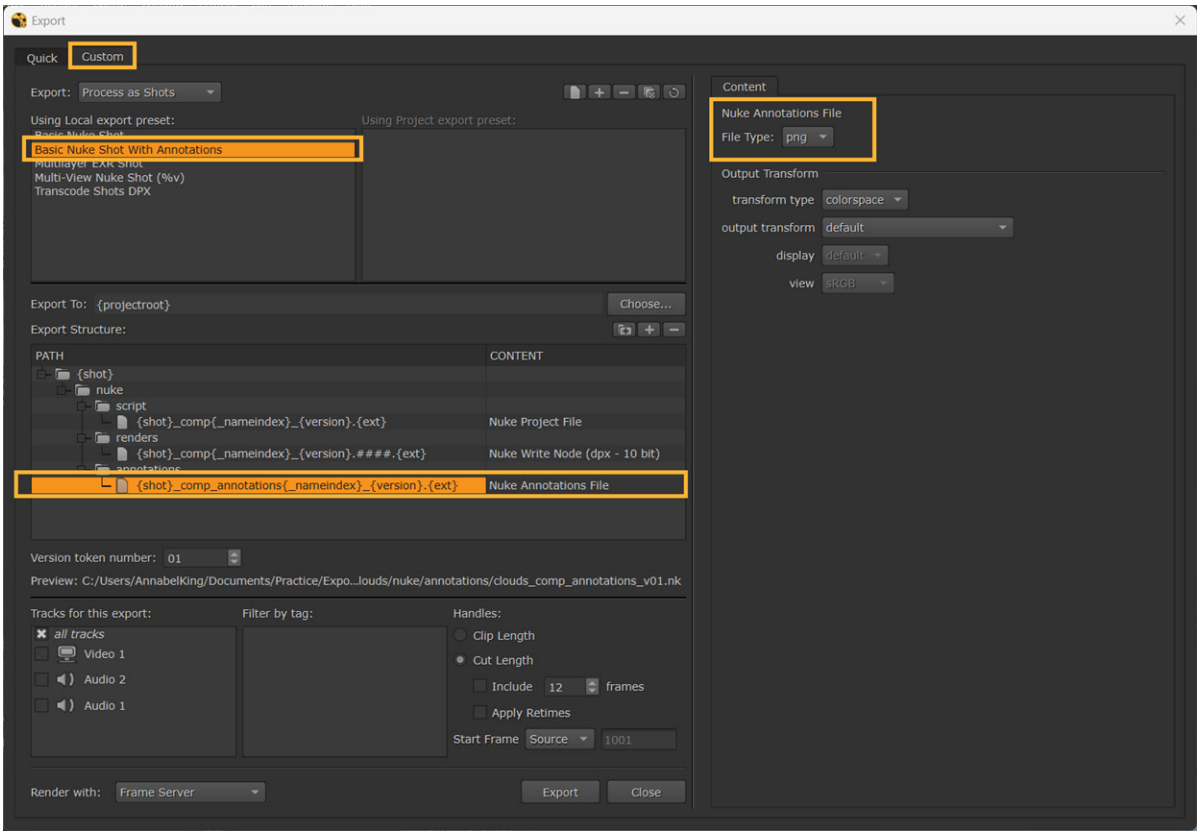
Export to PNG or JPEG

You can also export to **.png** or **.jpeg** by using the **Custom** Export options.

To do this, go to **File** > **Export**, or use **Ctrl** + **Shift** + **E** to open the Export Dialog.

In the **Custom** tab, select **Basic Nuke Shot with Annotations**. Choose your Annotation File under the **Export Structure**, and then select a **File Type** on the right-hand side (either **png** or **jpeg**).

Selecting **Export** will then write the annotation images using the selected file format. The frames **.png** or **.jpg** files will be saved in your **/annotations** folder.

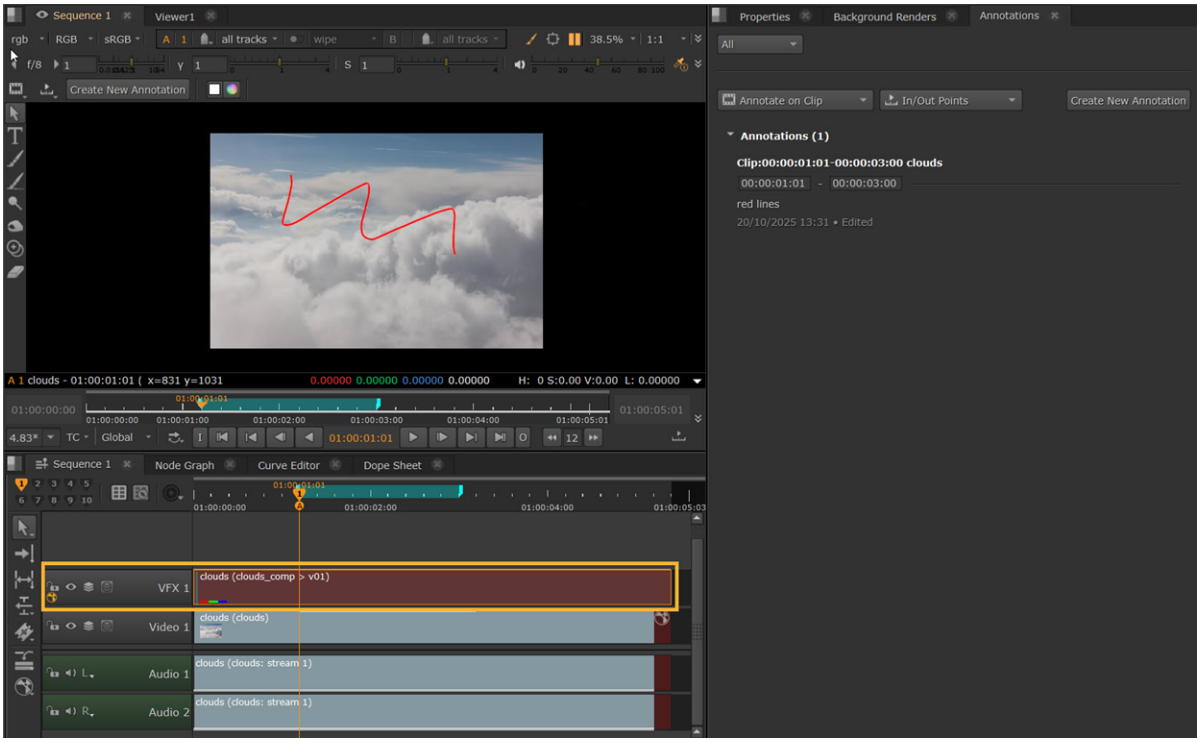


Create Comp with Annotations

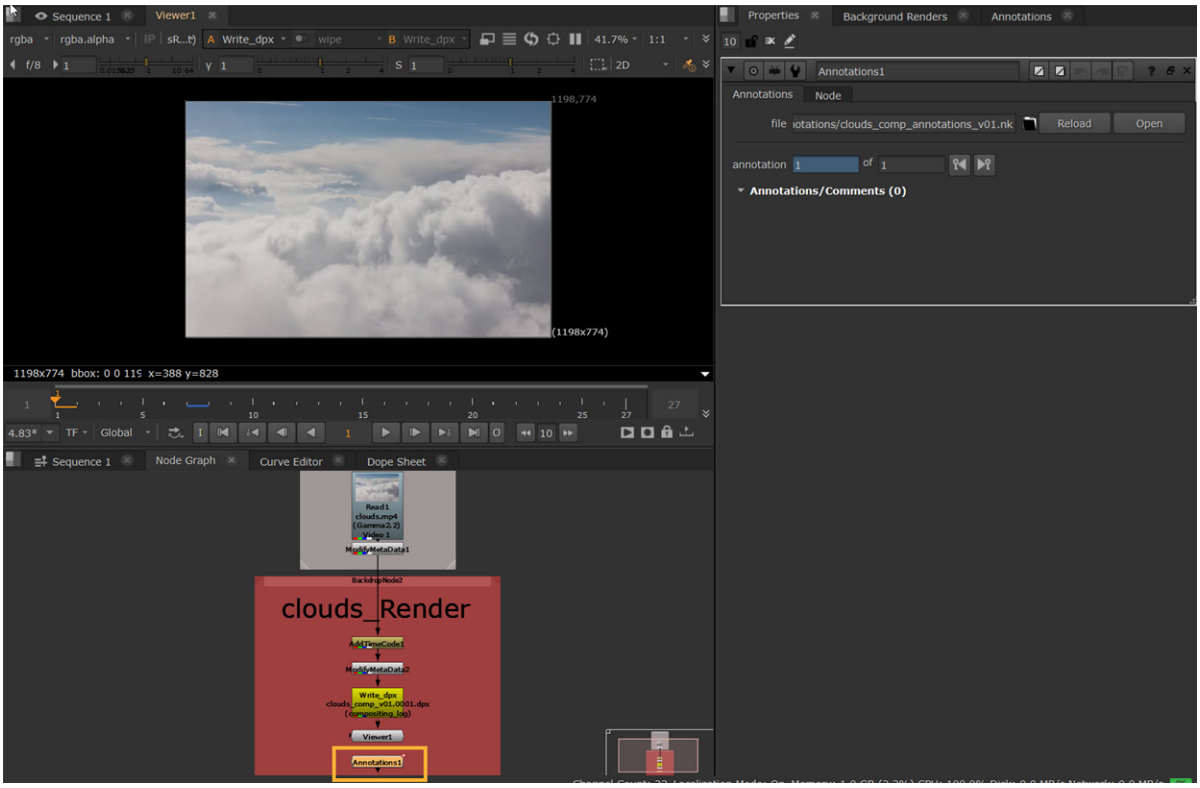
To view your annotation comments in a Compositing environment, you can use **Create Comp**.

1. Right click on clip in timeline and select **Create Comp >Create Comp**.

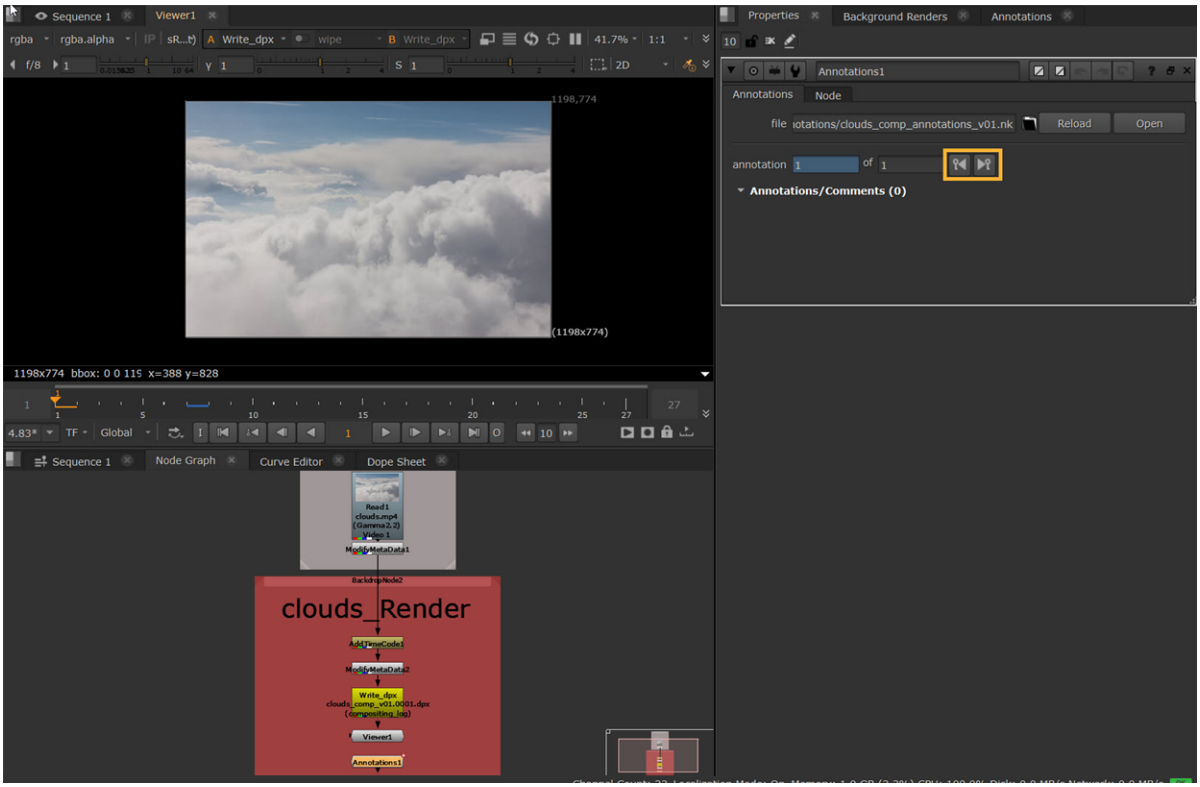
2. The Comp track will be added to your timeline.



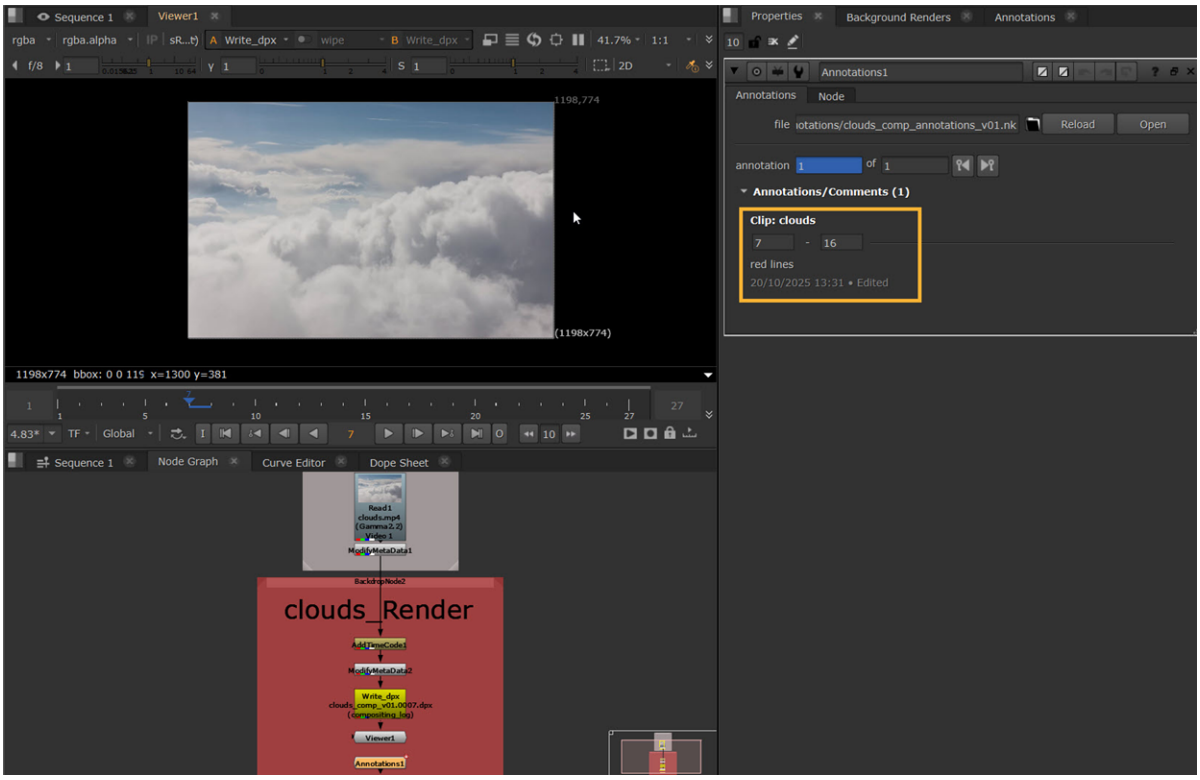
3. Double click the Comp track to open it in the Compositing environment.
4. Here you will find the Node Graph populated with the relevant nodes for the Comp. The Annotations node stores your annotation comments.



5. The Annotations node contains controls to jump between where annotations are in the Comp (if you have more than one).



6. In the **Annotations/Comments** section you can also see the annotations that are active on that frame.



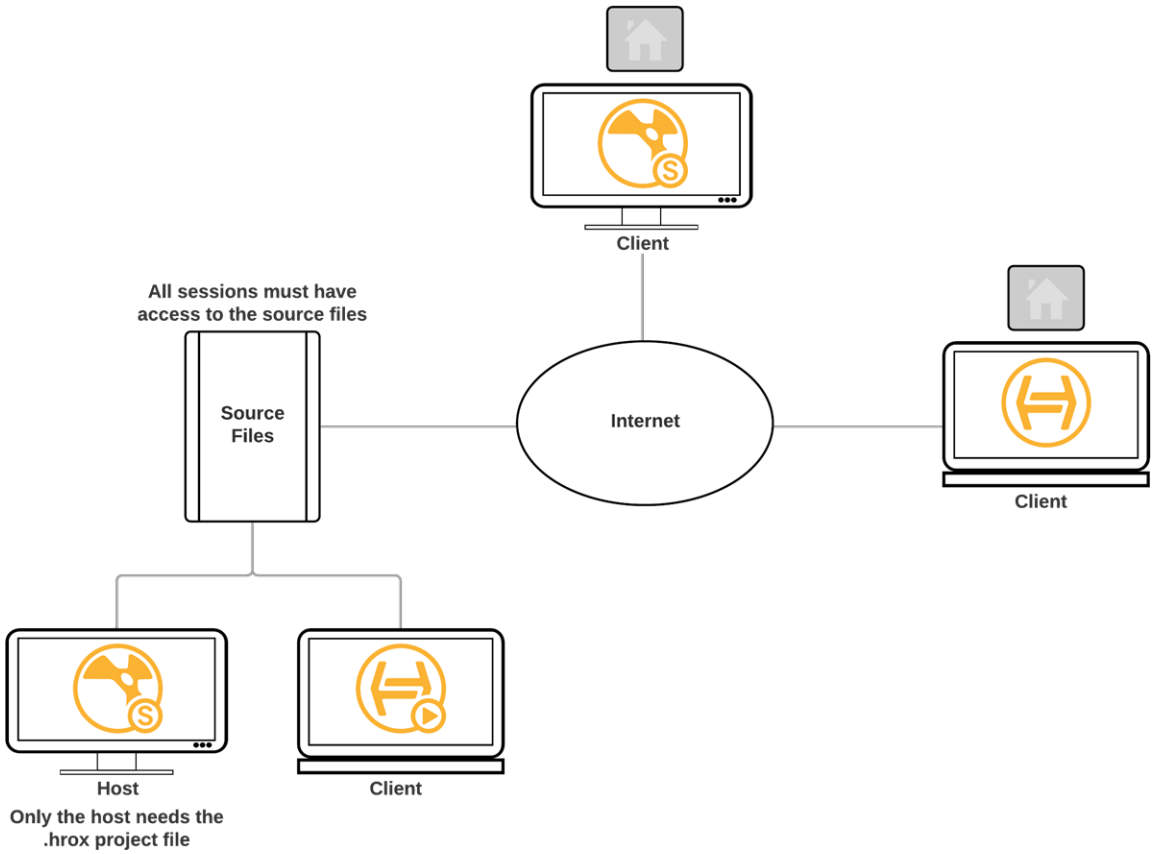
Note: See more details about Create Comp workflows at Create Comp.

Sync Review

Sync Review allows multiple participants to collaborate on a single project in Nuke Studio, Hiero, or HieroPlayer. The host allows multiple clients to join the review session and all participants have access to the playback controls, annotations, and versioning system.

Warning: All participants must use the same version of the application to run a sync session. For example, if the host is using 17.0v3, clients can only connect to the session if they are using 17.0v3 as well.

Sync Review works by connecting clients to the host using the host's IP address or hostname and an open port number. The project **.hrox** file only needs to exist on the host machine, but the source clips referenced by the shots on the timeline must be available for all participants either locally or from a central server.



Hosting a Sync Review Session

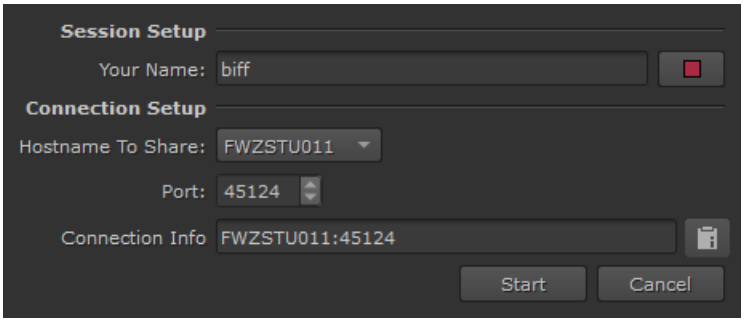
Nuke Studio, Hiero, and HieroPlayer can all act as host in Sync Review sessions, though you only need one host to connect multiple clients.

To start a host session:

1. Go to **File** > **Open Project** to load the project you want to review.

> **Note:** Only the host loads the project, client sessions don't require the **.hrox** file.

2. Go to **Workspace** > **Sync Session** to load a dedicated review workspace,
   OR
   Go to **Window** > **Sync Session** to open the **Sync Session** panel.
3. Click **Host**.
   The **Session Setup** dialog is displayed.



4. Enter the name you want to display on participant sessions and pick a cursor color by clicking on the color swatch.

Tip: You can set your name and cursor color for new sessions in the **Preferences** under **Project Defaults** > **Sync Review**.

5. Select the host name and port to which you want the client sessions to connect.

   Your selection is displayed in the **Connection Info** field.

Tip: Click the copy to clipboard button to automatically copy the connection details so that you can pass them on to participants.

6. Click **Start** to begin hosting.

   The **Sync Session** panel is populated with your details automatically.
7. Send the connection details to the participants that intend to join the review session.

Tip: If you didn't copy them to the clipboard earlier, click **Sync Panel** > **Copy Info**.

Connecting to a Sync Review Session

Nuke Studio, Hiero, and HieroPlayer can all connect as clients in Sync Review sessions, all you need is the connection information from the host.

Warning: All participants must use the same version of the application to run a sync session. For example, if the host is using 17.0v3, clients can only connect to the session if they are using 17.0v3 as well.

To connect a client session to a host:

1. Ensure that you have access to the source files used in the host project.

   The files can be on a server on the same network as the host machine or accessed over the internet using suitable security, such as VPN.



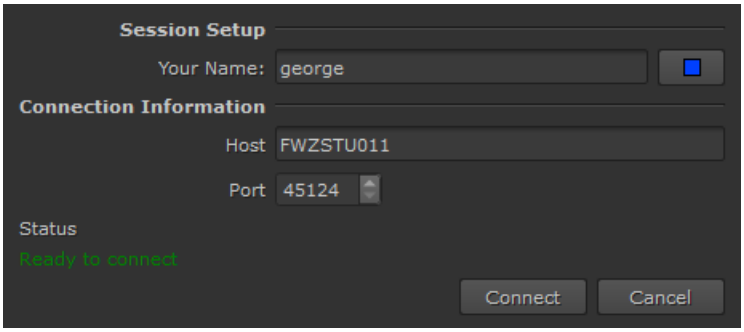
Note: Only the host loads the project, client sessions don't require the **.hrox** file.

2. Go to **Workspace** > **Sync Session** to load a dedicated review workspace,

   OR

   Go to **Window** > **Sync Session** to open the **Sync Session** panel.
3. Click **Connect**.

   The **Session Setup** dialog is displayed.

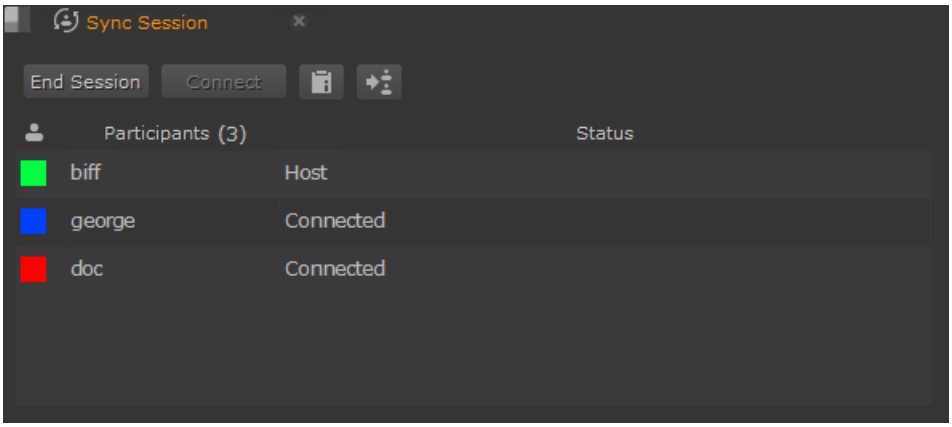


4. Enter the name you want to display on client sessions and pick a cursor color by clicking on the color swatch.

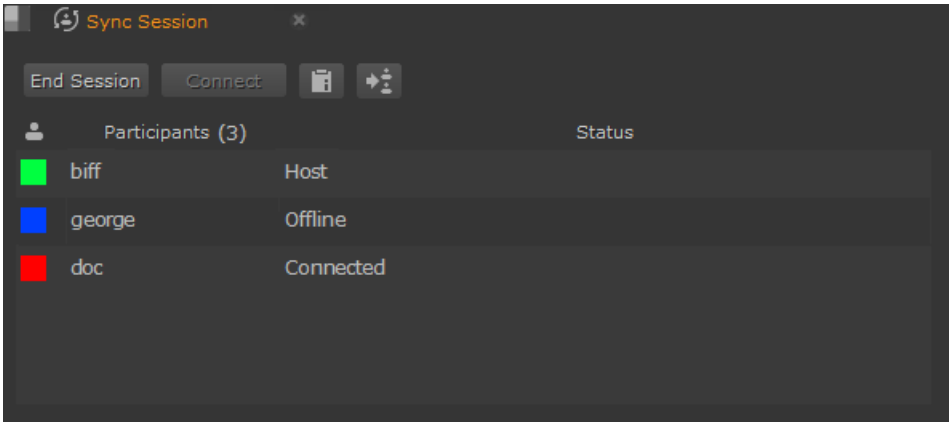
Tip: You can set your name and cursor color for new sessions in the **Preferences** under **Project Defaults** > **Sync Review**.

5. Enter the **Host** name and **Port** supplied by the Sync Review session host.
6. Click **Connect** to join the review.

   A list of participants is displayed in the **Sync Session** panel.



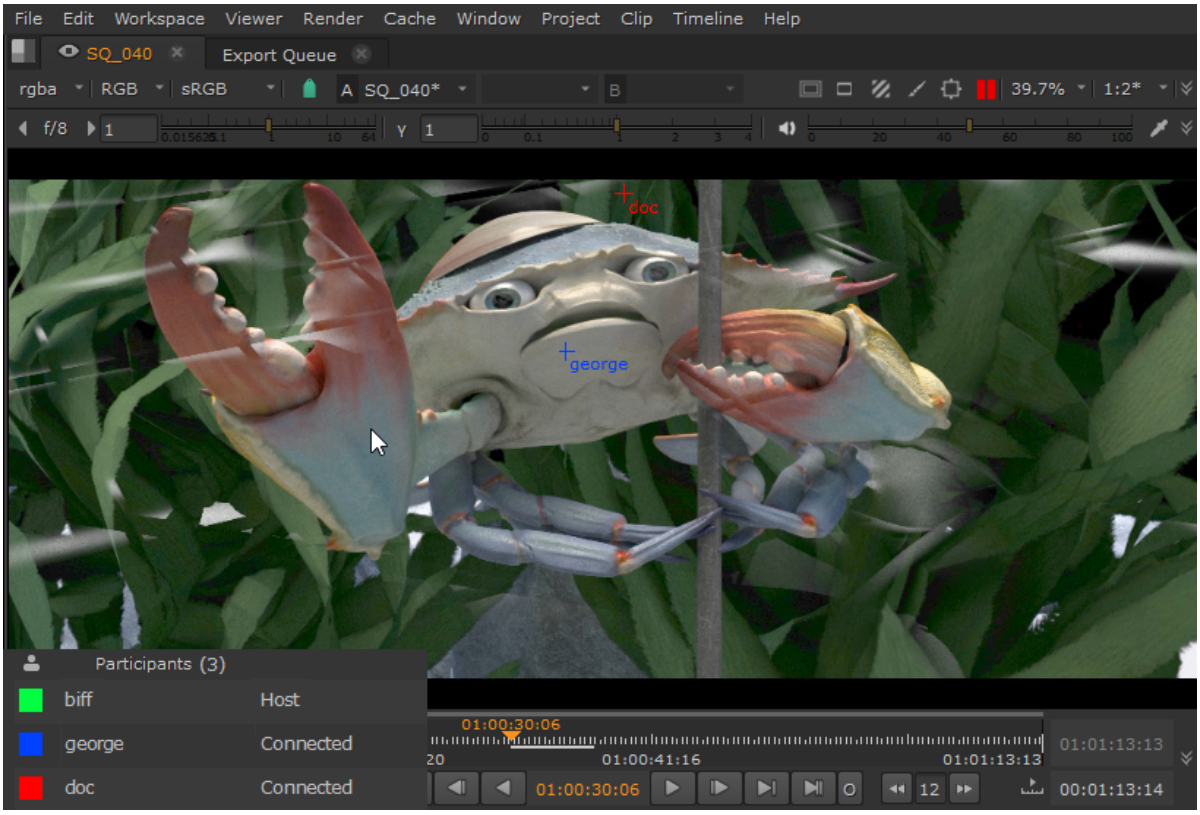
Client sessions can connect or disconnect at any time and the **Status** updates as required.



If the Host ends the session, all client sessions are also ended.

Collaborative Review

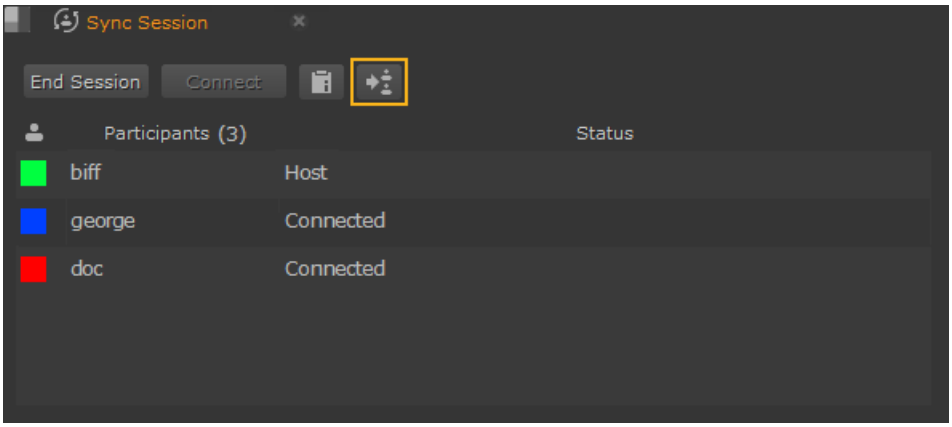
Participants in a Sync Review session can all contribute equally to the review. Each participant has a named and color-coded cursor that other participants can see in the Viewer. In the example, biff's cursor is shown as normal because it's their session.



Most participant actions available in the Viewer, **Project** panel, and timeline are synced automatically during the session, including:

- Viewer - Layer and channel selection, TC/Frames option, guides and Viewer selection options, Zoom in/out.
- Project panel - Importing files, creating new sequences, tags, and bins and creating, renaming, moving or deleting project items.
- Timeline - Creating new tracks, renaming shots, adding or deleting soft effects, changes in soft effects parameters, retimes and transitions, Sync Lock Status of tracks, trimming and moving clips in the timeline.

If participant sessions are out of sync, you can push actions to other participants manually. Click the update button in the **Sync Session** panel or use the **Ctrl**/**Cmd**+**P** keyboard shortcut to force an update across all participant sessions.


Sync Session
End Session
Connect
Participants (3)
Status
biff
Host
george
Connected
doc
Connected

Timeline Editing Tools

The timeline editing tools allow you to manipulate your shots directly in the timeline, in single- or multi-view projects, using a series of modal editorial tools that complement the **Multi Tool**. Select the tool you need for the job and then select a new tool and continue editing.

The timeline editing tools are grouped for convenience - each tool group contains several tools and you can cycle between them by clicking the tool or using keyboard shortcuts. The editing tools work the same way in single- and multi-view timelines.

Icon	Tools	Description
	Multi Tool	The **Multi Tool**'s functionality is equivalent to most of the other tools combined, but doesn't require modal tool selection.
	Move/Trim	The **Move/Trim** tool allows you to manipulate the position of a shot or its output by adding or removing handles.
	Select	The marquee **Select** tool allows you to make multiple selections quickly by lassoing shots. Hold **Shift** to add to the selection and **Alt** to subtract from the selection.
	Effects Selector	The **Effects Selector** only allows selection of soft effects. Shots and other timeline objects cannot be selected. This is extremely useful when you want to adjust effects in complex timelines with multiple tracks and shots without affecting other objects.
	Selection by Track	The track selection tools allow you to quickly select multiple items depending on the initial selection. For example, the **Select Track to Right** tool selects all shots to the right of the target shot, within a single track.
	Slip Clip	The **Slip Clip** tool allows you to shift a shot's In and Out points by the same amount and in the same direction, retaining the original duration but altering the timeline

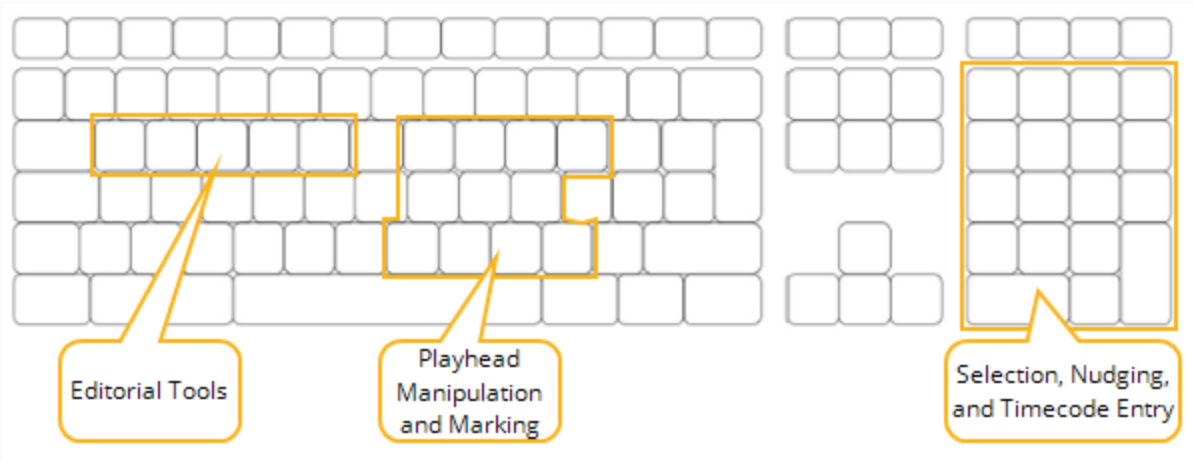
Icon	Tools	Description
		output.
	Slide Clip	The **Slide Clip** tool allows you to move a shot in relation to the item before and/or after the target item, without changing its length or timeline output.
	Roll Edit	The **Roll Edit** tool enables you to roll a single edit within the available handles, shortening one shot while lengthening the other, but keeping the overall duration the same.
	Ripple Edit	The **Ripple Edit** tool operates similarly to the trim function of the **Move/Trim** tool, except that downstream shots are rippled to automatically close any resulting gaps in the timeline.
	Retime Clip	The **Retime Clip** tool allows you to trim a shot's In or Out point and automatically retime the clip to fill the new shot duration.
	Razor	The **Razor** and **Razor All** tools allow you to cut shots in to separate parts so you can remove sections or rearrange items on the timeline.
	Join	The **Join** tool can only be used on edit points between two razored shots, denoted by the yellow arrows at the edit.

Tip: Enabling **Preferences** > **Panels** > **Timeline** > **show frame end marker** draws an extra line on the timeline to the right of the playhead, indicating the end of the current frame.

The modal editorial tools are mapped to the **Q**, **W**, **E**, **R**, and **T** keyboard shortcuts when the timeline is the active tab.



Note: For a full list of keyboard shortcuts, please see Appendix B: Keyboard Shortcuts.

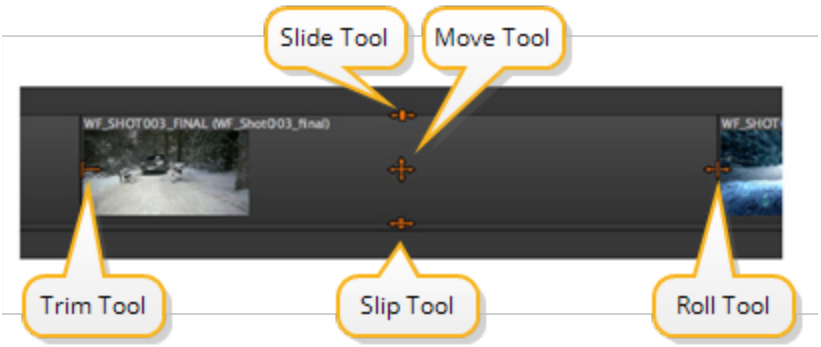


Pressing a keyboard shortcut multiple times selects the tools within each mode. For example, pressing **E** twice, rapidly in succession activates **Slide Clip**. Pressing them slowly in succession does not achieve the same result, but instead, remains on the first item in the menu. This allows you to activate a tool without knowing the current state of tool selection.

- mapped to **Q**, cycles through **Multi Tool**, **Move/Trim**, **Select**, and **Effects Selector**.
- mapped to **W**, cycles through **Track Selection** tools.
- mapped to **E**, cycles through **Slip Clip** and **Slide Clip**.
- mapped to **R**, cycles through **Roll Edit**, **Ripple Edit**, and **Retime Clip**.
- mapped to **T**, cycles through **Razor**, **Razor All**, and **Join**.

Using the Multi Tool

Unlike the other editing tools available, the **Multi Tool** changes function depending on the position of your pointer in relation to the shot(s) selected.



The **Multi Tool**'s functionality is equivalent to most of the other tools combined, but doesn't require modal tool selection:

- **Move** - placing the mouse in the center of a shot activates the tool. Drag the selected shot to the required position on the timeline.
- **Trim** - placing the mouse at the left or right of the shot activates the tool. Drag the edit point to the new position and release the mouse to complete the trim.

  See Using the Move/Trim Tool for more information.
- **Select** - click-and-drag to marquee select clips. Hold **Shift** to add to your selection or **Alt** to subtract.

  See Using the Selection Tools for more information.
- **Slip** - placing the mouse at the bottom of the shot activates the tool and displays the slip handles. Drag the shot to the new position and release the mouse to complete the slip.

  See Using the Slip Clip Tool for more information.
- **Slide** - placing the mouse at the top of the shot activates the tool and displays the slide handles. Drag the shot to the new position and release the mouse to complete the slide.

  See Using the Slide Clip Tool for more information.
- **Roll** - placing the mouse on the edit between shots activates the tool and displays the handles. Drag the edit to the new position and release the mouse to complete the roll.

  See Using the Roll Edit Tool for more information.
- **Razor** - when using the Multi Tool, Razor cuts are menu driven. Navigate to **Timeline** > **Razor Selected** or **Razor All** to make cuts at the current playhead position.

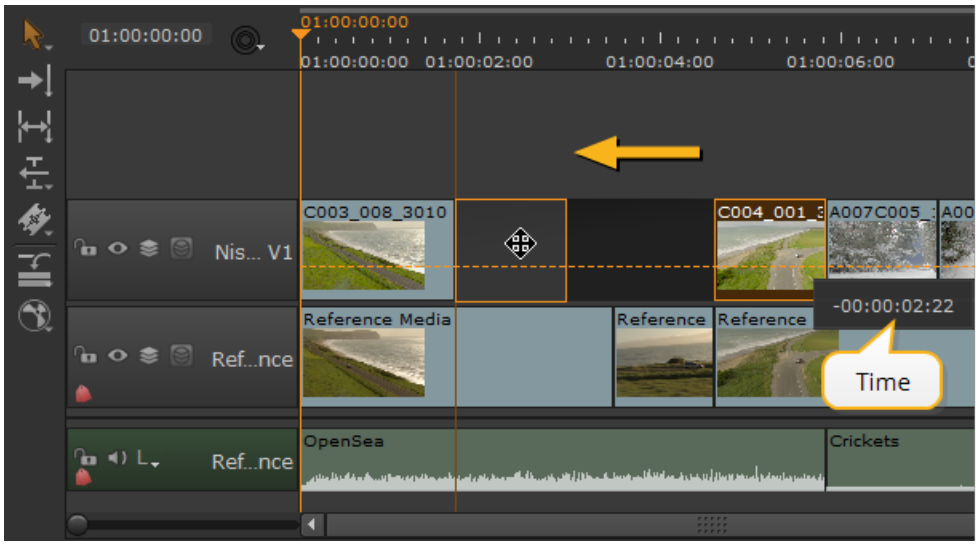
  See Using the Razor and Join Tools for more information.

Using the Move/Trim Tool

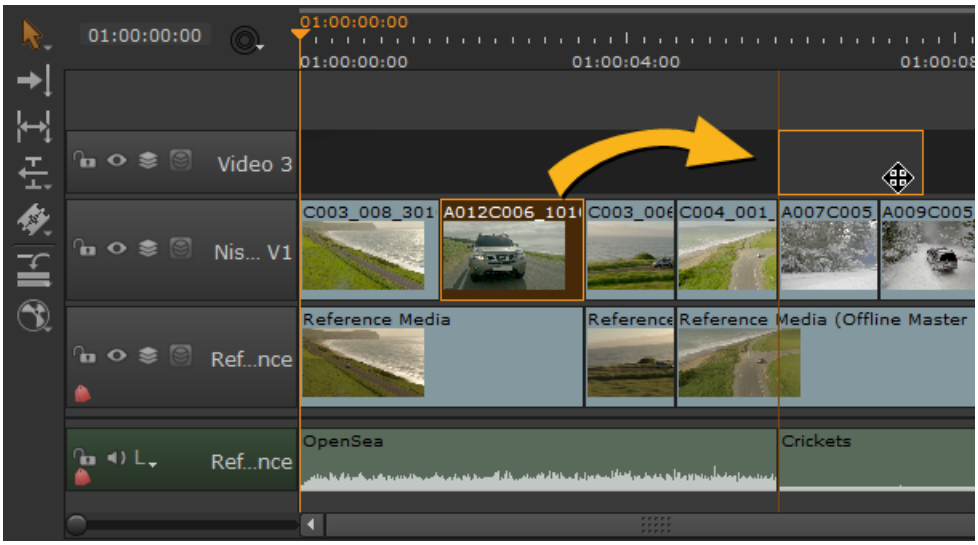
The **Move/Trim** tool allows you to manipulate the position of a shot or its output by adding or removing handles. Activate the **Move/Trim** tool by clicking the tool or pressing **Q** twice.

Moving Shots

Click and drag the selected shot(s) to the required position on the timeline. A time count popup, in conjunction with the snap to edit function, helps you to reposition the shot(s) accurately.



You can also move shots up and down the track hierarchy using drag-and-drop or the **Alt**+**,** (comma) and **Alt**+**.** (period) keyboard shortcuts.



The following table describes the **Move/Trim** tool's modifiers and actions:

Mode	Method	Description
Overwrite	drag-and-drop	The default move mode. The dragged shot overwrites any items that are present in the move location.
Ripple	drag then hold **Alt** and drop	Drag-and-drop shots on top of other items without overwriting content - items are pushed down the timeline to accommodate the move.

Mode	Method	Description
Duplicate	hold **Alt** and drag then release **Alt** and drop	Copy the shot, then drag-and-drop on top of other items overwriting existing content - items are **not** pushed down the timeline to accommodate the move.
Ripple and Duplicate	hold **Alt** then drag and drop while holding **Alt**	Copy the shot, then drag-and-drop items on top of others without overwriting content - items are pushed down the timeline to accommodate the move.



Note: On Linux, hold **Ctrl**+**Alt** for **Duplicate** and **RippleDuplicate** modifiers.

Action	Keyboard Shortcut	Description
Delete	**Backspace**	Delete the selected shot(s) or gap(s).
Ripple Delete	**Shift** + **Backspace**	Remove the selected shot(s) and ripple items down stream to close gaps in the timeline.  **Note:** The ripple effect may not close gaps entirely, because Nuke Studio does not allow linked tracks to become desynchronized during rippling.

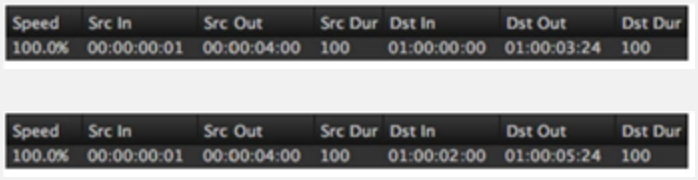
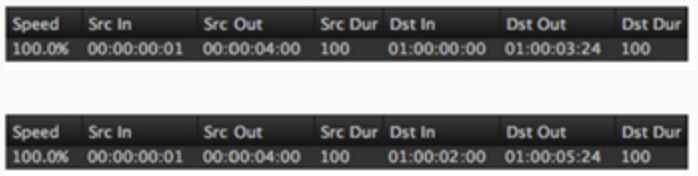
If you need to nudge shots horizontally by just a frame or two, you can select the items on the timeline and press **,** (comma) to nudge it left or **.** (period) to nudge it right. Press **Shift**+**,** (comma) or **.** (period) to nudge the shot horizontally by the **FrameIncrement** set under the Viewer.



Note: You cannot overwrite other shots on the timeline horizontally using the nudge keys. However, vertical nudging (**Alt**+**,** and **Alt**+**.**) overwrites other tracks.

To Move Shots Using the Spreadsheet View:

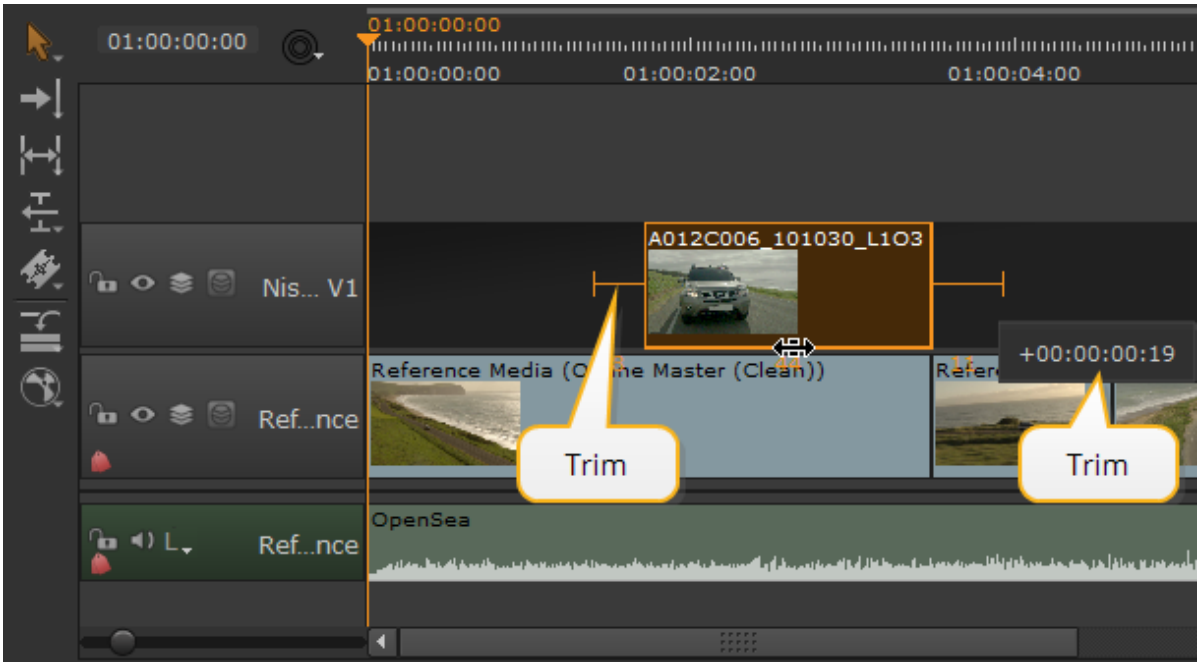
1. Select the required events in the spreadsheet view.
2. Click the cog icon and select the required **Time Edit Behaviors**:

Modify	Using	Result
Dst In	Move Destination	Adjusts the event's **Dst In** and **Out** by the same amount, moving the shot's position on the timeline by the specified amount, while maintaining Speed. Before and after a 2 second **Dst In** increase: 
Dst Out	Move Destination	Adjusts the event's **Dst Out** and **In** by the same amount, moving the shot's position on the timeline by the specified amount, while maintaining Speed. Before and after a 2 second **Dst Out** increase: 

3. Adjust the **Dst In** or **DstOut** to move the shot(s) by the specified amount.

Trimming Shots

Click-and-drag the edit point to the new position and release the mouse to complete the trim.

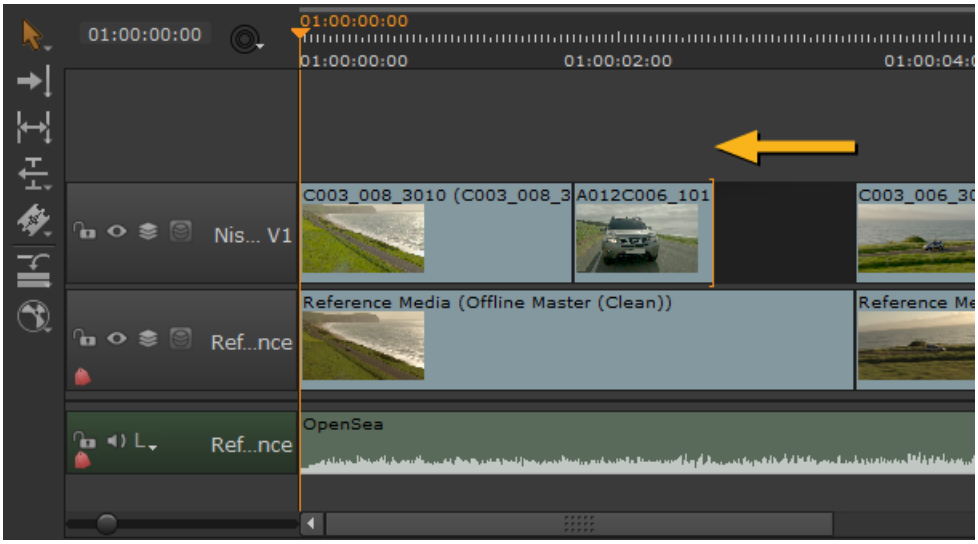


Tip: Use the **Ripple Edit** tool, activated by pressing **R** twice, to ripple downstream shots automatically.

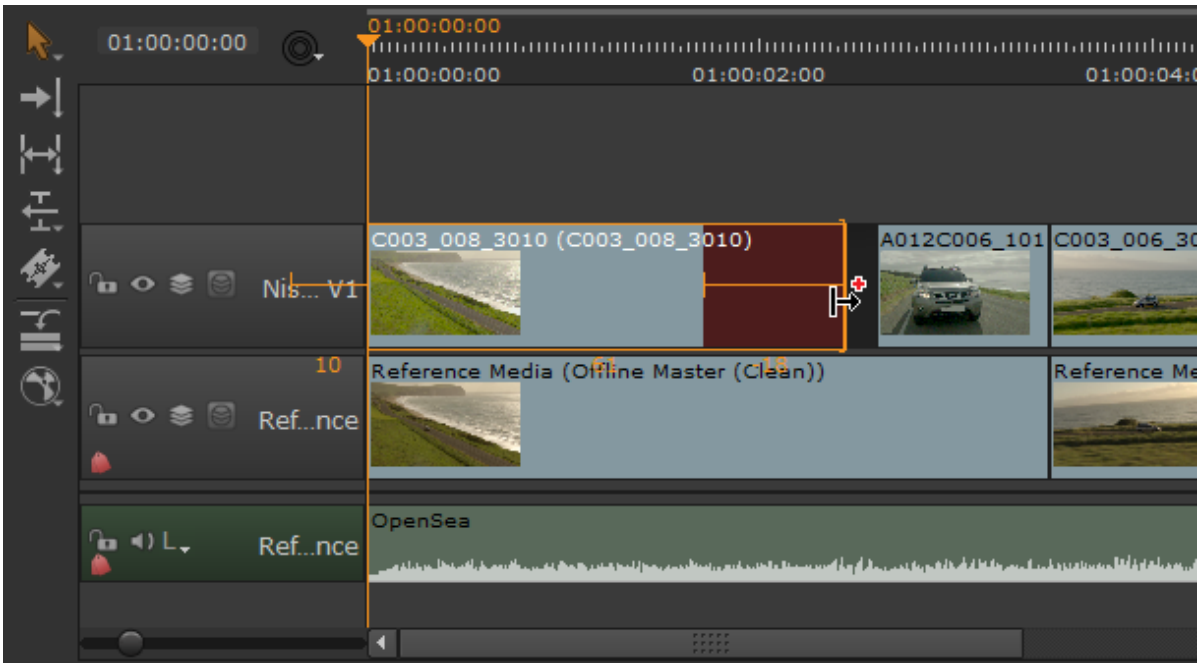
The Viewer displays the new In or Out point (depending on whether you're adjusting the beginning or end of the shot), allowing you to accurately gauge the new output.

Note: Trimming multiple shots simultaneously trims each item by the same amount and in the same direction.

Alternatively, click an edit point and nudge the edit using the **Ctrl/Cmd**+←→ (numeric pad) keys or hold **Shift** to nudge by the **Frame Increment** set under the Viewer.

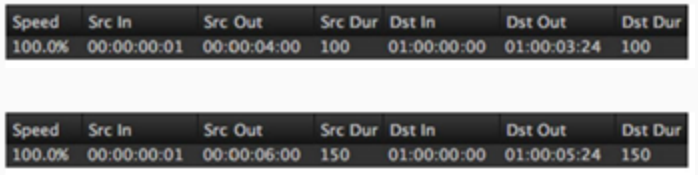
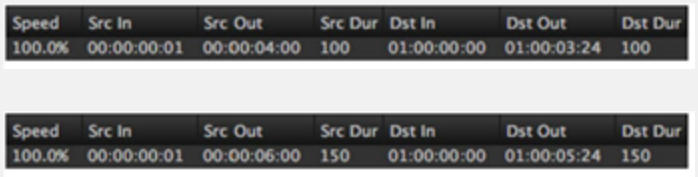
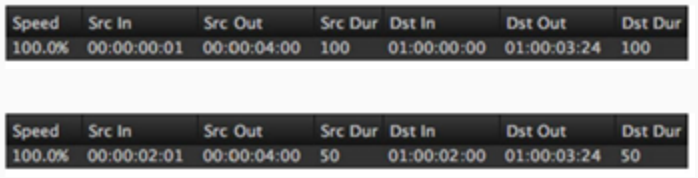


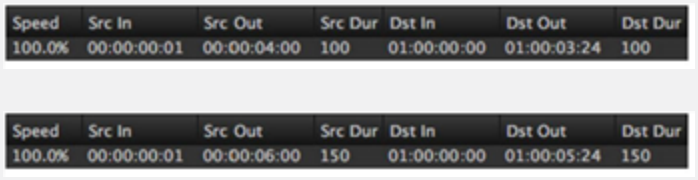
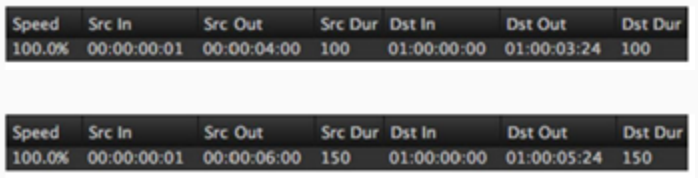
By holding **Ctrl/Cmd** and dragging an edit, you can add blank frames past the end of the shot's handles. Blank frames are colored red on the timeline for clarity:



To Trim Shots Using the Spreadsheet View:

1. Select the required events in the spreadsheet view.
2. Click the cog icon and select the required **Time Edit Behaviors** depending on whether you're using the In or Out points or duration:

Modify	Using	Result
Src In	Trim In	Trims the event's **Src In**, **Dst In**, and **durations** while maintaining speed. Before and after a 2 second **Src In** increase: 
Src Out	Trim Out	Trims the event's **Src Out**, **Dst Out**, and **durations** while maintaining speed. Before and after a 2 second **Src Out** increase: 
Src Dur	Trim Out	Trims the event's **Src Dur**, **Dst Dur**, and **Src/Dst Out** while maintaining speed. Before and after a 50 frame **Src Dur** increase: 
Dst In	Trim In	Trims the event's **Dst In**, **Src In**, and **durations** while maintaining speed. Before and after a 2 second **Dst In** increase: 
Dst Out	Trim Out	Trims the event's **Dst Out**, **Src Out**, and **durations** while maintaining speed. Before and after a 2 second **Dst Out** increase:

Modify	Using	Result
		
Dst Dur	Trim Out	Trims the event's **Dst Dur**, **Src Dur**, and **Dst/Src Out** while maintaining speed. Before and after a 50 frame **Dst Dur** increase: 

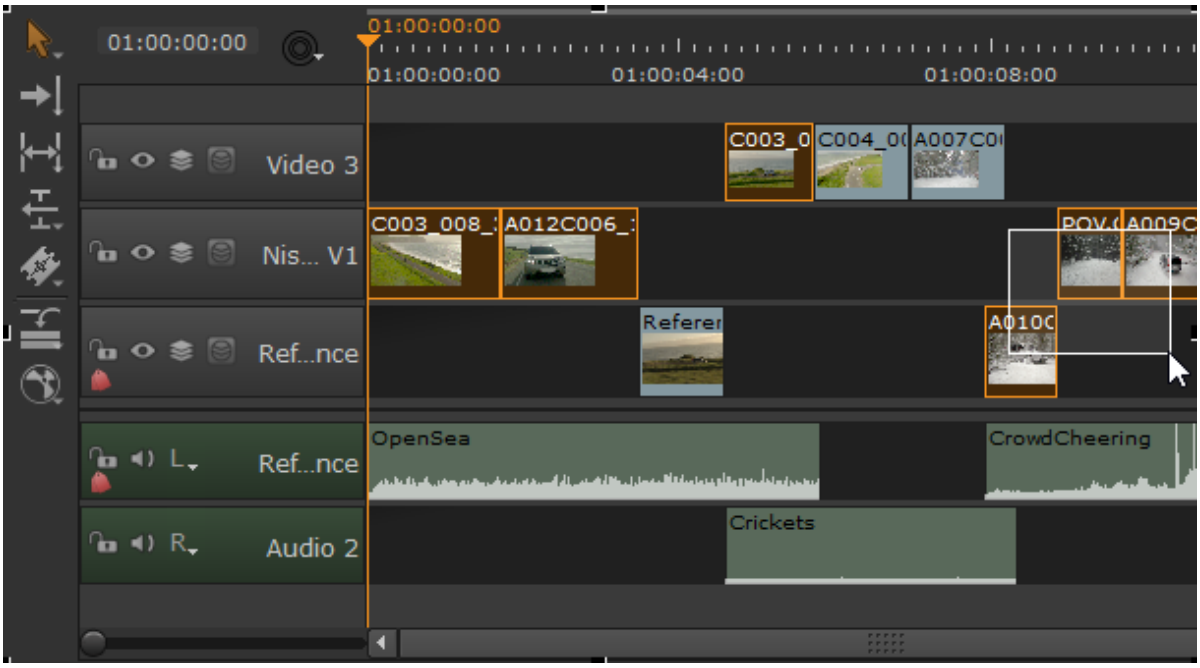
3. Adjust the values as required to trim the shot(s) by the specified amount.

Using the Selection Tools

The timeline editing tools include a marquee selection tool and several context dependent track selection tools.

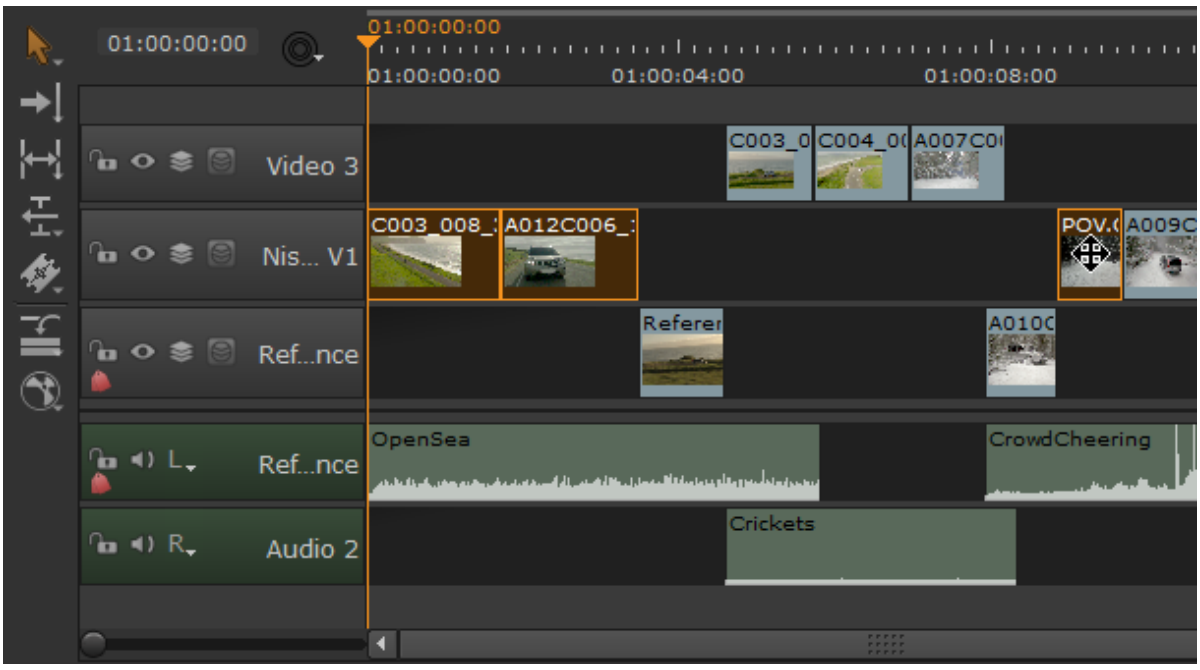
The marquee **Select** tool, activated by clicking the tool or pressing **Q** three times, allows you to make multiple selections quickly by lassoing shots.

Hold **Shift** to add to the selection:

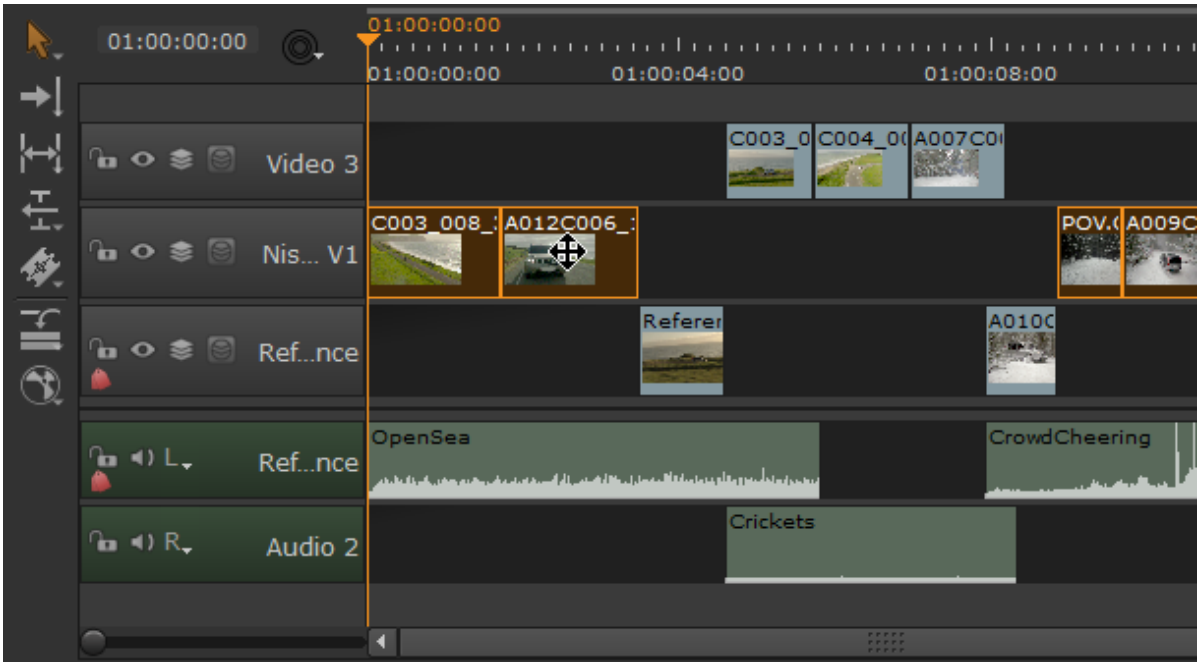


The track selection tools, activated by clicking the tool or by pressing **W**, selects multiple items depending on the initial selection:

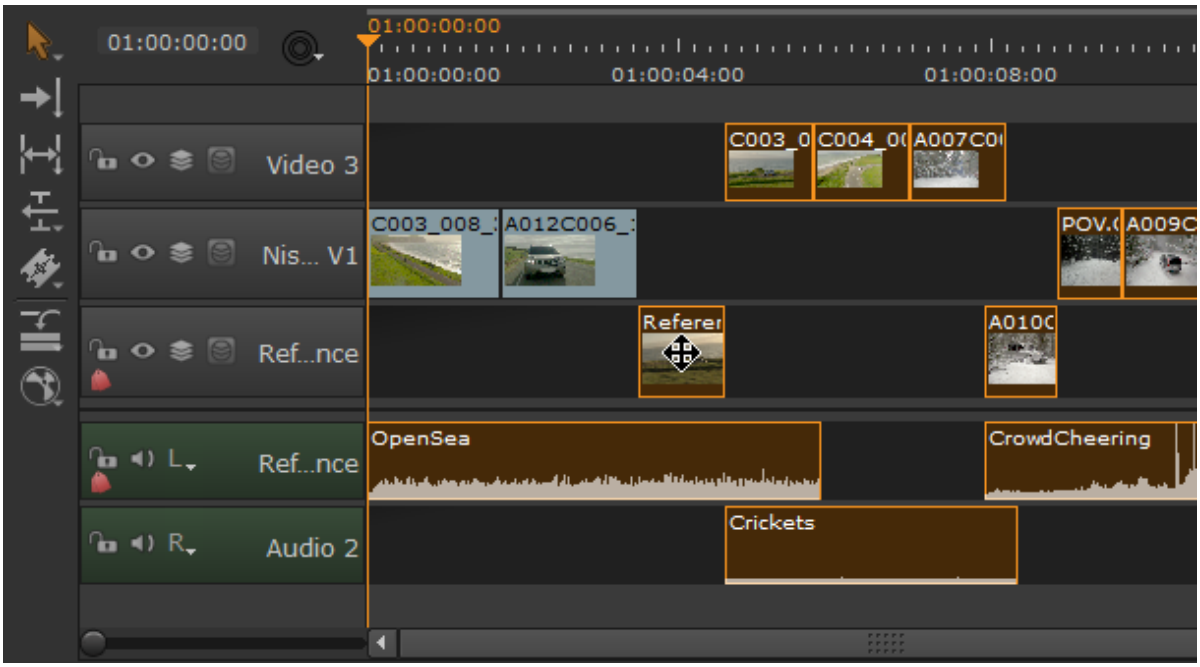
• **Select Track to Right** or **Left** - all shots right or left of the target shot are selected, within a single track.



• **Select All in Track** - all shots on the target track are selected, regardless of the item selected.



- **Select All Tracks Right** or **Left** - all shots right or left of the target item are selected, regardless of which track they occupy.



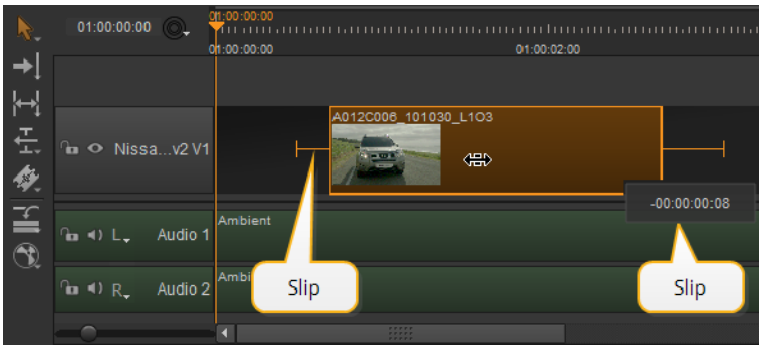
Using the Slip Clip Tool

The **Slip Clip** tool allows you to shift a shot's In and Out points by the same amount and in the same direction, retaining the original duration but altering the timeline output. Activate the **Slip Clip** tool by clicking the tool or pressing **E**.

Note: The target shot must have handles to use the **Slip** tool.

The **Slip Clip** tool displays different Viewer previews depending on whether the playhead is on the target shot or not, but the basic principles are the same.

Click the target clip to display the available handles and then drag the shot to the new position. Release the mouse to complete the slip.



Note: Using the **Slip Clip** tool does not move the shot on the timeline, only the output is changed.

Alternatively, nudge the slip using the **,** (comma) or **.** (period) keys or hold **Shift** to nudge by the **Frame Increment** set under the Viewer.

Tip: If you're using the **Multi Tool**, you can nudge using the "slip bar" by clicking at the bottom the shot.

The following Viewer previews are displayed, depending on the playhead position:

Note: The Viewer background always displays the playhead's current position.

- When the playhead is not currently on the target shot, the Viewer displays the **In** frame (1) and **Out** frame (2), allowing you to accurately gauge the new output.



- When the playhead is on the target shot, the Viewer displays the **In** frame (1), the **current** frame (2), and **Out** frame (3), allowing you to accurately gauge the output of the shot against the current frame.



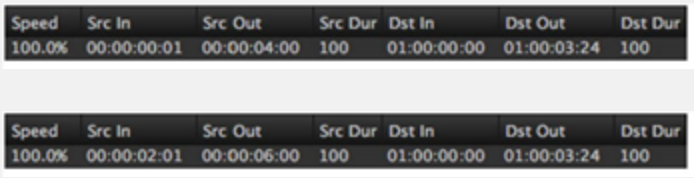
- When the playhead is on the target shot and A/B compare is active, the Viewer displays the target shot (1) and the reference shot (2), allowing you to synchronize your working track against the reference track.



Slipping Using the Spreadsheet View

You can slip shots using the **Src In** and **Src Out** columns of the spreadsheet:

1. Select the required event in the spreadsheet view.
2. Click the cog icon and select the required **Time Edit Behaviors** depending on whether you're using the In or Out point:

Modify	Using	Result
Src In	Slip Source	Adjusts the **Src In** and **Src Out** by the same amount, slipping the event while maintaining speed. Before and after a 2 second **Src In** increase: 
Src Out	Slip Source	Adjusts the **Src Out** and **Src In** by the same amount, slipping the event while maintaining speed. Before and after a 2 second **Src Out** increase: 

3. Adjust the **Src In** or **Src Out** to slip the shot(s) by the specified amount.

Using the Slide Clip Tool

The **Slide Clip** tool allows you to move a shot in relation to the item before and/or after the target item, without changing its length or timeline output. Activate the **Slide Clip** tool by clicking the tool or pressing **E** twice.

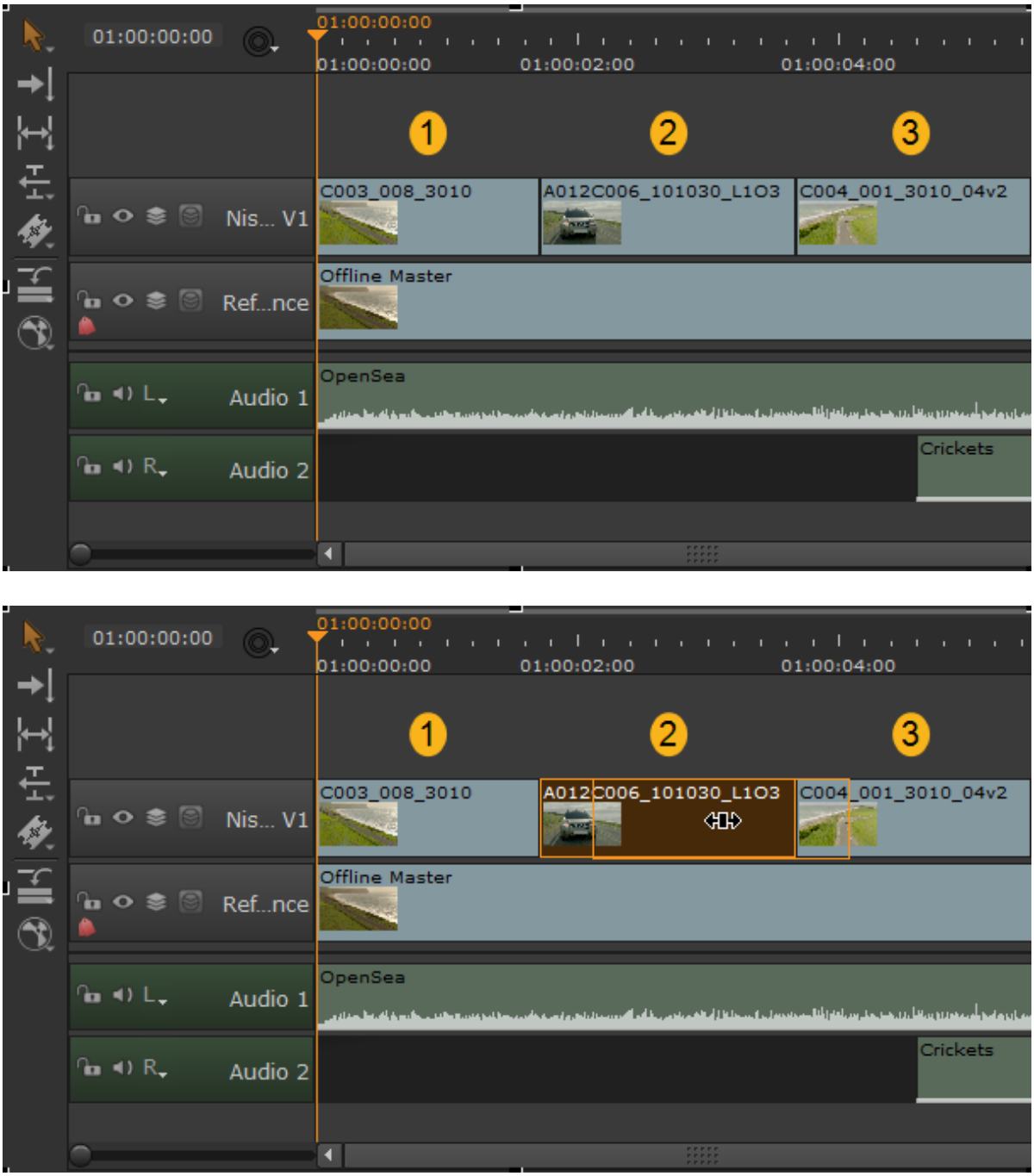
The shot either side of the target are shortened or lengthened within the limits of their handles to accommodate the slide.

Note: The surrounding shots must have handles to use the **Slide** tool.

Click the target shot and then drag it to the new position and release the mouse to complete the slide.

For example, if you slide the target shot (2) five frames to the right, the preceding item (1) ends five frames later and the next item (3) starts five frames later.

The first image shows a timeline containing three shot, and the second shows the same shots with the target (2) sliding to the right.



The Viewer displays the new end point of the previous shot on the left and the new start point of the next shot on the right, allowing you to accurately gauge the slide.

The two center images (2) represent the start and end frames of the target shot, which don't change.



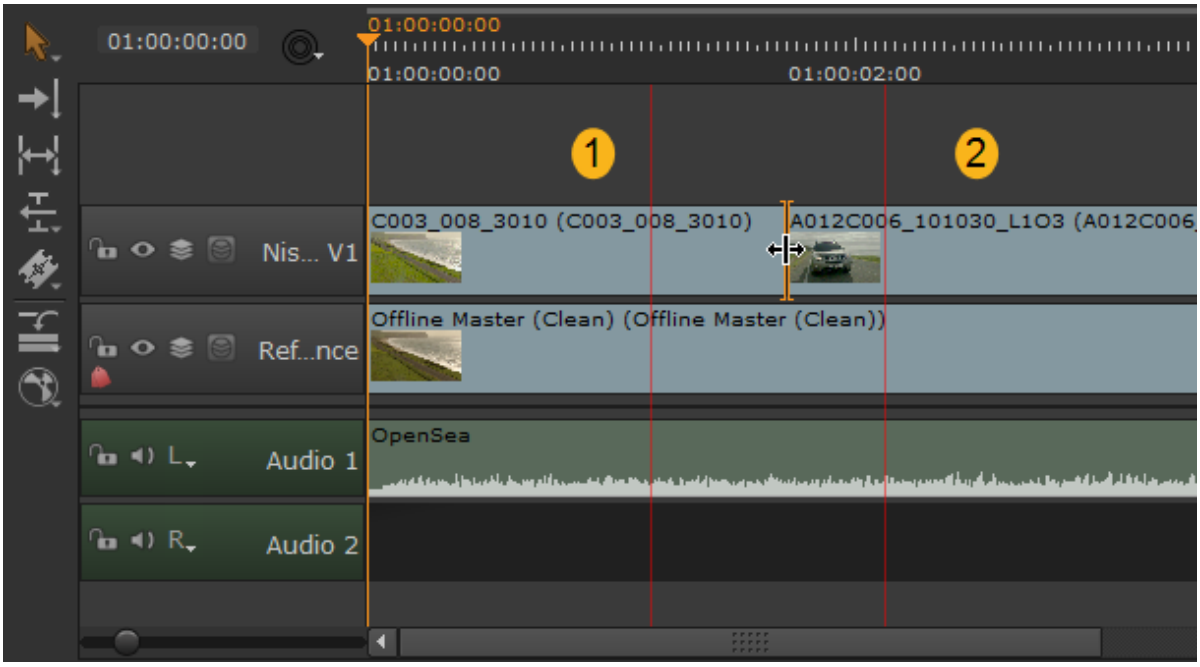
Using the Roll Edit Tool

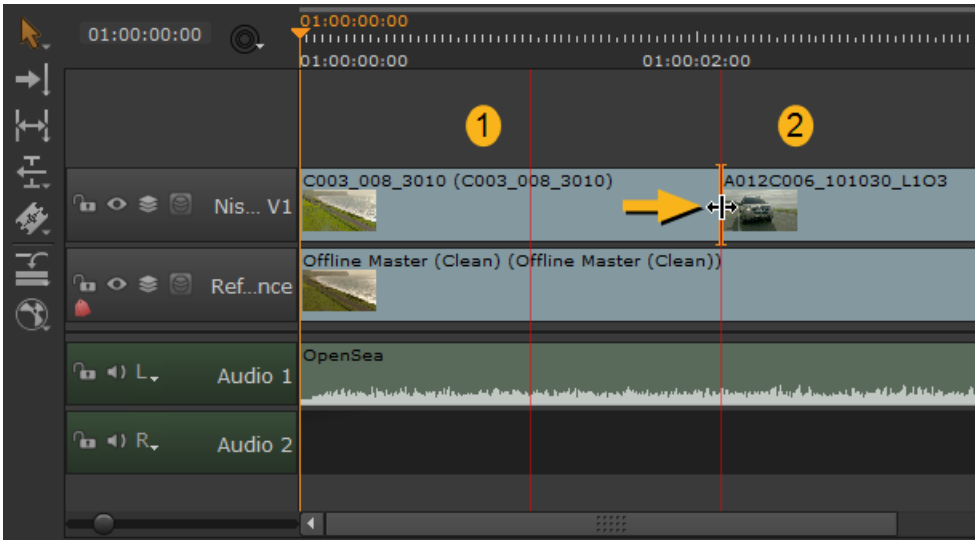
The **Roll Edit** tool enables you to roll a single edit within the available handles, shortening one shot while lengthening the other, but keeping the overall duration the same. Activate the **Roll Edit** tool by clicking the tool or pressing **R**.

Note: At least one of the target items must have handles to use the **Roll** tool.

1. Click an edit point between two shots to display the available handles as a red overlay.
2. Drag the edit to the new position and release the mouse to complete the roll.

   For example, if you roll a number of frames at the end of one shot (1), the next item (2) starts that number of frames later. The first image shows a timeline containing two shots, and the second shows the same items with the edit point "rolled" to the right.





The Viewer displays the pre-edit shot on the left and the post-edit item on the right, allowing you to accurately gauge the new position of the edit.

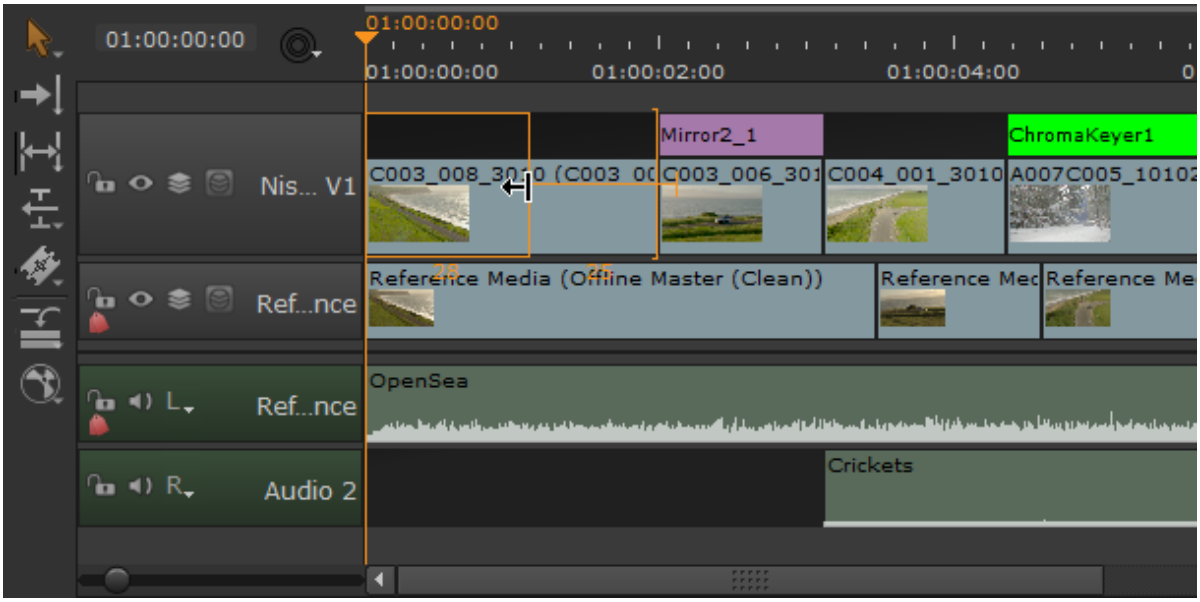


Alternatively, click the edit point between the shot and nudge the edit using the **,** (comma) or **.** (period) keys or hold **Shift** to nudge by the **Frame Increment** set under the Viewer.

Using the Retime Clip Tool

The **Retime Clip** tool allows you to trim a shot's In or Out point and automatically retime the clip to fill the new shot duration. Activate the **Retime Clip** tool by clicking the tool or pressing **R** three times.

Click-and-drag the edit point to the new position and release to complete the trim and retime. For example, trimming a 50 frame shot to 25 frames retimes the clip to 200%.





Alternatively, click an edit point and nudge the edit using the **,** (comma) or **.** (period) keys or hold **Shift** to nudge by the **Frame Increment** set under the Viewer.

Tip: By holding **Ctrl/Cmd** and dragging an edit, you can retime past the end of the shot's handles.

Using the Razor and Join Tools

The **Razor** tools allow you to cut shots in to separate parts so you can remove sections or rearrange items on the timeline. Activate **Razor** and **Razor All** by clicking the tool or pressing **R**.

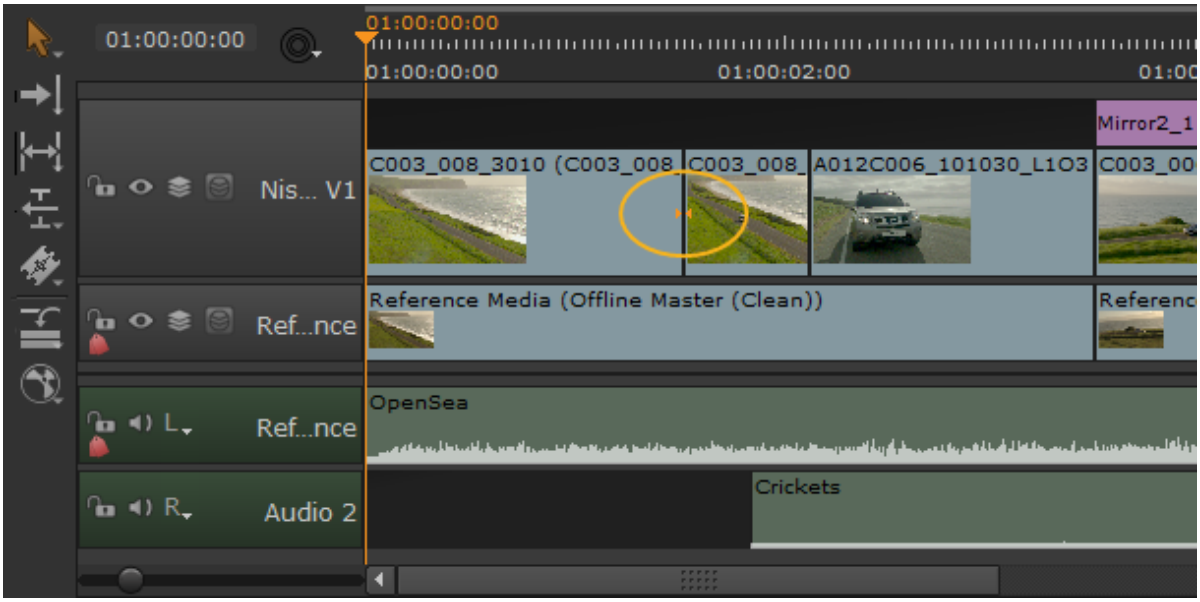
Place the cursor on the target shot, and if the cut is permissible, click to razor the shot or all shots depending on which tool you have selected.

Tip: The Razor cursor indicates whether a cut is permissible or not, such as on existing edits.

You can also apply cuts at the playhead position from the menu bar using **Timeline** > **Razor Selected**, or all tracks using **Timeline** > **Razor All**.

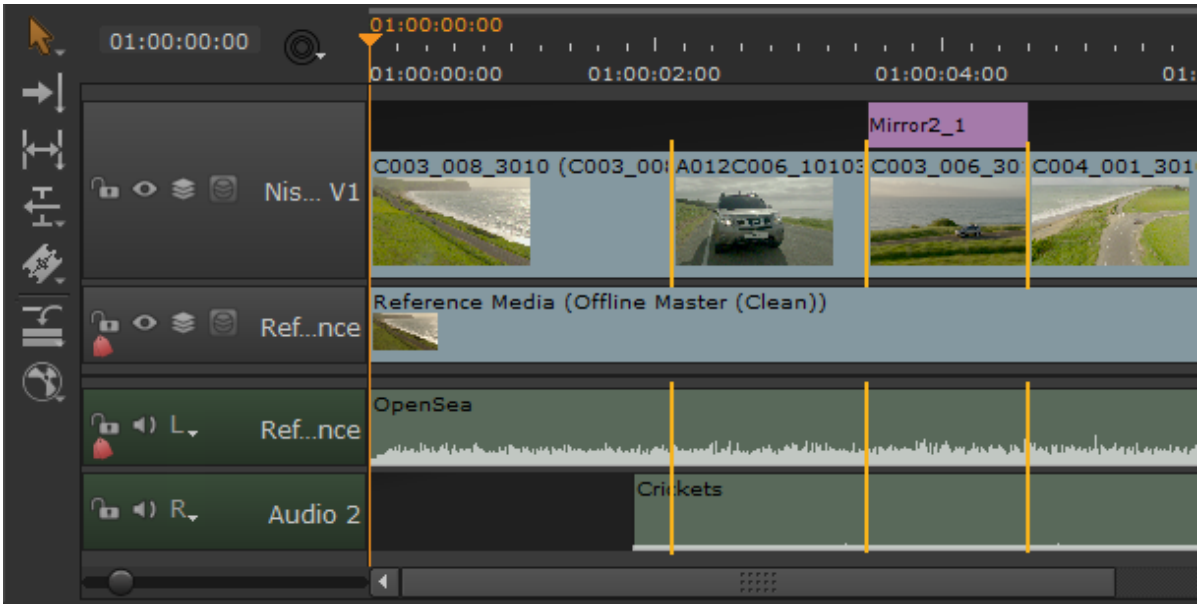
Tip: Use the **C** (with the shot under the playhead selected) and **Shift**+**C** keyboard shortcut, or the right-click context menu, to perform the respective cuts.

The **Join** tool can only be used on edit points between razored shots, denoted by the yellow arrows at the edit.



Copying Cuts Between Tracks

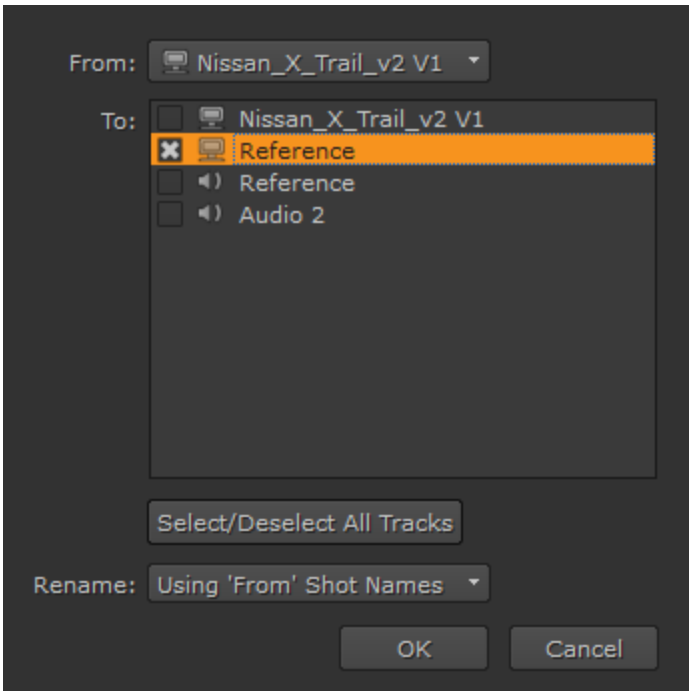
The **Copy Cuts** function allows you to quickly apply cuts from one track to other tracks on the timeline. For example, in the timeline shown, you could copy the cuts from the second video track to the Reference audio tracks.



To copy cuts:

1. Select the shots containing the cuts to copy, or if you intend to copy all the cuts from a track you don't need to make a selection.
2. Right-click in the timeline and select **Editorial** > **Copy Cuts**.

   The **Copy Cuts** dialog displays.



3. If you made a selection on the timeline, use the dropdown to select **Copy All Cuts** or **Copy Selected Cuts** as required.

   This dropdown is not displayed if no shots were selected.

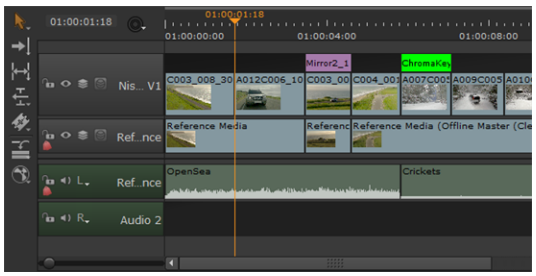
4. Click the **From** dropdown to select the source track.
5. Check all the destination tracks in the **To** field to which you want to copy the cuts.
6. Choose whether or not the resulting shots are named identically to the source track. Selecting **None** retains the destination clip name.
7. Click **OK** to copy the cuts to the destination track(s).

Insert, Overwrite, and 3-Point Editing

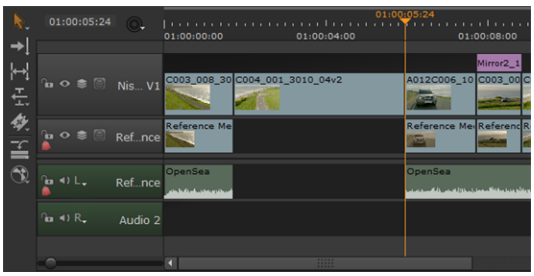
Insert and **Overwrite** edits are applied at the current playhead position by default, but the use of In and Out points in the clip Viewer and/or sequence Viewer can give you greater control over the result. 3-point editing, makes use of In and Out points in the clip Viewer and an In or Out in the sequence Viewer to control where the clip is placed on the timeline.

Inserting Clips

By default, **Insert** places the entire contents of the clip Viewer into the timeline at the current playhead position, on the lowest available track. All shots downstream of the playhead are rippled to make room for the clip. No items are overwritten or removed.



Pre-insert timeline



Post-insert timeline

Note: If the playhead is not positioned at an edit point, or there are shots on other tracks, the **Insert** action cuts the shot(s) at the playhead and ripples the cut downstream.
For example, the Post-insert image shows the audio shot being cut and rippled, even though it doesn't reside on the same track.

You can select a track before inserting if you don't want to target the lowest available track. Even if the target track is empty, shots on all other unlocked tracks are rippled by the same amount.

You can also use In and Out points to control where the clip is inserted and how many frames are included. See 3-Point Editing for more information.

To insert a clip at the playhead:

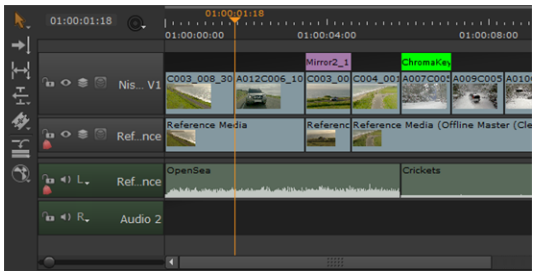
1. Navigate to **Workspace** > **Editing** to display the 2-up Viewer layout.
2. Double-click your sequence in the bin view to load it into the right-hand sequence Viewer .
3. Double-click the source clip to load it into the left-hand clip Viewer .
4. Place the playhead at the required edit point or timecode and select the target track, if necessary.
5. Navigate to **Clip** > **Insert**, or press **N**, to insert the clip into the timeline.
   All shots downstream of the clip are rippled to make room for the duration of the edit.

To insert a clip at an In or Out point:

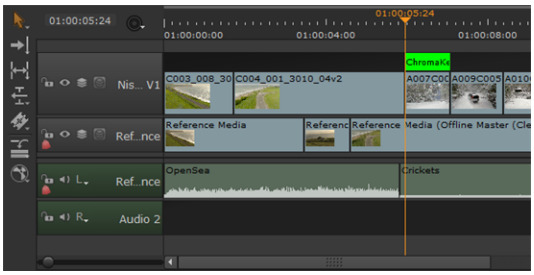
1. Navigate to **Workspace** > **Editing** to display the 2-up Viewer layout.
2. Double-click your sequence in the bin view to load it into the right-hand sequence Viewer .
3. Double-click the source clip to load it into the left-hand clip Viewer .
4. Place an In or Out point on the timeline to determine the clip's position:
   - **In point** - the source clip is inserted so that the first frame is at the In point specified.
   - **Out point** - the source clip is inserted so that the last frame is at the Out point specified.
5. Navigate to **Clip** > **Insert**, or press **N**, to insert the clip into the timeline.
   All shots downstream of the In or Out point are rippled to make room for the duration of the edit.

Overwrite Edits

Unlike inserting, **Overwrite** does not incorporate downstream ripple and doesn't alter the length of your sequence. Any shots you overwrite are destroyed, though they can easily be recovered from the source clips in the bin view.



Pre-overwrite timeline



Post-overwrite timeline

You can select a track before overwriting if you don't want to target the lowest available track.

You can also use In and Out points to control what the clip overwrites and how many frames are included. See 3-Point Editing for more information.

To overwrite at the playhead:

1. Navigate to **Workspace** > **Editing** to display the 2-up Viewer layout.
2. Double-click your sequence in the bin view to load it into the right-hand sequence Viewer .
3. Double-click the source clip to load it into the left-hand clip Viewer .
4. Place the playhead at the required edit point or timecode and select the target track, if necessary.
5. Navigate to **Clip** > **Overwrite**, or press **M**, to overwrite from the playhead for the duration of the source clip.

   All shots under the source clip are overwritten.

To overwrite from an In or Out point:

1. Navigate to **Workspace** > **Editing** to display the 2-up Viewer layout.
2. Double-click your sequence in the bin view to load it into the right-hand sequence Viewer .
3. Double-click the source clip to load it into the left-hand clip Viewer .
4. Place an In or Out point on the timeline to determine the clip's behavior:
   - **In point** - the source clip begins overwriting from its first frame at the In point specified downstream for the duration of the clip.
   - **Out point** - the source clip begins overwriting from its last frame at the Out point specified upstream for the duration of the clip.
5. Navigate to **Clip** > **Overwrite**, or press **M**, to overwrite from the In or Out point for the duration of the source clip.

   All shots under the source clip are overwritten.

3-Point Editing

Setting the output of a source clip and then editing the clip into a timeline at a specific point is sometimes referred to as 3-point editing. Using this method, you can insert and overwrite edits in an existing timeline or quickly construct scratch timelines from your source clips.

Firstly, set the output of your source clip using In and Out points in a clip Viewer, then set the reference In or Out point on your timeline to determine the clip's position. Finally, add the clip to the timeline using **Insert** or **Overwrite**.

Tip: You can set both In and Out points on the timeline, but bear in mind that there may be insufficient source frames for the range specified. If this is the case, blank frames are added and highlighted in red.

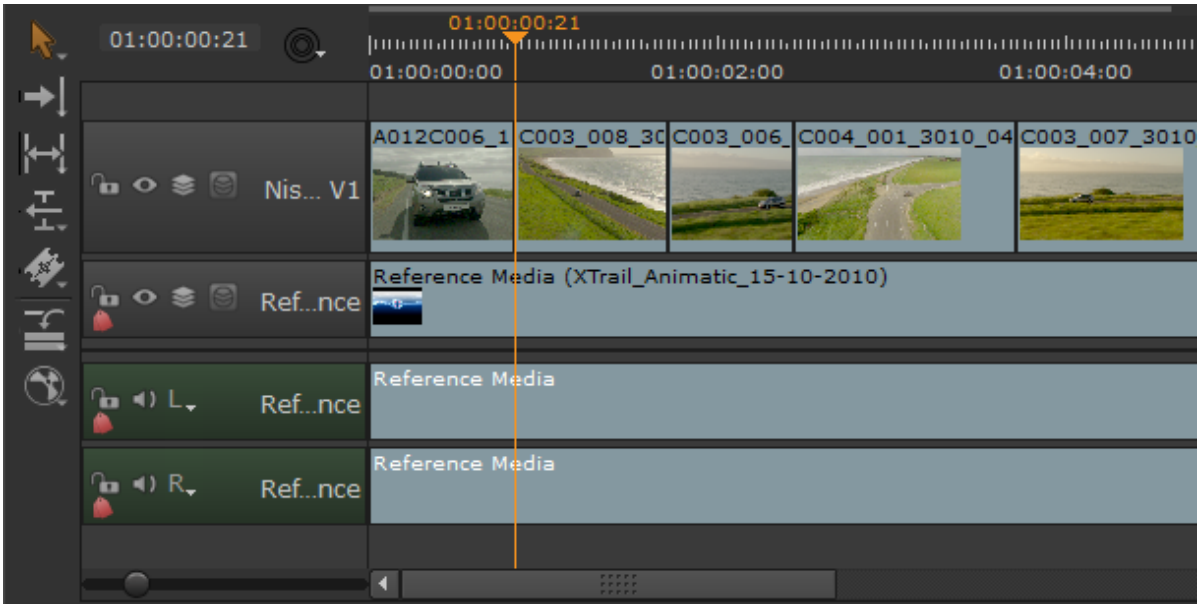
You can select a track before editing if you don't want to target the lowest available track. When inserting, even if the target track is empty, shots on all other unlocked tracks are rippled by the same amount.

1. Navigate to **Workspace** > **Editing** to display the 2-up Viewer layout.
2. Double-click the required source clip to load it into the left-hand clip Viewer .
3. Set the required frame range using In and Out points.
4. Double-click your sequence in the bin view to load it into the right-hand sequence Viewer .
5. Set In and/or Out points on the timeline to specify where the clip should be added and use **Insert** (N) or **Overwrite** (M) as required.

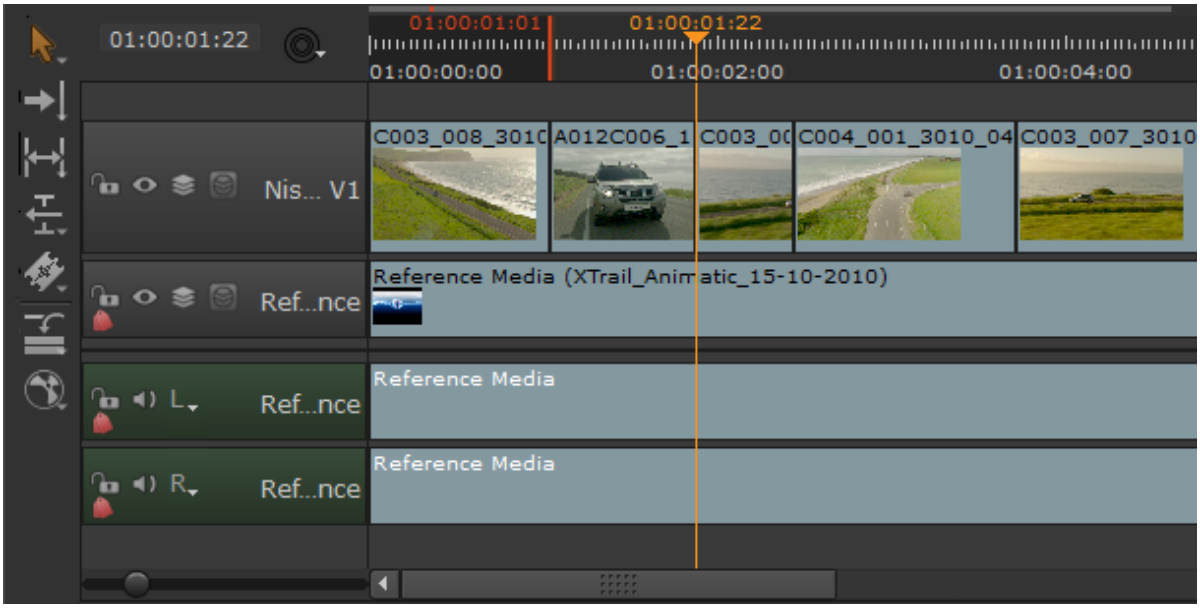
   As an example, assuming your clip Viewer and timeline are represented by the following image, and the Overwrite function is used:



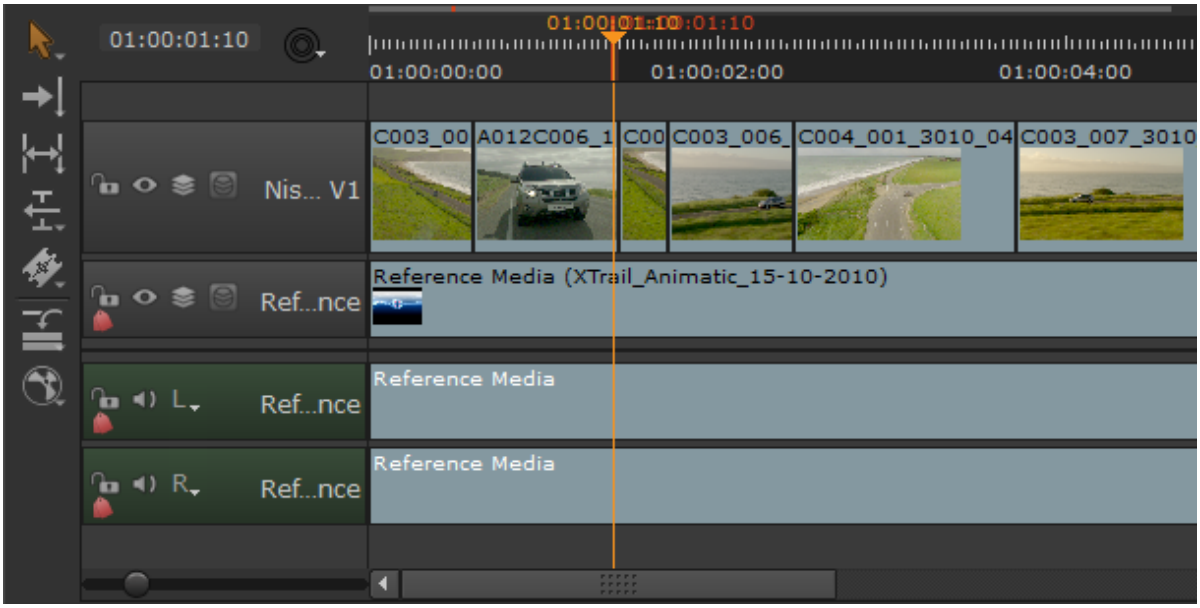
- **No In or Out points** - insert or overwrite at the current playhead position, for the range currently set in the clip Viewer.



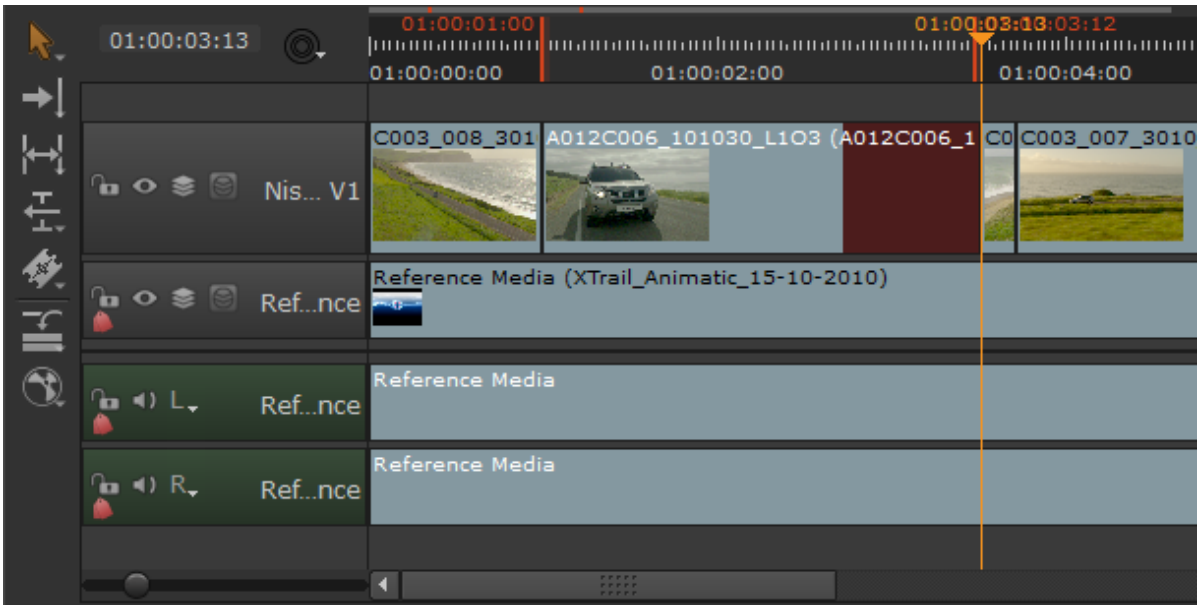
- **In point but no Out point** - insert or overwrite from the In point position downstream, for the range currently set in the clip Viewer.



- **Out point but no In point** - insert or overwrite from the Out point position upstream, for the range currently set in the clip Viewer.



- **In and Out points** - insert or overwrite at the current In point position, for the duration set by the timeline's In and Out points. If there are insufficient source frames for the range specified, blank frames are added highlighted in red.


01:00:03:13
01:00:01:00
01:00:03:13
01:00:03:12
01:00:00:00
01:00:02:00
01:00:04:00
C003_008_301
A012C006_101030_L1O3 (
A012C006_1
C0
C003_007_3010
Nis... V1
Reference Media (XTrail_Animatic_15-10-2010)
Ref...nce
Reference Media
L
Ref...nce
Reference Media
R
Ref...nce

Versions and Snapshots

In addition to the regular project save and restore options, Nuke Studio can record the different states of your workflow as you progress using versions and snapshots.

- **Versions** are children of clips. You can have any number of versions per clip as long as they follow the correct naming conventions, as shown in Using Versions. Versions can only be applied to source clips and shots and can be swapped in and out without overwriting existing work. Source clips have three states: Linked (default), Unlinked and Mixed, and you can define if a shot is linked to its source clip or not. See Version Linking for more information.
- **Snapshots** are time-stamped copies of a sequence, allowing you to save its current state without the inconvenience of saving the entire project. When you restore a snapshot, a warning displays prior to the restore reminding you that edits since the snapshot was taken are lost. See Using Snapshots for more information.

Using Versions

Versions can be added to source clips and shots to allow greater flexibility in your workflow. You can have as many versions as required and cycle through them quickly using keyboard shortcuts.



Note: You cannot use versions when a clip is opened as a timeline, that is, by using the right-click **Open In** > **Timeline View** option.

The application relies on specific file naming or directory structure conventions to discover versions:

Convention	Description	Example
File name constants		
Clip name	The file name base must remain the same.	**myClip**_v1.0001.dpx **myClip**_v2.0001.dpx

Convention	Description	Example
		myClip_v3.0001.dpx
Version prefix	The delineation between the file name and version information must be either _ (underscore) or **.** (period) and remain the same for all versions.	myClip_v1.0001.dpx myClip_v2.0001.dpx myClip_v3.0001.dpx
File name variables		
Version padding	The version number padding in the clip name can be increased or decreased.	myClip_**v1**.0001.dpx myClip_**v002**.0001.dpx myClip_**v03**.0001.dpx
Frame padding	The frame padding in the clip name can be increased or decreased.	myClip_v1.**01**.dpx myClip_v1.**1**.dpx myClip_v1.**0001**.dpx
Extension	The file format is interchangeable. See Appendix C: Supported File and Camera Formats for more information.	myClip_v1.01.**png** myClip_v1.0001.**dpx** myClip_v1.**mov**
Directory name constants		
Root directory	The root directory name must remain the same for all directories containing versions.	**~/version/**v1/myClip_v1.0001.dpx **~/version/**v2/myClip_v2.0001.dpx **~/version/**v3/myClip_v3.0001.dpx
Directory name variables		
Version padding	The version number padding in the directory name can be increased or decreased.	09_WF_Shot004_**v1**

Note: If the file extension is a movie format, such as **.r3d** or **.mov**, the **Frame padding** can be omitted.

Convention	Description	Example
		09_WF_Shot004_**v002** 09_WF_Shot004_**v03**

Version Linking

Control linking between source clip versions and shot versions.

Versions in Bins

Apply versioning to **Project** bin source clips.

Versions in Sequences

Apply versioning to shots in a sequence.

Version Linking

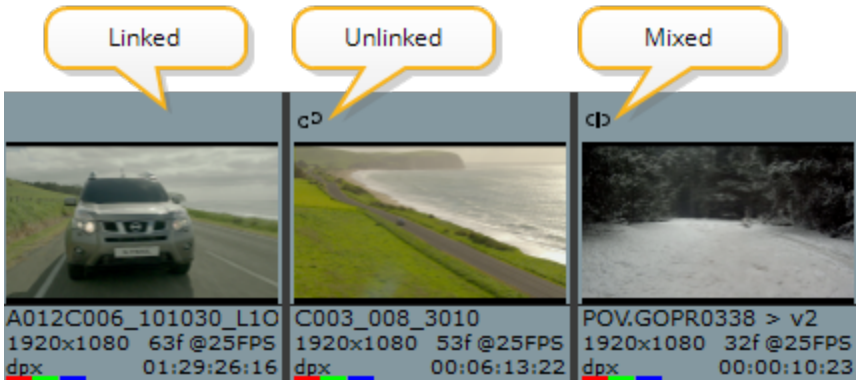
Version Linking allows linking and unlinking of versioning between source clips and shots. You can define if a shot version is linked to its source clip version or not, and there are three clip states:

- **Linked** - the source clip and all instances of shots referencing the clip are linked. Versioning up or down on any linked item affects all items.



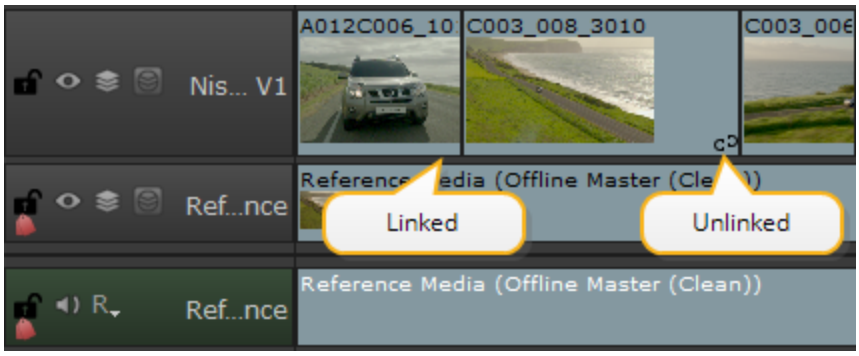
Note: Linked is the default state for all clips and shots, but you can set the default in the **Preferences** under **Project Defaults** > **General** > **Link bin and track item version**.

- **Unlinked** - the source clip and all instances of shots referencing the clip are unlinked. Versioning up or down on any clip or shot only affects that item.
- **Mixed** - the source clip is linked to some instances of shots referencing the clip, but not all of them. Versioning up or down on any clip or shot only affects linked items. Unlinked items are unchanged.

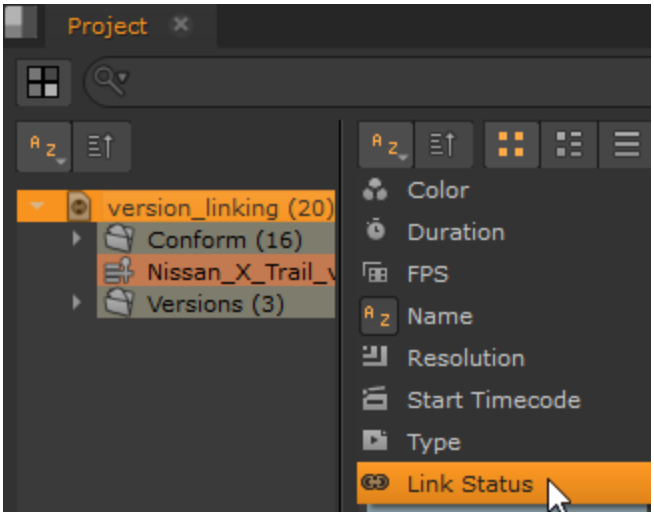


Two states govern shots in sequences:

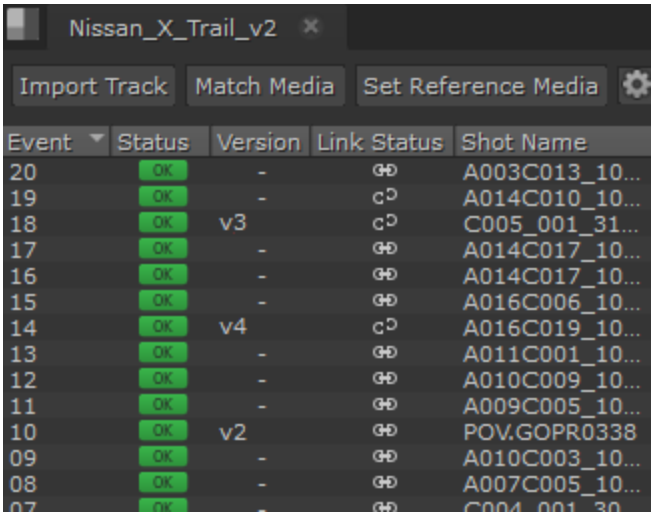
- **Linked** - The shot is linked to the version used by the source clip it references. Versioning up the shot or the source clip affects both.
- **Unlinked** - The shot is not linked to the version used by the source clip it references. Versioning up the shot or the source clip only affects the versioned item.



You can order the **Project** bin by **Link Status** to keep track of clip status and the spreadsheet view includes columns for **Version** and **Link Status** to keep track of shot status.



Source clip link status.

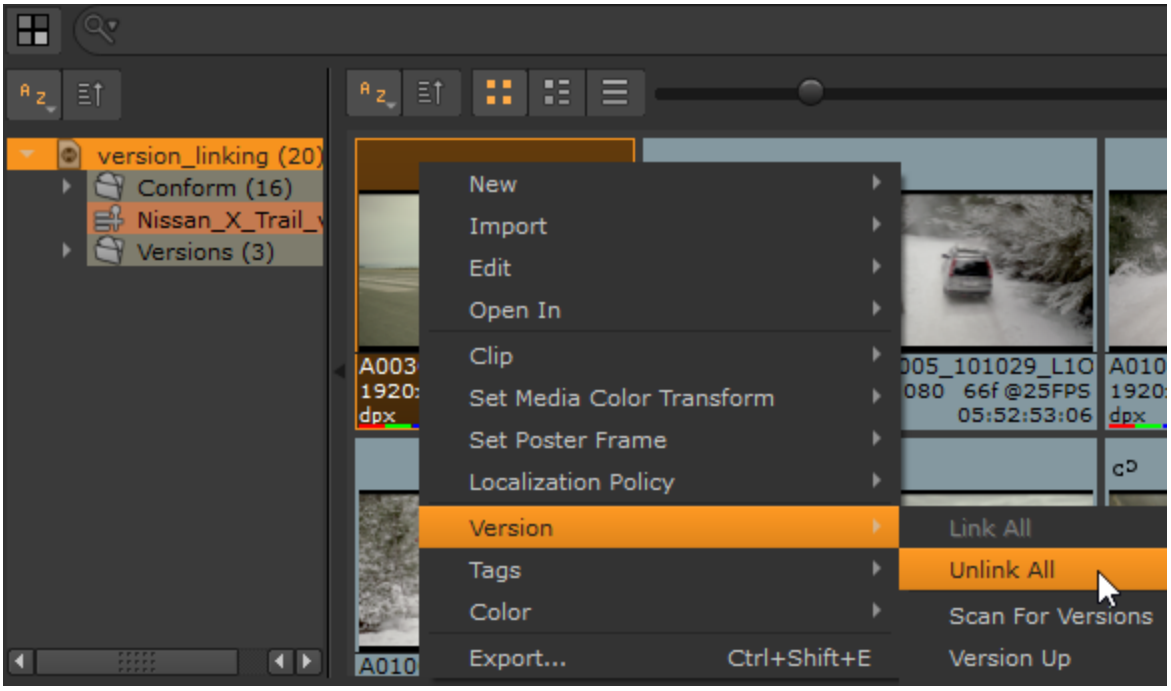


Shot version and link status.

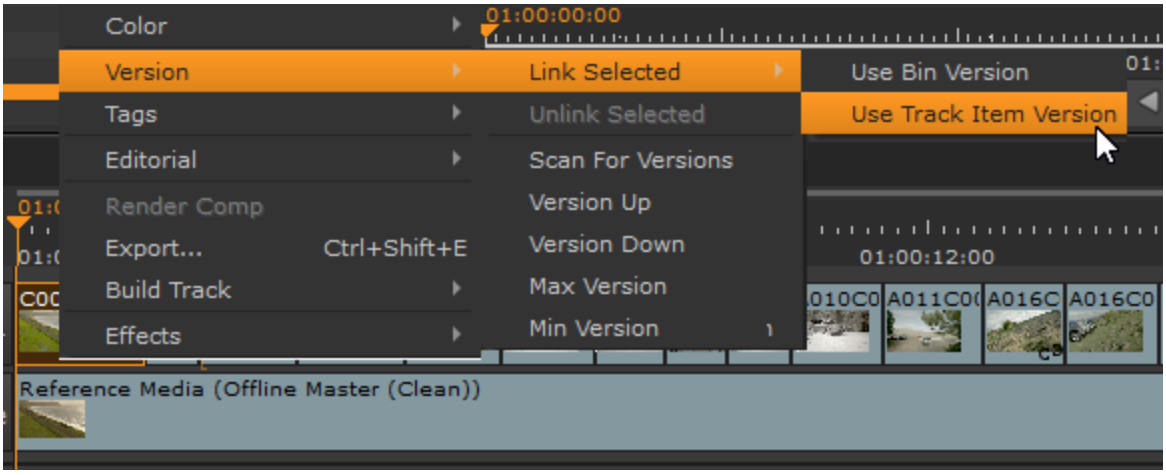
Linking and Unlinking Clips and Shots

Clip and shot versions are linked by default so that versioning up or down on any linked item affects all other items. You can unlink all clips and shots in new projects by disabling **Project Defaults** > **General** > **Link bin and track item version** in the **Preferences**. If you want more control over linking, you can right-click a source clip or shot and set the link state individually.

Source clips allow you to link or unlink all instances of shots in one action. **Link All** and **Unlink All** affect the version of all shots that reference the selected source clip.



Shots can be linked or unlinked to their source clip's version individually. If you choose to link a shot, you can choose to use the shot's current version or the clip's current version as the linked version.



Versions in Bins

Versions behave similarly in both bins and sequences, and in both cases, you first have to ingest an existing version.

Ingest and locate the versioned clip, then:

1. Right-click and select **Version** > **Scan for Versions** to search for available versions.

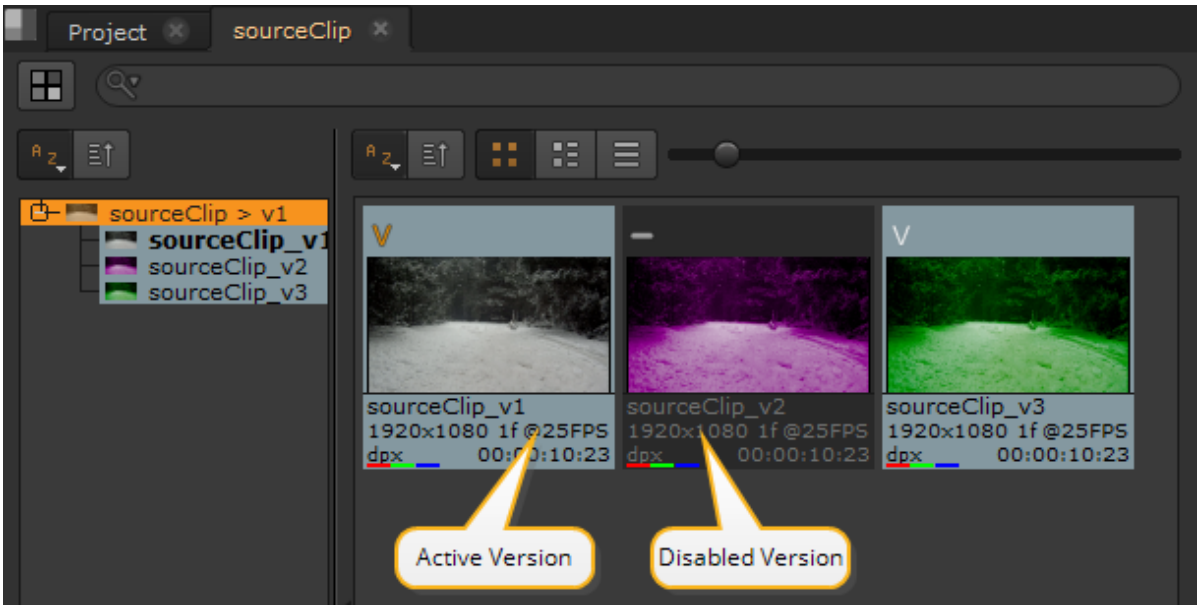
   A dialog box lets you know how many versions were discovered.
2. Use the right-click **Version** menu to:
   - Go to the next **Version Up** or **Version Down**.
   - Go to the **Minimum** or **Maximum Versions**.

Tip: You can also use the **Alt**+**Up**/**Down Arrow** keyboard shortcuts to increment versions or **Alt**+**Shift**+**Up**/**Down Arrow** to go to the maximum or minimum.

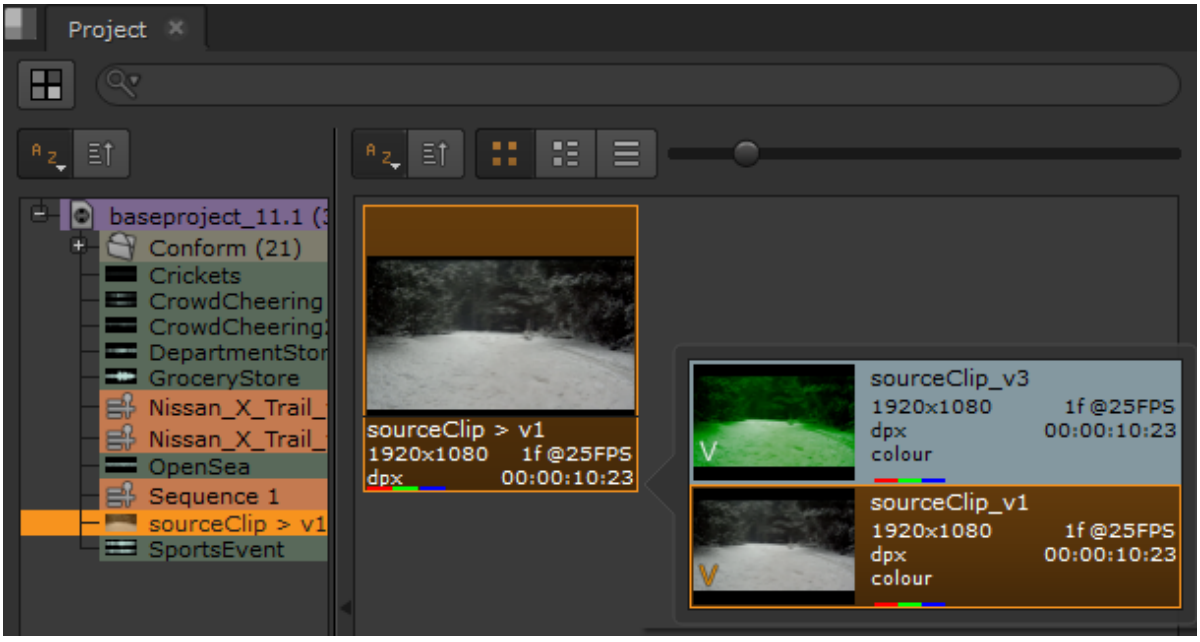
   When you reach the end of the discovered versions, incrementing the version automatically scans for new versions that may have become available.
3. For source clips only, you can right-click the clip and select **Open In** > **Versions Bin** to display all discovered versions of the target clip.

   The versioning conventions may allows clips into the Version Bin that you weren't expecting. You can disable versions by selecting them and pressing **D** or by selecting the **Set Active Version** of a clip using the right-click **Version** menu.

The **Active Version** is the version displayed when you drag the source clip to the timeline, denoted by the orange **V** in the top left-hand corner of the thumbnail.



4. Once you've sorted all the available versions, select a clip in the bin view and press **V** to display all versions for that clip in a convenient window. Disabled versions are not displayed.



5. Select the required clip to set the **Active Version** and apply it to the clip.

Versions in Sequences

As mentioned previously, versions behave similarly in both bins and sequences, but swapping versions in sequences allows you to compare results more easily.

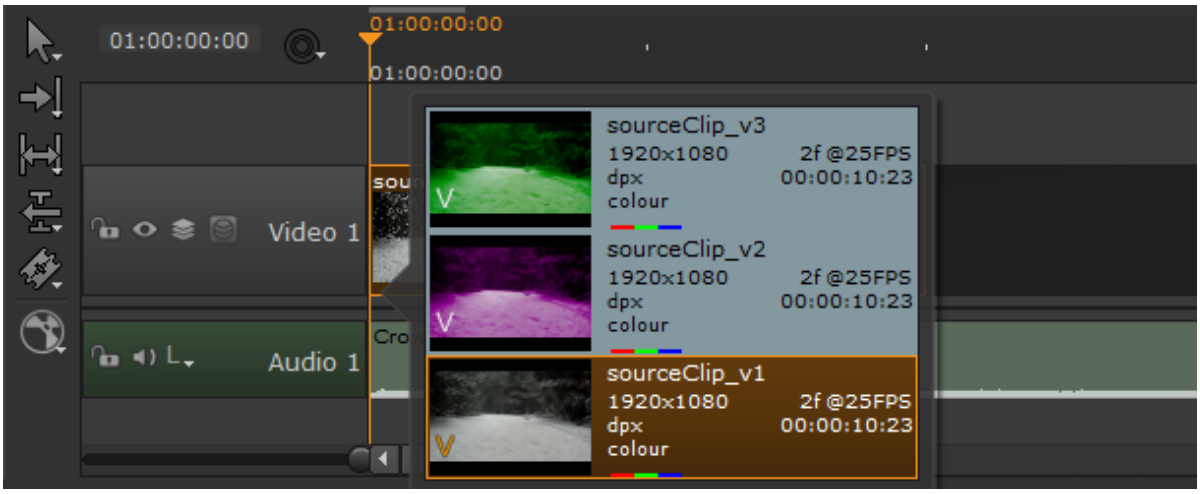
Note: You cannot use versions when a clip is opened as a sequence, that is, using the right-click **Open In** > **Timeline View** option.

Locate the ingested version clip and drag it to the timeline, right-click and select the **Version** menu:

- **Scan For Versions** to locate new versions of the clip.
- **Version Up** or **Version Down** to increment the version by one.
- Go to the **Minimum** or **Maximum Version**.

Tip: You can also use the **Alt** +**Up**/**Down Arrow** keyboard shortcuts to increment versions or **Alt**+**Shift**+**Up**/**Down Arrow** to go to the maximum or minimum.

Once you've scanned for versions, select a shot on the timeline and press **V** to display all available versions for that item in a convenient window.

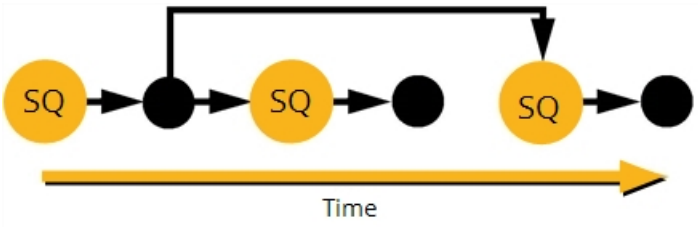


Select the required shot version to set the **Active Version**.

Using Snapshots

Within a project you can save the current state of a sequence as a snapshot, including a comment or tag to describe that snapshot. You can see what snapshots exist for a sequence in the bin view and flip it back to any previously saved state.

An example workflow might appear as follows:

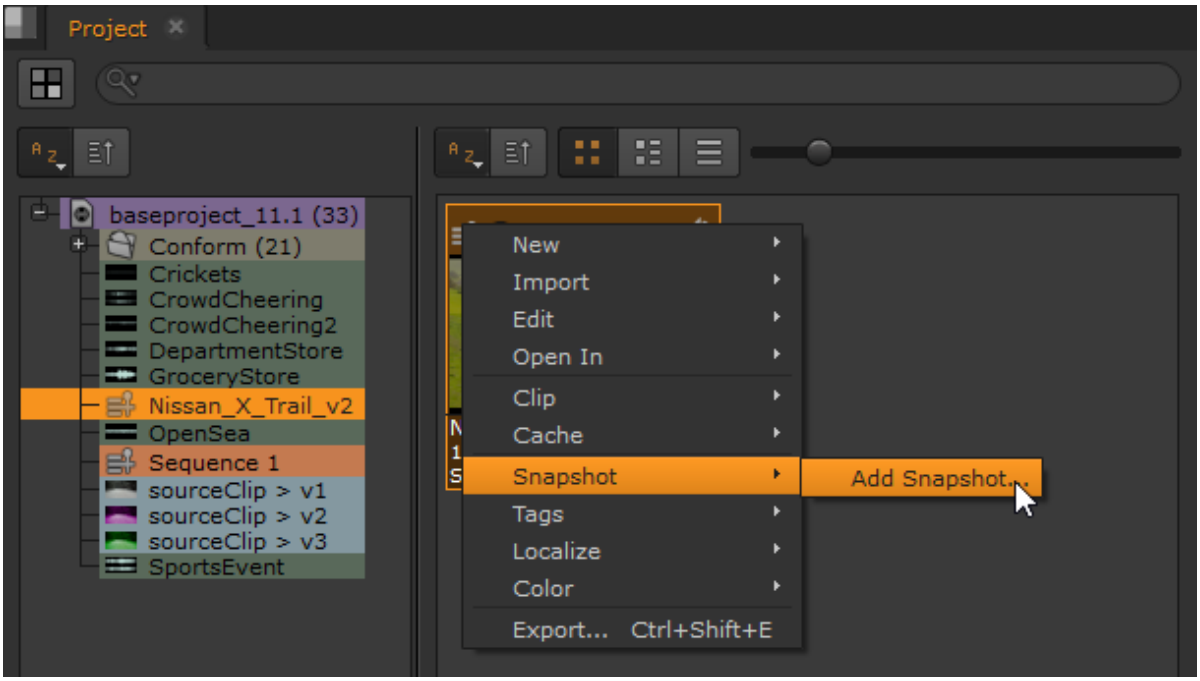


1. Two snapshots of the sequence (SQ) are recorded after edits. See Creating Snapshots.
2. Snapshot 1 is then restored. See Restoring Snapshots.
3. Further edits are made, then the sequence is recorded as snapshot 3.

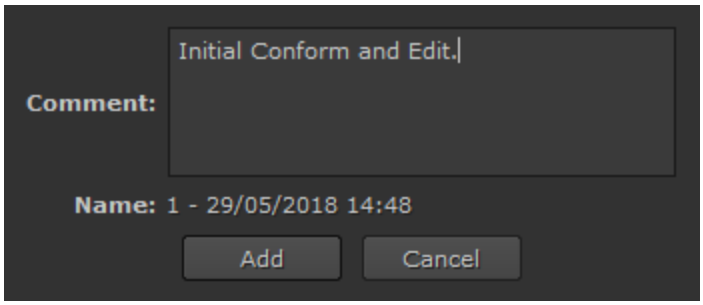
Creating Snapshots

To create a snapshot for a sequence:

1. Locate the sequence in the bin view.
2. Right-click the sequence and select **Snapshot** > **Add Snapshot**.

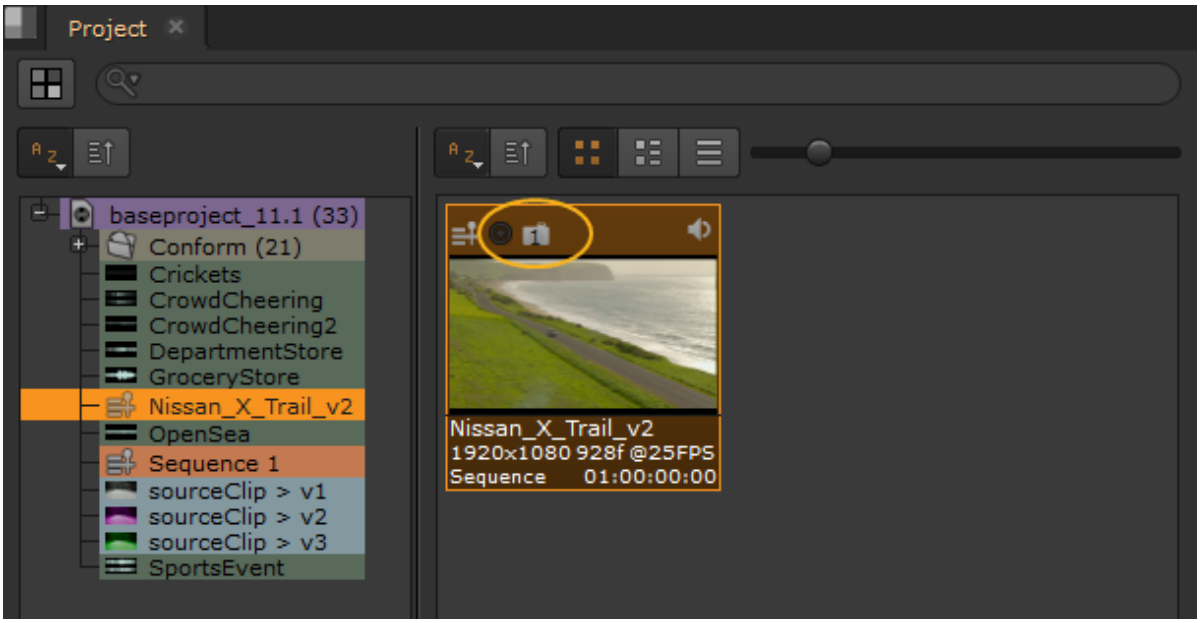


The **Add new snapshot** dialog box displays.



3. Enter a comment, or use the default date and time supplied.
4. Click **Add** to create the snapshot.

   Snapshots are indicated in the bin view with a camera icon containing the number of snapshots available.

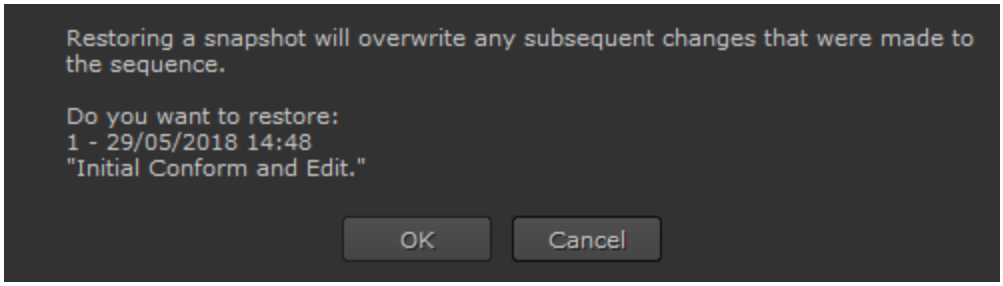


Restoring Snapshots

To restore a snapshot:

1. Locate the sequence in the bin view.
2. Right-click the sequence and select **Snapshot** > **Restore Snapshot**.
3. Select the required snapshot from the list.

   A warning displays reminding you that edits since the snapshot was taken are lost.



4. Click **OK** to restore the sequence to the point at which the snap was recorded.

Exporting from Nuke Studio

This section deals primarily with shot management and export functionality when you're farming out shots or sequences to other artists. It also deals with the presets, which dictate how **Create Comp** passes data between the Timeline environment and Compositing environment.

The export suite can transcode, export clip selections from a timeline or bin, write out EDL, OTIO, and XML files, or bake out an entire timeline as a single clip in your required delivery format. The **Export** presets are also used to manage how **Create Comp** sends clips back and forth between Nuke Studio's Timeline environment and Compositing environment using **Local** and **Project Presets**.

Export Dialog

There are two tabs on the Export Dialog - **Quick** and **Custom**. The **Quick** tab allows you to export your timeline to video format, without additional steps. The **Custom** tab allows you to change more settings such as export presets, files, tracks, and handles. See more at Introduction to the Export Dialog.

Export Options

- Exporting Sequences with Quick Export - an accelerated renderer that allows for fast exporting of sequences to .MOV formats.
- Exporting Sequences and Shots - the process of preparing a sequence or individual shots for export and paving the way for VFX work to come back into Nuke Studio.
- Transcoding - converts one file format to another. You can transcode sequences, timeline selections, and clips from the bin view.
- Ad Hoc Exports - an umbrella covering exports that you might not perform on a per project basis, such as EDL, OTIO, or XML exports.
- Create Comp - edit or create the presets used during **Create Comp**, passing source clips or shots to the Compositing environment and sending rendered Write nodes from the Node Graph back to the Timeline environment.

With the addition of Python bindings to perform the same functions, this infrastructure provides a massive amount of flexibility, whether your pipeline is GUI or command line orientated.

Note: Nuke Studio is non-destructive and can slot into your pipeline if you setup your shot template to mirror the existing file structure.

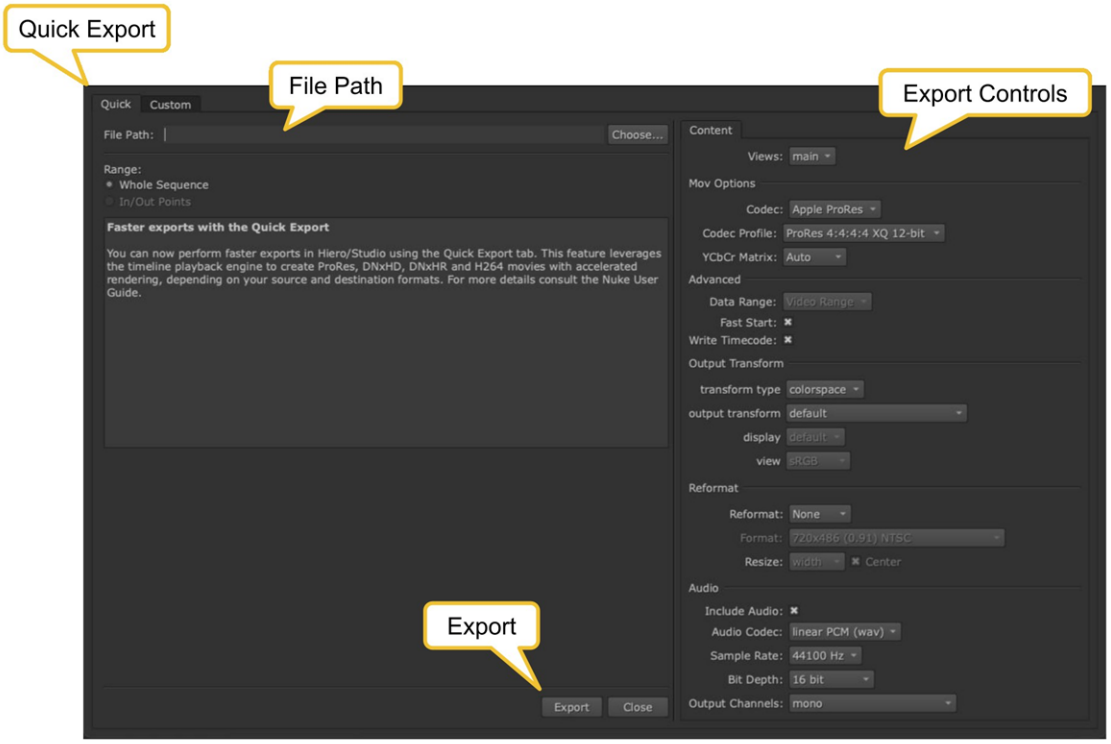
Introduction to the Export Dialog

Nuke Studio uses presets and shot templates to perform export operations, including round-tripping and EDL/OTIO/XML creation. The **Export** dialog controls what is exported and where, and whether or not to expect versioned clips as part of a round-trip from Nuke.

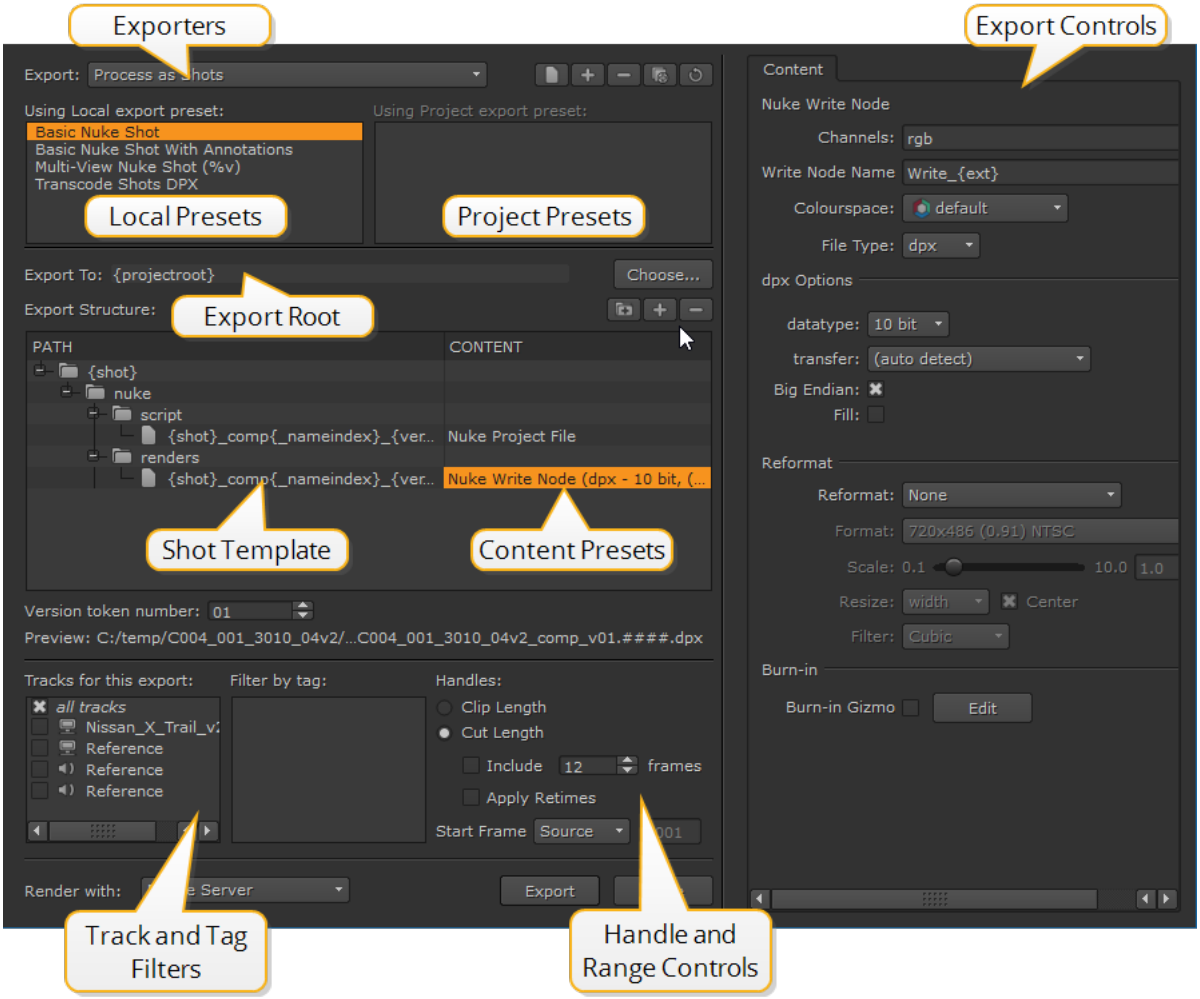
The **Export** dialog is accessed from the **File** menu, from the right-click bin and timeline menus, or by using the keyboard shortcut **Ctrl**/**Cmd**+**Shift**+**E**.

There are two tabs on the Export dialog - **Quick** and **Custom**. The **Quick** tab allows you to export your timeline to video format, without additional steps. The **Custom** tab allows you to change more settings such as export presets, files, tracks, and handles.

The Quick tab:



The Custom tab:



By default you will be taken to the **Quick** tab if the item you are exporting is compatible, otherwise you'll be taken to the **Custom** tab and the Quick tab will be greyed out.

If the source to be exported is compatible with both the Quick and Custom export options then you can switch freely between both. Remember that only the Quick export option leverages the accelerated rendering mechanism.

Note: See more about Quick Export at Exporting Sequences with Quick Export.

For Custom export, the Shot Template is also used to create the presets used during **Create Comp**, passing shots from the Compositing environment and sending rendered Write nodes from the Node Graph back to the Timeline environment.

Nuke Studio uses **Content Presets** in all shot templates, enabling you to create commonly used export conditions, which are then available across all projects. Some presets are only available with certain exporters, for example, the **EDL Exporter** preset cannot be used with **Process as Shots** exports.

You can filter your exports using the **Tracks for this Export** and **Filter by Tag** lists, exporting only certain tracks or shots marked with a particular tag. See Using Tags for more information.

For your convenience, Nuke Studio ships with a number of ready-made **Content Presets**, but you can edit these as required:

- **Audio Export** - copies any audio tracks to **.wav** files in a specified location.
- **EDL Exporter** - used to export a sequence to the EDL format.
- **External Render** - inserts a placeholder in the template so that other tasks can find the location of external renders.
- **Nuke Annotations File** - defines the script name and paths used by Nuke Write nodes and Precomp group during a round-trip or **Create Comp**.
- **Nuke Project File** - defines the script name and paths used by Nuke Read and Write nodes during a round-trip or **Create Comp**.
- **Nuke Write Node** - defines the render format for Nuke Write nodes. Add multiple Nuke Write Node presets to create multiple Write nodes in the resulting Nuke script.
- **OTIO Exporter** - used to export a sequence to OTIO format.
- **SymLink Generator** - creates symlinks to the location of the source files, rather than making copies.
- **Transcode Images** - defines transcode parameters allowing you to save your most-used file type conversions.
- **XML Exporter** - used to export a sequence to XML format.



Note: See more about export options at Exporting from Nuke Studio.

Exporting Sequences with Quick Export

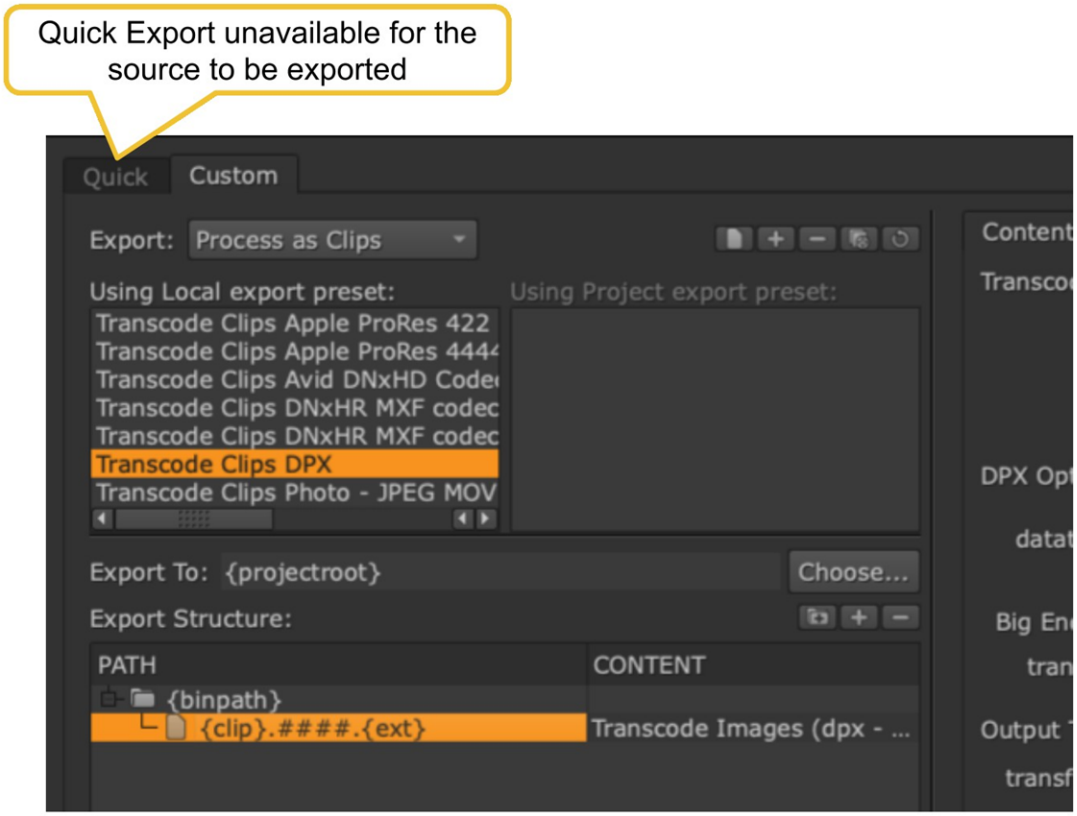
To export your sequence quickly to a compressed video file, for example to show to reviewers, you can use **Quick Export**. This exports a sequence from Nuke Studio or Hiero to a variety of video formats at a fast speed.

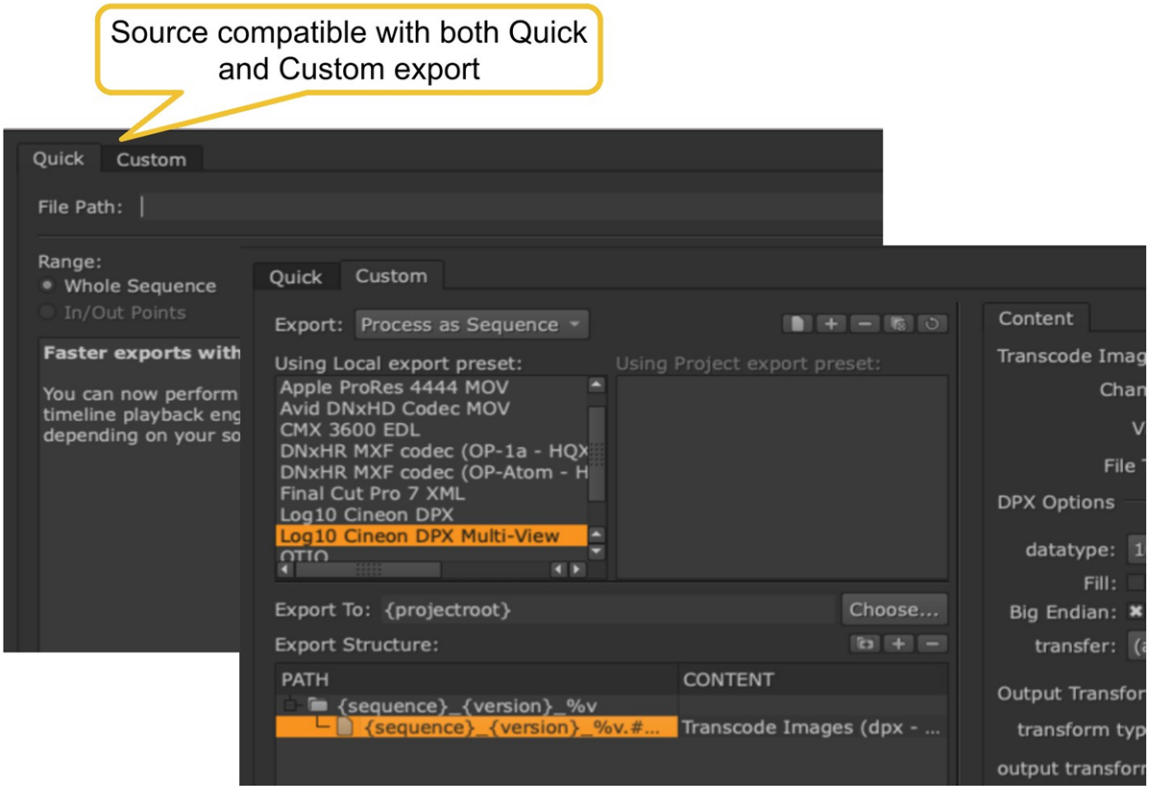
You can output your sequence to .MOV formats in the following codecs:

Apple ProRes	• 4:4:4:4 XQ 12-bit • 4:4:4:4 12-bit • 4:2:2 HQ 10-bit • 4:2:2 10-bit • 4:2:2 LT 10-bit • 4:2:2 Proxy 10-bit
Avid DNxHD	• DNxHD 444 10-bit 440Mbit • DNxHD 422 10-bit 220Mbit • DNxHD 422 8-bit 220Mbit • DNxHD 422 10-bit 145Mbit • DNxHD 422 10-bit 36Mbit
Avid DNxHR	• 4:4:4 12-bit • HQX 4:2:2 12-bit • HQ 4:2:2 8-bit • SQ 4:2:2 8-bit • LB 4:2:2 8-bit
H.264	• High 4:2:0 8-bit • Main 4:2:0 8-bit

By default you will be taken to the **Quick** tab if the item you are exporting is compatible, otherwise you'll be taken to the **Custom** tab and the Quick tab will be grayed out.

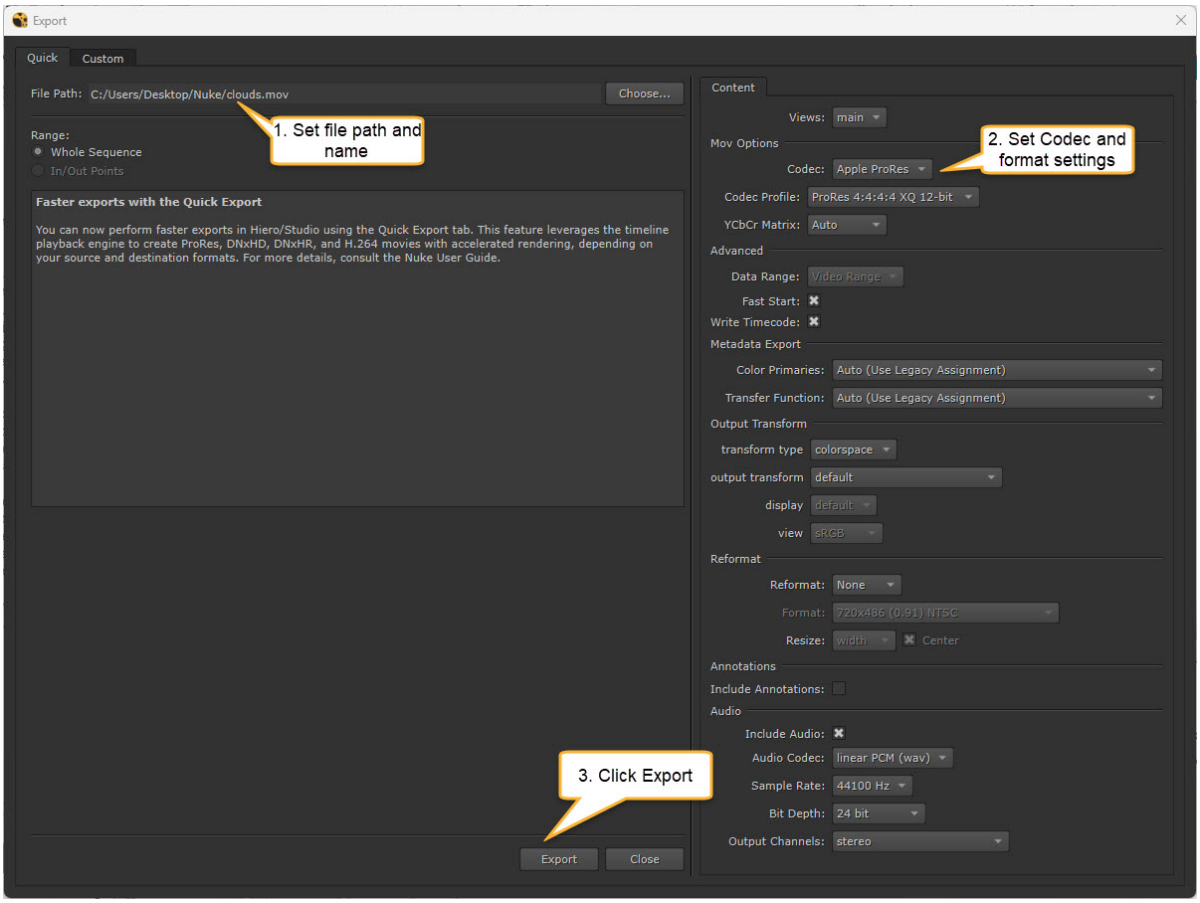
If the source to be exported is compatible with both the **Quick** and **Custom** export options then you can switch freely between both. Remember that only the Quick export option leverages the accelerated rendering mechanism.


Quick Export unavailable for the source to be exported
Quick
Custom
Export: Process as Clips
Using Local export preset:
Using Project export preset:
Transcode Clips Apple ProRes 422
Transcode Clips Apple ProRes 444
Transcode Clips Avid DNxHD Code
Transcode Clips DNxHR MXF codec
Transcode Clips DNxHR MXF codec
Transcode Clips DPX
Transcode Clips Photo - JPEG MOV
Export To: {projectroot}
Choose...
Export Structure:
PATH
CONTENT
{binpath}
{clip}.####.{ext}
Transcode Images (dpx - ...

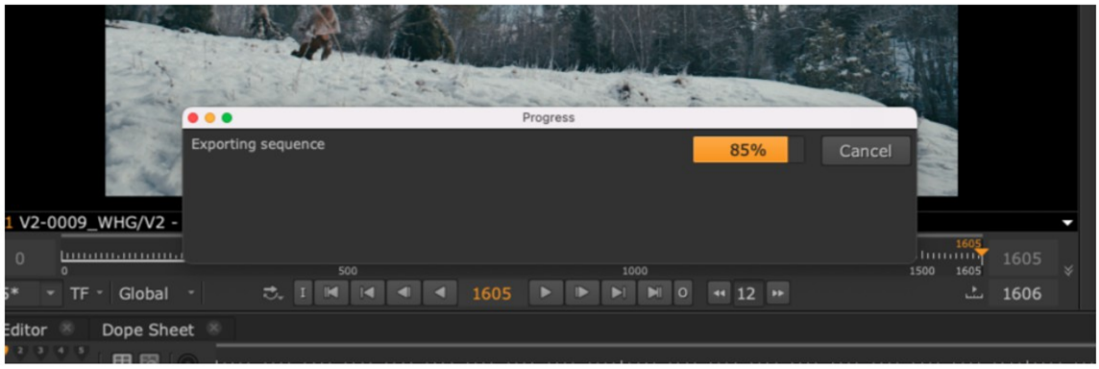

Source compatible with both Quick and Custom export
Quick
Custom
File Path:
Range:
Whole Sequence
In/Out Points
Faster exports with
You can now perform
timeline playback eng
depending on your so
Quick
Custom
Export: Process as Sequence
Using Local export preset:
Using Project export preset:
Apple ProRes 4444 MOV
Avid DNxHD Codec MOV
CMX 3600 EDL
DNxHR MXF codec (OP-1a - HQX
DNxHR MXF codec (OP-Atom - H
Final Cut Pro 7 XML
Log10 Cineon DPX
Log10 Cineon DPX Multi-View
Export To: {projectroot}
Choose...
Export Structure:
PATH
CONTENT
{sequence}_{version}_%v
{sequence}_{version}_%v.#...
Transcode Images (dpx - ...
Content
DPX Options
datatype:
Fill:
Big Endian:
transfer:

To use Quick Export:

1. Open the sequence that you wish to export in the sequence Viewer. Ensure that the layers you wish to be included in the export are enabled.
   Any video or audio tracks that are disabled will not be included in the final exported video file. Likewise any clips in the sequence that are disabled will also be ignored.*
   (*On Mac and Linus disabled Audio tracks or disabled audio clips will be ignored. On Windows all audio will be exported regardless of track or clip status.)
2. Select **File** > **Export**, or right click the sequence in the project view and choose **Export**. This can also be done by using the **Ctrl + Shift + E** hotkey combination.
3. This will open up the **Export** dialog.
4. In the **Quick** tab there will be a subset of options that are applicable to formats supported.
5. Choose a destination for your file path, configure the codec and format options. You can then select the **Export** button.



6. A progress popup will appear and update based on the percentage of the export process that is complete.



7. When the export is complete, you can choose to either open the file folder location or to close the window and return to the Studio UI.

Note: During export, the studio interface will be locked to prevent any changes to the sequence being outputted. It will be possible to cancel the export but a warning will appear notifying that any partially exported files will be deleted if it is canceled.

Quick Export - Additional Considerations

Limitations

- Currently only Whole Sequence in the Range options is available.
- Only on Windows; disabled audio tracks or audio clips are not ignored and will be exported.
- Currently there is no provision to save custom presets.

Behavior

- Currently only the {sequence} token is available in the file path name.
- The last used file path is persistent within the current session.

Using Local and Project Presets

Presets are containers for export preferences, such as file structure and format, and filters for tracks, tags, and frame range. Two types of **Presets** are available to construct commonly used export tasks:

- **Local Presets** - these presets are used to set up round-trips between artists on different platforms and also to manage passing files between the Timeline and Compositing environment. See Create Comp for more details. Local Presets are saved in a Task Presets folder using the XML file format.

- **Project Presets** - you can drag-and-drop **Local Presets** into this panel to save the preset within a project **.hrox** file. This option is designed for collaborative work, allowing you to quickly share your export presets.

You can edit these options in the Custom tab of the **Export dialog**. See Introduction to the Export Dialog.

Using the Shot Template

The shot template sets up the folder hierarchy and naming conventions for export presets such as **Basic Nuke Shot with Annotations** and **Transcode Clip DPX**, and how **Create Comp** sends clips back and forth between the Timeline environment and Compositing environment. Any folders added to the template are created during export unless they already exist, in which case the export writes to the existing structure.

Nuke Studio ships with default templates for your convenience, but you can quickly create custom templates using folders and "tokens", which are replaced with the relevant information during export.

Tip: Clicking an entry in the shot template displays a preview file path with the tokens resolved under the **Version token number** field.

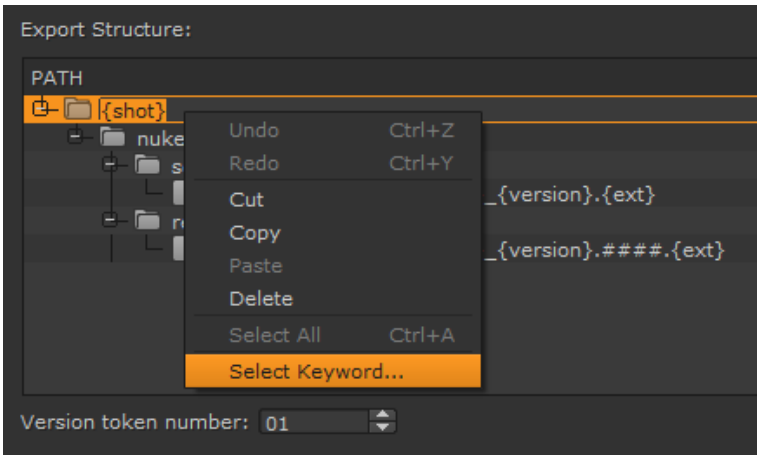
Exports can resolve the following tokens:

Token	Resolves to
_nameindex	The index of the shot name in the sequence, preceded by _ (underscore), to avoid clashes with shots of the same name.
{ampm}	The local equivalent of either AM or PM.
{binpath}	The bin structure to preserve. Including this token recreates your bin structure up to the nearest parent bin.
{clip}	The name of the clip used in the shot processed.
{day}	The local weekday name, abbreviated to Mon, Tue, and so on.
{DD}	The day of the month as a decimal, 01, 02, and so on.
{event}	The timeline event number associated with the shot to process.
{ext}	The extension of the file to output, such as **.dpx** or **.mov**
{filebase}	The base of the clip name to process. For example, the filebase of Shot01_####.dpx is Shot01_####.

Token	Resolves to
{fileext}	The format of the clip to process, such as **.dpx** or **.mov**
{filehead}	The source clip filename not including frame padding or extension. For example, the filehead of Shot01_####.dpx is Shot01.
{filename}	The source clip name of the media to process.
{filepadding}	The source filename padding, which you might use for formatting frame indices.
{filepath}	The full file path to the source media referenced in the export.
{fullbinpath}	The full bin structure to preserve. Including this token recreates the bin structure up to the project level.
{fullday}	The local full weekday name.
{fullmonth}	The local full month name.
{hierotemp}	The temp directory as specified in the Preferences.
{hour12}	The export start time hour component (12-hour clock).
{hour24}	The export start time hour component (24-hour clock).
{MM}	The month of the year as a decimal, 01, 02, and so on.
{minute}	The export start time minute component.
{month}	The local month name, abbreviated to Jan, Feb, and so on.
{project}	The name of the parent project of the export item.
{projectroot}	The root export file path as specified in the Timeline Environment Project Settings.
{second}	The export start time second component.
{sequence}	The sequence name to process.
{shot}	The name of the shot to process.
{timestamp}	The export start time in the 24-hour clock format (HHMM).

Token	Resolves to
{track}	The name of the track to process. Exporting EDLs using this token generates a separate EDL for each track.
{user}	The current username.
{version}	The string **v#**, defined by the number (#) set in the Version section of the export dialog
{YY}	The year of the century as a decimal, 01, 02, and so on.
{YYYY}	The year, including century.

Tip: Double-click the path column, right-click, and then choose **Select Keyword** to display a list of available export tokens, though only valid tokens for the current selection are listed.



Token substrings are valid if you need to extract a certain part of an evaluated string. For example, if **{shot}** resolves to JB100, then:

- {shot [0:2] } - resolves to JB
- {shot [-3:] } - resolves to 100

Similarly, anything within the outer brackets is evaluated as a Python string. For example, if **{shot}** resolves to JB_10_20, then:

- {shot.split('_') [0] } - resolves to JB
- {shot.split('_') [2] } - resolves to 20

Custom Shot Templates

The shot template enables you to create as many Nuke Read and Write nodes as required for a project. A typical use case might be creating **.jpg** clips for review and **.dpx** resolution clips for finishing.

Multi-format Exports

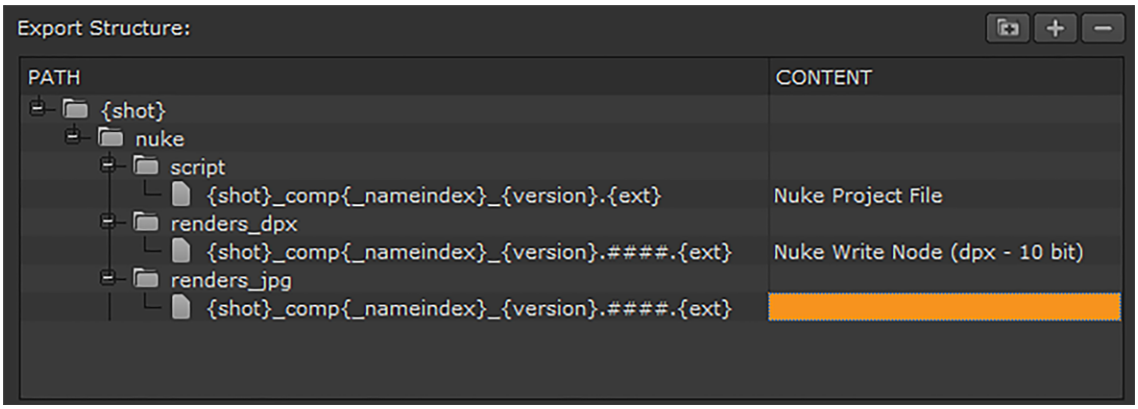
The following example describes how to build a shot template to export a sequence of **.mov** clips, create **.dpx** and **.jpg** Write nodes in Nuke, and bring the **.dpx** clips back into the timeline.

1. In the **Export** dialog, select **Basic Nuke Shot** in the **Local Presets** panel to auto-complete the shot template with the preset values.
2. Click **Duplicate selected preset** and give the new preset a name.
3. Rename the **renders** folder **renders_dpx**.
4. Select the **nuke** folder and click the folder icon to add a new folder. Name the new folder **renders_jpg**.
5. Select the **renders_jpg** folder and click  to add a new entry.
6. Replace the **{filename}** token with **{shot}_comp{_nameindex}_{version}.####.{ext}**, the same as the existing entry under **renders_dpx**.

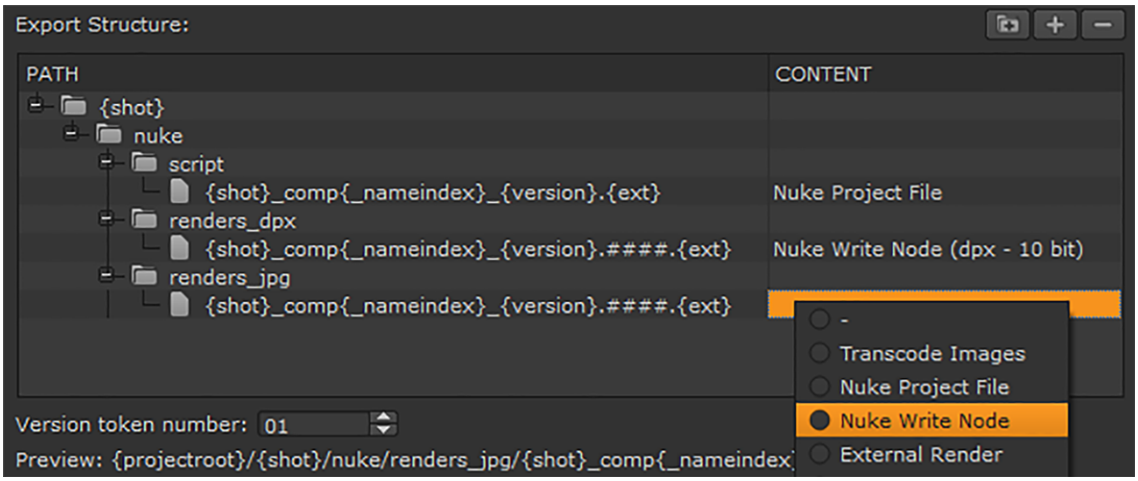


Note: The **####** marks represent frame numbers for image sequences. If you were creating **.mov** clips, they'd be omitted.

The shot template should look something like this:

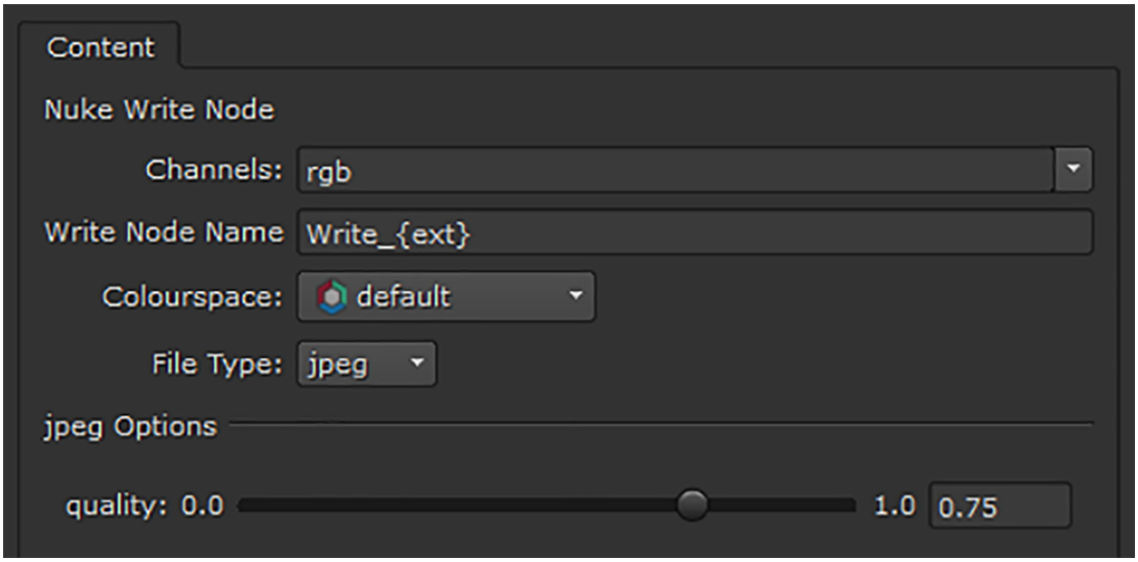


7. Click the **Content** column and select **Nuke Write Node**.



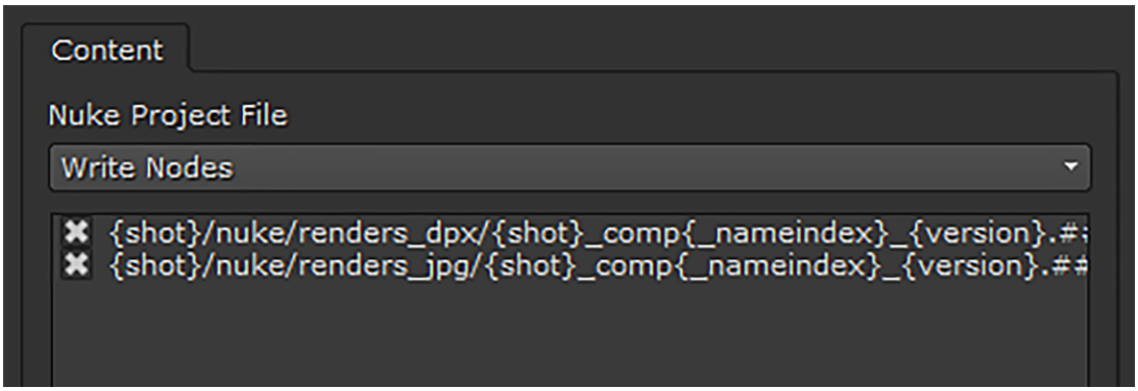
Note: When using a third party application to produce the VFX work, select **ExternalRender** instead of **Nuke Write Node**.

8. In the **Content** settings tab, use the **File Type** dropdown to select **jpeg**.



Notice that the settings available change depending on the **File Type** selected.

9. Click **Nuke Project File** in the shot template and check that both **Write** nodes are enabled.



10. Set up the rest of the export as described in Exporting Sequences and Shots and click **Export**.

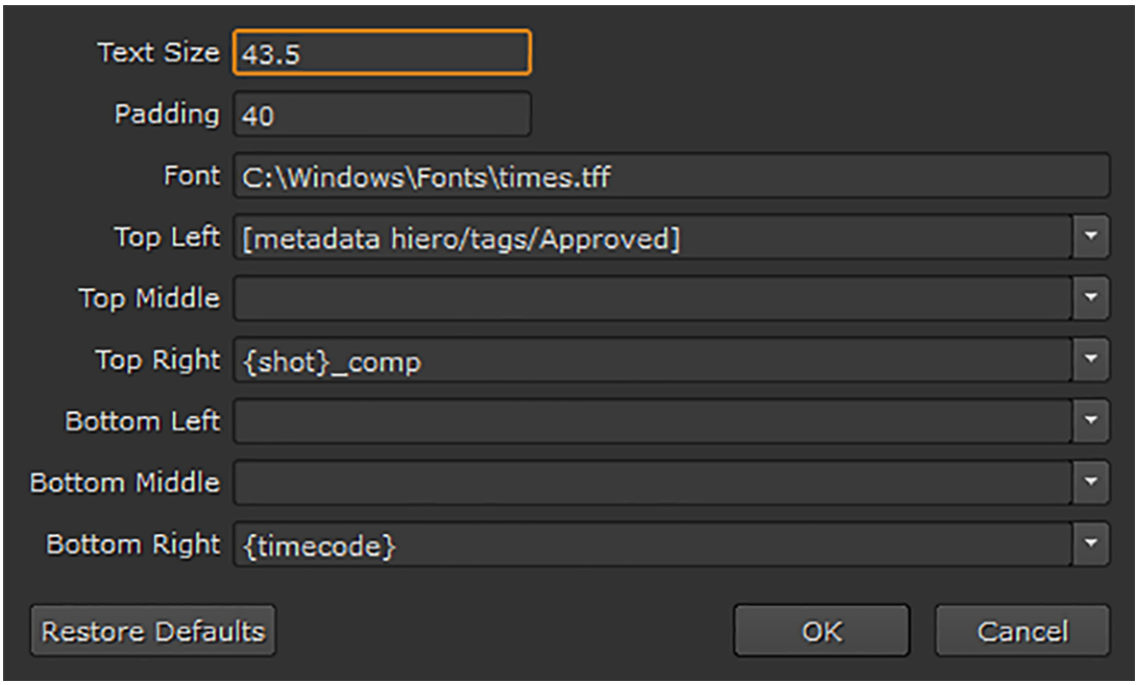
Adding Burn-in Text to Exports

Nuke Studio can burn-in text during the export process using a simple Nuke gizmo. The gizmo is accessed from the the **Export's dialog**'s Custom tab. Here you can find the Write Node preset's **Content** panel and edit the **Burn-in** section.

The gizmo contains controls for the font style and fields denoting the position of the text. You can also add burn-in directly on timeline using the Burn-In soft effect. See Adding Effects on the Timeline for more information.

Note: The **Font** field only accepts the full file path and name of the font file to use. For example, on Mac ~/Library/Fonts/Verdana.ttf

Click **Edit** to display the available controls.



You can mix-and-match the following methods to create burn-in text:

- Enter text manually, what you see is what you get in the burn-in.
- Use any of the tokens valid in the shot template as burn-in tokens. For example:

  ```
  {shot}_comp
  ```

  Extracts the shot nam from the timeline and appends **_comp**.

  See Using Local and Project Presets for more information.
- Use metadata from tags applied to clips and shots. For example:

  ```
  [metadata hiero/shot/tags/Approved]
  ```

  Extracts the Approved tag from the clip or shot. You can also append **note** to include any notes associated with the tag:

  ```
  [metadata hiero/shot/tags/Approved/note]
  ```



Note: You must precede spaces and slashes in the tag name with \\ (backslashes) to enable Nuke Studio to process the tag name correctly. For example: **[metadata hiero/shot/tags/Blue\\ Screen/note]**

Tip: If you're not sure what metadata keys and values are available on a shot, you can add a Text soft effect containing the Tcl expression **[metadata values]** to display all metadata in the Viewer. See Adding Effects on the Timeline for more information.

Adding Additional Nodes During Export

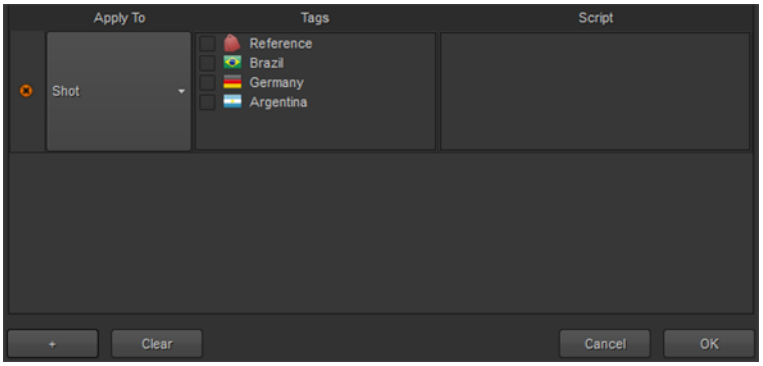
Nuke Studio can include additional nodes, in any **Nuke Project File** or **Transcode** export in the Shot Template, by simply copying and pasting scripts from the Node Graph.

You can add nodes to shots, tracks, or sequences, or include them as unconnected ad hoc nodes in the script, filtered by tags if necessary.

1. In the Export dialog's Custom tab, find the **Content** tab, scroll down to the **Additional Nodes** control and click **Edit**.

   The **Additional Nodes Setup** dialog displays.

2. Click  to add an entry.

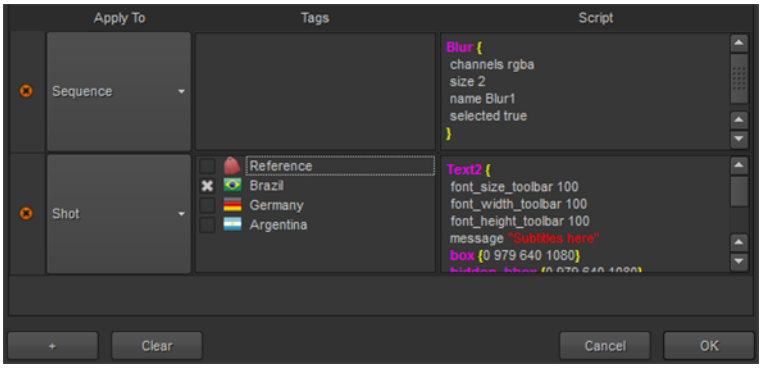


3. Click the **Apply To** field and select what the current entry applies to:
   - **Shot** - the additional nodes are added to the script for each shot in the export.
   - **Track** - the additional nodes are added to the script for each track in the export.
   - **Sequence** - the additional nodes are added to the script for the entire sequence.
   - **Unconnected** - the additional nodes are added to the script, but are not connected to the export node tree.
   - **None** - temporarily disables the current entry.
4. Select the **Tags** that you intend to use to filter which items receive the additional nodes.

If you want to affect only the Reference track, for example, select the Reference tag. All items without that tag are ignored.

5. Copy and paste a node from the Node Graph into the **Script** panel.

Note: If you need more than one node, you might consider creating a Group in the Node Graph and pasting that into the **Script** panel.



6. Click **OK** to accept the additional nodes.
7. Select the **Additional Nodes** checkbox and complete the export process as described Exporting Sequences and Shots.

Using the Frame Server on Render Machines

Although Nuke Studio is capable of rendering frames internally, running the Frame Server on a render machine can accelerate the process considerably by sharing work across a network of machines.

Note: The Frame Server requires a Nuke Studio license (nukestudio_i) on the main workstation, but only a Nuke render license (nuke_r) on the render machines. Local Frame Server processes use ports 5558-5662.

If you want to use an interactive license (nuke_i) on the render machines, add the **--useInteractiveLicense** argument to the **runframeserver.py** command described below.

Configuring the Frame Server on Render Machines

Nuke Studio's Frame Server can be set up on a render machine (or a number of machines) to render from your Nuke Studio session. To do this, you need to run the **runframeserver.py** script on the render machines, found inside the Python site-packages, with specific command line arguments.



Warning: In order for everything to work smoothly, you need to ensure that both your external render machines and main Nuke Studio session can read and write files to a shared location, such as an NFS share.

Depending on platform this can be done by manipulating your default **umask** setting, but be aware that this alters the permissions of the created files.

Additionally, Mac and certain Linux distributions, such as RHEL, can not function as the main workstation if the firewall is blocking the communication port 5560. You can configure the firewall to allow certain ports through the firewall using the **iptables** command, but use caution when doing so. For example:

```
sudo iptables -I INPUT -p tcp --dport 5560 --syn -j ACCEPT
```

Please refer to the documentation on firewalls for your particular platform for more information.

The Frame Server uses a number of worker processes on the render machine, each of which requires allocated resources, such as threads, memory, and so on. There are a number of arguments that you must pass to **runframeserver.py** for the server to work correctly:

- **--numworkers** - this is the number of concurrent Nuke processes that are launched when you run this server render node.
- **--nukeworkerthreads** - the number of threads that each worker is allocated. This is similar to setting the **-m** argument when running Nuke from the command line.
- **--nukeworkermemory** - the amount of memory, in MB, allocated to each frame server worker.
- **--workerconnecturl** - the TCP port address of the main workstation you want to serve. For example:

  ```
  tcp://bob:5560
  ```

  where **bob** is the resolved hostname of a machine you wish to serve. You can also use an IP address.

Tip: To ensure that you're entering a valid URL, try using the **ping** command to see if you get a response.

• **--nukepath** - the path to the Nuke application on the render machine.

Tip: On Windows, if there are spaces in the file path, remember to place the path in quotes. For example, **--nukepath="C:\Program Files\Nuke17.0v3\Nuke17.0.exe"**

On a Linux render machine, an example command prompt entry running from the application directory might look like this:

```
./python3 ./pythonextensions/site-
packages/foundry/frameserver/nuke/runframeserver.py --numworkers=2 --
nukeworkerthreads=4 --nukeworkermemory=8096 --workerconnecturl=tcp://bob:5560 -
-nukepath=./Nuke17.0
```

On a Windows render machine, an example command prompt entry running from the application directory might look like this:

```
python.exe pythonextensions\site-
packages\foundry\frameserver\nuke\runframeserver.py --numworkers=2 --
nukeworkerthreads=4 --nukeworkermemory=8096 --workerconnecturl=tcp://bob:5560 -
-nukepath=Nuke17.0.exe
```

On a macOS render machine, an example command prompt entry running from the application directory might look like this:

```
./python ../Resources/pythonextensions/site-
packages/foundry/frameserver/nuke/runframeserver.py --numworkers=2 --
nukeworkerthreads=4 --nukeworkermemory=8096 --workerconnecturl=tcp://bob:5560 --
nukepath=/Applications/Nuke17.0v3/Nuke17.0v3.app/Contents/MacOS/Nuke17.0
```

In the examples, we specify that the render machine uses two Nuke workers, with four threads and 8 GB RAM each, and are linked to the main Nuke Studio workstation running on **bob**.

Tip: If your render machines run a different OS than your main Nuke Studio machine, you can use the **--remap** command line argument to convert file paths between them. The host file path is read first followed by the render machine file path. Nuke Studio expects all file paths to use **/** (forward slash) between directories. For example:
`--remap "P:/,/mnt/renders/"`
converts host paths beginning with **P:/** (Windows style) to render machine paths beginning with **/mnt/renders/** (Linux style).

You can check that the Frame Server and workers are connected by running the following lines in the Script Editor on the main workstation:

```
from hiero.ui.nuke_bridge.FnNsFrameServer import frameServer
print([worker.address for worker in frameServer.getStatus(1).workerStatus])
```

Successful connections should report something similar to the following in the output panel:

```
['Worker 0 - henry.local - 192.168.1.11', 'Worker 0 - bob.local -
192.168.1.111', 'Worker 1 - henry.local - 192.168.1.11']
```

Where **henry.local** is the name of the render machine, and **bob.local** is the name of the main Nuke Studio session.

Note: If the workers cannot contact the Frame Server, an exception is printed in the Script Editor's output panel.

Frame Server Logs

Broker and Worker logging can to help diagnose Frame Server issues. The logs are written to NUKE_TEMP_DIR/logs by default, and take the form:

```
broker.log
worker-0.log
worker-1.log
worker-2.log
```

Note: Running the Frame Server using Python, as described above, always writes log files to the specific OS temporary directory. For example, on Windows **C:\temp** is used.

Tip: You can use the FRAMESERVER_LOG_DIR environment variable to force Frame Server logs into a different location. See Environment Variables in the Nuke Online Help for more information.

Exporting Sequences and Shots

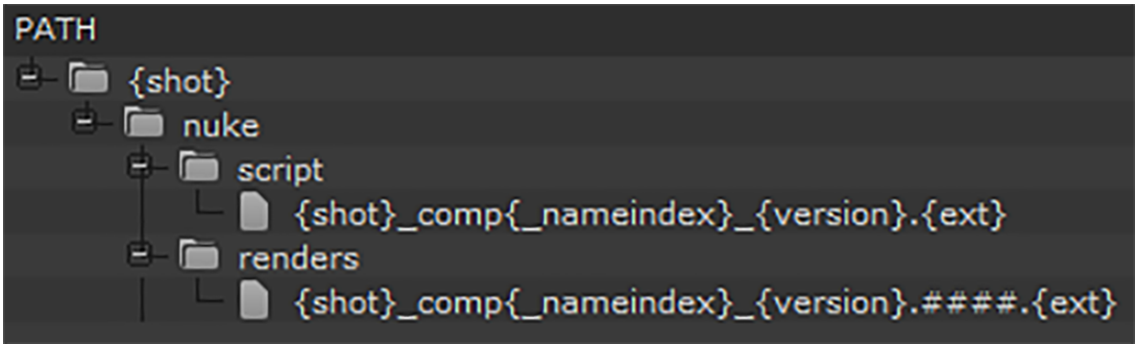
1. Select an entire sequence in the bin view, or shots in the timeline, and navigate to **File** > **Export...**

   The **Export** dialog displays.

2. If you simply want to export your current sequence to a compressed video file, you can export using the default settings on the **Quick** tab. If you wish to amend more settings, go to the **Custom** tab. Select **Process as Shots** from the **Export** dropdown.
3. Enter the **Export To** directory or click **Choose...** and browse to the location.

   The **Export To** directory is the starting point from which the shot template builds your shot hierarchy.
4. Select the **Basic Nuke Shot** preset under **Local Presets** to auto-complete the shot template or build a custom shot template by copying an existing template and editing as required using **Path** tokens, the **Contents** field, and the folder and +/- buttons.

   Basic Nuke Shot creates a folder for each clip, or shot, containing **nuke**, **script**, and **renders** folders.



   The tokens in the **Basic Nuke Shot** template break down as follows:

   - **{shot}** simply extracts the shot names as they appear in the timeline.
   - **{shot}_comp{_nameindex}_{version}.nk** extracts the shot name for each clip and the version selected in the **Tracks and Handles** controls. For example, **Shot01_comp_v03.nk**
   - **{shot}_comp{_nameindex}_{version}.####.{ext}** appends padding and the specified file extension. For example, **Shot01_comp_v03.0001.dpx**

Note: The **{_nameindex}** token is included to avoid conflicts with non-unique shot names.

Tip: Select a file entry in the shot template to display a preview of the file path with all the tokens resolved.

Article: See Knowledge Base article Q100195 for information on customizing exported Nuke scripts.

5. Proceed to Nuke Project File Settings to determine the Nuke script's behavior.

Nuke Project File Settings

Note: Custom shot presets can only be selected from the **Project Settings** if they contain a **Nuke Project File** and **Nuke Write Node** Content preset.

1. Open the Export Dialog's **Custom** tab. In the **Export Structure** section, click the **Nuke Project File** Content preset to display the script settings.
2. In the **Content** pane, select **Write Nodes** from the dropdown and check which path from the shot template should be used for the Nuke Write node. For example:

   {shot}/nuke/renders/{shot}_comp{_nameindex}_{version}.####.{ext} to resolve the render path where Nuke Studio expects to find the files when they're rendered.

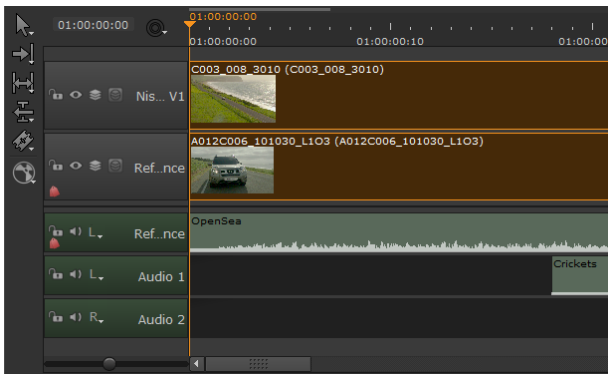
Note: If you included a **Nuke Annotations File** Content preset, enable the **Annotations** Precomp creator. See Annotations for more information.

3. If you're exporting retimed media, set how you want the Nuke script to handle the retime:
   - **None** - no retime is applied.
   - **Motion** - vector interpolation is used to calculate the in between frames. This is the most accurate retime method, but takes longer to render.
   - **Frame** - the nearest original frame is displayed.
   - **Blend** - a mix between two frames is used for the in between frames. This is quick to render and is useful when tweaking the timing in the Curve Editor before setting the method to **Motion**.
4. Soft Effects added to shots in your export are included in the resulting Nuke script by default. If you don't need the soft effects, disable **Include Effects** to omit them from the script. See Adding Effects on the Timeline for more information.
5. Select the required Reformatting options:
   - **Plate Resolution** - exports at the clip's original resolution, regardless of what is set in the timeline.
   - **To Sequence Resolution** - exports at the resolution set in the timeline **Sequence** panel **Output Resolution** dropdown.
   - **Custom** - activates the **Reformat** controls allowing you to customize the export resolution.
6. Enable **Collate Shot Timings** or **Collate Shot Name** to create additional Nuke Read nodes in the same script for clips that would normally be hidden by clips higher up the track hierarchy or clips on the same track with the same shot name.

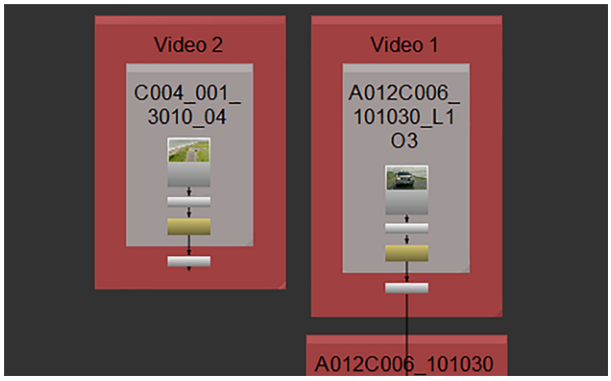
Note: If you have a Read node selected, you can't enable the **Collate** functions.

For example:

- **Collate Shot Timings** - Items on track 1 that would otherwise be hidden by track 2.



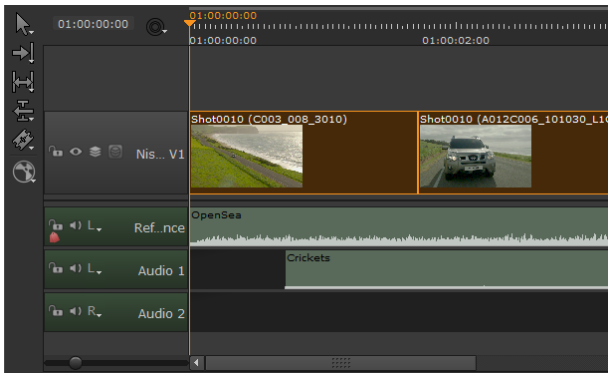
Timeline environment



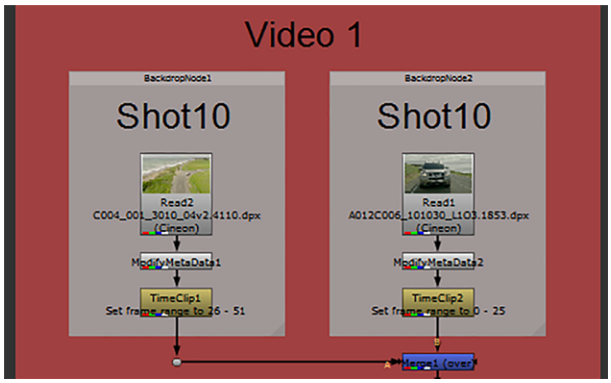
Compositing environment

Note: Shots on different tracks are not connected by default. If you want all the exported clips to be connected to the Nuke script Write node, enable **Connect Tracks**.
This applies to stereo and multi-view sequences as well when you use separate files for the tracks per view. See Stereoscopic and Multi-View Projects for more information.

- **Collate Shot Name** - Two items on the same track with the same shot name.



Timeline environment



Compositing environment

7. If you want to add additional nodes to the script on export, enable **Additional Nodes** and click **Edit**. See Adding Additional Nodes During Export for more information.
8. Proceed to Nuke Write Node Settings to determine the Write node's behavior.

Nuke Write Node Settings

Note: Custom shot presets can only be selected from the **Project Settings** if they contain a **Nuke Project File** and **Nuke Write Node** Content preset.

1. In the Export dialog's Custom tab, find the **Content** presets. Click the **Nuke Write Node** Content preset to display the write settings.
2. Set the following controls common to all file types:
   - **Channels** - set the channels to export using the dropdown. If you want to export a non-standard channel, type the name of the channel into the field manually.
   - **Write Node Name** - if you intend to create more than one Nuke Write node, define the name here. The default, **Write_{ext}**, appends the individual Write nodes with the file extension being written. You can, however, use any of the tokens Nuke Studio recognizes.
   - **Output Transform** - use the dropdown to set the colorspace to render, such as **linear**, **REDLog**, or **raw**.
3. Select the **file type** to render using the dropdown and complete the relevant fields, dependent on the **file type** selected.

Note: Selecting **mov** from the dropdown provides additional controls, allowing you to choose a **Codec**, **Codec Profile** or **Pixel Format**, and in some cases, **YCbCrMatrix**. The matrix control enables you to use the **Rec 601** and **Rec 709** encoding methods.

Similarly, selecting **.exr** provides an additional metadata dropdown allowing you to export or round-trip selected metadata along with your **.exr** output.

See the Nuke Reference Guide for more information.

4. **Create Directories** is enabled by default, which enables the corresponding control in the **.nk** script's Write node. This control allows Nuke to create the required directories when you render out a new version from the Write node.

   Disabling this control causes versioned renders to fail because the target directories don't exist. You can manually create the correct directories or enable **Create Directories** in the script's Write node if this happens.
5. Use the **Reformat** controls to determine how the Write node is set up in the Nuke script:
   - **None** - the clip or sequence resolution is used, no additional formatting is applied during export.

- **To Sequence Resolution** - exports at the resolution set in the timeline **Sequence** panel **Output Resolution** dropdown. This option also allows you to set the **Filter** used to resize the output.



Note: The filters available are generally listed in order of quality and processing time. **Cubic** can be faster to render, but **Lanczos4** may be produce better results. See Choosing a Filtering Algorithm for more information on filters in Nuke.

- **To Scale** - activates all the **Reformat** controls, except **Format**, allowing you to customize the export resolution.
- **Custom** - activates all the **Reformat** controls, except **Scale**, allowing you to customize the export resolution.

Reformat Control	To Sequence Resolution	To Scale	Custom
Format - sets the format to render out in Nuke, such as **1920x1080 HD 1080**.			●
Scale - sets the proportion by which to scale the output format.		●	
Resize - sets the method by which you want to preserve or override the original aspect ratio: • **width** - scales the original until its width matches the format's width. Height is then scaled in such a manner as to preserve the original aspect ratio. • **height** - scales the original until its height matches the format's height. Width is then scaled in such a manner as to preserve the original aspect ratio. • **fit** - scales the original until its smallest side matches the format's smallest side. The original's longer side is then scaled in such a manner as to preserve original aspect ratio. • **fill** - scales the original until its longest side matches the format's longest side. The input's shorter side is then scaled in such a manner as to preserve original aspect ratio. • **distort** - scales the original until all its sides match the lengths specified by the format. This option does not preserve the original aspect ratio, so distortion may occur.		●	●

Reformat Control	To Sequence Resolution	To Scale	Custom
Center - when enabled, transform image to the center of the output. When disabled, the image is aligned with the bottom-left corner of the output.		●	●
Filter - sets the filtering algorithm used to transform pixels. See Choosing a Filtering Algorithm for more information on filters in Nuke.	●	●	●

6. You can apply text burn-in to the media using a Nuke gizmo. Enable **Burn-in Gizmo** and click **Edit** to define the information applied. See Adding Burn-in Text to Exports for more information.
7. Proceed to Tracks, Range, and Handles Settings to select which items are processed during export.

Tracks, Range, and Handles Settings

The tracks, tags, and handles controls in the **Export dialog**'s Custom panel allow you to select the frame range or shots to export.

1. Set the **Version** number for the export, if applicable. Use the arrows to increment the version number and the +/- buttons to increase or decrease the padding. You can also type directly into the numeric field.



Note: See Using Versions for more information on how versioning works in Nuke Studio.

2. Select **Tracks for this export** by enabling or disabling the tracks in the list. Nuke Studio exports **all tracks** by default.
3. Enable or disable tags using the **Filter by tag** panel. Click the checkbox to cycle through the available tag states.
4. If you're exporting a sequence, set the **Range** controls as required:
   - Select **Whole Sequence** or **In/Out Points** to export only the selected frames.
   - Set how clip **Start Frames** are derived using the dropdown menu:
     - **Sequence** - use the sequence's start frame.
     - **Custom** - specify a start frame for all clips using the field to the right.
5. If you're exporting shots, set the **Handles** controls as required:

- **Clip Length** - exports the full clip length available, as if the clip was opened as a Viewer.
- **Cut Length** - exports only the cuts included on the timeline.



Note: Selecting **Cut Length** allows you to add handles to each clip, up to the maximum available source clip length.

- Check **Apply Retimes** to export any retimes present on the timeline.



Note: When **Apply Retimes** is disabled, which is the default state for **Create Comp**, any TimeWarp soft effects are not included in the resulting Nuke script. When the new shot is created through **Create Comp** or **Build Track from Export Tag**, TimeWarp soft effects are copied from the original shot to the new one.

- Set how clip **Start Frames** are derived using the dropdown menu:
  - **Source** - use the source clip's start frame.
  - **Custom** - specify a start frame for all clips using the field to the right.

6. Set how Nuke Studio should render your export using the **Render with** dropdown. Nuke Studio provides the following options:
   - **Frame Server** - uses multiple Nuke processes to speed up render times and shares a render queue with any Nuke Comp renders in the timeline, improving resource management.

     See Using the Frame Server on Render Machines for more information.
   - **Single Render Process** - uses a single Nuke process to render your export. Rendering QuickTimes falls back to this setting, but it's also used when a problem is detected with the Frame Server.
   - **Custom Render Process** - uses a custom render process. Nuke Studio requires a Python script to pass exports to your render farm of choice. Scripts must be located in specific directories, dependent on platform, as listed in Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts. If no scripts exist, the dropdown only contains the default render processes.
7. Click **Export**.

   The **Export Queue** window displays an estimate of how long each component of the export is expected to take.

   Once the export is complete, the file structure specified in the shot template is created. You can then import the Nuke clips on a separate track when they're ready.



Tip: Click the magnifying glass icon to reveal the file structure in a browser window.

When clips are exported from Nuke Studio, they are marked with a Nuke tag flagging which clips have an export history. Clips tagged in this way can be used to build VFX tracks quickly as described in Building VFX Tracks and Comp Clips.

Building VFX Tracks and Comp Clips

When the compositing work is complete, the clips are ready to re-ingest. The shot template defines where the Nuke files reside, so all you need to do is instruct Nuke Studio to build tracks from previous exports. See Building Tracks From Export Structure for more information.

Alternatively, if you have a history of rendered VFX clips, different versions and so on, you can also build tracks from export tags to select from a list of available clips. This method allows you to add **Comp Clips** to the timeline, which act as containers for Nuke **.nk** scripts, or placeholders for Nuke renders. See Building Comp Clips From Export Tags or Building Render Placeholders From Export Tags for more information.

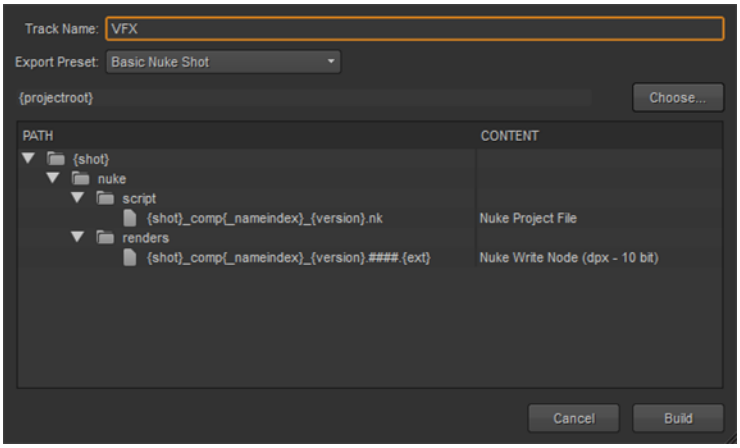
Building Tracks From Export Structure

1. Select the required clips on the timeline and right-click to display the context sensitive menu.



Tip: You may find it easier to select clips in the spreadsheet and then right-click on the timeline.

2. Click **Build Track** > **From Export Structure**.
   The **Build Track From Export Structure** dialog displays.



3. Enter a **Track Name** or use the default **VFX**.

4. Select an **Export Preset** using the dropdown menu. In this case, select the same preset used during the export.

> **Tip:** You may find that the **{_nameindex}** token cannot locate shots due to case differences in the file name on disk, depending on your operating system. If your export can't find the files required, you can use the **Rename Shots** function by right-clicking the shots in the timeline and navigating to **Editorial** > **Rename Shots** and selecting **Case Change** from the dropdown.

5. Enter the file path of the **Export Root** directory or click **Choose** and browse to the location.

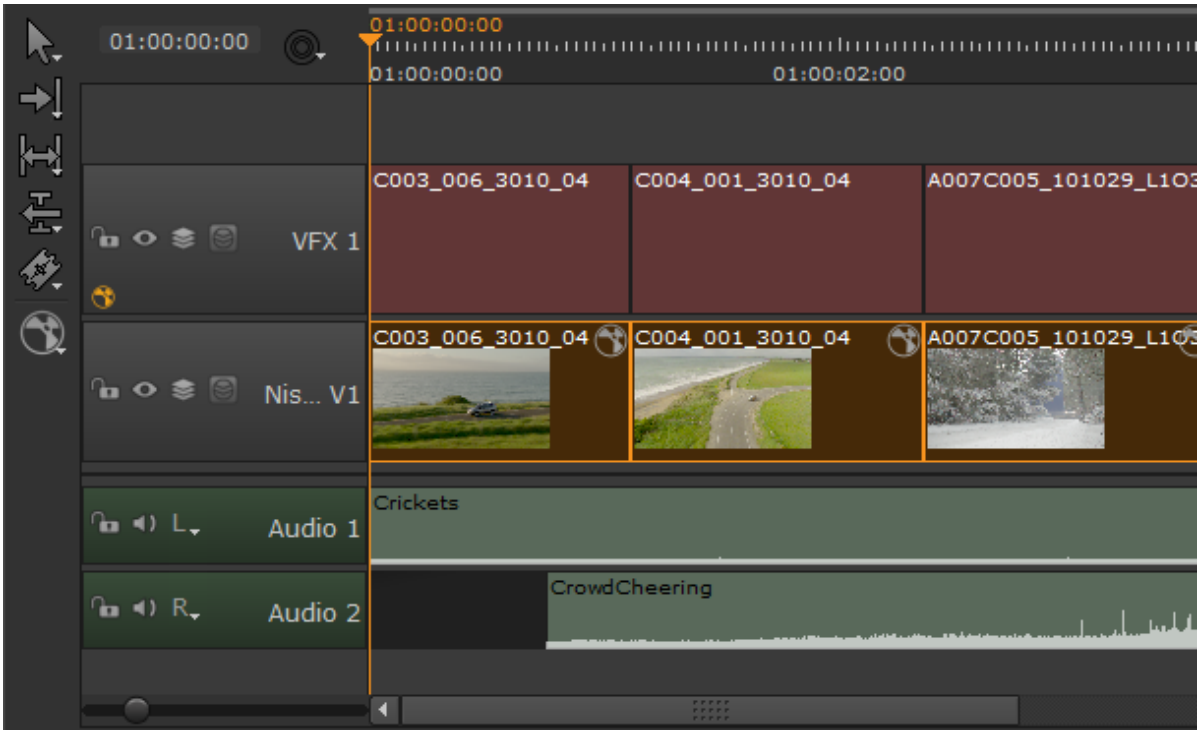
> **Note:** The root directory is the location entered in **Export To** when exporting the project.

6. Select the Content preset you intend to ingest from the shot template. In this case, the **Nuke Write Node**.
7. Click **Build** to create the VFX track.

> **Note:** Nuke Studio warns you if no selection is made in the **Content** column.

Nuke Studio automatically creates a new track containing the VFX clips, if they exist, or offline place holders if the clips are a work in progress.

If a shot already exists in any of the target tracks, a new track is created to hold the new shots.



The clips are automatically updated when work is complete as long as they are saved with the expected name and location, as specified in the shot template.

Building Comp Clips From Export Tags

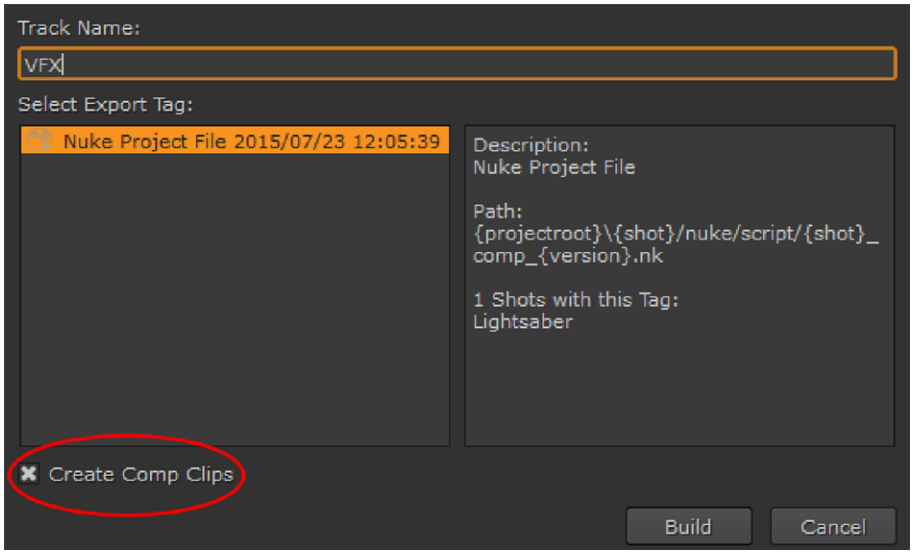
When you build a track from an export tag, Nuke Studio imports Comp Clips containing the Nuke script, by default. Comp Clips are shots that reference Nuke scripts, rather than being placeholders for offline clips as in the Building Render Placeholders From Export Tags workflow.

1. Select the required clips on the timeline and right-click to display the context sensitive menu.



Tip: You may find it easier to select clips in the spreadsheet and then right-click on the timeline.

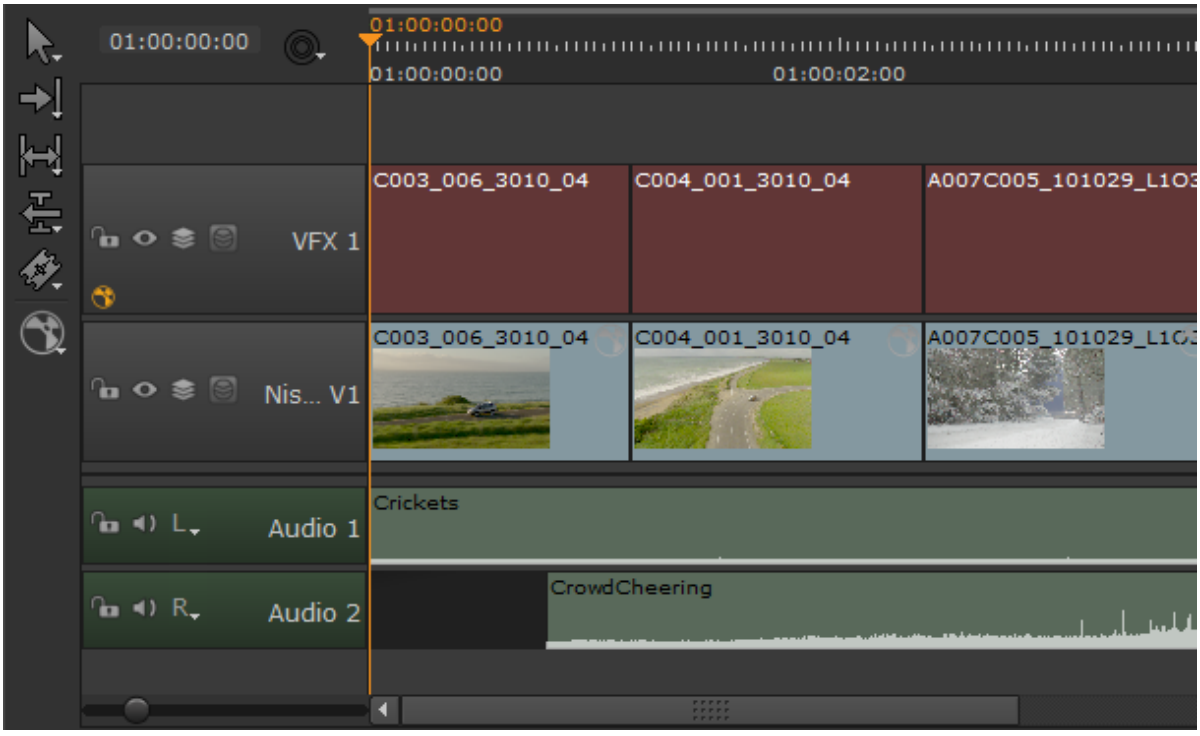
2. Click **Build Track** > **From Export Tag**.
   The **Build Track From Export Tag** dialog displays.
3. Enter a **Track Name** or use the default **VFX**.
4. Select the required export tag in the left-hand panel to display tag information in the right-hand panel.
   Nuke Studio imports the **.nk** Comp Clip by default.



If you just want to import the offline renders when they're finished, disable the **Create Comp Clips** checkbox. See Building Render Placeholders From Export Tags for more information.

5. Click **Build** to create the VFX track.

   Nuke Studio automatically creates a new track containing the Comp Clips. If a shot already exists in any of the target tracks, a new track is created to hold the new shots.



6. You can double-click Comp Clips to open them in Nuke Studio's Compositing environment to make edits as required.

Building Render Placeholders From Export Tags

When you build a track from an export tag, you can choose to import the renders from the **.nk** script, rather than Comp Clips which contain the Nuke script.

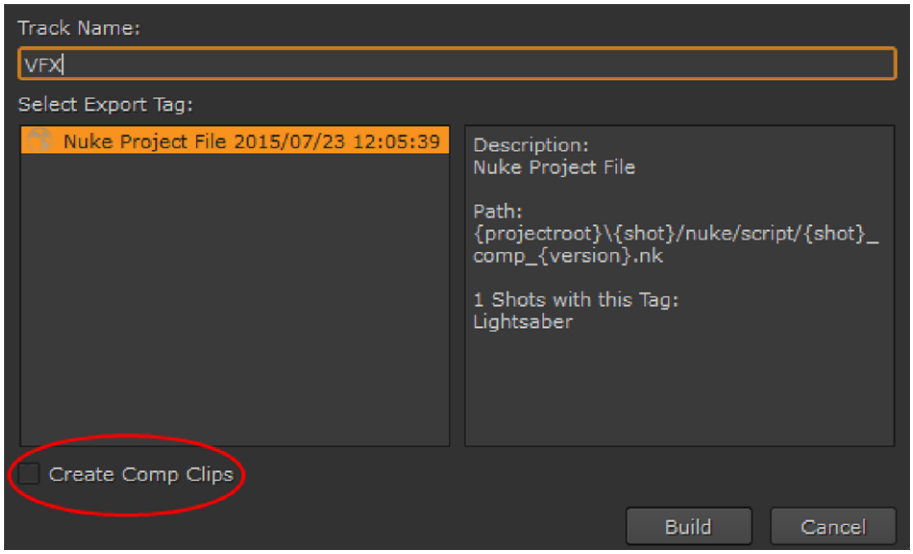
1. Select the required clips on the timeline and right-click to display the context sensitive menu.



Tip: You may find it easier to select clips in the spreadsheet and then right-click on the timeline.

2. Click **Build Track** > **From Export Tag**.

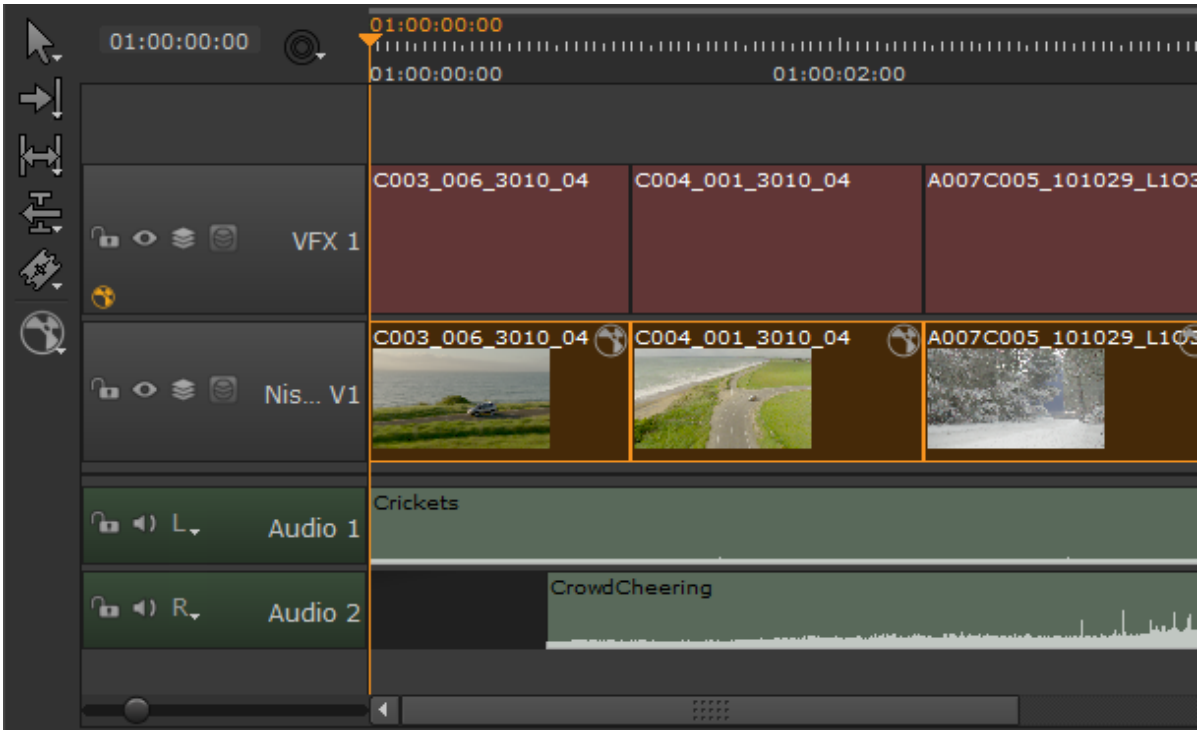
   The **Build Track From Export Tag** dialog displays.
3. Enter a **Track Name** or use the default **VFX**.
4. Select the required export tag in the left-hand panel to display tag information in the right-hand panel.
5. Disable the **Create Comp Clips** checkbox to import the offline renders when they're finished.



   If you want to import the **.nk** Comp Clips, enable the **Create Comp Clips** checkbox. See Building Comp Clips From Export Tags for more information.
6. Click **Build** to create the VFX track.

   Nuke Studio automatically creates a new track containing the VFX clips, if they exist, or offline place holders if the clips are a work in progress.

   If a shot already exists in any of the target tracks, a new track is created to hold the new shots.



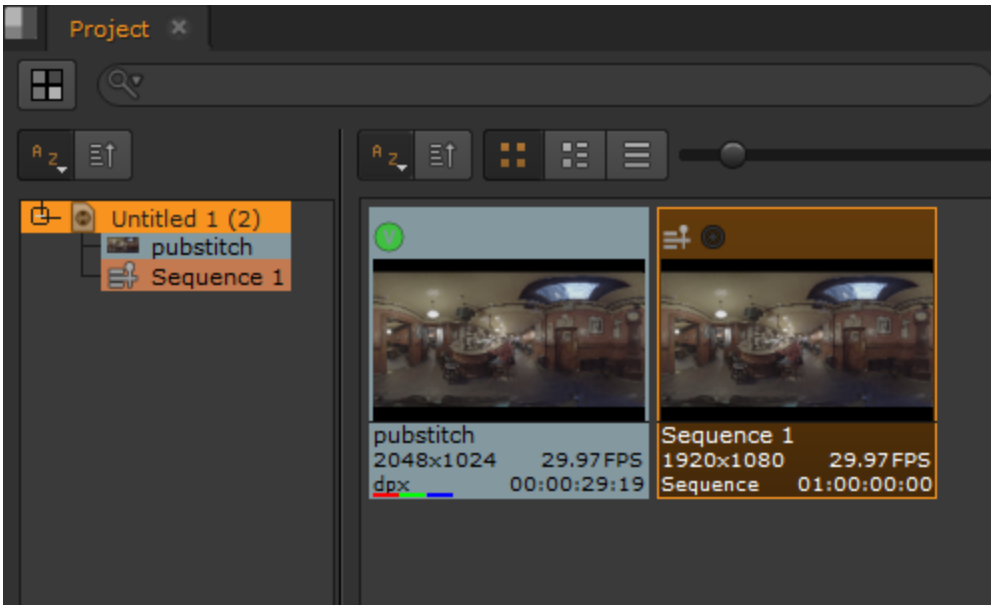
Exporting Multi-View Source Clips

Multi-view exports are similar to regular exports, but the **DPX Multi-View** or **Multi-View Nuke Shot** templates are used to created the required export tree using %v functionality, just like Nuke's Node Graph. The **DPX Multi-View** example preset is designed for exporting sequences and the **Multi-View Nuke Shot** preset is designed for shots.

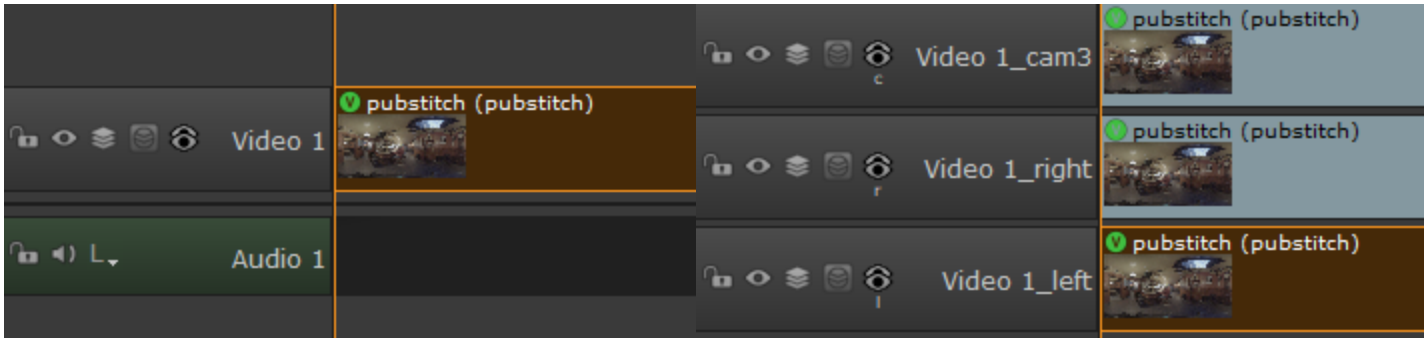
See Stereoscopic and Multi-View Projects for more information on working with multi-view footage.

1. Right-click on the sequence or shot that you want to export.

   Sequences are exported from the bin view.



Shots can be a multi-view or single views split into separate tracks. See Displaying Views in the Timeline for more information.



For separate tracks, you can right-click a single track to export all views or select all the per-track views and right-click.

2. Select **Export**.

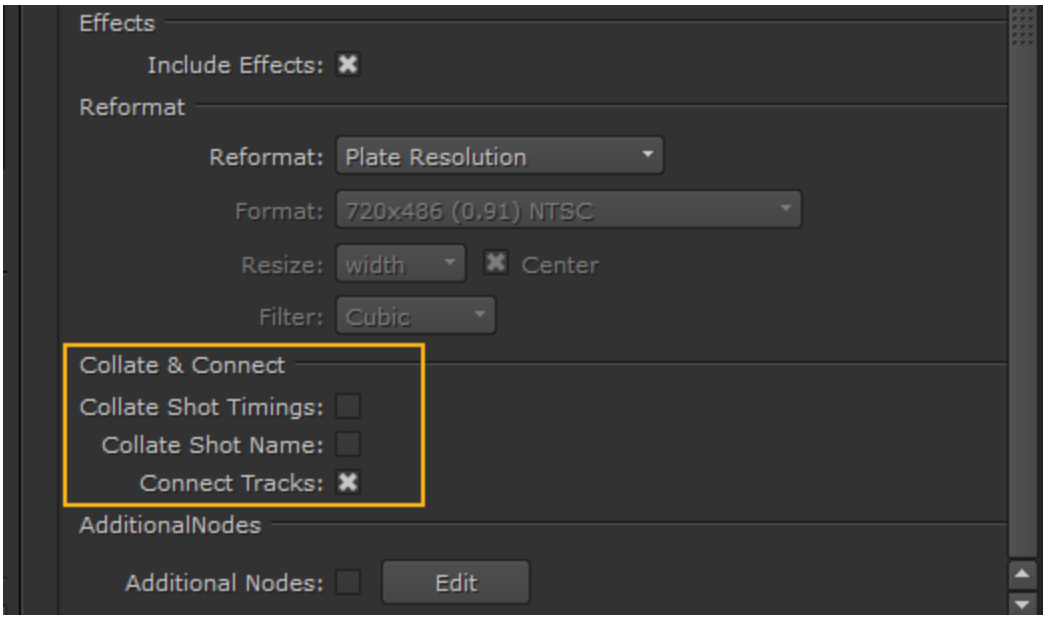
   The **Export** dialog is displayed. Go to the **Custom** tab.

3. Select **Process as Sequence** or **Process as Shots** from the **Export** dropdown.

   If you're exporting a sequence, select the **Log10 Cineon DPX Multi-View** example preset. If you're exporting shots, select the **Multi-View Nuke Shot (%v)** example preset.

4. Set the **Nuke Project File** and **Nuke Write Node** presets as described in Nuke Project File Settings and Nuke Write Node Settings.

   If you're using independent files per track, that is without importing multi-view files or using %V functionality, the separate tracks in the export script are not connected by default. If you want the tracks to be connected in the script, enable **Connect Tracks** in the **Nuke Project File** preset.



5. Set the track and handle preferences as described in Tracks, Range, and Handles Settings and then click **Export**.

   Once the export is complete, the file structure specified in the shot template is created. You can then import the Nuke clips on a separate track when they're ready.

Tip: Click the magnifying glass icon in the **Export Queue** to reveal the file structure in a browser window.

When clips are exported from Nuke Studio, they are marked with a Nuke tag flagging which clips have an export history. Clips tagged in this way can be used to build VFX tracks quickly as described in Building VFX Tracks and Comp Clips.

Transcoding

Transcoding in Nuke Studio uses a background render process to convert one file format to another. You can transcode sequences, timeline selections, and clips from the bin view.

Transcoding a Sequence

1. Select a sequence in the bin view and navigate to **File** > **Export...**

   The **Export** dialog displays. Open the **Custom** tab.
2. Select **Process as Sequence** and the preset you intend to use, or use the default **dpx** preset.

3. Enter the **Export To** directory or click **Choose...** and browse to the location.
4. Click the **Content** column in the shot template to display the transcode options.
5. Set the following controls common to all file types:
   - **Channels** - set the channels to export using the dropdown. If you want to export a non-standard channel, type the name of the channel into the field manually.
   - **Output Transform** - use the dropdown to set the colorspace to render, such as **linear**, **REDLog**, or **raw**.
6. Select the **File Type** to render using the dropdown and complete the relevant fields, dependent on the **File Type** selected.

   For a full description of the file type options available, see the Write node topic in the Nuke Reference Guide.



Note: Selecting **mov** from the dropdown provides additional controls, allowing you to choose a **Codec**, **Codec Profile** or **Pixel Format**, and in some cases, **YCbCrMatrix**. The matrix control enables you to use the **Rec 601** and **Rec 709** encoding methods.

Similarly, selecting **exr** provides an additional metadata dropdown allowing you to export or round-trip selected metadata along with your **.exr** output.

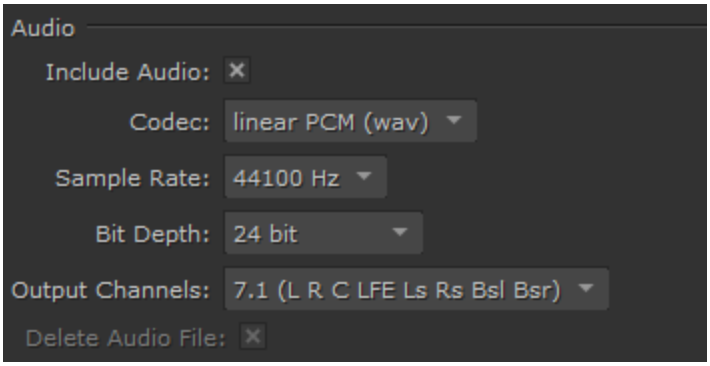
See the Nuke Reference Guide for more information.

7. Use the **Reformat** controls to determine how the Write node is set up in the Nuke script:
   - **None** - the clip or sequence resolution is used, no additional formatting is applied during export.
   - **To Scale** - activates all the **Reformat** controls, except **Format**, allowing you to customize the export resolution.
   - **Custom** - activates all the **Reformat** controls, except **Scale**, allowing you to customize the export resolution.

Reformat Control	To Scale	Custom
Format - sets the format to render out in Nuke, such as **1920x1080 HD 1080**.		●
Scale - sets the proportion by which to scale the output format.	●	
Resize - sets the method by which you want to preserve or override the original aspect ratio:	●	●

Reformat Control	To Scale	Custom
• **width** - scales the original until its width matches the format's width. Height is then scaled in such a manner as to preserve the original aspect ratio. • **height** - scales the original until its height matches the format's height. Width is then scaled in such a manner as to preserve the original aspect ratio. • **fit** - scales the original until its smallest side matches the format's smallest side. The original's longer side is then scaled in such a manner as to preserve original aspect ratio. • **fill** - scales the original until its longest side matches the format's longest side. The input's shorter side is then scaled in such a manner as to preserve original aspect ratio. • **distort** - scales the original until all its sides match the lengths specified by the format. This option does not preserve the original aspect ratio, so distortion may occur.		
Center - when enabled, transform image to the center of the output. When disabled, the image is aligned with the bottom-left corner of the output.	●	●
Filter - sets the filtering algorithm used to transform pixels. See Choosing a Filtering Algorithm for more information on filters in Nuke.	●	●

8. Complete the general controls common to all file types:
   - Select the **Retime Method** to apply, if applicable.
   - Soft Effects added to shots in your export are included in the resulting Nuke script by default. If you don't need the soft effects, disable **Include Effects** to omit them from the script. See Adding Effects on the Timeline for more information.
   - Any annotations added to shots in your export can be burnt-in by checking **Include Annotations**. See Annotations for more information.
   - **Include Audio** - when enabled, any audio tracks are exported alongside the video.



- **Delete Audio File** - when disabled, delete the **.wav** file used to create the export. This control only applies to formats that support audio, such as **.mov** files.

9. Enable the **Burn-in Gizmo** to burn-in text using a Nuke gizmo. Click **Edit** to define the information applied during burn-in. See Adding Burn-in Text to Exports for more information.
10. Specify any **Additional Nodes** required during export by clicking **Edit**. See Adding Additional Nodes During Export for more information.
11. Check **Keep Nuke Script** if you require the **.nk** files after the transcode operation.

Note: The following controls may improve render times for certain exports:

- For **.dpx** exports, you can enable **Read All Lines**, which can speed up transcode times for I/O heavy scripts.
- For machines with multiple CPU sockets using a **Single Render Process**, you may find that limiting the render with **Use Single Socket** may improve render times.

Tracks and Range Settings

1. Set the **Version** number for the export, if applicable. Use the arrows to increment the version number and the +/- buttons to increase or decrease the padding. You can also type directly into the numeric field.

Note: See Using Versions for more information on how versioning works in Nuke Studio.

2. Select **Tracks for this export** by enabling or disabling the tracks in the list. Nuke Studio exports **all tracks** by default.
3. If you set in and out point on the sequence, enable In/Out Points to export only the selected frames.
4. Set how clip **Start Frames** are derived using the dropdown menu:
   - **Sequence** - use the sequence's start frame.
   - **Custom** - specify a start frame for all clips using the field to the right.
5. Set how Nuke Studio should render your export using the **Render with** dropdown. The following options are available:
   - **Frame Server** - uses multiple Nuke processes to speed up render times and shares a render queue with any Nuke Comp renders in the timeline, improving resource management.

     See Using the Frame Server on Render Machines for more information.
   - **Single Render Process** - uses a single Nuke process to render your export. Rendering QuickTimes falls back to this setting, but it's also used when a problem is detected with the Frame Server.

- **Custom Render Process** - uses a custom render process. Nuke Studio requires a Python script to pass exports to your render farm of choice. Scripts must be located in specific directories, dependent on platform, as listed in Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts. If no scripts exist, the dropdown only contains the default render processes.

6. Click **Export**.

   The **Export Queue** window displays an estimate of how long each component of the export is expected to take.

   Once the export is complete, the file structure specified in the shot template is created containing the transcoded files.



Tip: Click the magnifying glass icon to reveal the exported file in a browser window.

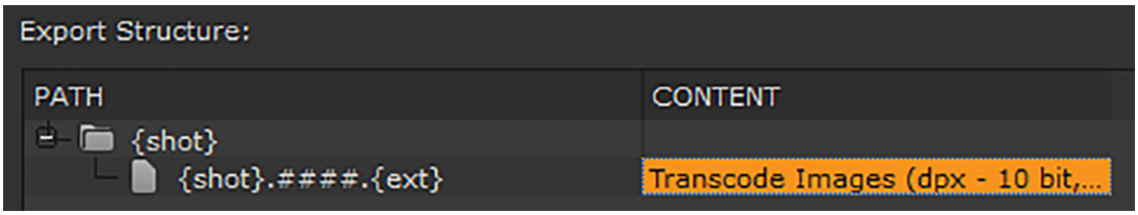
Transcoding a Sequence as Shots

1. Select the required sequence in the bin view and navigate to **File** > **Export...**

   The **Export** dialog displays. Open the **Custom** tab.

2. Select **Process as Shots** and use the default, **Transcode Shots DPX**, or build a shot template using the **Path** and **Contents** fields and the folder and +/- buttons.

   The default:



   Creates a folder for each shot, containing a clip with the {shot} name and the required file padding (####) and extension {ext}.

3. Enter the **Export To** directory or click **Choose...** and browse to the location.

   The **Export To** directory is the starting point from which the shot template builds your shot hierarchy.

4. In the **Content** tab, complete the **File Type** specific and general controls common to all file types as described in Transcoding a Sequence.

5. Click the **Tracks and Handles** tab, select the **Tracks For This Export** by enabling or disabling the tracks in the list. Nuke Studio exports **all tracks** by default.

6. Enable or disable tags using the **Filter by Tag** panel. Click the checkbox to cycle through the available tag states.

7. Set the **Range** and **Handles**, as required:

- **Clip Length** - exports the full clip length available, as if the clip was opened as a Viewer.
- **Cut Length** - exports only the cuts included on the timeline.



Note: Selecting **Cut Length** allows you to add handles to each clip, up to the maximum available source clip length.

8. Check **Apply Retimes** to export any retimes present on the timeline.



Note: When **Apply Retimes** is disabled, which is the default state for **Create Comp**, any TimeWarp soft effects are not included in the resulting Nuke script. When the new shot is created through **Create Comp** or **Build Track from Export Tag**, TimeWarp soft effects are copied from the original shot to the new one.

9. Set how clip **Start Frames** are derived using the dropdown menu:
   - **Source** - use the source clip's start frame.
   - **Custom** - specify a start frame for all clips using the field to the right.
10. Set the **Version** number for the export, if applicable. Use the arrows to increment the version number and the +/- buttons to increase or decrease the padding. You can also type directly into the numeric field.



Note: SeeUsing Versions for more information on how versioning works in Nuke Studio.

11. Set how Nuke Studio should render your export using the **Render with** dropdown. Nuke Studio provides the following options:
    - **Frame Server** - uses multiple Nuke processes to speed up render times and shares a render queue with any Nuke Comp renders in the timeline, improving resource management.

      See Using the Frame Server on Render Machines for more information.
    - **Single Render Process** - uses a single Nuke process to render your export. Rendering QuickTimes falls back to this setting, but it's also used when a problem is detected with the Frame Server.
    - **Custom Render Process** - uses a custom render process. Nuke Studio requires a Python script to pass exports to your render farm of choice. Scripts must be located in specific directories, dependent on platform, as listed in Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts. If no scripts exist, the dropdown only contains the default render processes.
12. Click **Export**.

    The **Export Queue** window displays an estimate of how long each component of the export is expected to take.

Once the export is complete, the file structure specified in the shot template is created. You can then import the Nuke clips on a separate track when they're ready.

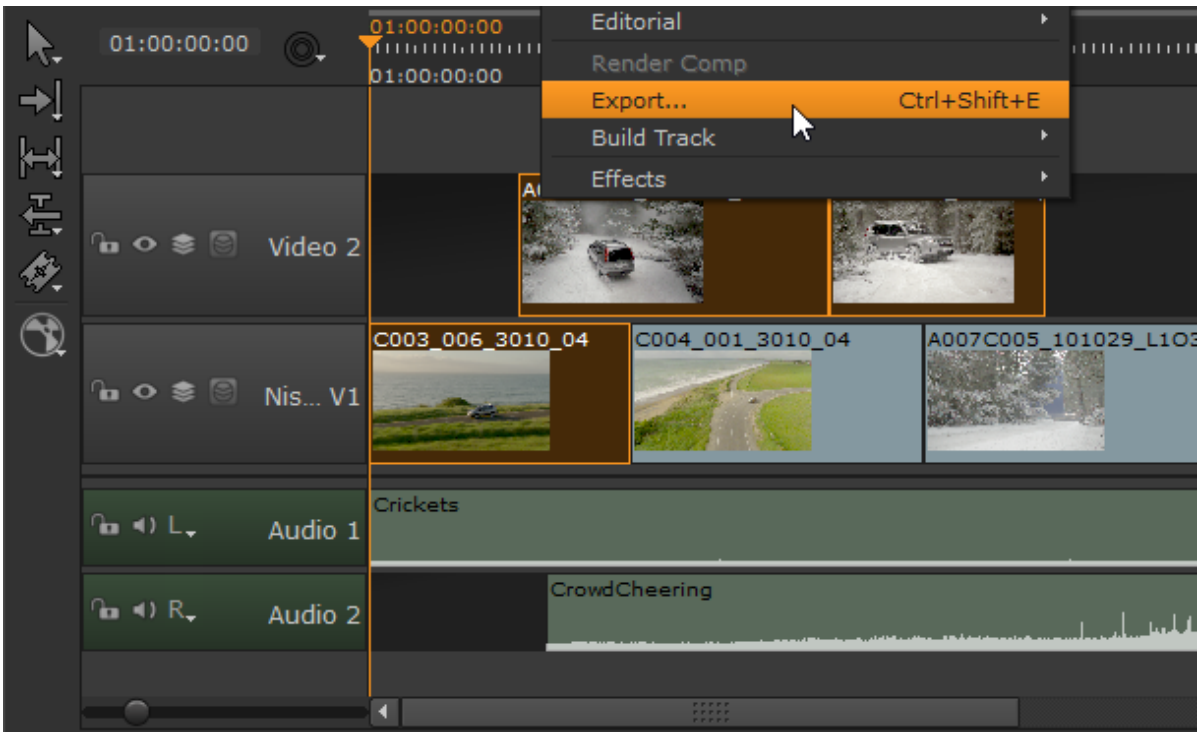
Tip: Click the magnifying glass icon to reveal the file structure in a browser window.

Transcoding Timeline Selections

Transcoding an entire timeline can be time consuming, or even unnecessary, if all you're looking for is a new version of a selection of shots.

To transcode a selection of clips from a timeline:

1. Select the required shots on the timeline.
2. Right-click a highlighted item and select **Export...**



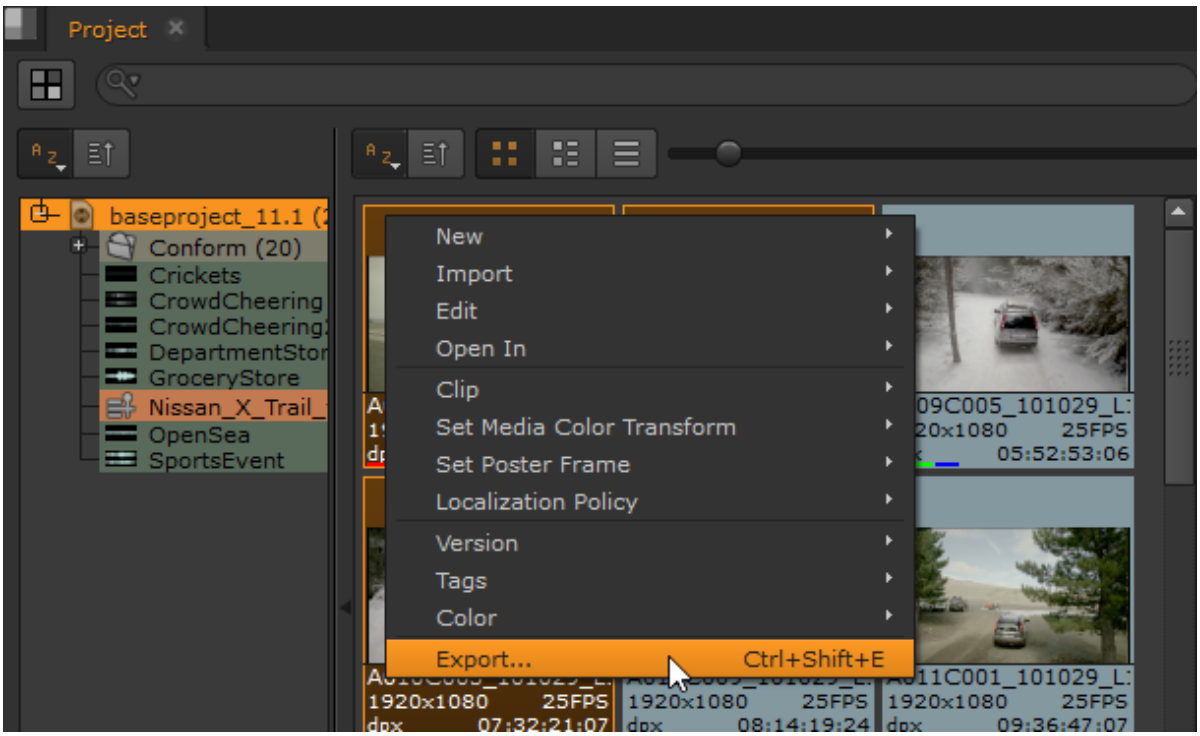
The **Export** dialog displays.

3. Refer to Transcoding a Sequence to complete the export.

Transcoding from the Bin View

To transcode directly from the bin view:

1. Select the bin(s) to export from the bin view.
2. Right-click a highlighted bin and select **Export...**



   The **Export** dialog displays. Open the **Custom** tab.
3. Select **Process as Clips** and modify the shot template, if required.
4. Follow the steps under Transcoding a Sequence to complete the export.

Ad Hoc Exports

This section covers exports that you might not perform on a per project basis, such the EDL, OTIO, or XML Exporters and Copy Exporter. Exporters are available for sequences, shots, and clips as described in the following table.

Exporter	Source		
	Sequences	Shots	Source Clips
EDL Exporter	●		
OTIO Exporter	●		
XML Exporter	●		
Audio Exporter	●	●	●
Copy Exporter		●	●
SymLink Exporter		●	●

Exporting Sequences as EDL, OTIO, and XML

Nuke Studio supports export to EDL, OTIO, and XML using very similar methods, the main difference being that EDL doesn't support multiple video tracks in a single file whereas OTIO and XML does.

Note: Nuke Studio can read AAF files, but not write them out.

To export to EDL, OTIO, or XML:

1. Select a sequence in the bin view and navigate to **File** > **Export...**
   The **Export** dialog displays. Open the **Custom** tab.
2. Select **Process as Sequence** from the **Export** dropdown.
3. Select the **CMX 3600 EDL**, **OTIO**, or **Final Cut Pro XML** preset, or duplicate one and create your own preset.



Note: EDLs only support one video track per file. If you have more than one track, include the **{track}** token in the shot template to write out an EDL for each track preset.

For example, **{filename}_{track}.{ext}** might produce a separate EDL for each track on your timeline called **myTimeline_Video1.edl**, **myTimeline_Video2.edl**, and so on.

4. Enter the **Export To** directory or click **Choose...** and browse to the location.

   The **Export To** directory is the starting point from which the shot template builds your shot hierarchy.
5. If you're exporting to EDL, set the additional EDL Exporter controls in the **Content** tab, if required:
   - **Reel Name** - define the reel name written into the EDL, independent of the clip's reel name.

     Enter text or standard shot-level tokens in this field. See Using the Shot Template for more information.

     If the field is left blank, the reel name from the clip is used or the name of the shot, if no reel name exists.
   - **Truncate Reel Name** - restricts the **Reel** name to eight characters.
   - **Use Absolute Path** - adds the full file path for each clip to the EDL comments field.
   - **From Clip Name** - define the text appended to "from" comment fields in EDLs, such as ***FROM CLIP NAME**. Text and/or standard shot-level tokens are valid in this field: {shot}, {clip}, {track}, {sequence}, {event}, {fps}, and the default {filename}.

     OR

     If you're exporting to XML, you can enable **Include Markers** to convert any frame tags present in the sequence to markers in Final Cut Pro or Premiere. See Tagging Using the Viewer for more information on adding tags to frames.
6. Set the **Version token number** for the export, if applicable. Use the arrows to increment the version number and the +/- buttons to increase or decrease the padding. You can also type directly into the numeric field.



Note: See Using Versions for more information on how versioning works.

7. Select the **Tracks For This Export** by enabling or disabling the tracks in the list. Nuke Studio exports **all tracks** by default.
8. If you set in and out point on the sequence, enable **In/Out Points** to export only the selected frames.
9. Set how clip **Start Frames** are derived using the dropdown menu:
   - **Sequence** - use the sequence's start frame.
   - **Custom** - specify a start frame for all clips using the field to the right.
10. Set how Nuke Studio should render your export using the **Render with** dropdown. Nuke Studio provides the following options:

- **Frame Server** - uses multiple Nuke processes to speed up render times and shares a render queue with any Nuke Comp renders in the timeline, improving resource management.

  See Using the Frame Server on Render Machines for more information.
- **Single Render Process** - uses a single Nuke process to render your export. Rendering QuickTimes falls back to this setting, but it's also used when a problem is detected with the Frame Server.
- **Custom Render Process** - uses a custom render process. Nuke Studio requires a Python script to pass exports to your render farm of choice. Scripts must be located in specific directories, dependent on platform, as listed in Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts. If no scripts exist, the dropdown only contains the default render processes.

11. Click **Export**.

    The **Export Queue** window displays an estimate of how long each component of the export is expected to take.

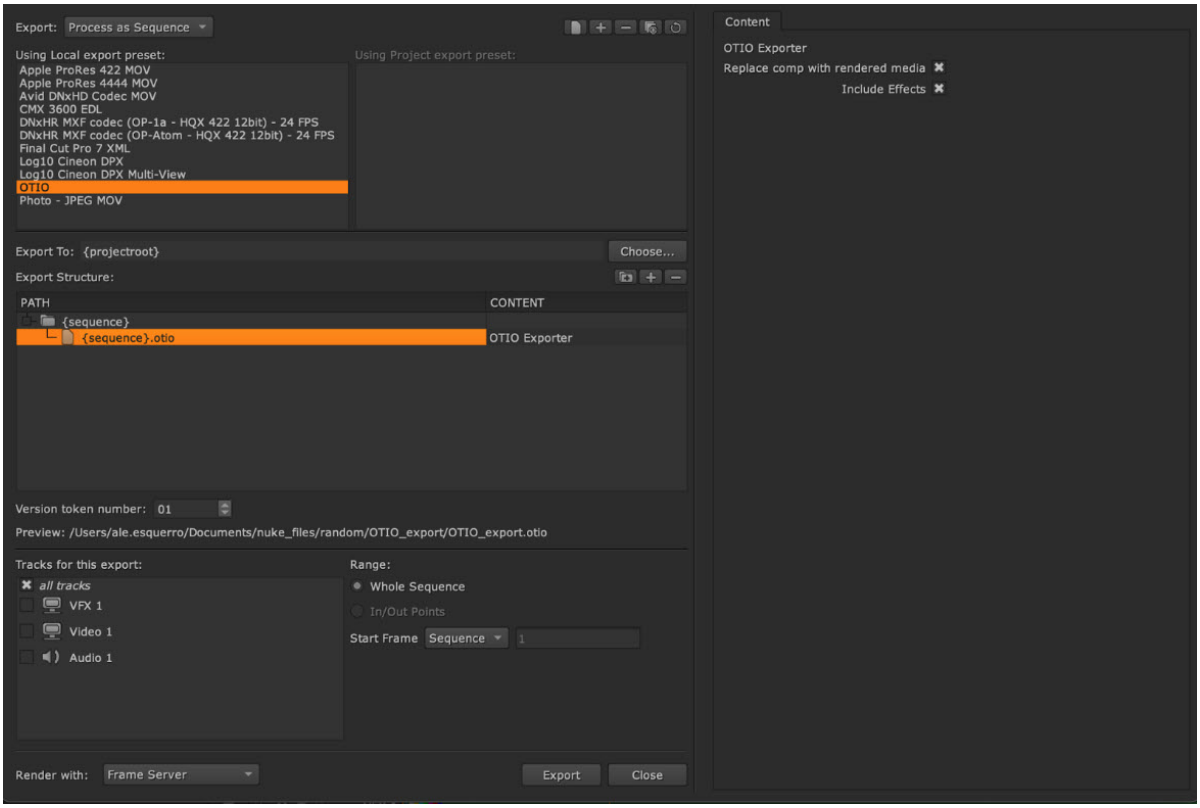
    Once the export is complete, the file structure specified in the shot template is created.



Tip: Click the magnifying glass icon to reveal the exported file in a browser window.

Custom OTIO Export Options

Use the following custom options when you export an OTIO file from Nuke Studio.



Replace comp with rendered media

- On by default. This option is for sending the OTIO file to a generic OTIO reader/editor. The setting replaces the .nk path with a literal path to the file location of the rendered media.
- Turn this option off when sending an OTIO to another Nuke Studio. This setting keeps the Create Comp workflows .nk path to automatically pick up comps and versions of renders.

Include Effects

- On by default. Includes all Nuke Studio effects - soft effects, track blend, tags, locked/disabled clip states - in custom OTIO metadata. Another Nuke Studio/Hiero reads and makes the effects editable, whereas a 3rd party OTIO reader/editor ignore s Nuke Studio custom effects.
- Turn off this option to make a simplified standard OTIO without any custom metadata. It won't include any soft effects, tags etc. This option allows for compatibility in cases where your exported OTIO with effects won't open in a 3rd party app.

Using the Audio Exporter

The Audio Exporter allows you write audio to separate **.wav** files. You can extract audio from whole sequences, shots, and source clips.

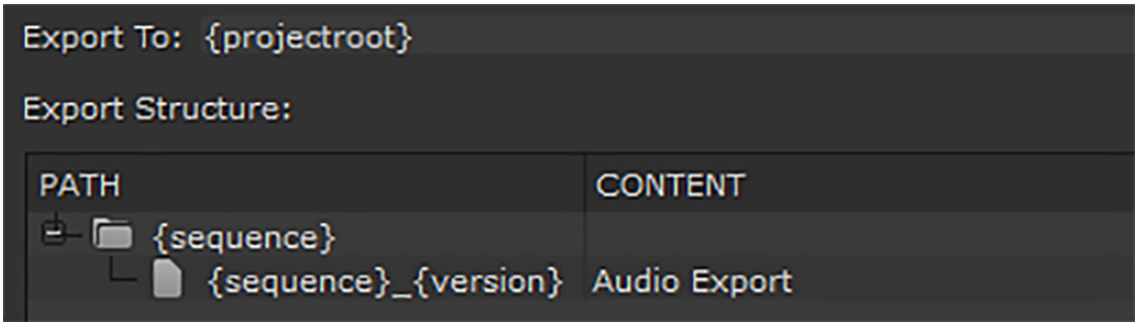
Exporting Audio from Sequences

1. Select a sequence in the bin view and navigate to **File** > **Export...**

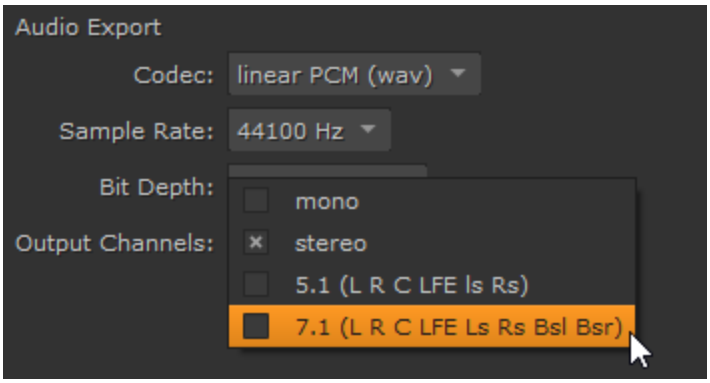
   The **Export** dialog displays. Open the **Custom** tab.
2. Select **Process as Sequence** from the **Export** dropdown and select a preset, duplicate it, and give it a name.
3. Enter the **Export To** directory or click **Choose...** and browse to the location.

   The **Export To** directory is the starting point from which the shot template builds your shot hierarchy.
4. Build a custom shot template using **Path** tokens and the folder and +/- buttons and then click the **Content** field and select **Audio Export** from the list.

   An example shot template is shown below:



5. Click the **Audio Export** field and select the required export options, including the **Output Channels** for multi-channel audio.



6. Set the **Version** number for the export, if applicable. Use the arrows to increment the version number and the +/- buttons to increase or decrease the padding. You can also type directly into the numeric field.

Note: See Using Versions for more information on how versioning works in Nuke Studio.

7. Select **Tracks for this export** by enabling or disabling the tracks in the list. Nuke Studio exports **all tracks** by default.
8. Set the **Range** controls as required:
   - Select **Whole Sequence** or **In/Out Points** to export only the selected frames.
   - Set how clip **Start Frames** are derived using the dropdown menu:
     - **Sequence** - use the sequence's start frame.
     - **Custom** - specify a start frame for all clips using the field to the right.
9. Set how Nuke Studio should render your export using the **Render with** dropdown. Nuke Studio provides the following options:
   - **Frame Server** - uses multiple Nuke processes to speed up render times and shares a render queue with any Nuke Comp renders in the timeline, improving resource management.

     See Using the Frame Server on Render Machines for more information.
   - **Single Render Process** - uses a single Nuke process to render your export. Rendering QuickTimes falls back to this setting, but it's also used when a problem is detected with the Frame Server.
   - **Custom Render Process** - uses a custom render process. Nuke Studio requires a Python script to pass exports to your render farm of choice. Scripts must be located in specific directories, dependent on platform, as listed in Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts. If no scripts exist, the dropdown only contains the default render processes.
10. Click **Export**.

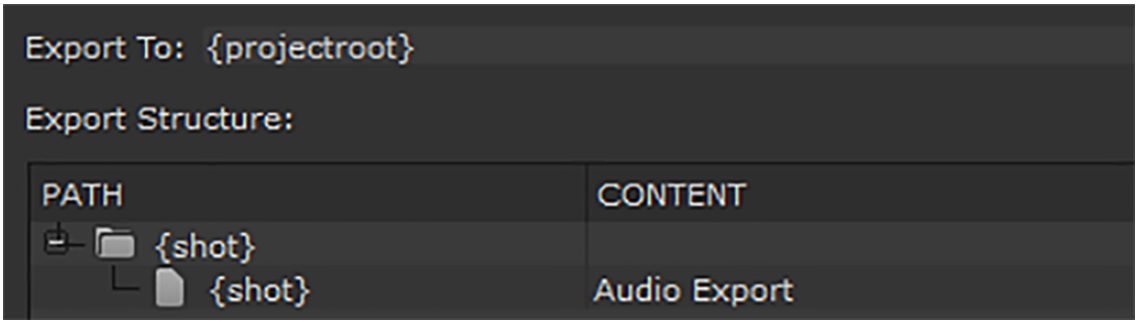
    The **Export Queue** window displays an estimate of how long each component of the export is expected to take.

    Once the export is complete, the file structure specified in the shot template is created.

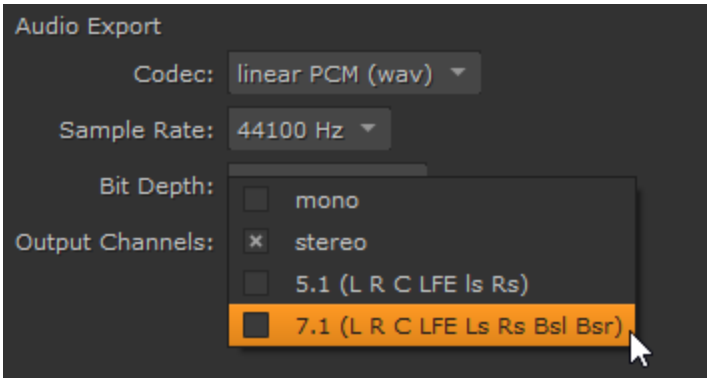
Tip: Click the magnifying glass icon to reveal the exported file in a browser window.

Exporting Audio from Shots

1. Select the required shots in the timeline and navigate to **File** > **Export...**
   The **Export** dialog displays.
2. Select **Process as Shots** from the Export list and select a preset, duplicate it, and give it a name.
3. Enter the **Export To** directory or click **Choose...** and browse to the location.
   The **Export To** directory is the starting point from which the shot template builds your shot hierarchy.
4. Build a custom shot template using **Path** tokens, the **Contents** field, and the folder and +/- buttons.
   An example shot template is shown below:



5. Click the **Audio Export** field and select the required export options, including the **Output Channels** for multi-channel audio.



6. Set the **Version** number for the export, if applicable. Use the arrows to increment the version number and the +/- buttons to increase or decrease the padding. You can also type directly into the numeric field.

Note: SeeUsing Versions for more information on how versioning works in Nuke Studio.

7. Select **Tracks for this export** by enabling or disabling the tracks in the list. Nuke Studio exports **all tracks** by default.
8. Enable or disable tags using the **Filter by tag** panel. Click the checkbox to cycle through the available tag states.
9. If you're exporting a sequence, set the **Range** controls as required:
   - Select **Whole Sequence** or **In/Out Points** to export only the selected frames.
   - Set how clip **Start Frames** are derived using the dropdown menu:
     - **Sequence** - use the sequence's start frame.
     - **Custom** - specify a start frame for all clips using the field to the right.
10. If you're exporting shots, set the **Handles** controls as required:
    - **Clip Length** - exports the full clip length available, as if the clip was opened as a Viewer.
    - **Cut Length** - exports only the cuts included on the timeline.



Note: Selecting **Cut Length** allows you to add handles to each clip, up to the maximum available source clip length.

11. Set how clip **Start Frames** are derived using the dropdown menu:
    - **Source** - use the source clip's start frame.
    - **Custom** - specify a start frame for all clips using the field to the right.
12. Set how Nuke Studio should render your export using the **Render with** dropdown. Nuke Studio provides the following options:
    - **Frame Server** - uses multiple Nuke processes to speed up render times and shares a render queue with any Nuke Comp renders in the timeline, improving resource management.

      See Using the Frame Server on Render Machines for more information.
    - **Single Render Process** - uses a single Nuke process to render your export. Rendering QuickTimes falls back to this setting, but it's also used when a problem is detected with the Frame Server.
    - **Custom Render Process** - uses a custom render process. Nuke Studio requires a Python script to pass exports to your render farm of choice. Scripts must be located in specific directories, dependent on platform, as listed in Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts. If no scripts exist, the dropdown only contains the default render processes.
13. Click **Export**.

    The **Export Queue** window displays an estimate of how long each component of the export is expected to take.

    Once the export is complete, the file structure specified in the shot template is created.

Tip: Click the magnifying glass icon to reveal the exported file in a browser window.

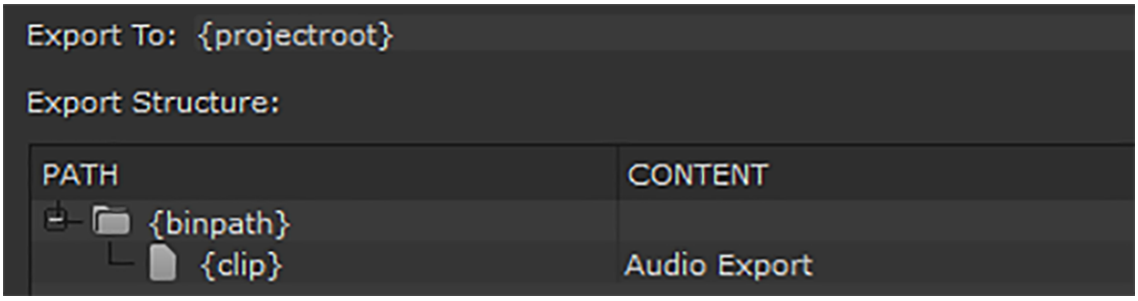
Exporting Audio from Source Clips

1. Select the required source clips and navigate to **File** > **Export...**

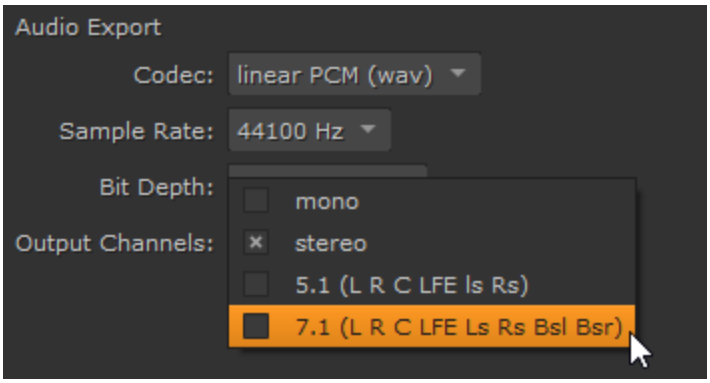
   The **Export** dialog displays.
2. **Process as Clips** is selected automatically from the Export list, so select a preset, duplicate it, and give it a name.
3. Enter the **Export To** directory or click **Choose...** and browse to the location.

   The **Export To** directory is the starting point from which the shot template builds your shot hierarchy.
4. Build a custom shot template using **Path** tokens, the **Contents** field, and the folder and +/- buttons.

   An example shot template is shown below:



5. Click the **Audio Export** field and select the required export options, including the **Output Channels** for multi-channel audio.



6. Set the **Version** number for the export, if applicable. Use the arrows to increment the version number and the +/- buttons to increase or decrease the padding. You can also type directly into the numeric field.

> **Note:** See Using Versions for more information on how versioning works in Nuke Studio.

7. Set how clip **Start Frames** are derived using the dropdown menu:
   - **Source** - use the source clip's start frame.

- **Custom** - specify a start frame for all clips using the field to the right.

8. Set how Nuke Studio should render your export using the **Render with** dropdown. Nuke Studio provides the following options:
   - **Frame Server** - uses multiple Nuke processes to speed up render times and shares a render queue with any Nuke Comp renders in the timeline, improving resource management.

     See Using the Frame Server on Render Machines for more information.
   - **Single Render Process** - uses a single Nuke process to render your export. Rendering QuickTimes falls back to this setting, but it's also used when a problem is detected with the Frame Server.
   - **Custom Render Process** - uses a custom render process. Nuke Studio requires a Python script to pass exports to your render farm of choice. Scripts must be located in specific directories, dependent on platform, as listed in Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts. If no scripts exist, the dropdown only contains the default render processes.
9. Click **Export**.

   The **Export Queue** window displays an estimate of how long each component of the export is expected to take.

   Once the export is complete, the file structure specified in the shot template is created.



Tip: Click the magnifying glass icon to reveal the exported file in a browser window.

Using the Copy Exporter

Copying media from various locations is very time consuming and can waste disk space. The Copy Exporter allows you to consolidate sequences containing only your project media in a named file structure using the shot template.

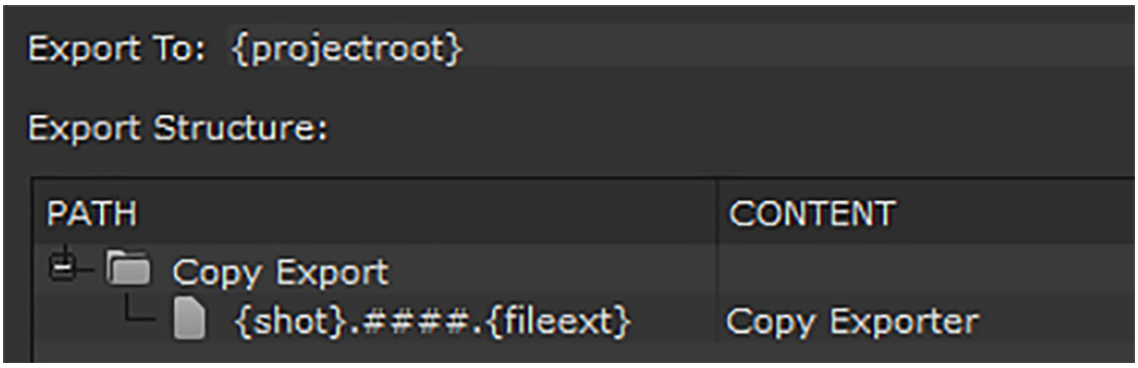
To copy media to a named location:

1. Select a sequence in the bin view and navigate to **File** > **Export...**

   The **Export** dialog displays. Open the **Custom** tab.
2. Select **Process as Shots** from the **Export** dropdown.
3. Select the **Transcode Shots DPX** preset, duplicate it, and give it a name.
4. Enter the **Export To** directory or click **Choose...** and browse to the location.

   The **Export To** directory is the starting point from which the shot template builds your shot hierarchy.
5. Build a custom shot template using **Path** tokens, the **Contents** field, and the folder and +/- buttons.

   An example shot template is shown below:



6. Set the **Version** number for the export, if applicable. Use the arrows to increment the version number and the +/- buttons to increase or decrease the padding. You can also type directly into the numeric field.

Note: See Using Versions for more information on how versioning works in Nuke Studio.

7. Set how clip **Start Frames** are derived using the dropdown menu:
   - **Source** - use the source start frame.
   - **Custom** - specify a start frame for all clips using the field to the right.
8. Set how Nuke Studio should render your export using the **Render with** dropdown. Nuke Studio provides the following options:
   - **Frame Server** - uses multiple Nuke processes to speed up render times and shares a render queue with any Nuke Comp renders in the timeline, improving resource management.

     See Using the Frame Server on Render Machines for more information.
   - **Single Render Process** - uses a single Nuke process to render your export. Rendering QuickTimes falls back to this setting, but it's also used when a problem is detected with the Frame Server.
   - **Custom Render Process** - uses a custom render process. Nuke Studio requires a Python script to pass exports to your render farm of choice. Scripts must be located in specific directories, dependent on platform, as listed in Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts. If no scripts exist, the dropdown only contains the default render processes.
9. Click **Export**.

   The **Export Queue** window displays an estimate of how long each component of the export is expected to take.

   Once the export is complete, the file structure specified in the shot template is created.

Tip: Click the magnifying glass icon to reveal the exported file in a browser window.

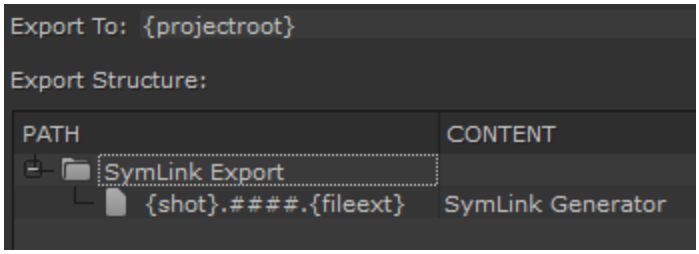
Using the SymLink Generator

The SymLink Generator allows you to create symbolic links to your project media in a named file structure using the shot template.

Note: Windows - you may also need administrator privileges to create symbolic links on Windows.

To create symbolic links to a named location:

1. Select a sequence in the bin view and navigate to **File** > **Export...**
   The **Export** dialog displays. Open the **Custom** tab.
2. Select **Process as Shots** from the **Export** dropdown.
3. Select the **Transcode Shots DPX** preset, duplicate it, and give it a name.
4. Enter the **Export To** directory or click **Choose...** and browse to the location.
   The **Export To** directory is the starting point from which the shot template builds your shot hierarchy.
5. Build a custom shot template using **Path** tokens, the **Contents** field, and the folder and +/- buttons.
   An example shot template is shown below:



6. Set the **Version** number for the export, if applicable. Use the arrows to increment the version number and the +/- buttons to increase or decrease the padding. You can also type directly into the numeric field.

Note: See Using Versions for more information on how versioning works in Nuke Studio.

7. Set how clip **Start Frames** are derived using the dropdown menu:
   - **Source** - use the source start frame.
   - **Custom** - specify a start frame for all clips using the field to the right.
8. Set how Nuke Studio should render your export using the **Render with** dropdown. Nuke Studio provides the following options:

- **Frame Server** - uses multiple Nuke processes to speed up render times and shares a render queue with any Nuke Comp renders in the timeline, improving resource management.

  See Using the Frame Server on Render Machines for more information.
- **Single Render Process** - uses a single Nuke process to render your export. Rendering QuickTimes falls back to this setting, but it's also used when a problem is detected with the Frame Server.
- **Custom Render Process** - uses a custom render process. Nuke Studio requires a Python script to pass exports to your render farm of choice. Scripts must be located in specific directories, dependent on platform, as listed in Loading Gizmos, NDK Plug-ins, and Python and Tcl Scripts. If no scripts exist, the dropdown only contains the default render processes.

9. Click **Export**.

   The **Export Queue** window displays an estimate of how long each component of the export is expected to take.

   Once the export is complete, the file structure specified in the shot template is created.



Tip: Click the magnifying glass icon to reveal the exported file in a browser window.

Contact Sheet in Nuke Studio

This section covers how Contact Sheet works in Nuke Studio, Hiero, and HieroPlayer.

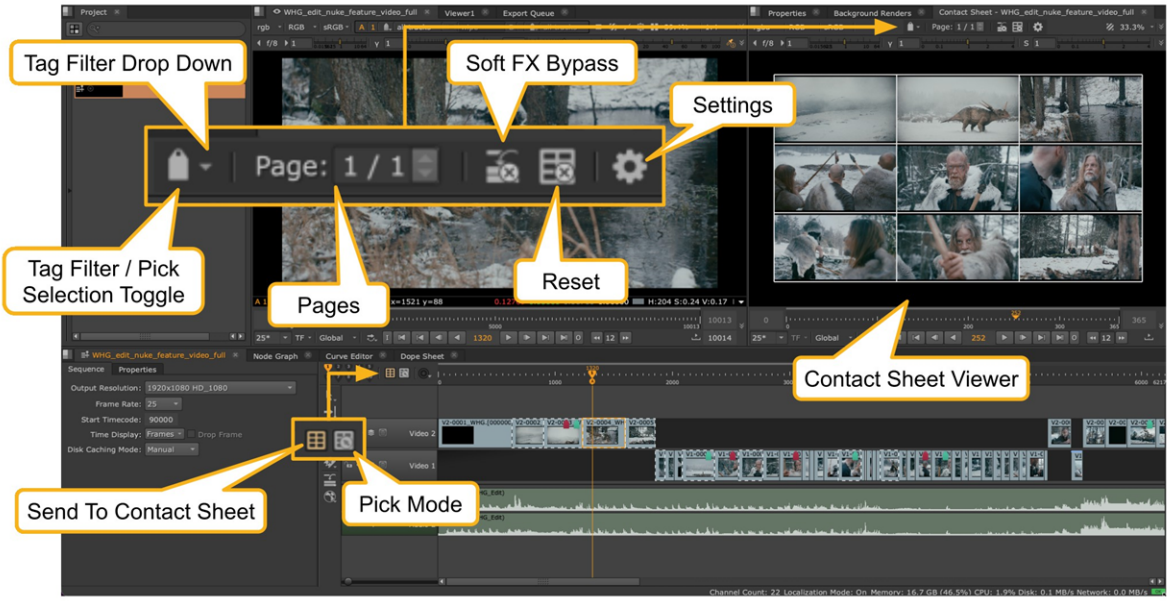
Contact Sheet allows you to quickly load multiple shots from a sequence, for artistic, comparison, and review purposes.

Contact Sheet is designed to offer a fluid and dynamic user experience, with the ability to add and filter shots to the **Contact Sheet Viewer**.

Contact Sheet Overview

In the example below we have a sequence open and some shots have been selected to be displayed in the Contact Sheet. The Timeline viewer and the **Contact Sheet Viewer** can be seen side by side.

The Contact Sheet buttons and menu items are shown below:



- **Send To Contact Sheet** - The Send To Contact Sheet button is used to make the current sequence eligible to populate the Contact Sheet.

  The button has two states:

  - **Inactive (gray)** - The sequence is not populating the Contact Sheet
  - **Active (highlighted)** - The sequence is being used by the Contact Sheet

> **Note:** Once the button is active, you cannot disable it unless another sequence is loaded and sent to the Contact Sheet or the Contact Sheet Viewer is closed. More on this in the Using Contact Sheet section.

- **Pick Mode** - The Pick Mode button lets you manually choose shots in the sequence to be displayed in the Contact Sheer Viewer. The Pick Mode button will only be available once the Send To Contact Sheet button is active for the sequence.

  The button has two states:

  - **Inactive (gray)** - Normal hard selection and editorial tools are active in the Edit Tool bar. Any shots selected will obey the usual hard selection methodology and will not be displayed in the Contact Sheet Viewer
  - **Active (highlighted)** - This state is used to explicitly select shots in the sequence that you want displayed in the Contact Sheet viewer. Any selections made in Pick Mode are only for Contact Sheet use and will not interfere or affect any current hard selections made in the sequence before entering Pick Mode

- **Tag Filter/Pick Toggle & Tag Filter Drop Down**  - This button enables you to switch between displaying either manually selected shots from the sequence in the Contact Sheet Viewer, or toggling to display shots that are using tag filtering. This functionality is provided by the left hand Tag icon. The right hand side drop down icon, when pressed, will show a list of all tags currently being used in the sequence and check boxes to allow for boolean operations.

  The left hand Tag button has two states:

  - **Inactive (gray)** - This state will display any shots that have been selected via the Pick Mode button
  - **Active (highlighted)** - This state will toggle the Contact Sheet Viewer to display shots that match the criteria set in the Tag Filter Drop Down menu



Note: Tag Filtering is available in Nuke Studio and Hiero, but not in HieroPlayer.

- **Pages** 

- This button works in conjunction with the **Max Row Count** and **Max Column Count** options found in the Contact Sheet Settings. This is set by default to a 5x5 grid able to accommodate 25 shots per page. If a selection of more than 25 shots are selected to be displayed in the Contact Sheet in this particular example, then any shot count above 25 will be automatically added to a second page until this is populated. The page count will then increment to display how many pages the selected are contained in and up and down arrows to be able to navigate between the pages. You can also use the scroll functionality found in other number fields in Nuke, to also navigate the additional pages.
- **Soft FX Bypass**  - The Soft FX Bypass button will bypass only any track based Soft effects to enable the user to quickly disable these effects without having to modify the sequence. This will only affect the Contact Sheet Viewer. Viewing the sequence in the Timeline Viewer will still show any track based Soft effects.

  The button has two states:

  - **Inactive (gray)** - This state will display any track based Soft effects in the sequence being used to populate the Contact Sheet
  - **Active (highlighted)** - This state will bypass any track based Soft effects in the sequence being used to populate the Contact Sheet

Note: Soft Effects are available in Nuke Studio and Hiero, but not in HieroPlayer.

- **Reset**  - The Reset button will quickly clear the Contact Sheet of all clips and reset to its initial state. Pushing this button will clear all active clips in the Pick Mode and as well as all tag filter states. The function is not undoable.

- **Contact Sheet Settings**  - The Contact Sheet Settings drop down displays options that affect the way the Contact Sheet functions.

  Currently this includes:

  - **Contact Sheet Output Resolution** - This gives the user the ability to customize the Contact Sheet resolution. Modifying this can affect the quality of the images currently being displayed. You can also set custom resolutions that can be useful when used in conjunction with the **Max Row** and **Column** count options.
  - **Max Row & Column Count** - These two values allow you to customize the layout of the Contact Sheet. Together the two values will effectively set a maximum amount of shots that can be displayed per page, i.e the default of 5 rows and 5 columns allows for a Contact Sheet of 25 shots per page.

Tip: By adjusting the Max Row & Columns Count in combination with Contact Sheet Output Resolution, you can optimize the amount of shots and their scale displayed and create bespoke layouts for review.

Using Contact Sheet

To learn how to use the Contact Sheet, see the following pages:

- Using Contact Sheet: Manual Pick Mode
- Using Contact Sheet: Playback
- Using Contact Sheet: Tag Filtering
- Using Contact Sheet: Workflows & Soft Effects
- Using Contact Sheet: Preferences

Additional Considerations

Performance

The performance of the Contact Sheet feature in Studio 17.0v3 will depend on the specification of the host computer that Studio is being run on, storage and network performance as well as the format of media sources being used.

Currently, there is no specific Contact Sheet Disk Caching so real time playback will depend on the amount of shots being displayed in conjunction with the available RAM cache allocated in preferences.

Integration with Existing Features

Contact Sheet is compatible with existing timeline Soft Effects, although multiple stacked Soft Effects and BlinkScript effects may impact responsiveness.

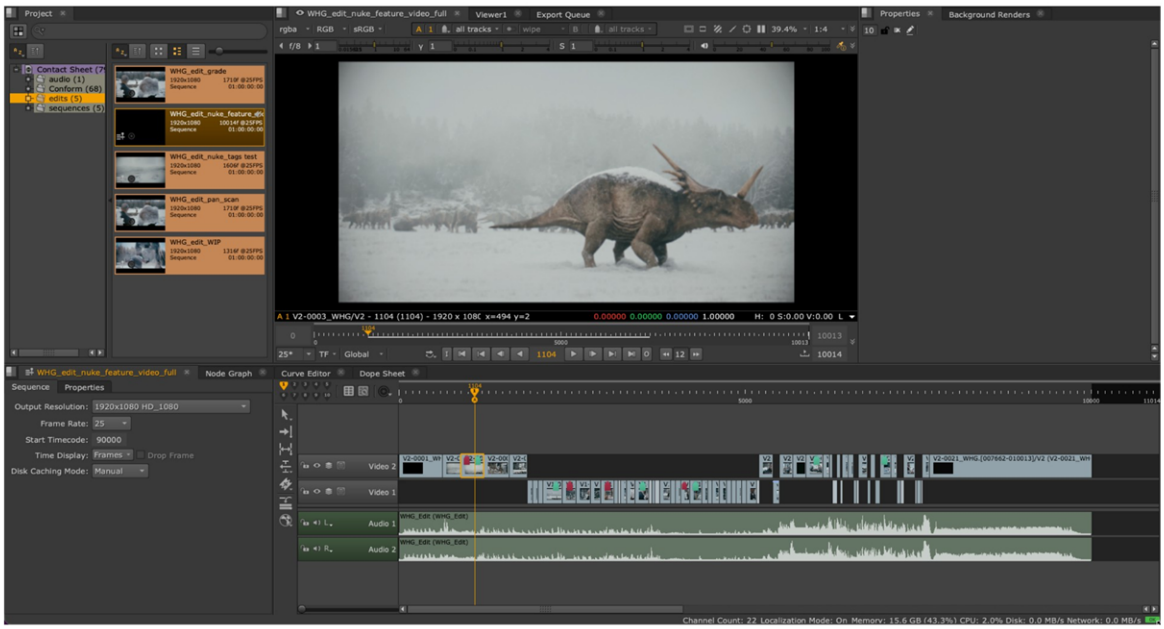
Annotations are not currently displayed and it is not possible to annotate directly onto the Contact Sheet Viewer.

A/B compare modes are not supported in the Contact Sheet Viewer.

Using Contact Sheet: Manual Pick Mode

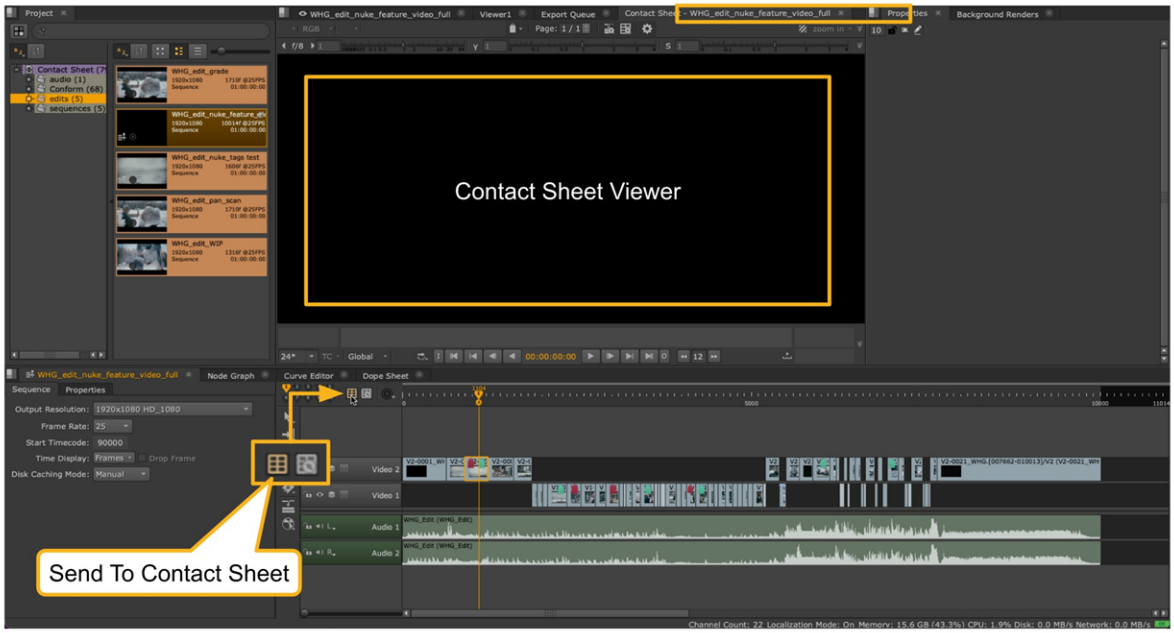
To get started with Contact Sheet, you first have to open a sequence in the Timeline View. Only shots that are residing in a sequence can be used to populate the Contact Sheet view.

Here is an example of a project to use with Contact Sheet.

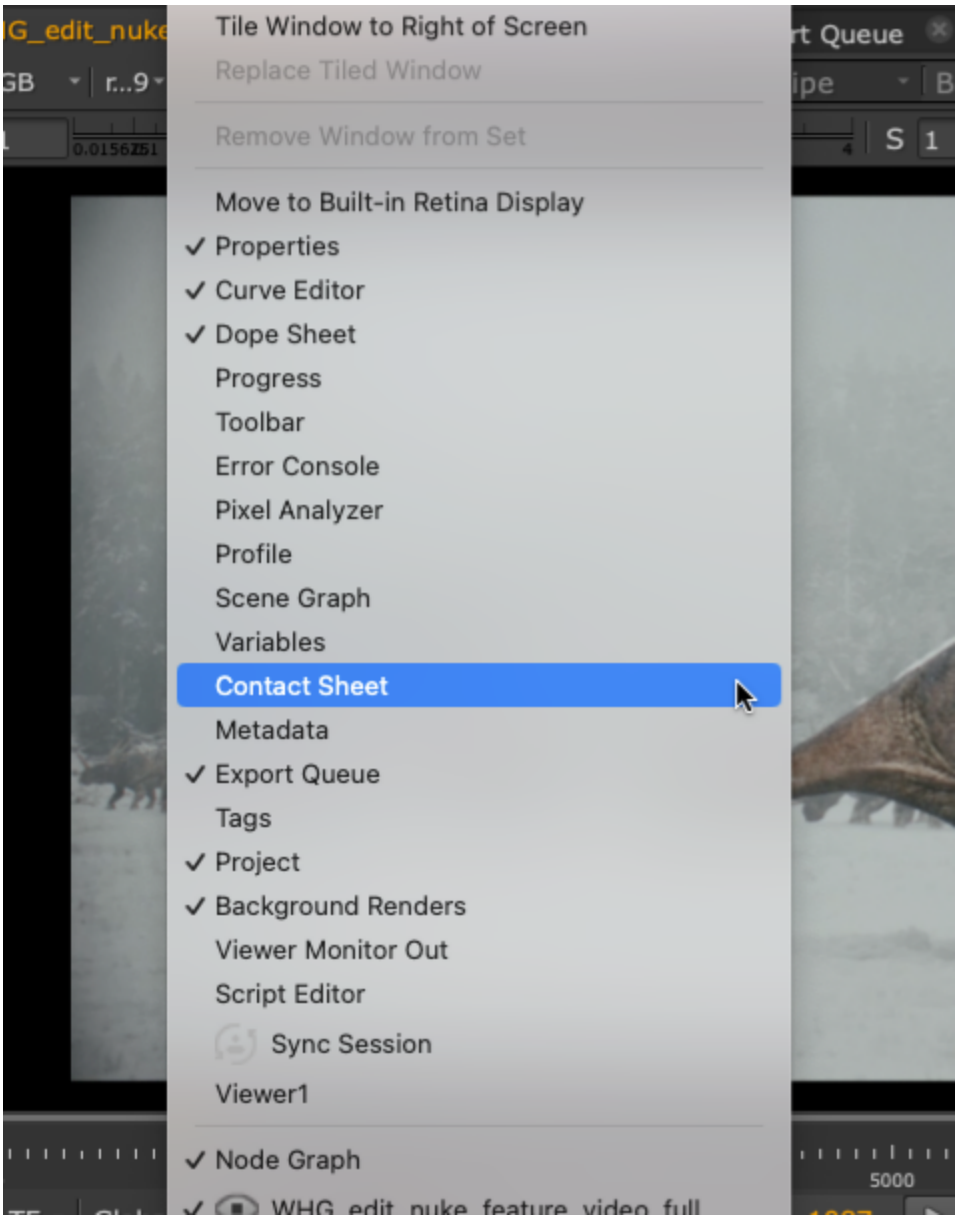


1. The first step is to create a new Contact Sheet Viewer which can be populated with the shots you'd like to review.

To do this, click the **Send To Contact Sheet** button in the Timeline View. Doing this will create a new Contact Sheet Viewer named after the sequence.



You can also use the Window drop down menu to create a new Contact Sheet Viewer as well.

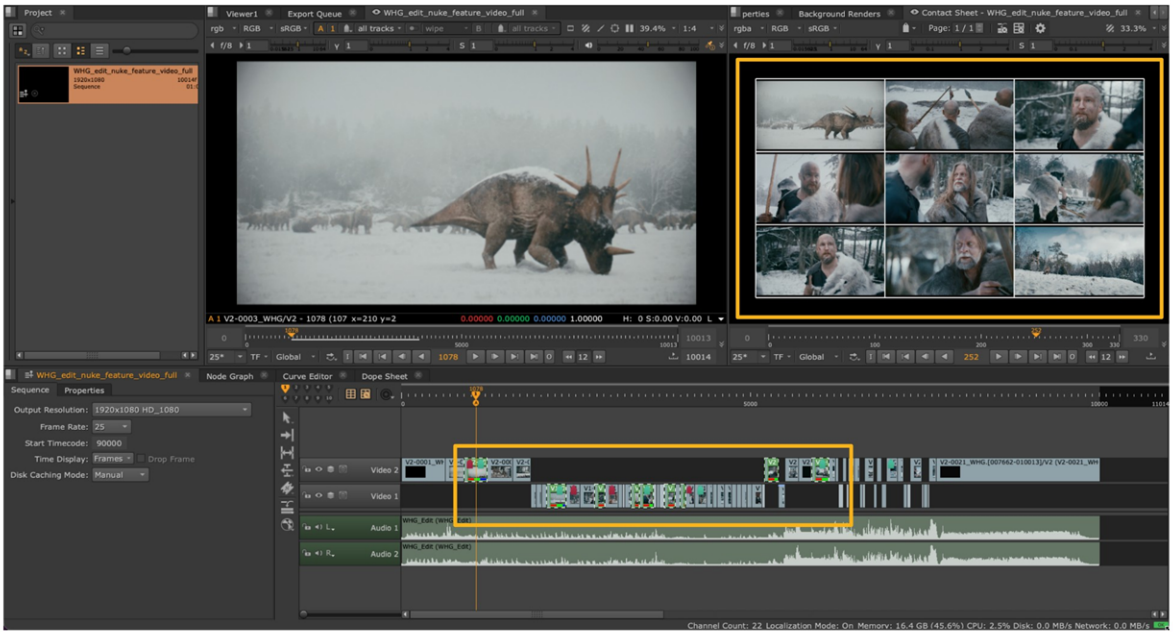


If you have a second monitor you can drag the Contact Sheet Viewer to be displayed on the second screen or move it to a new pane alongside the standard Viewer.

2. You can now start to populate the Contact Sheet Viewer with clips from the sequence. You can do this by directly selecting the clips from the sequence. First go into the **Pick Mode** by using the button next to the Send To Contact Sheet button.

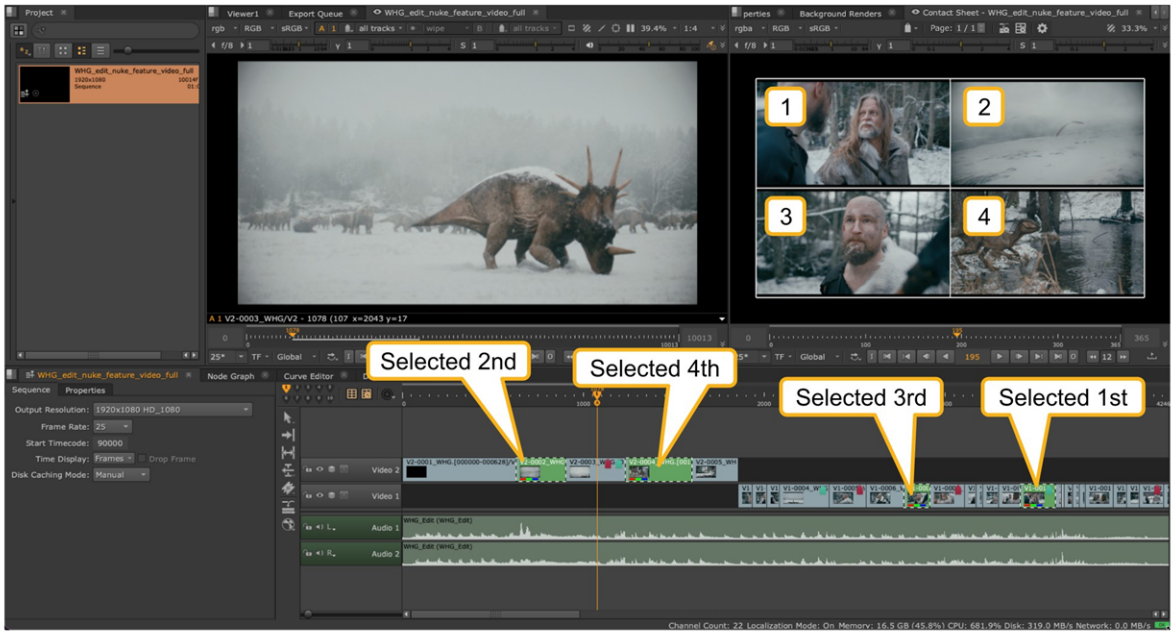


3. Once you are in Pick Mode you can select shots directly from the sequence view and they will dynamically populate the Contact Sheet Viewer.

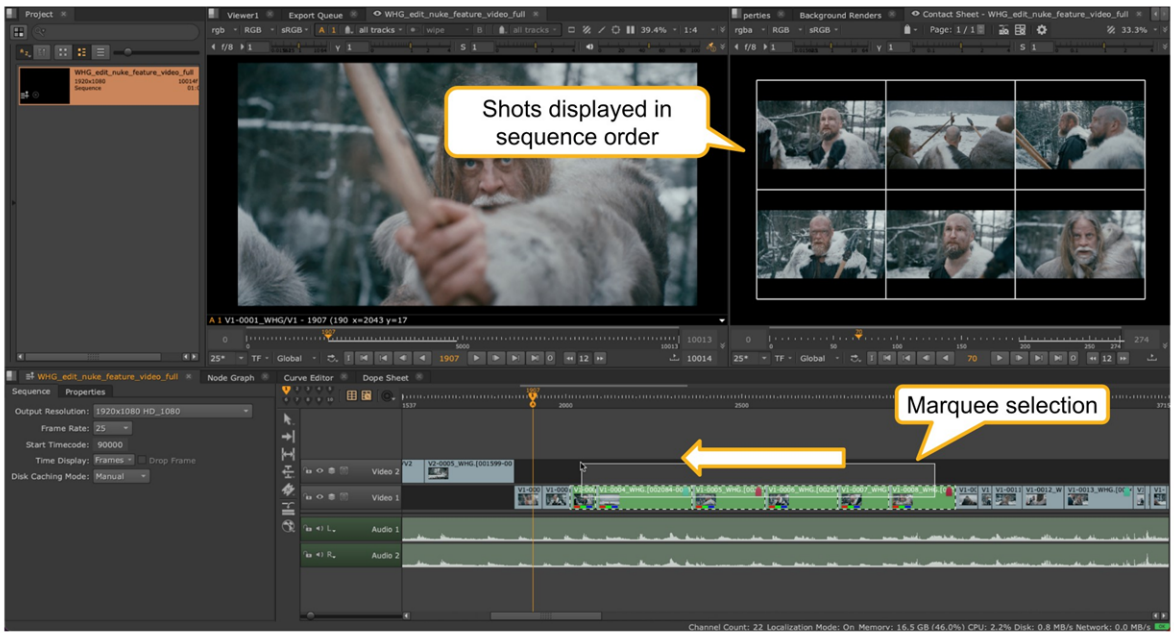


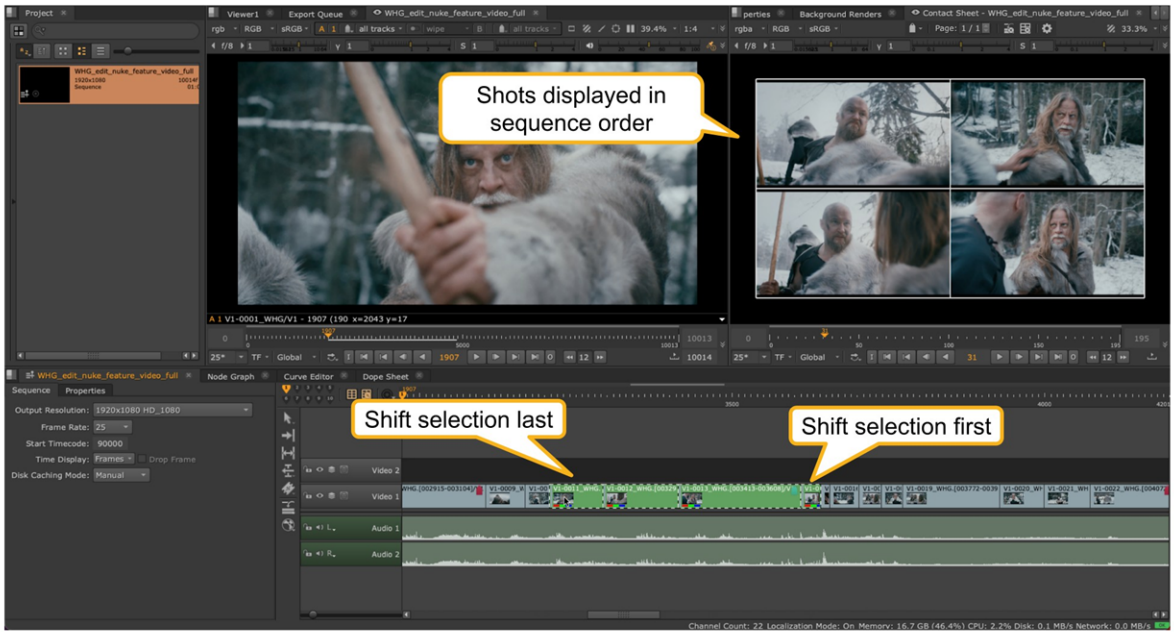
You can use all the standard selection methods like **Ctrl+Left Click** to select non contiguous clips, **Left Click+Drag Marquee** to select a range or **Left Click** on a clip and **Shift+Left Click** on a different clip to select all clips in between.

The Pick Mode works in the same fashion as Hard Selections; so if, for instance, you want to deselect all your currently selected clips being displayed in Contact Sheet simply make sure that you are in Pick Mode and tap down in an empty area in the sequence and your selection will be cleared.
It's worth noting that when you are selecting non-contiguous clips using the **Ctrl+Left Click** method, the order you select the clips defines the order that they appear in the Contact Sheet Viewer, as shown in the example below.

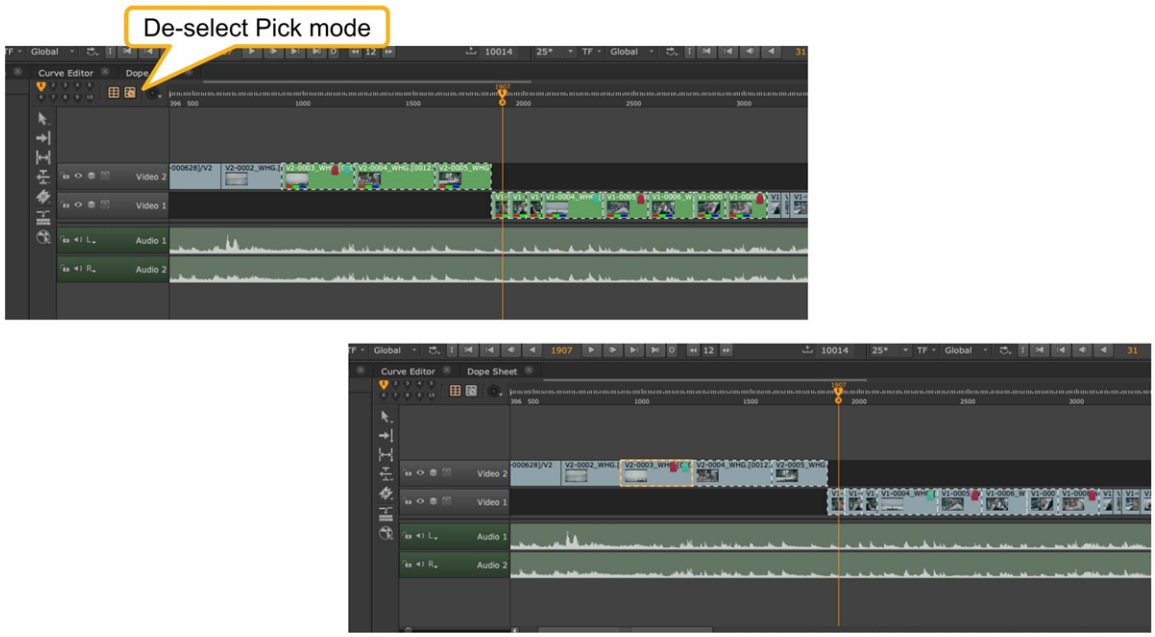


When selecting via either the **Left Click+Drag Marquee** or **Shift+Left Click** methods, the direction of the selection is ignored and clips are displayed in the Contact Sheet in the order they appear in the sequence.





4. Once you've finished adding the shots you want displayed in the Contact Sheet deactivate the Pick Mode button to return to normal editing and selection mode. You'll note that when you do this that your chosen shots are highlighted with a **dashed border** so that you can visually keep track of what shots are being used in the Contact Sheet. In the normal editing mode you cannot affect the Contact Sheet selection and will need to go back into the Pick Mode if you want to add or remove shots.

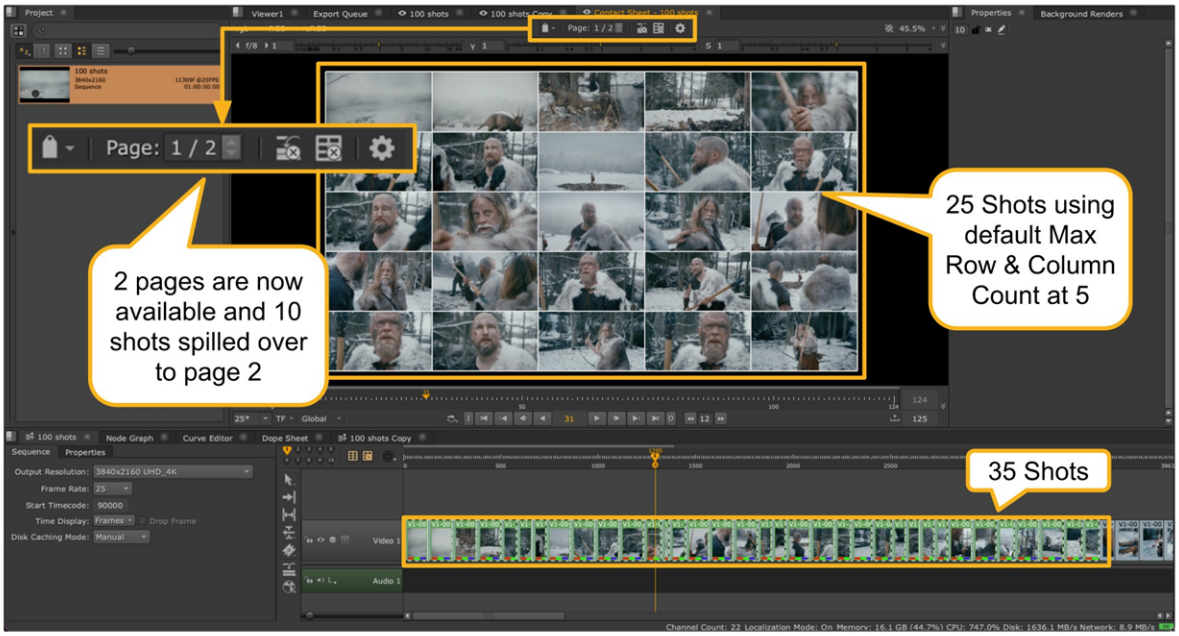


5. At this point, depending on how many clips you have selected, additional pages may be shown available in the Contact Sheet menu bar **Pages** area. The default **Max Rows & Max Columns Count** is set to 5 by default, meaning that up to 25 shots can be displayed per page before the remaining are automatically

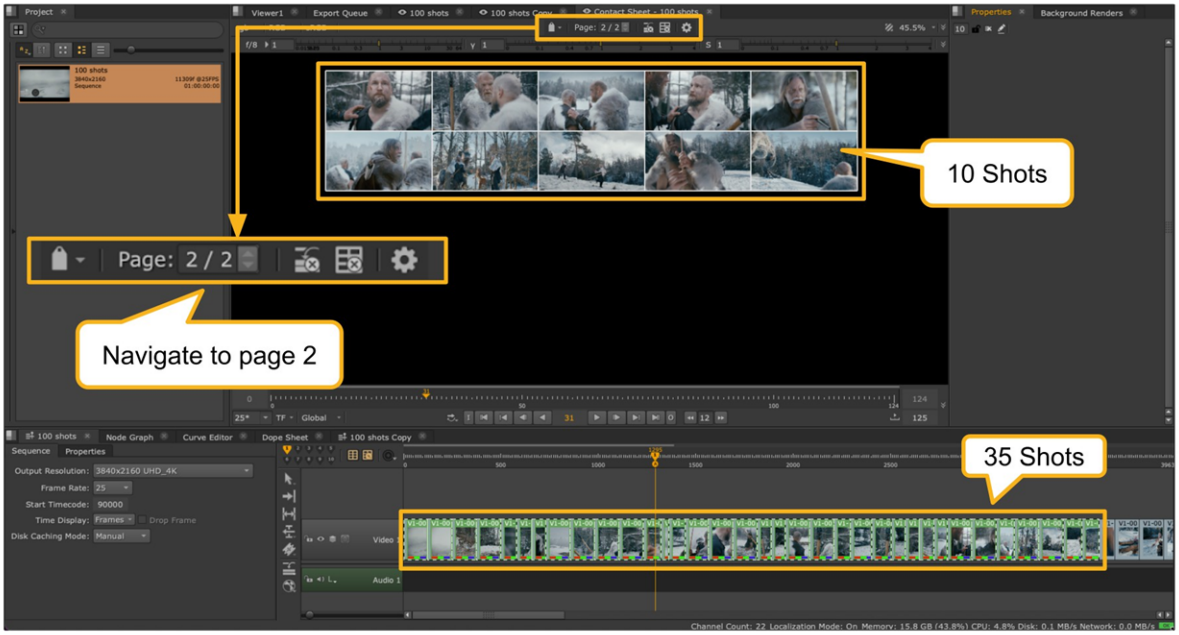
spilled over to additional pages. In the screenshot below with these default settings 15 shots are selected to be displayed.



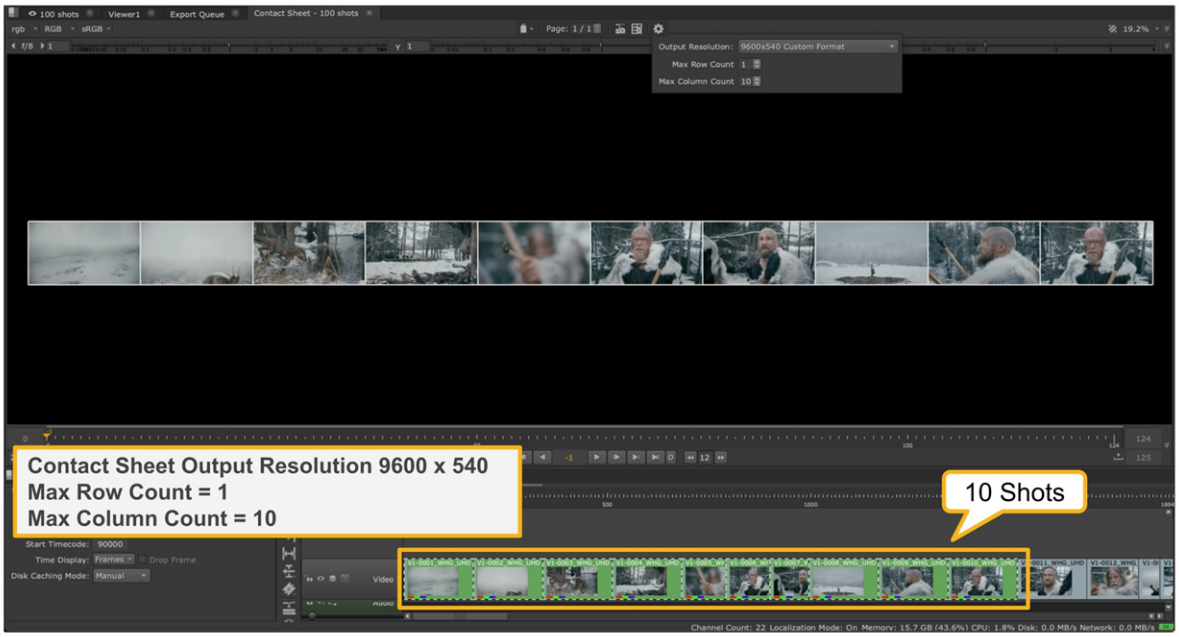
If you now select a further 20 shots, making a total of 35, then a second page is created as you've exceeded the maximum allowed determined by the **Row & Column Count**, and 10 shots are placed on page 2. The page navigation box is updated accordingly.

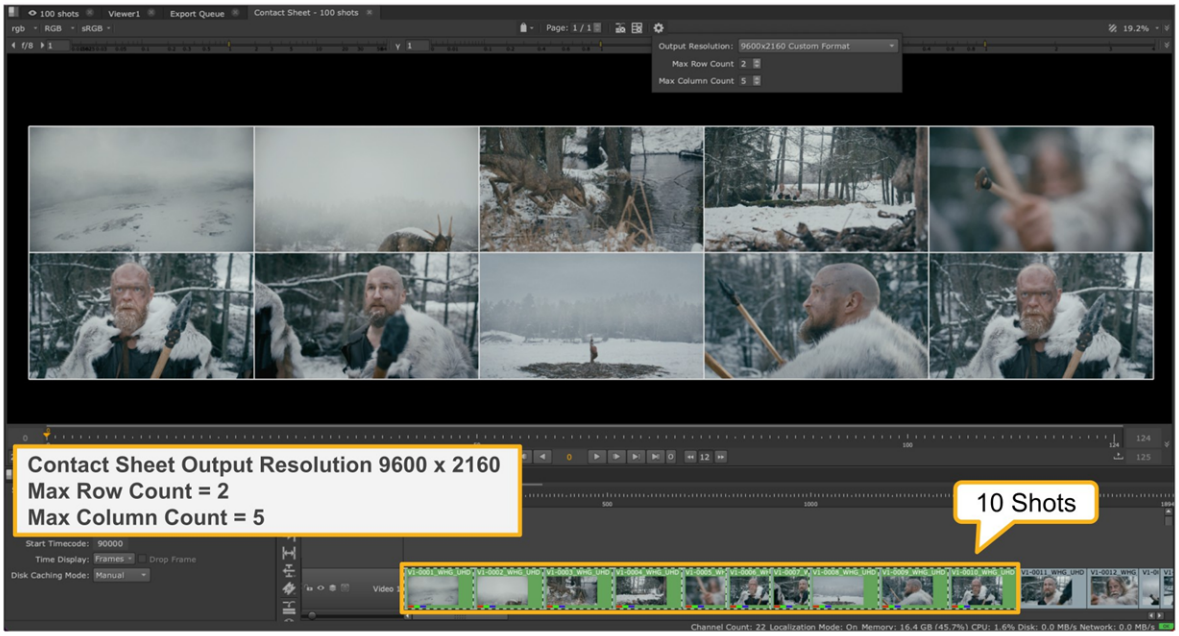


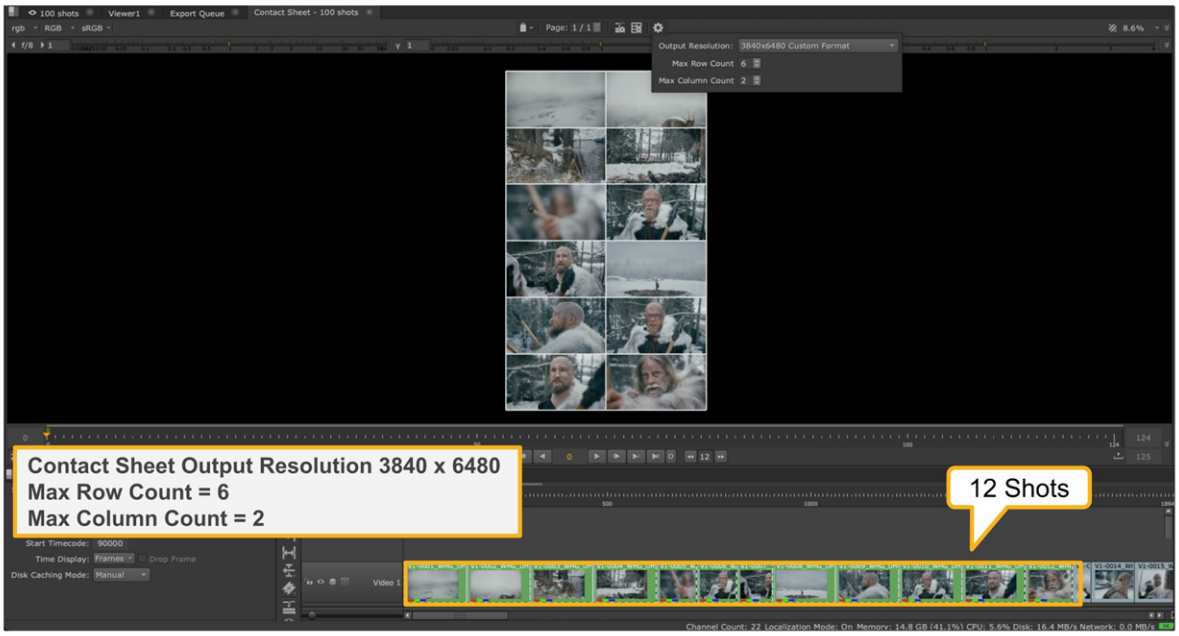
So now using the up and down arrows you can navigate between the pages; this is what is seen on page 2.

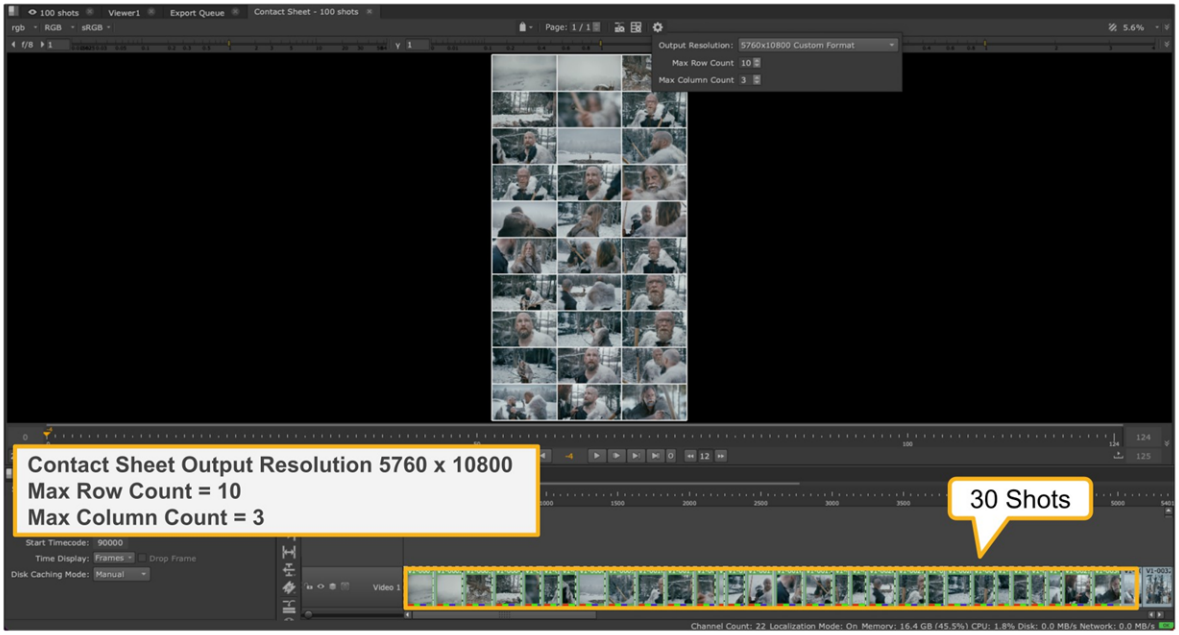


Using the Max Rows & Columns count you can optimise the amount of shots and their scale when working with large amounts of content in the Contact Sheet, which will also help playback performance, but in combination with the Contact Sheet Output Resolution create bespoke layouts a few examples of which can be seen below.




Contact Sheet Output Resolution 9600 x 2160
Max Row Count = 2
Max Column Count = 5
10 Shots


Contact Sheet Output Resolution 3840 x 6480
Max Row Count = 6
Max Column Count = 2
12 Shots

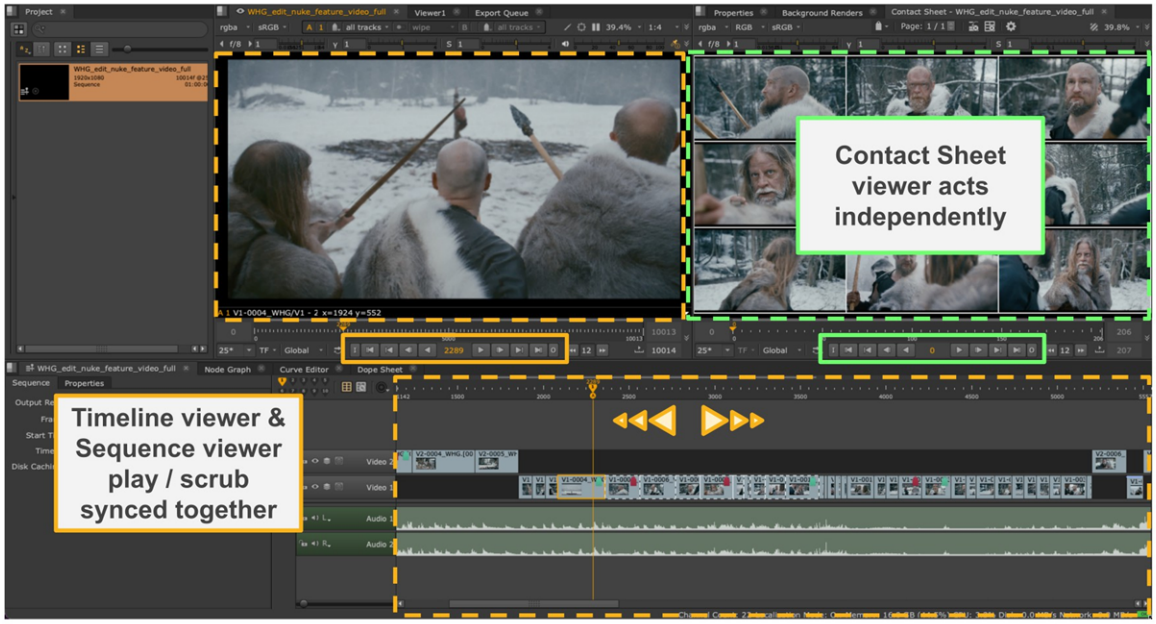


Using Contact Sheet: Playback

Once you have populated the Contact Sheet Viewer with some clips, there are several things to note in regard to playback, navigation and quality in the Contact Sheet Viewer.

Time

Looking at the timebar under the Contact Sheet Viewer, you'll notice that the clips in the Contact Sheet do not have a temporal relationship to the where they exist in the sequence. In fact the Contact Sheet and Timeline viewers act independently of each other when it comes to playback. Playing, scrubbing and stepping through frames in the sequence or timeline viewer does not affect the Contact Sheet Viewer and vice versa.



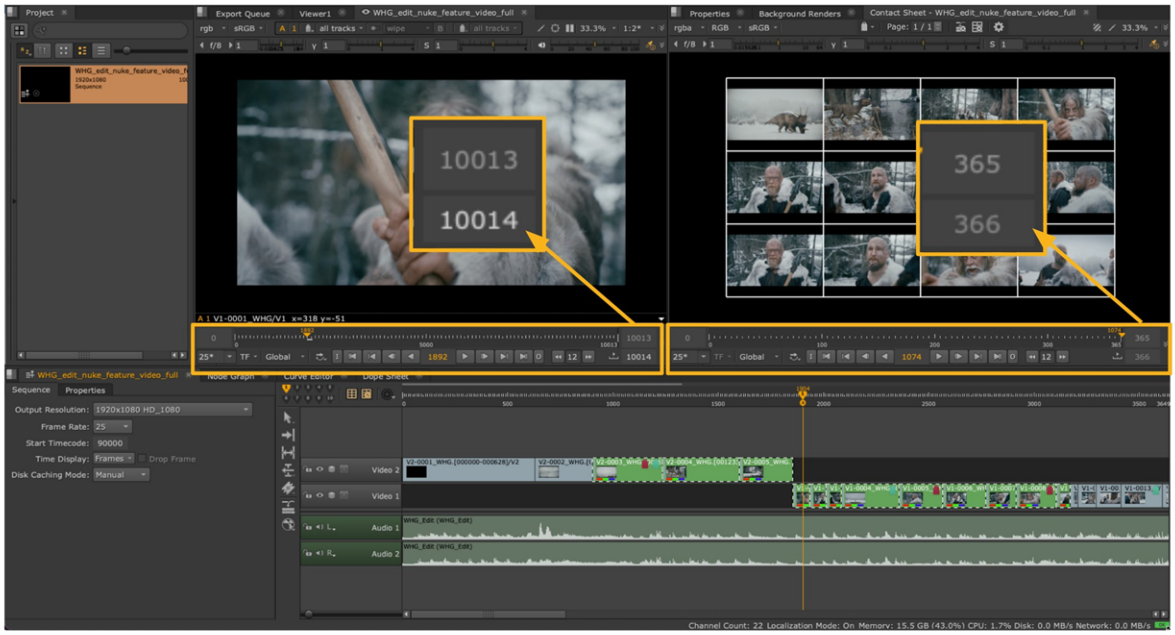
Cache

The playback cache is automatically split in half when a Contact Sheet Viewer is opened. For example, if you have assigned 20GB of RAM in your playback cache preferences 10GB will be allocated to the Contact Sheet Viewer and 10GB to the Timeline Viewer when using Contact Sheet.

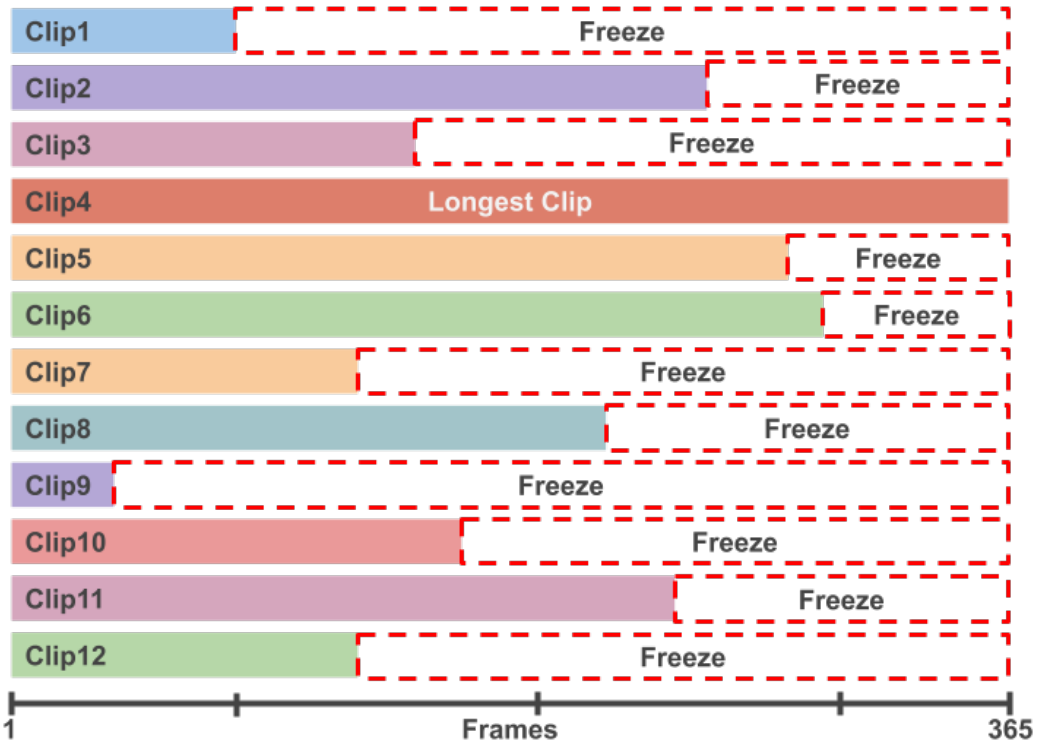
On closing a Contact Sheet Viewer, this RAW allocation will be returned to the active Viewer.

Playback range

If you look at the duration of the Timeline Viewer showing the sequence and the Contact Sheet Viewer duration you'll notice that they are considerably different.



This is because clips displayed in the Contact Sheet Viewer are all synced to frame 1 and the Contact Sheet Viewer playback range is dynamically determined by the the longest clip in the current selection. This also means that you will see shorter clips freeze on the last frame when you playback or scrub, and held to the duration of the longest clip.



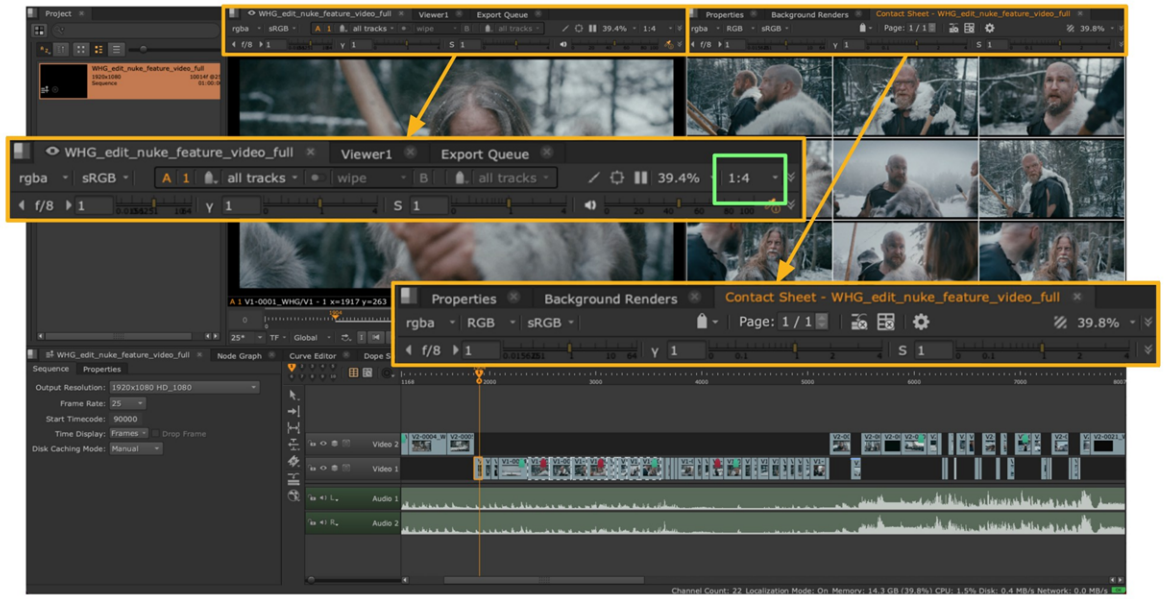
As mentioned above, this means that the Contact Sheet playback range will constantly change according to the clips loaded.

Navigation

Navigation around the Contact Sheet Viewer is exactly the same as the standard Viewer so you can use the normal zooming and panning gestures and home and fit hotkeys. You will also have a lot of the same Viewer controls as in the standard Viewer but with some exceptions.

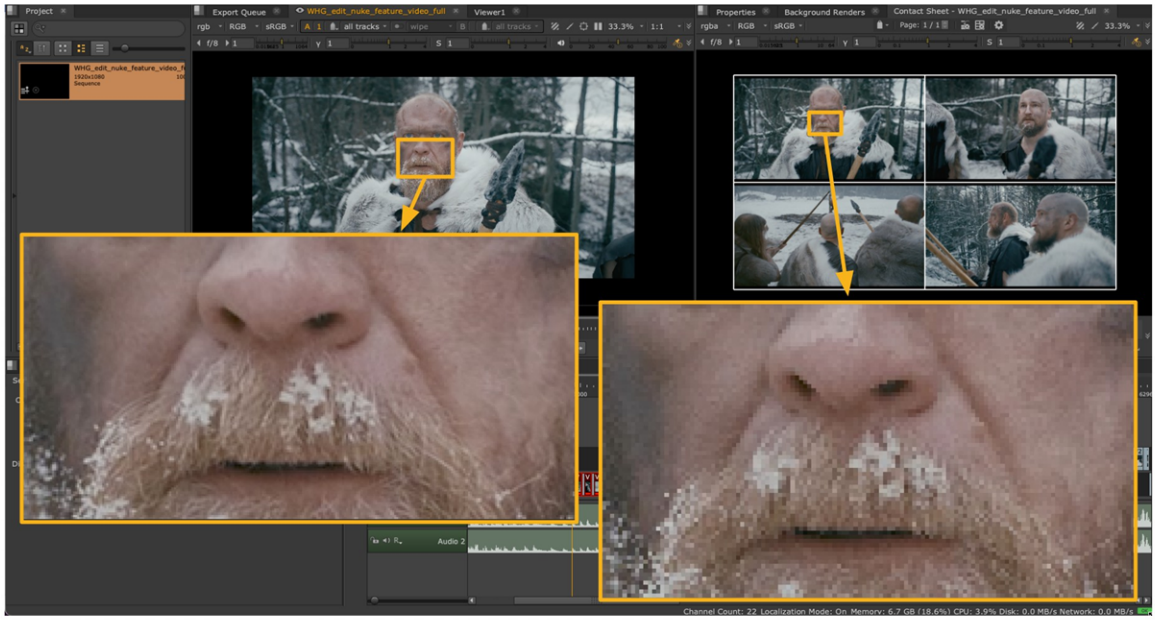
Quality

Most notably is the lack of a proxy or quality setting drop down. To enhance performance, the Contact Sheet will automatically set this for you.



By default the initial Contact Sheet Viewer resolution is set to HD 1920x1080 regardless of the sequence resolution. Depending on your display device this may be perfectly fine for general viewing and comparison tasks. But if you are using the Contact Sheet Viewer for more critical decisions then you will most likely require a higher fidelity.

Take the example here where we have loaded four clips into the Contact Sheet Viewer and zoomed into a similar area in both the timeline viewer on the left and the same clip in the Contact Sheet Viewer on the right. You can clearly see some pixelation happening in the Contact Sheet Viewer.

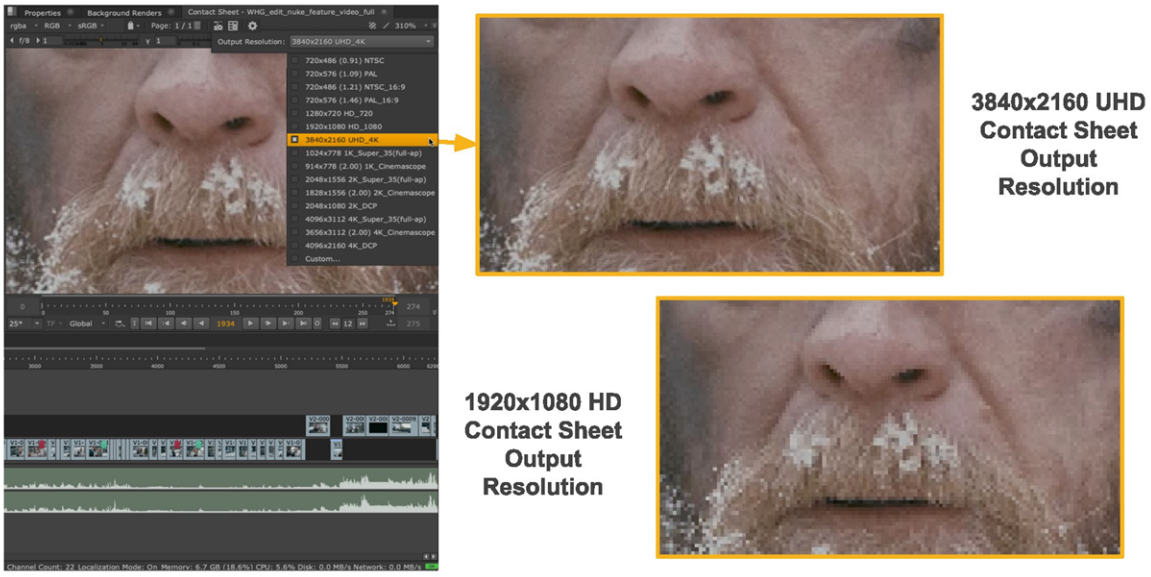


Improving quality

You can improve the quality of the Contact Sheet image by increasing the **resolution** of the Contact Sheet Viewer.

To do this, access the **Output Resolution** option found in the Contact Sheet Settings cog. This provides a familiar interface similar to the Sequence Output resolution setting, but again like the Viewer player timebar, this setting is particular to just the Contact Sheet Viewer.

If we change the Contact Sheet output resolution to UHD 3140x2160 4K we can now see a huge improvement of the zoomed in area compared to the Timeline Viewer.



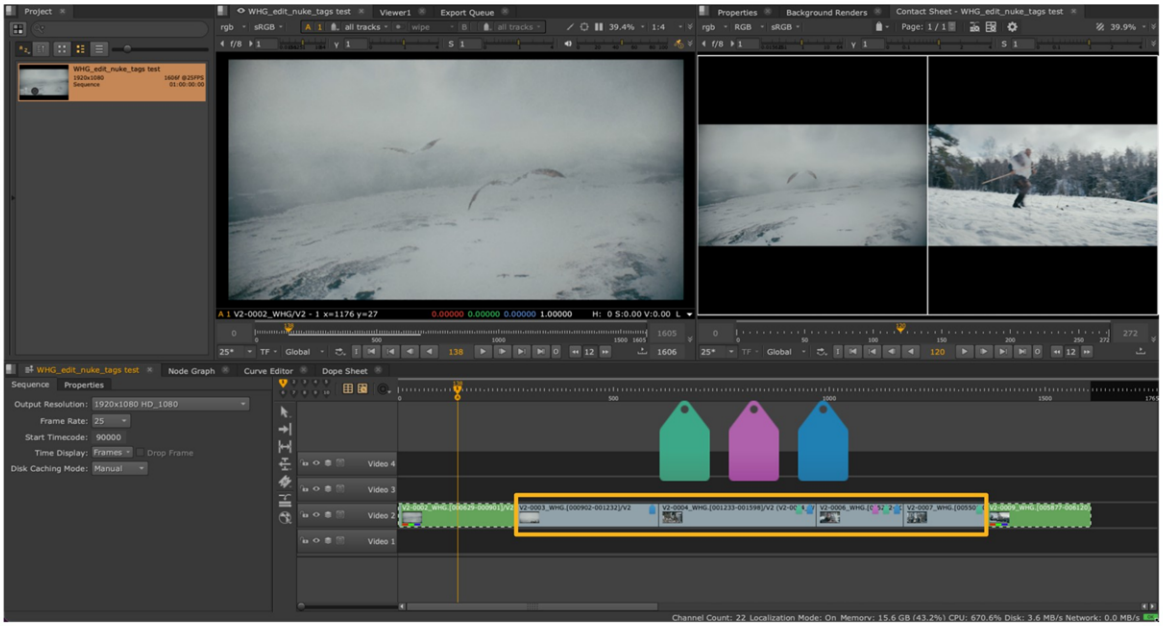
Note: Like the Viewer's playback behavior, the sequence resolution and the Contact Sheet's resolution are not directly linked which allows you a lot of flexibility.

Using Contact Sheet: Tag Filtering

In addition to manually choosing the shots with Pick Mode, you can also populate the Contact Sheet Viewer by using tags on shots in the Timeline.

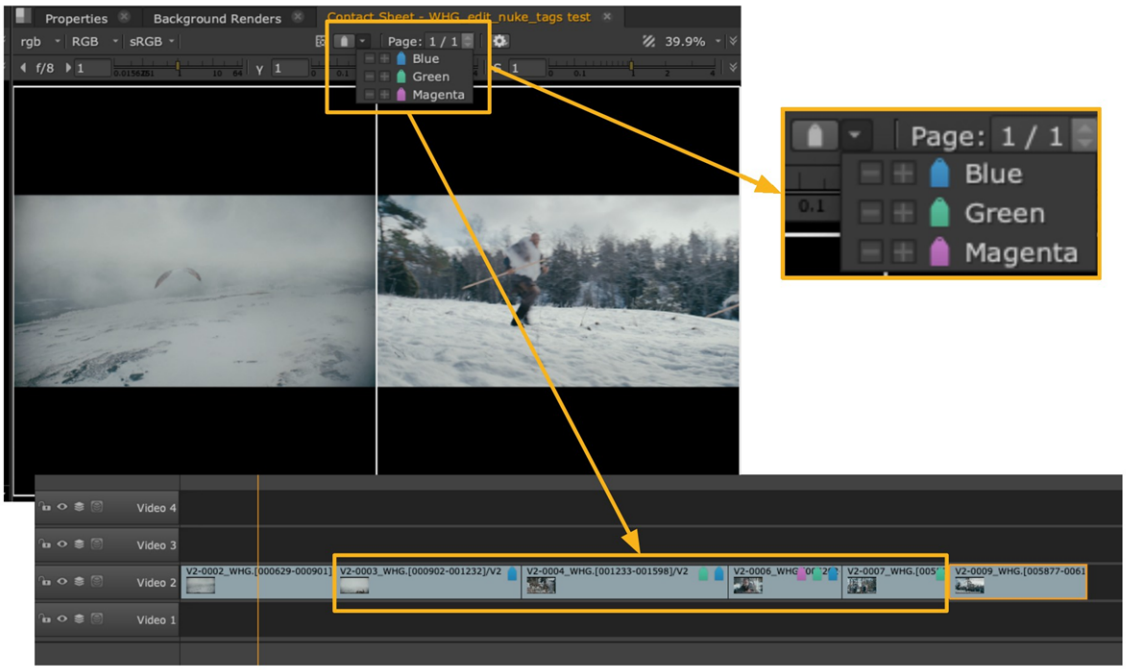
Note: Tag Filtering is available in Nuke Studio and Hiero, but not in HieroPlayer.

The example below shows a simplified sequence that has three tags applied in various ways to different clips. The beginning and end clip are also populating the Contact Sheet Viewer via manual Pick Mode selection.



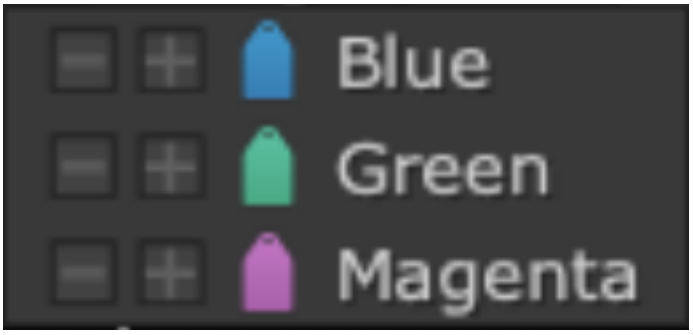
To start filtering the Contact Sheet by tags, you can simply select which tags and boolean operation you'd like to use.

Clicking the **Tag Filter** drop down in the Contact Sheet menu bar will display all the tags currently present on the sequence being sent to the Contact Sheet. Here you can also see check boxes indicating the boolean operation.



The two check boxes next to each tag entry have three states:

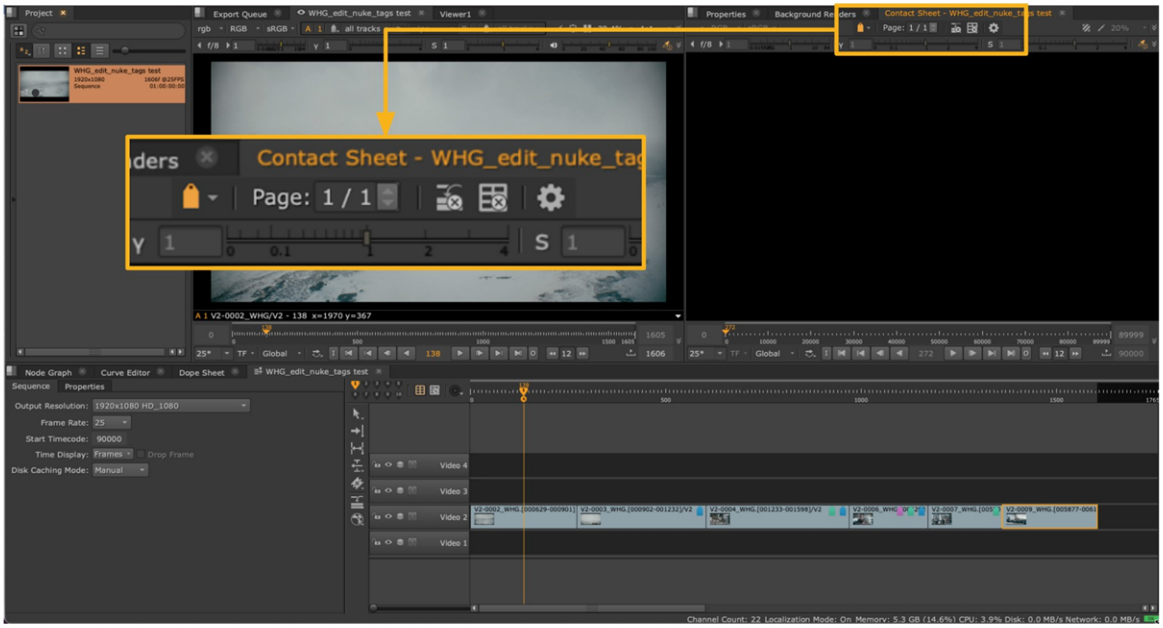
- **Inactive (gray)** - this tag is not be used to display a shot in Contact Sheet
- **Active (highlighted) '+' PLUS** - This tag is being used to include shots in the Contact Sheet and is performing an "AND" boolean
- **Active (highlighted) '-' MINUS** - This tag is being used to include shots in the Contact Sheet and is performing an "MINUS" boolean



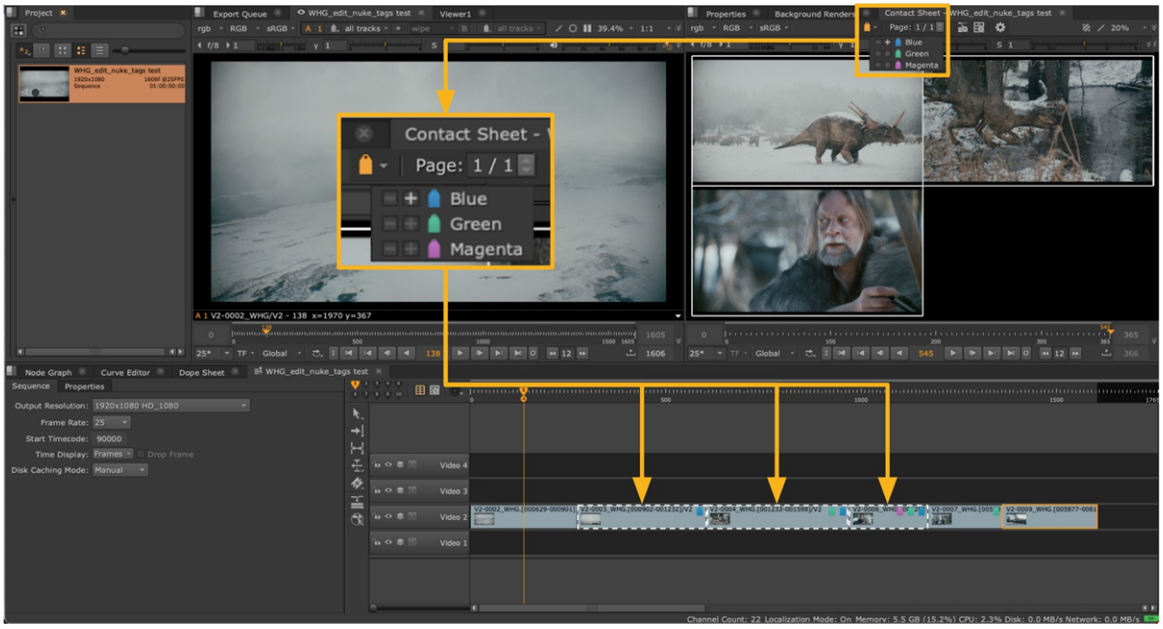
The check boxes make it quick to toggle between the different states and see your changes dynamically update the Contact Sheet Viewer. Note that you can only have one state for a particular tag enabled at a time. For instance if you had a '+' PLUS check box enabled for a tag then clicked the '-' MINUS check box the '+' PLUS state would automatically be disabled and the '-' MINUS checkbox enabled.

Clicking an active checkbox again will disable it, effectively removing that tag from the filtering.

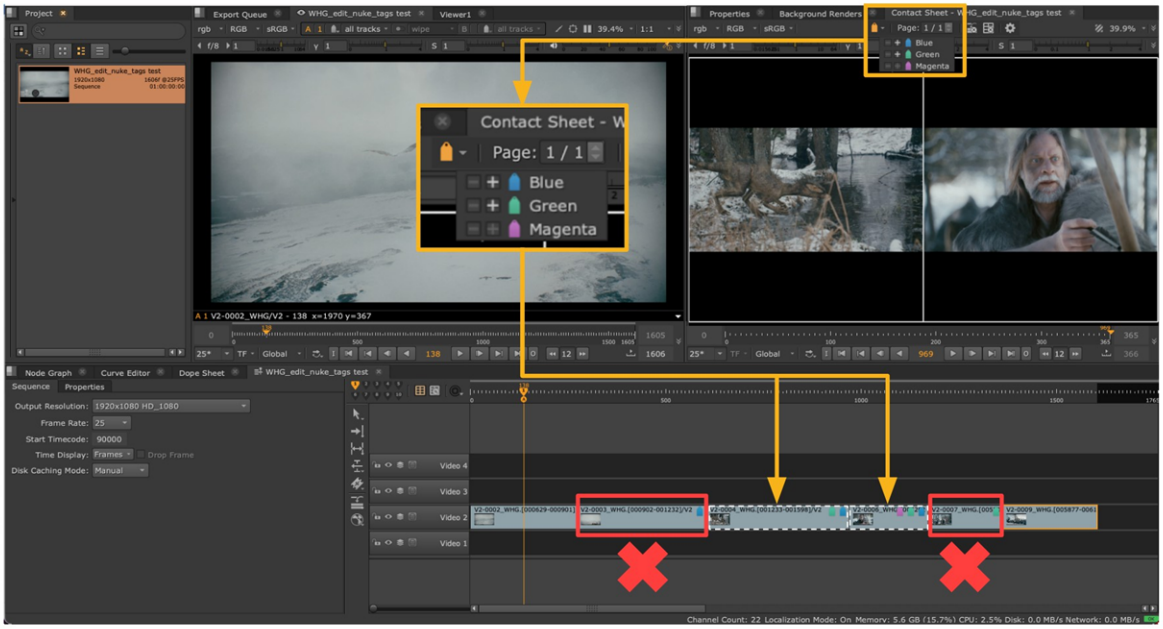
In this simple sequence we're first going to enable the Tag Filter Pick Toggle button on the left so we can see changes in the Contact Sheet Viewer as we change the tag filtering options. As we have no pre-chosen tags this will effectively show an empty Contact Sheet.



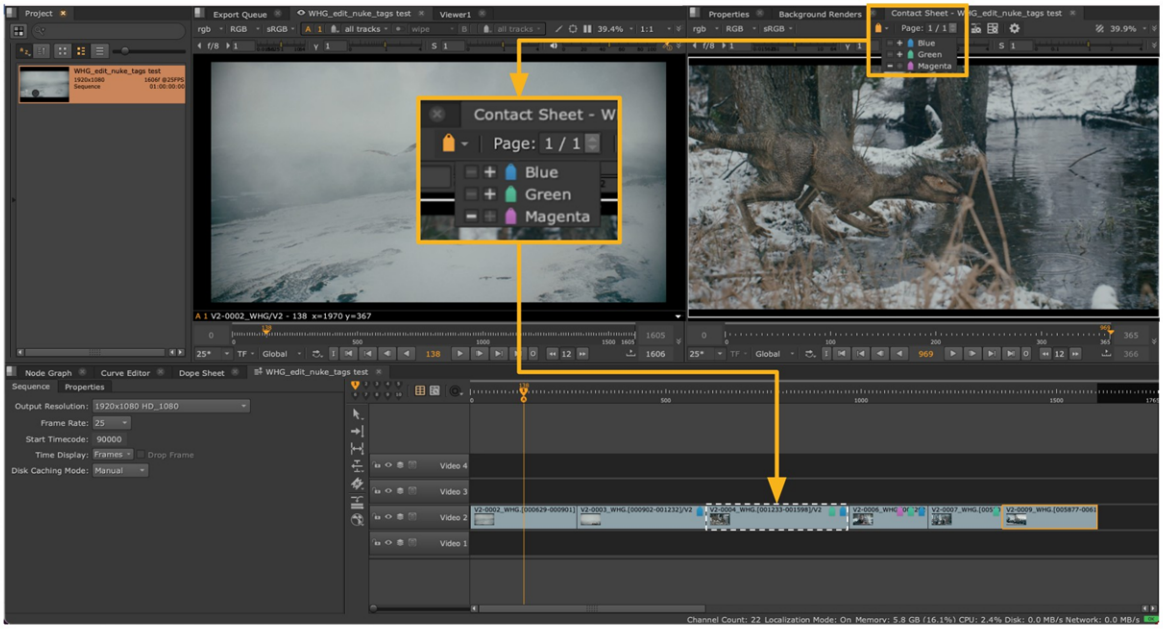
Now opening the tag filter drop down, check the '+' PLUS button next to the Blue tag. You'll see that three shots are added to the Contact Sheet because they contain a Blue tag.



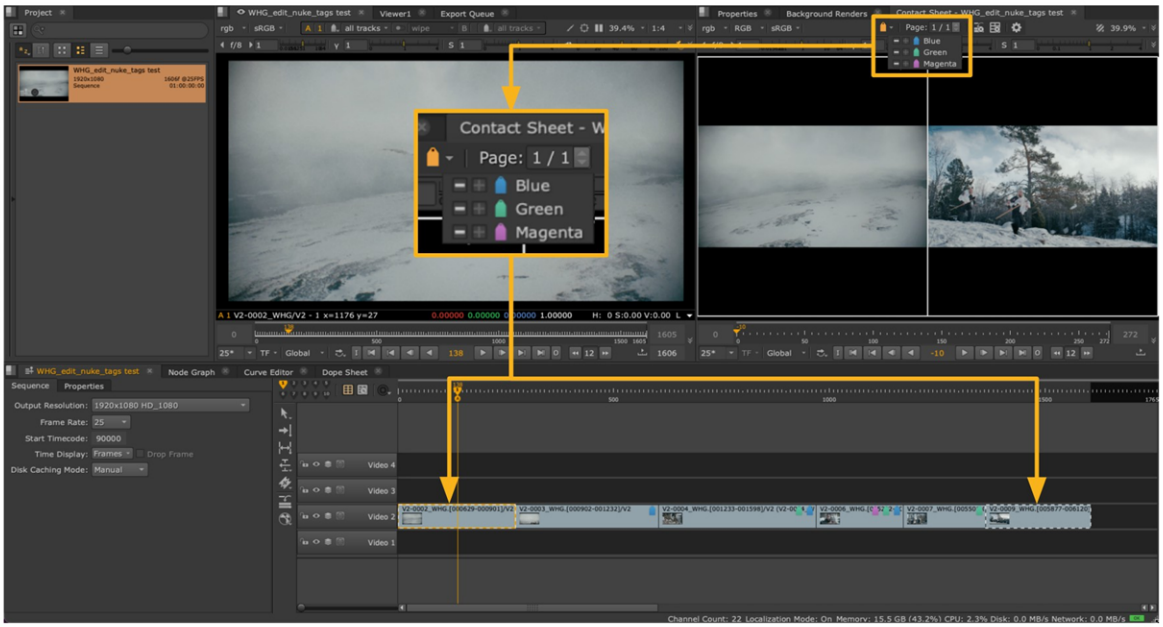
Now if you check the '+' PLUS button next to the Green tag only two shots are displayed because we are asking the Contact Sheet to ONLY filter and display shots that meet that "AND" boolean. In other words the clips MUST contain both a Blue AND a Green tag. You'll notice that there is a clip that contains just a Blue tag and just a Green tag that are not displayed.



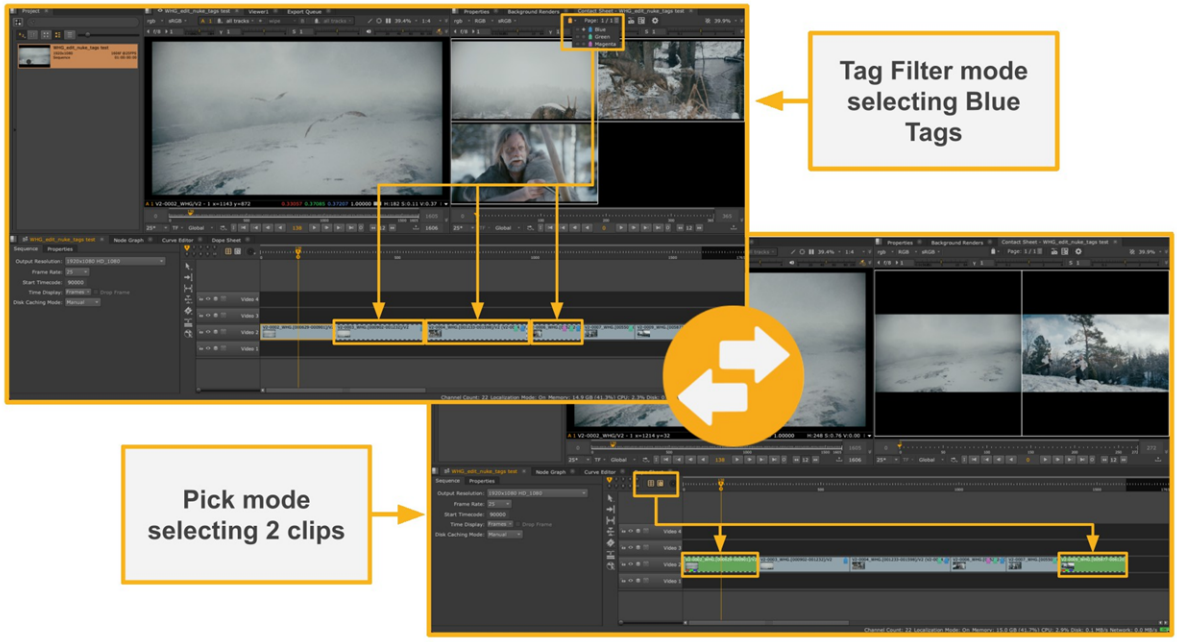
Now if you click the '-' MINUS button next to the Magenta tag you will only see the one shot which has a Blue AND Green tag but NOT a Magenta one.



Using this simple mechanism it should be possible to filter and display a lot of combinations. For instance if you wanted to see all shots that contained no tags then you could just switch to '-' MINUS on all the tags and you would then see what is left; in this case the two un-tagged shots at the beginning and end of the sequence.



The manual Pick Mode selection and Tag filter selections are persistent, meaning that you can use the Tag Filter/Pick Toggle button to quickly switch between the two modes and see what was last used.

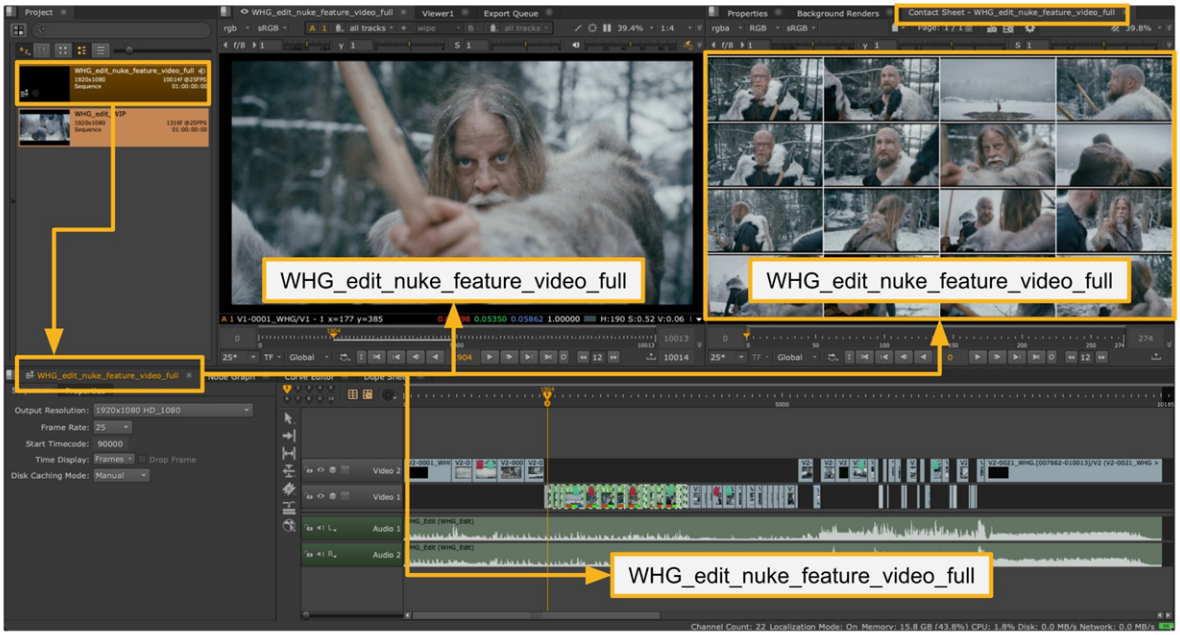


Using Contact Sheet: Workflows & Soft Effects

Display other sequences alongside Contact Sheet

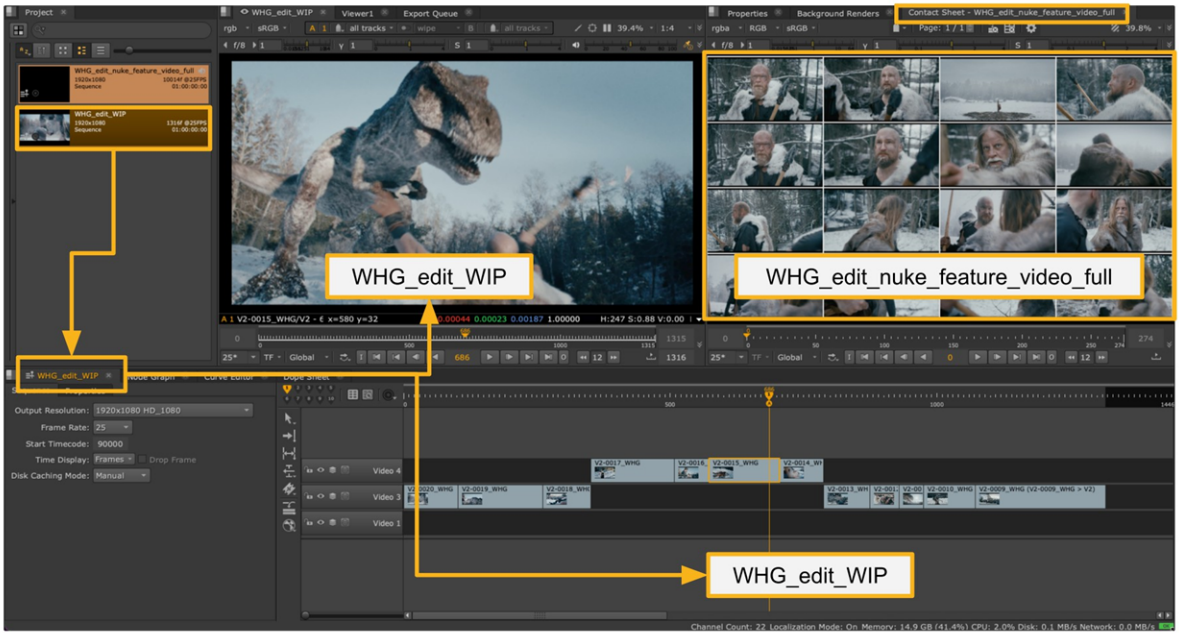
Although in this release Contact Sheet only supports loading clips from a single sequence it is still possible to have multiple sequences loaded and displayed whilst using a Contact Sheet Viewer from another sequence.

To do this, simply open a sequence and create a Contact Sheet Viewer populated with some clips. In this example note there is a sequence loaded called “WHG_edit_nuke_feature_video_full”.

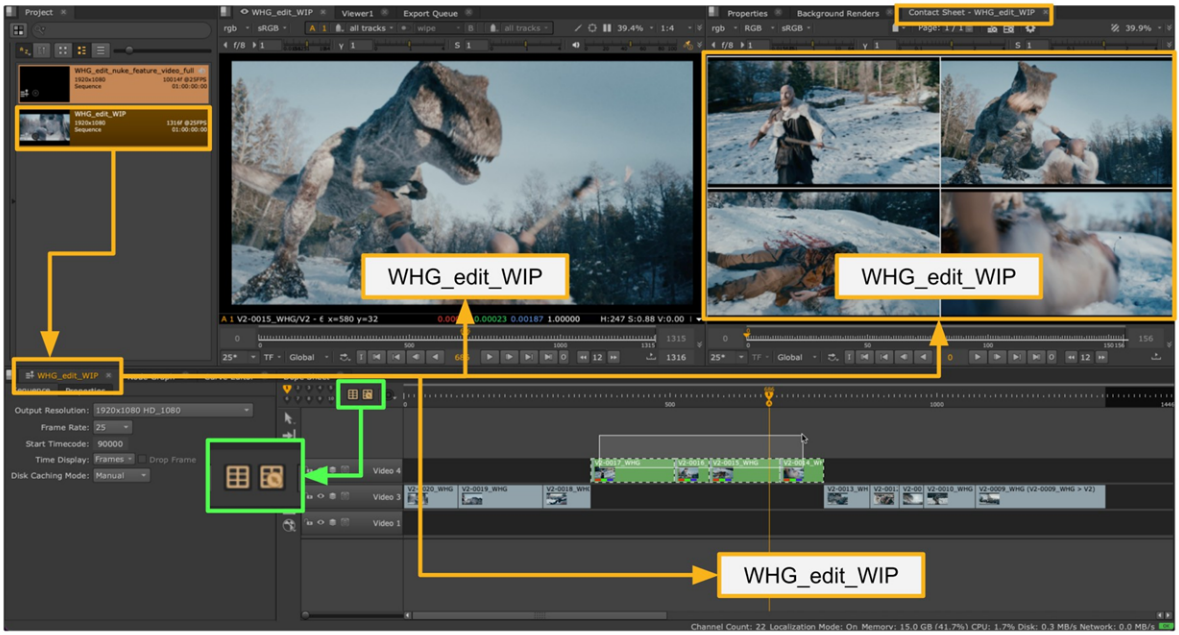


Once you've done that, you can now either double click a different sequence to open this into the existing Timeline Viewer, or open a sequence into a new Timeline Viewer. In this example the opened sequence is called WHG_edit_WIP.

Either way the end result is the same, the new sequence is displayed in a Timeline Viewer and the **original** sequence that was the source for the Contact Sheet is still displayed.



If you now want to have this NEW sequence populate the Contact Sheet you just need to click the Send To Contact Sheet button on the WHG_edit_WIP and this sequence will now replace the previously loaded one.



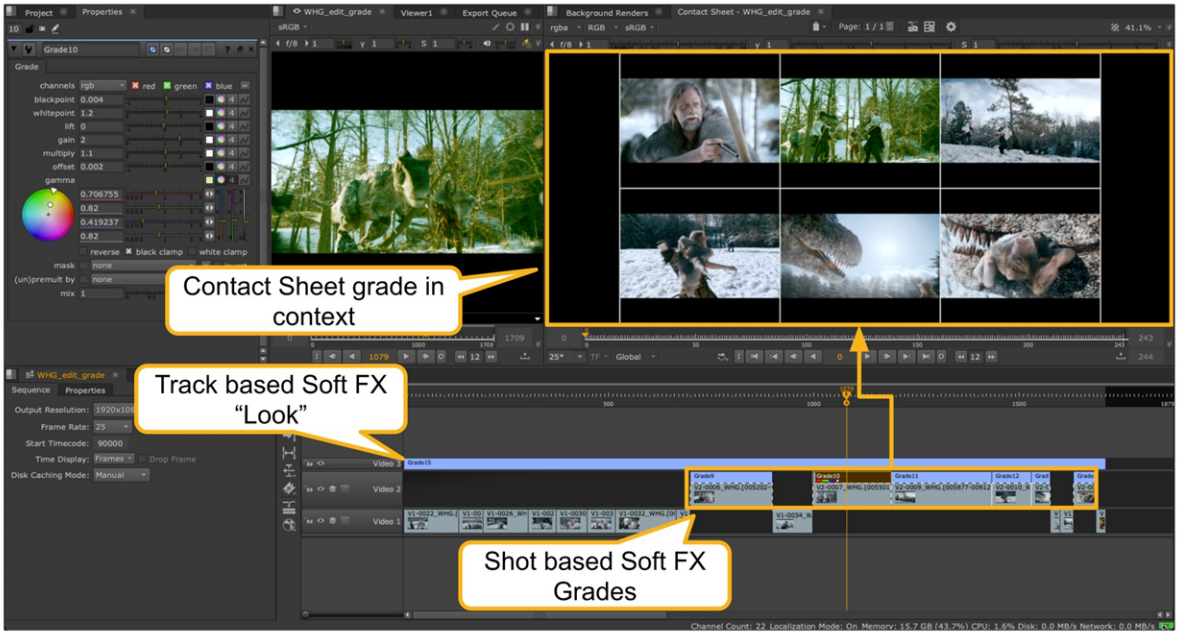
This functionality allows you to view one sequence's shots in a Contact Sheet against another sequence.

Display multiple shots with Soft Effects

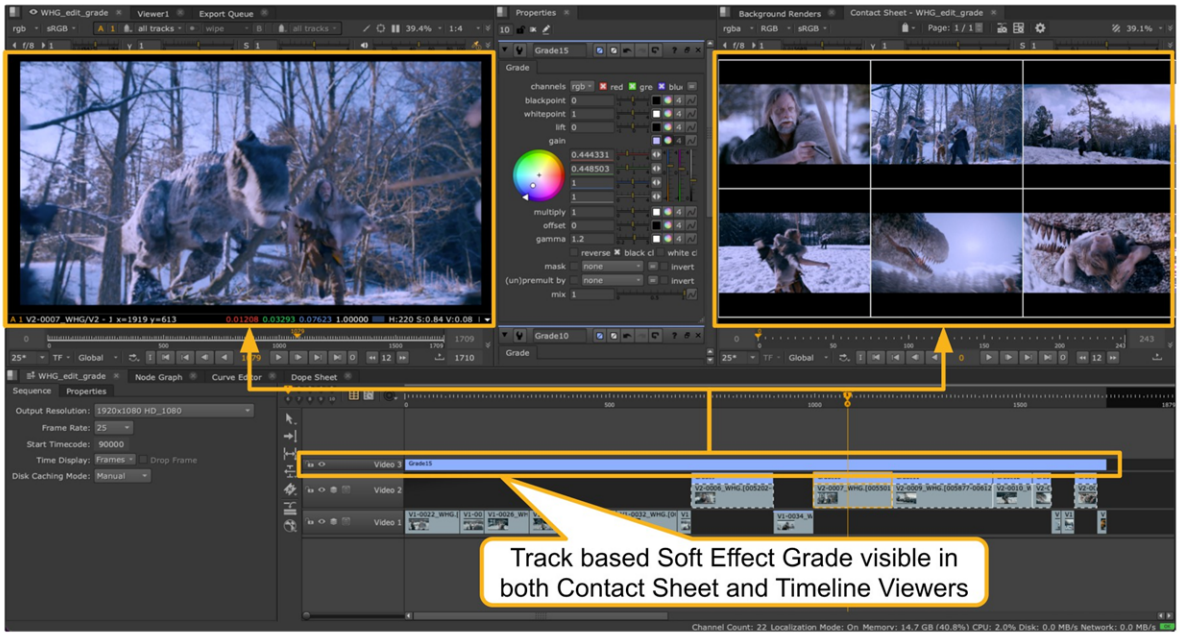
Another really useful workflow in Contact Sheet is the ability to display multiple shots that have clip-based and layer-based Soft Effects.

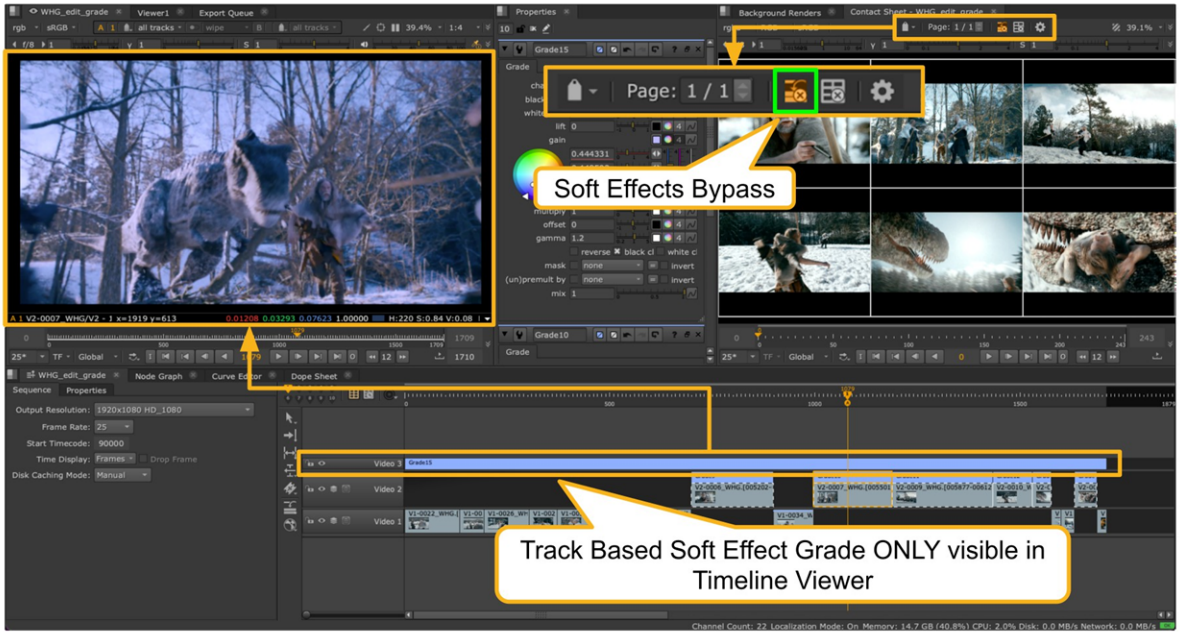
The example below demonstrates using a Grade Soft Effect which is being used to balance out shots. This effect then can be viewed dynamically over multiple shots in a Contact Sheet Viewer.

You could also use Contact Sheet to have a "show look" over everything and see the culmination of all these effects.



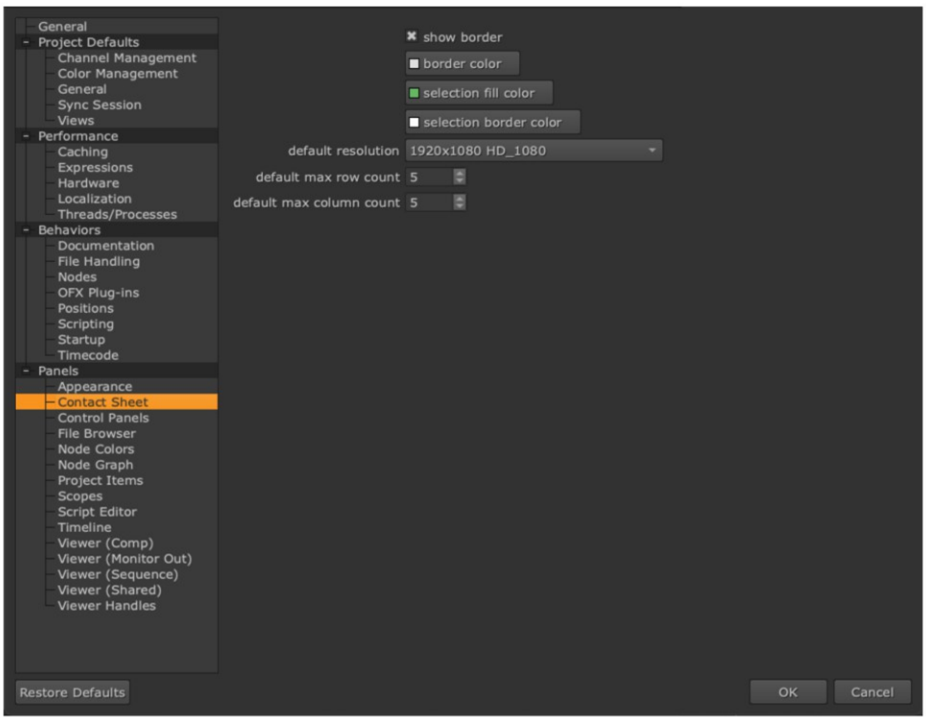
When the sequence has track based Soft Effects applied you can quickly bypass these for the Contact Sheer Viewer by clicking on the Soft FX Bypass button in the Contact Sheet menu bar. This will then hide these track based Soft Effects but **only** for the Contact Sheet Viewer; the Timeline Viewer will still display the track Soft Effects.





Using Contact Sheet: Preferences

There is a new Contact Sheet section in the Preferences panel.



The options are:

- **show border** - Allows you to toggle the Contact Sheet shot outline on and off. The default is enabled.
- **border color** - Allows you to customize the Contact Sheet shot outline color. The default is white.
- **selection fill color** - Allows you to customize the fill color of shots in the **sequence** being used in the Contact Sheet Viewer. Visible only in Pick mode. The default is green.
- **selection border color** - Allows you to customize the border color of shots in the sequence being used in the Contact Sheet Viewer. Visible in both Pick and Edit mode. The default is white.
- **default resolution** - Allows you to set a default Contact Sheet Output resolution that will affect the quality of the shots displayed when either zooming in close or when the Contact Sheet is displayed on a large monitor device. The default is 1920x1090 HD.
- **default max row count** - Sets the maximum number of rows on a Contact Sheet Page. The default is 5.
- **default max column count** - Sets the maximum number of columns on a Contact Sheet Page. The default is 5.

Advanced Compositing with NukeX and Nuke Studio

These pages explain in detail key features of NukeX and Nuke Studio. For more information on the differences between the various applications, see Nuke Products.

- **VectorGenerator, Kronos, and MotionBlur** - VectorGenerator, Kronos, and MotionBlur use Foundry's advanced motion estimation technology to produce images containing motion vector fields, slow down or speed up footage, and add motion blur. For more information, see Retiming and Motion Blur.
- **LensDistortion** - The LensDistortion node gives you multiple ways to analyze image sequences and lens grids, resulting in a lens model and the ability to warp and un-warp in order to compensate for lens distortion. For more information, see Working with Lens Distortion.
- **PlanarTracker** - The PlanarTracker is a powerful tool for tracking surfaces that lie on a plane in your source footage. You can use your tracking results to replace the tracked plane with another image for instance. For more information, see Tracking with PlanarTracker.
- **CameraTracker** - With the fully integrated 3D CameraTracker node, you can do you own camera solves and create reference geometry and cards positioned at tracked points in the 3D scene. For more information, see Camera Tracking.
- **MatchGrade** - The MatchGrade node allows you to automatically calculate a grade to match the colors in the **Source** input to the colors in the **Target** input. For more information, see Match Grades between Shots with MatchGrade.
- **Smart Vector Toolset** - The Smart Vector Toolset allows you to work on one frame in a sequence and then use motion vector information to accurately propagate work throughout the rest of the sequence. For more information, see Using the Smart Vector Toolset.
- **DepthGenerator** - The DepthGenerator node provides a method to produce a per-frame Z-depth map from the input 2D footage. It additionally requires a camera solve which can be obtained using the CameraTracker node. For more information, see Generating Depth Maps.
- **PointCloudGenerator** - You can create dense point clouds from your footage using the PointCloudGenerator and CameraTracker. For more information, see Creating Dense Point Clouds.
- **PoissonMesh** - With the PoissonMesh node, you can use a dense point cloud to create a 3D mesh from your 2D footage. For more information, see Using the PoissonMesh Node.
- **ModelBuilder** - ModelBuilder provides an easy way to create 3D models for a 2D shot, given a tracked camera. You can build a model by creating shapes and then editing them, and align models over your 2D footage by dragging vertices to their corresponding 2D location. For more information, see Using ModelBuilder.

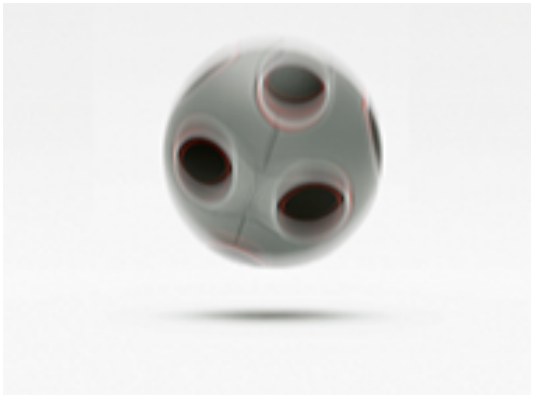
- **Particles** - The Particle node set is a solution for creating particles in a 3D environment. You can use the Particle nodes for emitting, manipulating and displaying limitless types of particles in your 3D scene. For more information, see Creating 3D Particles.
- **PrmanRender** - PrmanRender is a render node that works together with Pixar's PhotoRealistic RenderMan® Pro Server software to give you an even better quality render result. PrmanRender is an alternative to the ScanlineRender node for rendering 3D scenes, and it gives you control over features such as shadows, reflections, refractions and depth-of-field. For more information, see PrmanRender .
- **FurnaceCore** - This plug-in bundle consists of Foundry's best Furnace tools, including regraining, rig-removal, and more. For more information, see Using F_DeFlicker2, Using F_ReGrain, Using F_WireRemoval Using F_Align, Using F_RigRemoval, and Using F_Steadiness.
- **Machine Learning** - Use the CopyCat node copy sequence-specific effects, such as garbage matting, beauty repairs, or deblurring, from a small number of frames in a sequence and then train a network using machine learning to replicate this effect on a full sequence. For more information, see Train Neural Networks to Replicate Effects Using Machine Learning.
- **Using Pre-trained PyTorch Models in NukeX** - The CatFileCreator node creates custom **.cat** files from PyTorch TorchScript **.pt** files within Nuke, opening up a wide range of pre-existing, trained models. See Import Pre-Trained PyTorch Models for more information.
- **Using Nuke Server to Connect Unreal Editor and NukeX** - The UnrealReader node connects NukeX to Unreal Editor using the Nuke Server plug-in. UnrealReader makes it quick and easy for you to generate live renders from Unreal Engine and control the results in NukeX, by breaking objects into layers, separating passes, building environment maps, and tweaking shot framing. See Visualize Unreal Scenes in NukeX and Nuke Indie for more information.
- **BlinkScript** - The BlinkScript node runs Foundry's Blink framework enabling you to write your code once and run it on any supported device. For more information, see Using the BlinkScript Node.
- **ParticleBlinkScript** - This node works in a similar way to the BlinkScript node but allows you to write Blink scripts that operate on particles. This enables you to write your own Particle nodes, to create the specific behavior you need. For more information, see Using the ParticleBlinkScript Node.
- **CaraVR** - A sub-set of CaraVR's plug-ins allowing you to solve and stitch mono and stereo VR rigs and perform a number of mono 360 compositing tasks. For more information, see Stitching Rigs with CaraVR.

Retiming and Motion Blur

This chapter looks at creating motion vectors using VectorGenerator, retiming sequences using Kronos, and adding motion blur using MotionBlur.



Before MotionBlur.



After MotionBlur.

VectorGenerator

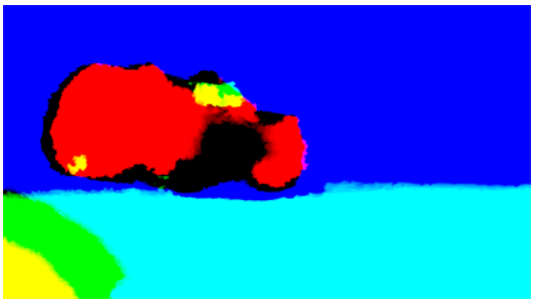
VectorGenerator allows you to produce images containing motion vector fields. A vector field for an image in a sequence has the same dimensions as the image, but contains an (x,y) offset per pixel. These offsets show how to warp a neighboring image onto the current image. Clearly, as most of the images in a sequence have two neighbors, each can have two vector fields:

1. The backward vector field: the x and y offsets per pixel that, when applied to the previous frame in the sequence, allow you to reconstruct an approximation to the current frame.
2. The forward vector field: the x and y offsets needed to transform the next frame into an approximation to the current one.

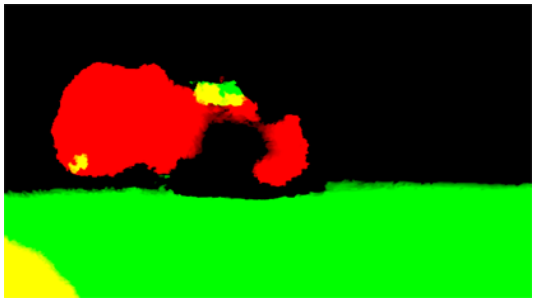
The output from VectorGenerator is stored in the vector channels. The images below show the different vector images for a clip.



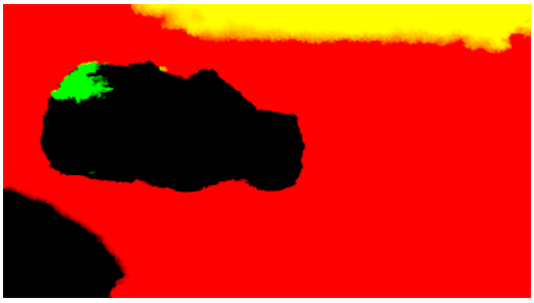
The source sequence.



Forward and backward motion vectors.



Forward motion vectors.



Backward motion vectors.

When viewing 'forward' or 'backward' motion vectors, the Viewer represents the x values as amounts of red and y values as amounts of green. Motion vectors can be positive or negative, where zero represents no motion.

In general, once you have generated a sequence of motion vector fields that describe the motion in a particular clip well, they are suitable for use in any nodes which can take vector inputs. These include Kronos and MotionBlur. If you are going to be using more than one of these effects in your project, it might be worth generating the vector fields beforehand with VectorGenerator, so that they can be reused.

Quick Start

Here's a quick overview of the workflow:

1. Add VectorGenerator to your node tree. See Connecting VectorGenerator below.
2. View and refine the results. See Viewing and Refining the Results.

Tip: You can check **Use GPU if available** to have the node run on the graphics processing unit (GPU) rather than the central processing unit (CPU). For more information on GPU support, see the Release Notes for your version of Nuke.

You can select the GPU to use in the Preferences. Press **Shift**+**S** to open the **Preferences** dialog, make sure you're viewing the **Preferences** > **Performance** > **Hardware** tab, and set **default blink device** to the device you want to use. You must restart Nuke for the change to take effect.

If you are using a render license of Nuke, you need to add **--gpu** on the command line.

Connecting VectorGenerator

To connect VectorGenerator:

1. Select **Time** > **VectorGenerator** to insert a VectorGenerator node after the sequence from which you want to generate motion vectors.
2. Attach a Viewer to the output of the VectorGenerator node.
3. If your sequence is composed of a foreground object moving over a background, the motion estimation is likely to get confused at the edge between the two. To fix this, add a matte of the foreground region to the **Matte** input. Then, use **Matte Channel** in the VectorGenerator properties to select which component of the matte to use.

   This helps the motion estimation algorithm inside VectorGenerator understand what is foreground and background in the image, so that the dragging of pixels between overlapping objects can be reduced.

   White areas of the matte are considered to be foreground, and black areas background. Gray areas are used to attenuate between foreground and background
4. If you supplied a foreground matte in the previous step, you can set **Output** in the VectorGenerator properties to:

   • **Foreground** - to output the vectors for the foreground regions.

   • **Background** - to output the vectors in the background regions.
5. Proceed to Viewing and Refining the Results.

Viewing and Refining the Results

To view and refine the results:

1. Select the required vector calculation type from the **Method** dropdown:
   - **Local** - uses local block matching to estimate motion vectors. This method is faster to process, but can lead to artifacts in the output.
   - **Regularized** - uses semi-global motion estimation to produce more consistent vectors between regions.



Note: Scripts loaded from previous versions of Nuke default to **Local** motion estimation for backward compatibility. Adding a new VectorGenerator node to the Node Graph defaults the **Method** to **Regularized** motion estimation.

2. To view the generated motion vector fields, click the channels dropdown menu on top of the Viewer and select:
   - **motion** - to view both forward and backward motion vectors.
   - **forward** - to view the forward motion vectors.
   - **backward** - to view the backward motion vectors.
3. If the calculated motion vector fields do not produce the results you're after when used with other nodes (such as Kronos and MotionBlur), try adjusting **Vector Detail** in the VectorGenerator properties.

   This determines the resolution of the vector field. The larger vector detail is, the greater the processing time, but the more detailed the vectors should be. A value of 1.0 generates a vector at each pixel. A value of 0.5 generates a vector at every other pixel. For some sequences, a high vector detail near 1.0 generates too much unwanted local motion detail, and often a low value is more appropriate.
4. If you're using the **Regularized** vector calculation method, adjust the **Strength** control to determine the strength of pixel matching between frames. Higher values allow you to accurately match similar pixels in one image to another, concentrating on detail matching even if the resulting motion field is jagged. Lower values may miss local detail, but are less likely to provide you with the odd spurious vector, producing smoother results.



Note: The default value should work well for most sequences.

5. If you're using the **Local** vector calculation method, adjust the **Smoothness** control to improve your results. A high smoothness can miss lots of local detail, but is less likely to provide you with the odd spurious vector, whereas a low smoothness concentrates on detail matching, even if the resulting field is jagged.



Note: The default value should work well for most sequences.

6. If there are variations in luminance and overall flickering in your Source sequence, enable **Flicker Compensation** to avoid problems with your output.

   Examples of variable luminance include highlights on metal surfaces, like vehicle bodies, or bodies of water within a layer that reflects light in unpredictable ways.



Note: Enabling **Flicker Compensation** increases rendering time.

7. By default, VectorGenerator analyzes motion based on the brightness of the image. Specifically this is the mean of the red, green, and blue channels. However, you can bias this using the **Weight Red**, **Weight Green**, and **Weight Blue** controls under **Tolerances**. For example, if you set the **Weight Red** and **Weight Green** parameters to zero, VectorGenerator only looks for motion in the blue channel.
8. Once you're happy with the results, we recommend that you insert a Write node after VectorGenerator to render the original images and the vector channels as an **.exr** file. This format allows for the storage of an image with multiple layers embedded in it. Later, whenever you use the same image sequence, the motion vector fields are loaded into Nuke together with the sequence.

Kronos

Kronos is NukeX's retimer, designed to slow down or speed up footage. It works by calculating the motion in the sequence in order to generate motion vectors. These motion vectors describe how each pixel moves from frame to frame. With accurate motion vectors, it is possible to generate an output image at any point in time throughout the sequence by interpolating along the direction of the motion.



Simple mix of two frames to achieve an in-between frame.



Kronos vector interpolation of the same two frames.

By default, Kronos is set to perform a half-speed slow down. This is achieved by generating a new frame at position 0.25 and 0.75 between the original frames at 0 and 1. Frames are created at a quarter and three

quarters instead of zero (an original frame) and a half so as not to include any original frames in the re-timed sequence. This avoids the pulsing that would otherwise be seen on every other frame on a half-speed slow down, and can introduce motion blur.

Kronos only interpolates between input frames, as it cannot extrapolate images before the first frame or after the last frame. A retime with a constant speed **s** "stretches" the output time by a factor **1/s**, and generates the required images to fill in all intervals between the input frames.

In the following table, **|** denotes where we need to have images to fill the video sequence. In this example, we assume that the input sequence has 5 frames, denoted by **X**, and we want to retime it with constant speed 0.5 (the default setting). This operation corresponds to stretching the time between the input frames by a factor of 2, which leaves a number of gaps that Kronos fills by generating the images denoted by **O**.

Samples	|	|	|	|	|	|	|	|	|
Input	X	X	X	X	X				
Input Stretched	X		X		X		X		X
Output	X	O	X	O	X	O	X	O	X

The output sequence is **number_of_intervals** / **speed** + **1** frames long. For example, if speed is **0.5** and the input frame is in the range [**1,10**], the output sequence length will be (**10 - 1**) / **0.5** + **1** = **19** frames long.

Example Input Speed Calculation

Let's say that the input sequence is [**t_start, t_end**]. Retiming the input sequence in the **Timing** > **Input Speed** mode varies the speed at which time flows in the input sequence, so that a point in time **tau**, in the input time frame, maps to the output time **t_o** as follows:

$$t_o(\tau) = t_{start} + \int_{t_{start}}^{\tau} \frac{1}{s(x)} dx$$

The start frame **t_start** is the same for both the input and output sequence. If the speed parameter **s** is not animated, and therefore **s** is a constant, we get:

$$t_o(\tau) = t_{start} + \frac{1}{s} \int_{t_{start}}^{\tau} dx = t_{start} + \frac{1}{s} \cdot (\tau - t_{start})$$

So, if we want to know where the last frame of the sequence maps to, we use:

$$t_o(t_{end}) = t_{start} + \frac{1}{s} \cdot (t_{end} - t_{start})$$

Let's look at an example. If the input sequence is in the range [**1, 10**] and speed is **0.5**, the output sequence has a range [**1, 19**], as follows:

t_o_start = **1**

t_o_end = **1** + **2** * (**10** - **1**) = **19**

Quick Start

Here's a quick overview of the workflow:

1. Create a node tree with Kronos. See Connecting Kronos.
2. Retime your footage. See Retiming a Sequence.
3. If you're not happy with the results, adjust the vector generation parameters. See Refining the Results.
4. If necessary, add motion blur to the retimed footage. See Adding Motion Blur.

Tip: You can check **Use GPU if available** to have the node run on the graphics processing unit (GPU) rather than the central processing unit (CPU). For more information on GPU support, see the Release Notes for your version of Nuke.

You can select the GPU to use in the Preferences. Press **Shift**+**S** to open the **Preferences** dialog, make sure you're viewing the **Preferences** > **Performance** > **Hardware** tab, and set **default blink device** to the device you want to use. You must restart Nuke for the change to take effect.

If you are using a render license of Nuke, you need to add **--gpu** on the command line.

Connecting Kronos

To connect Kronos:

1. Select **Time** > **Kronos** to insert a Kronos node after the sequence you want to retime.

The **Input Range** is set automatically by the source clip when you first create the node. After that, it is only updated if you click **Reset**.

2. If you intend to use **Output Speed** or **Frame** timing, attach a Viewer to the output of the Kronos node,

   OR

   If you intend to use **Input Speed** timing, attach a Viewer before the Kronos node.
3. Enable **Use GPU if available** to render output on the **Local GPU** specified, if available, rather than the CPU.



Note: For more information on GPU support, see the Release Notes for your version of Nuke.

4. Select the channels you want to apply the retime to using the **channels** dropdown. Kronos retimes **all** channels by default.
5. If your sequence is composed of a foreground object moving over a background, the motion estimation is likely to get confused at the edge between the two. To fix this, add a matte of the foreground region to the **Matte** input. Then, use **Matte Channel** in the Kronos properties to select which component of the matte to use.

   This forces the motion of the foreground to be calculated separately to the motion of the background and so should produce fewer artifacts in the retimed sequence.
6. If the motion in your input sequence has been estimated before (for example, using VectorGenerator or third-party software), you can supply the motion vectors to the Background Vectors (**BgVecs**) and Foreground Vectors (**FgVecs**) inputs to save processing time.



Note: The **BgVecs** input appears as an arrowhead on the side of the node.

   If you have separate vectors for the background and foreground, you should connect them to the appropriate inputs and supply the matte that was used to generate them to the **Matte** input. If you have a single set of vectors, you should connect it to the **FgVecs** input.
7. You can set **Output** in the Kronos controls to:
   - **Result** - displays both the foreground and background retimed footage.
   - **Matte** - displays the retimed **Matte** input.
   - **Foreground** - displays the retimed foreground. The background regions outside the matte input may show garbage.
   - **Background** - displays the retimed background. The foreground regions inside the matte input may show garbage.
8. If your **Source** sequence is very noisy and interfering with the motion estimation, you can supply a smoothed version of the sequence in the **MoSrc** input. When a **MoSrc** input is connected, motion vectors are calculated from that sequence and applied to the **Source** sequence.

9. Proceed to Retiming a Sequence.

Retiming a Sequence

Kronos allows you to retime a sequence by speed or by frame. Speed retimes describe the retime by a percentage of full speed, either against the output or input frames. For example, the default setting **Output Speed** = **0.5** is equal to a 50% retime on the output frames. Frame retimes describe the retiming in terms of 'at frame 100 in the output clip, I want to see frame 50 of the source clip'.



Note: **Input Range** in the Kronos properties defines which frames are used for the retime. When you first create the node, the range is automatically set to the frame range of the **Source** clip. If you change the **Source** clip later, you need to click the **Reset** button to make sure **Input Range** matches the current input.

Retiming a Sequence Using Output Speed

To apply a constant retime:

1. Ensure that the Viewer is connected to the Kronos node and **Timing** is set to **Output Speed**.
2. Select the required interpolation **Method** using the dropdown:
   - **Frame** - the nearest original frame is displayed.
   - **Blend** - a mix between two frames is used for the in-between frame. This is quick to render and is useful when tweaking the timing on a curve before setting the **Method** to **Motion**.
   - **Motion** - vector interpolation is used to calculate the in-between frame.
3. Enter a value for the **Output Speed** control. Values below 1 slow down the clip and values above 1 speed up the clip. The default value is **0.5**, creating a half-speed slow down. Quarter-speed would be **0.25**.
4. Play through the sequence to view the retime.
5. If you're not happy with the results of the retime, try Refining the Results.
6. If you'd like to add motion blur to your retimed footage, proceed to Adding Motion Blur.

To use varying retimes:

Tip: If you want to retime at a certain event in the clip, you may find that using **Input Speed** is easier to predict where keyframes should be placed.

1. Ensure that the Viewer is connected to the Kronos node and **Timing** is set to **Output Speed**.
2. Select the required interpolation **Method** using the dropdown:
   - **Frame** - the nearest original frame is displayed.
   - **Blend** - a mix between two frames is used for the in-between frame. This is quick to render and is useful when tweaking the timing on a curve before setting the **Method** to **Motion**.
   - **Motion** - vector interpolation is used to calculate the in-between frame.
3. Move the playhead to the frame you want to start retiming and enter the required **Output Speed**.
4. Click the Animation button and select **Set Key**.
5. Move the playhead to the frame you want to change the retime value and enter the required **Output Speed**.

   A new keyframe is added automatically.
6. Once you've added all the required keyframes, click the **Curve Editor** tab to view your keyframes in context of the sequence.
7. If you're not happy with the results of the retime, try Refining the Results.
8. If you'd like to add motion blur to your retimed footage, proceed to Adding Motion Blur.

Retiming a Sequence Using Input Speed

To apply a constant retime:

1. Ensure that the Viewer is connected to the tree before the Kronos node and **Timing** is set to **Input Speed**.
2. Select the required interpolation **Method** using the dropdown:
   - **Frame** - the nearest original frame is displayed.
   - **Blend** - a mix between two frames is used for the in-between frame. This is quick to render and is useful when tweaking the timing on a curve before setting the **Method** to **Motion**.
   - **Motion** - vector interpolation is used to calculate the in-between frame.
3. Enter a value for the **Input Speed** control. Values below 1 slow down the clip and values above 1 speed up the clip. The default value is **0.5**, creating a half-speed slow down. Quarter-speed would be **0.25**.
4. Connect the Viewer to the Kronos node, and then play through the sequence to view the retime.
5. If you're not happy with the results of the retime, try Refining the Results.
6. If you'd like to add motion blur to your retimed footage, proceed to Adding Motion Blur.

To use varying retimes:

1. Ensure that the Viewer is connected to the tree before the Kronos node and **Timing** is set to **Input Speed**.
2. Select the required interpolation **Method** using the dropdown:
   - **Frame** - the nearest original frame is displayed.
   - **Blend** - a mix between two frames is used for the in-between frame. This is quick to render and is useful when tweaking the timing on a curve before setting the **Method** to **Motion**.
   - **Motion** - vector interpolation is used to calculate the in-between frame.
3. Move the playhead to the frame you want to start retiming and enter the required **Input Speed**.
4. Click the Animation button and select **Set Key**.
5. Move the playhead to the frame you want to change the retime value and enter the required **Input Speed**.

   A new keyframe is added automatically.
6. Once you've added all the required keyframes, connect the Viewer to the Kronos node, and then click the **Curve Editor** tab to view your keyframes in context of the output sequence.
7. If you're not happy with the results of the retime, try Refining the Results.
8. If you'd like to add motion blur to your retimed footage, proceed to Adding Motion Blur.

Retiming a Sequence Using Frame

To apply a constant retime:

1. Ensure that the Viewer is connected to the Kronos node and **Timing** is set to **Frame**.
2. Select the required interpolation **Method** using the dropdown:
   - **Frame** - the nearest original frame is displayed.
   - **Blend** - a mix between two frames is used for the in-between frame. This is quick to render and is useful when tweaking the timing on a curve before setting the **Method** to **Motion**.
   - **Motion** - vector interpolation is used to calculate the in-between frame.
3. Go to a frame in the timeline, and set **Frame** to the input frame you want to appear at that output position. For a constant retime, you'll need two key frames to retime the clip.

   For example, to slow down a 50 frame clip by half, you can set **Frame** to 1 at frame 1, and to 50 at frame 100. To do a four times slow down, set **Frame** to 1 at frame 1, and to 25 at frame 100.
4. Play through the sequence to view the retime.
5. If you're not happy with the results of the retime, try Refining the Results.
6. If you'd like to add motion blur to your retimed footage, proceed to Adding Motion Blur.

To use varying retimes:

1. Ensure that the Viewer is connected to the Kronos node and **Timing** is set to **Frame**.
2. Select the required interpolation **Method** using the dropdown:

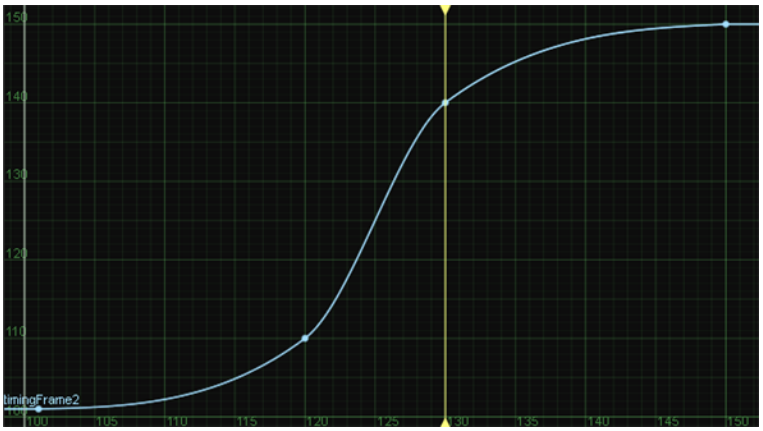
- **Frame** - the nearest original frame is displayed.
- **Blend** - a mix between two frames is used for the in-between frame. This is quick to render and is useful when tweaking the timing on a curve before setting the **Method** to **Motion**.
- **Motion** - vector interpolation is used to calculate the in-between frame.

3. Go to a frame in the timeline, and set **Frame** to the input frame you want to appear at that output position. You'll need atleast two key frames to retime the clip.

   For example, to speed up a 50 frame clip and then return to normal speed at the end:

   - Set **Frame** to 1 at frame 1,
   - Set **Frame** to 10 at frame 20,
   - Set **Frame** to 40 at frame 30,
   - Set **Frame** to 50 at frame 50.

   These keyframes produce the following curve in the Curve Editor.

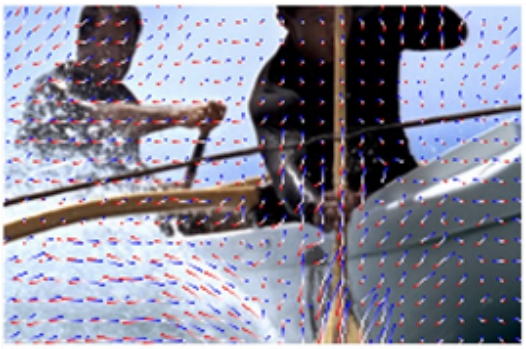


4. If you're not happy with the results of the retime, try Refining the Results.
5. If you'd like to add motion blur to your retimed footage, proceed to Adding Motion Blur.

Refining the Results

To refine the results:

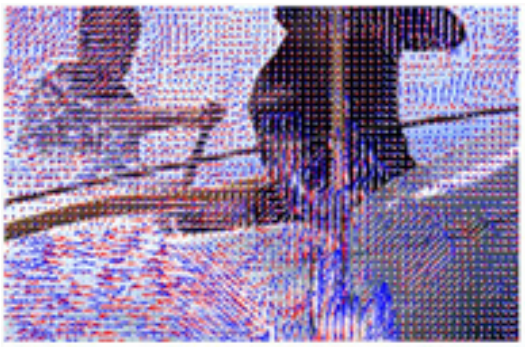
1. To have the motion vectors displayed in the Viewer, expand the **Advanced** control and enable **Overlay Vectors**. Forward motion vectors are drawn in red, and backward motion vectors in blue.



Overlay Vectors enabled.

Note: Motion vectors displayed in the Viewer are added to your output if you don't turn off the overlay before rendering.

2. To set the spacing between motion vectors displayed on the Viewer, adjust **Vector Spacing**. The default value of 20 means every 20th vector is drawn. Note that **Vector Spacing** only affects the Viewer overlay rather than the retimed result.



A **Vector Spacing** value of 4.



A **Vector Spacing** value of 40.

3. Adjust **Vector Detail** to vary the density of the vector field. The larger vector detail is, the greater the processing time, but the more detailed the vectors should be. A value of 1 generates a vector at each pixel. A value of 0.5 generates a vector at every other pixel. For some sequences, a high vector detail near 1 generates too much unwanted local motion detail, and often a low value is more appropriate.

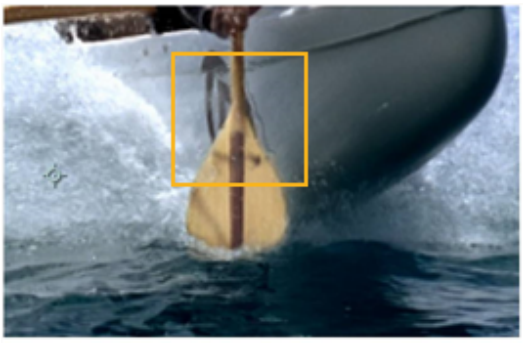


Areas of unwanted local motion detail.

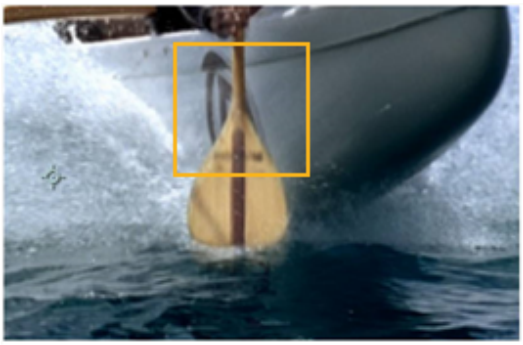


Lower **Vector Detail** is more appropriate in this case.

4. Vector fields usually have two important qualities: they should accurately match similar pixels in one image to another and they should be smooth rather than noisy. Often, it is necessary to trade one of these qualities off against the other. A high **Strength** misses lots of local detail, but is less likely to provide you with the odd spurious vector. A low **Strength** concentrates on detail matching, even if the resulting field is jagged. The default value of 1.5 should work well for most sequences.



Jagging as a result of a low **Strength** value.



Minimal jagging as a result of a high **Strength** value.

5. Set the type of resampling applied when retiming:
   - **Bilinear** - the default filter. Faster to process, but can produce poor results at higher zoom levels. You can use **Bilinear** to preview a retime before using one of the other resampling types to produce your output.
   - **Lanczos4** and **Lanczos6** - these filters are good for scaling down, and provide some image sharpening, but take longer to process.
6. If the overall brightness in your **Source** footage changes between frames, enable **Flicker Compensation**. This allows Kronos to take into account variations in luminance and overall flickering, which could otherwise cause problems with your output.

   Examples of variable luminance include highlights on metal surfaces, like vehicle bodies, or bodies of water that reflect light in unpredictable ways.

   Note that using **Flicker Compensation** increases rendering time.

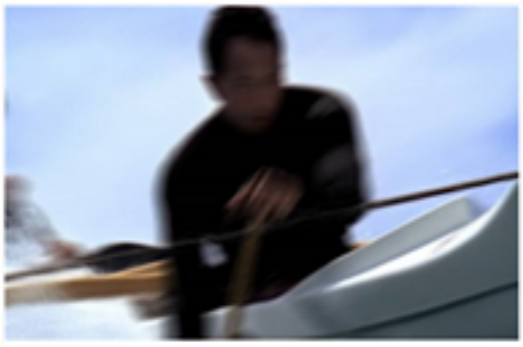
7. In order to reduce processing time, much of the motion estimation is done on luminance only - that is, using monochrome images. In most cases this is perfectly acceptable, but the parameters in the **Tolerances** group allow you to concentrate on a particular feature in an image by adding bias to individual colours. You may, for example, wish to increase **Weight Red** to allow the algorithm to concentrate on getting the motion of a primarily red object correct, at the cost of the rest of the objects in a shot.
8. If you'd like to add motion blur to your retimed footage, proceed to Adding Motion Blur.

Adding Motion Blur

To add motion blur:

1. In the **Shutter** controls, select a suitable **Shutter Time**, depending on the amount of blur you wish to add. This sets the equivalent shutter time of the retimed sequence. For example, a shutter time of 0.5 is equivalent to a 180 degree mechanical shutter, so at 24 frames per second the exposure time is 1/48th of a second.

   The larger the value, the more motion blur is produced.



A low **Shutter Time** value.



A high **Shutter Time**.

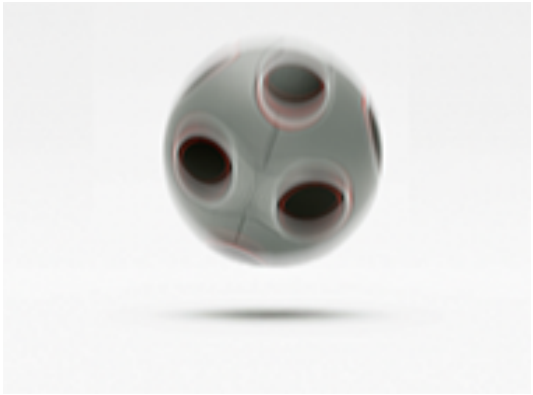
   Alternatively, you can enable **Automatic Shutter Time** to have Kronos automatically calculate the shutter time throughout the sequence. Note that this only produces motion blur when the retimed speed is greater than the original speed.
2. Render the sequence to see the motion blurred result.
3. If you can see that the motion blur has been created from a few discrete images, try increasing **Shutter Samples**. This results in more in-between images being used to generate the motion blur and so produces a smoother blur but takes longer to render.

MotionBlur

MotionBlur adds realistic motion blur to a sequence.



Before MotionBlur.



After MotionBlur.

It uses the same techniques and technology as the motion blur found in Kronos, but presents the controls in a less complex, more user friendly way. However, if you need precise control over the motion vectors used for adding blur, or a large temporal range (that is, a very high shutter time), you should use Kronos.

Quick Start

Here's a quick overview of the workflow:

1. Create a node tree with MotionBlur. See Connecting MotionBlur.
2. Adjust the amount and quality of the motion blur produced. See Adjusting MotionBlur Controls.

Tip: You can check **Use GPU if available** to have the node run on the graphics processing unit (GPU) rather than the central processing unit (CPU). For more information on GPU support, see the Release Notes for your version of Nuke.

You can select the GPU to use in the Preferences. Press **Shift**+**S** to open the **Preferences** dialog, make sure you're viewing the **Preferences** > **Performance** > **Hardware** tab, and set **default blink device** to the device you want to use. You must restart Nuke for the change to take effect.

If you are using a render license of Nuke, you need to add **--gpu** on the command line.

Connecting MotionBlur

To connect MotionBlur:

1. Select **Filter** > **MotionBlur** to insert a MotionBlur node after the clip you want to add motion blur to.
2. If your sequence is composed of a foreground object moving over a background, the motion estimation is likely to get confused at the edge between the two. To fix this, add a matte of the foreground region to the **Matte** input. Then, use **Matte Channel** in the MotionBlur controls to select which component of the matte to use.

   This forces the motion of the foreground to be calculated separately to the motion of the background and so should produce fewer artifacts in the motion blur.
3. If the motion in your input sequence has been estimated before (for example, using VectorGenerator or third-party software), you can supply the motion vectors to the Background Vectors (**BgVecs**) and Foreground Vectors (**FgVecs**) inputs to save processing time.

   If you have separate vectors for the background and foreground, you should connect them to the appropriate inputs and supply the matte that was used to generate them to the **Matte** input. If you have a single set of vectors, you should connect it to the **FgVecs** input.
4. Attach a Viewer to the output of the MotionBlur node.
5. Proceed to Adjusting MotionBlur Controls below.

Adjusting MotionBlur Controls

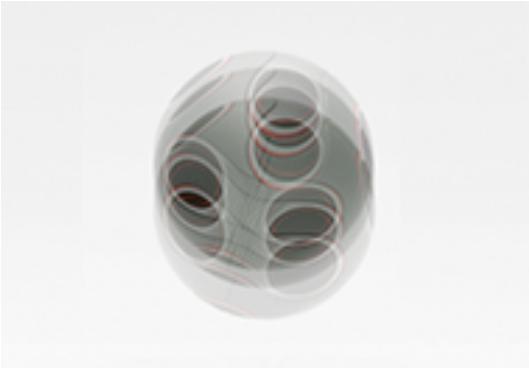
To adjust MotionBlur controls:

1. Select a suitable **Shutter Time**, depending on the amount of blur you wish to add. This sets the equivalent shutter time of the retimed sequence. For example, a shutter time of 0.5 is equivalent to a 180 degree mechanical shutter, so at 24 frames per second the exposure time is 1/48th of a second.

   The larger the value, the more motion blur is produced.



A low **Shutter Time**.



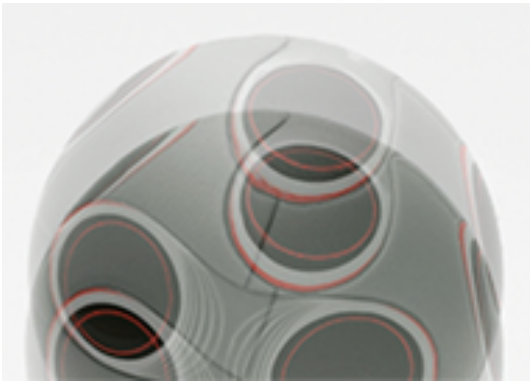
A high **Shutter Time**.

2. Set the method of calculating motion estimation vectors:
   - **Local** - uses local block matching to estimate motion vectors. This method is faster to process, but can lead to artifacts in the output.
   - **Regularized** - uses semi-global motion estimation to produce more consistent vectors between regions.

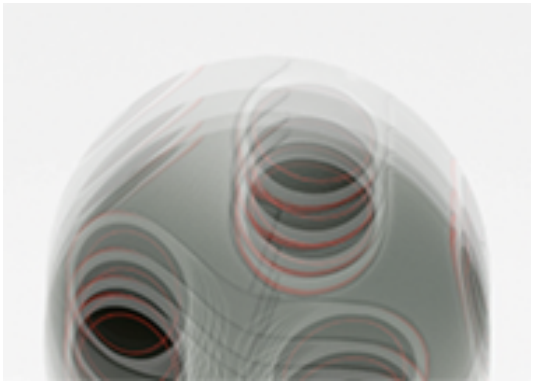
> **Note:** Scripts loaded from previous versions of Nuke default to **Local** motion estimation for backward compatibility. Adding a new MotionBlur node to the Node Graph defaults the **Method** to **Regularized** motion estimation.

3. Use **Vector Detail** to alter the density of the calculated motion vector field. A value of 1 generates a vector at each pixel, whereas a value of 0.5 generates a vector at every other pixel.

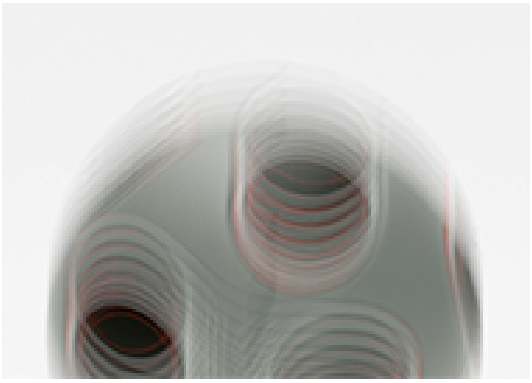
   Bear in mind that higher **Vector Detail** values require longer to process, but can pick up finer movement and enhance your results in some cases.
4. Set the type of re-sampling applied when retiming:
   - **Bilinear** - the default filter. Faster to process, but can produce poor results at higher zoom levels. You can use **Bilinear** to preview a motion blur before using one of the other re-sampling types to produce your output.
   - **Lanczos4** and **Lanczos6** - these filters are good for scaling down, and provide some image sharpening, but take longer to process.
5. Render the sequence to see the motion blurred result.
6. If you can see that the motion blur has been created from a few discrete images, try increasing **Shutter Samples**. This results in more in-between images being used to generate the motion blur and so results in a smoother blur.



Shutter Samples = 2



Shutter Samples = 4



Shutter Samples = 8



Shutter Samples = 30

Working with Lens Distortion

Lens distortion can make compositing work more difficult because the characteristics of the lens can cause areas of the footage to warp, making correct placement of assets problematic. One way around this problem is to undistort your plate before starting work on a comp and then redistort and merge only the VFX work to produce the final comp.



The original plate



The undistorted plate and comp



The comp after redistortion

It's a good idea to shoot grids with the same lenses you use to create your footage as this makes estimating distortion easier. If you have the lens properties, such as **Focal Length** and **Sensor Size**, and a grid shot with the same lens you'll find this process much less painful.

Nuke's LensDistortion node allows you to undistort or distort an image according to several radial distortion models that ship with Nuke or a custom model you define yourself. You can calculate the warp for use on the input image or output the warp to an STMap for use elsewhere in the script.

Quick Start

Here's a quick overview of the workflow:

1. Read in an input sequence, connect it to a LensDistortion node (**Transform** > **LensDistortion**), and connect the output to a Viewer.
2. If you have a grid shot using the same lens as your sequence, use **Grid Detection** to calculate the distortion, and then apply the warp to your sequence. See Estimating Lens Distortion Using a Grid and Removing Lens Distortion from an Image for more information.
3. If you do not have a grid, or **Grid Detection** did not work, draw features and lines manually to estimate distortion and undistort the sequence. See Estimating and Removing Lens Distortion Using Lines for more information.

Note: If you want, you can calculate the lens distortion on one image and apply that distortion to another image with the help of an STMap node. For more information, see Working with STMaps.

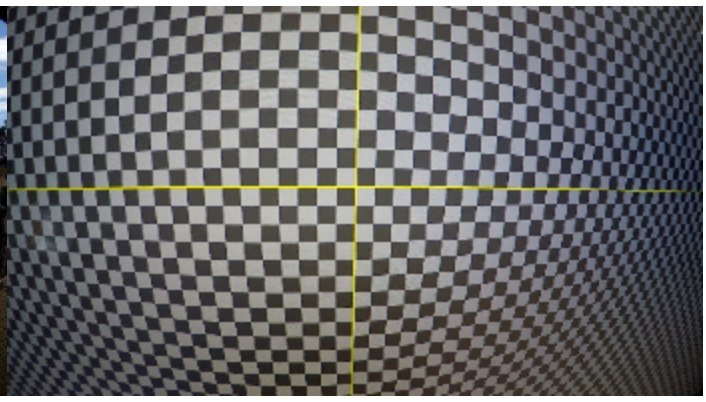
4. Add your VFX work to the undistorted sequence.
5. Redistort the VFX work and merge it over the original plate. See Applying Lens Distortion to an Image for more information.

Estimating Lens Distortion Using a Grid

Grid analysis estimates the distortion from a checkerboard or thin line grid. As a general rule, if you have a grid you can use to calculate your lens distortion, you should use grid analysis. **Grids** are constructed from **Features** and **Links**, which are calculated automatically.



The original distorted shot



Associated grid shot

To estimate distortion using a grid:

1. Read in your grid and connect a LensDistortion node, followed by a Viewer.
2. Set the **LensDistortion** > **Lens Type** and **Advanced** > **Projection Model** using the dropdowns in the LensDistortion **Properties** panel. In this example, the lens is **Spherical** and **None (Rectilinear)**, but there are a number of other presets included.
3. If you're using a fisheye lens, you need to 'defish' the lens. You can do this by selecting the correct **Advanced** > **Projection Model** setting, such as **Fisheye Equisolid**, entering the **Focal Length** and **Sensor Size**.

> **Tip:** If you don't know the **Focal Length**, set the Output **Mode** to **Undistort** and adjust the **Focal Length** until the curved lines in the image appear approximately straight. Don't forget to switch back to **Mode** > **STMap** before continuing.

4. Set the **Model Preset** to use for the estimation. Nuke ships with several models for use with CaraVR and 3DEqualizer, as well as a **NukeX Classic** model.

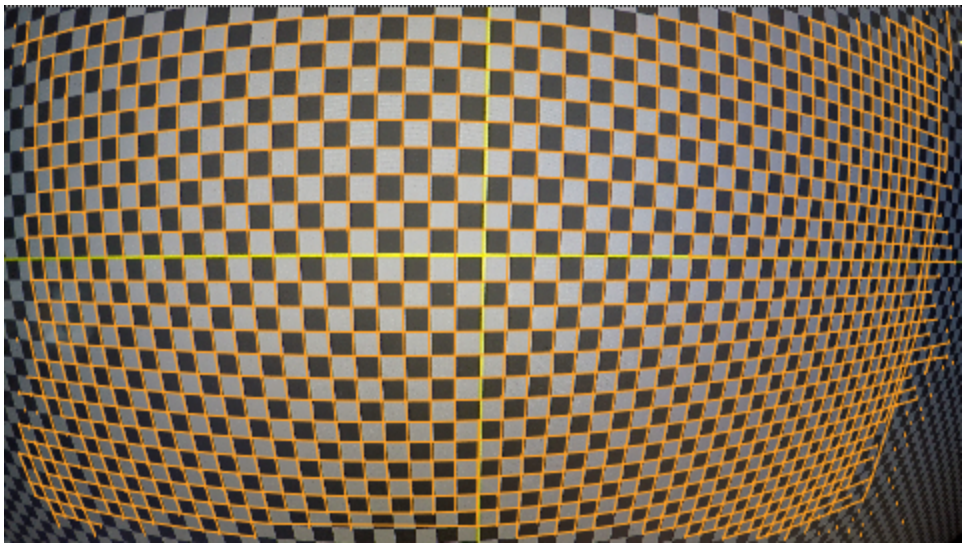
   The model you choose sets appropriate **Advanced** > **Distortion** controls and populates the read only **Equation** fields showing the math involved in the estimation.
5. The LensDistortion node adds a keyframe on the current frame by default when you click **Detect**, but you can add more using the  button or use the **Grid Detect** > **Every N Frames** option on the **Analysis** tab, instead of **Current Frame**.

   Adding keyframes can produce a better result for long sequences, but it takes longer to calculate the result.
6. On the **Analysis** tab, click **Detect** to start the grid calibration.
   By default, Nuke looks for **Features** on the grid and then creates **Links** between those features to create the distortion grid.

   You can enable **Preview** to view the features that the LensDistortion node is likely to find when you click the **Detect** button.

> **Tip:** For difficult shots, you can make adjustments between **Features** and **Links** detection to improve the results. See Adjusting Grid Detection Parameters for more information.

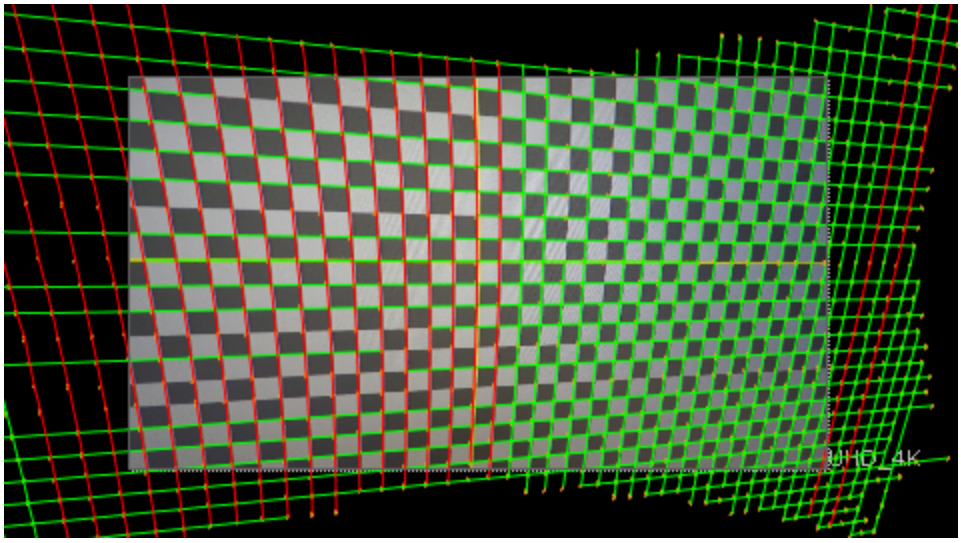
A grid overlay is displayed when the detection step is complete.



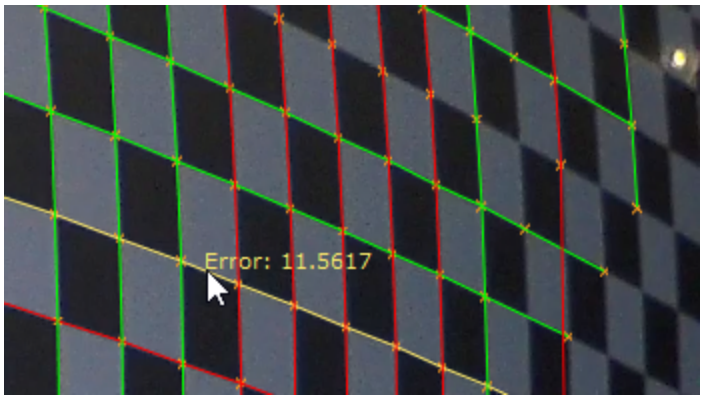
Tip: If the grid does not cover most of the image, try increasing the **Number of Features** value on the **Analysis** tab and clicking **Detect** again.

7. Click **Solve** to estimate the distortion using the grid.

   Nuke estimates the distortion and displays a new overlay and overall **Solve Error**. The default Output **Mode** is **Undistort**, so the inverse distortion is applied to the image automatically.



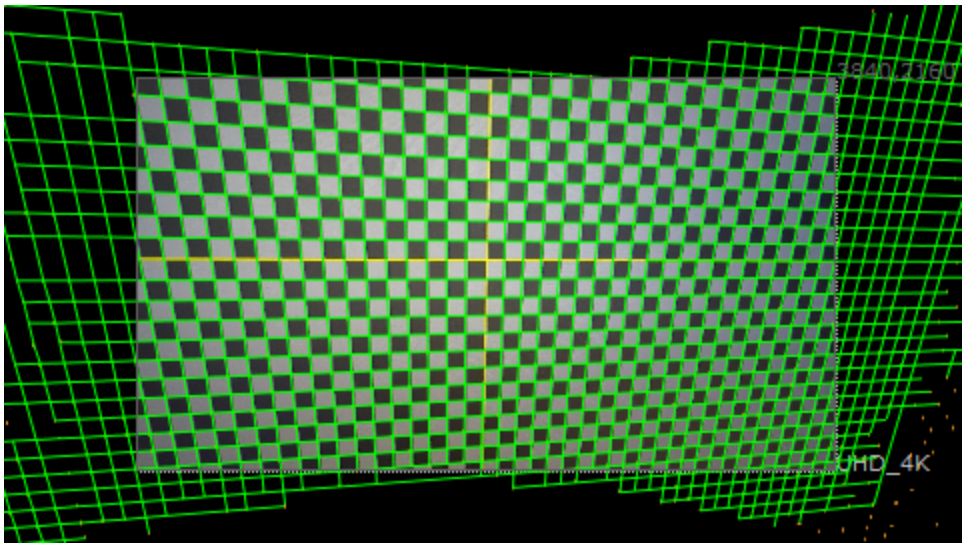
   Green lines represent links that fall within the **Detection Threshold** value and red lines those that fall outside the threshold. You can hover over links to display their solve error.



8. Clicking **Solve** again can improve the result in some cases. You can also refine the detection and linking before solving again using the controls on the **Analysis** tab. See Adjusting Grid Detection Parameters for more information.

Note: You can also output the distortion as an STMap for use in other images. STMaps contain the pre-computed warp data in the **motion** channel, allowing you apply the warp quickly and easily. See Working with STMaps for more information.

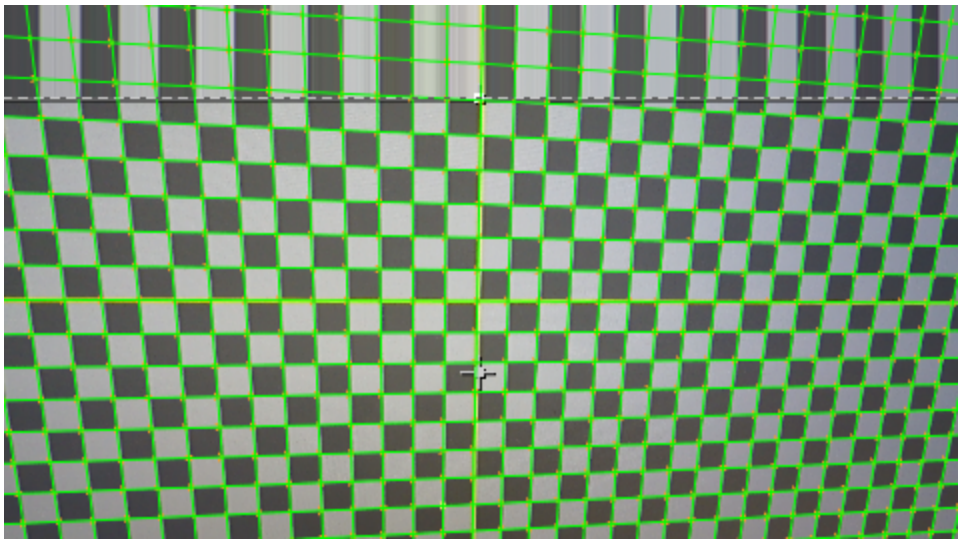
Nuke takes the estimated distortion and uses the result to 'straighten' the feature links.



The bounding box is increased beyond the format size to include all the image data from the input image.

9. You can control the bounding box manually using the **BBox** and **Output Format** controls on the **LensDistortion** tab.

   Any areas of the image where there is no data are overscanned, meaning the last pixel available is duplicated to the edge of the format. The area at the top of the image shows overscan.



You can also bring more image data into the format bounds using the **BBox** and **Output Format** controls.

10. Proceed to Removing Lens Distortion from an Image or Adjusting Grid Detection Parameters to fine tune your grid.

Adjusting Grid Detection Parameters

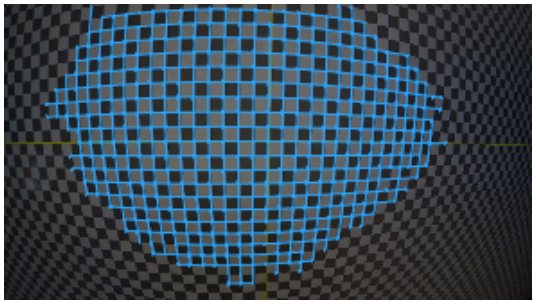
The default settings on the LensDistortion node's **Analysis** tab are suitable for a wide range of input images, but in some cases, you'll need to make some adjustments in order to achieve the required coverage of **Features** and viable **Links**. You'll need to click **Detect** again after making changes to the **Grid Detection** controls to update the results.

Tip: You can enable **Preview** on the **Analysis** tab to display a blue overlay of potential feature and link matches. The preview updates in real-time as you adjust the analysis **Settings**, so you only need to click **Detect** once.

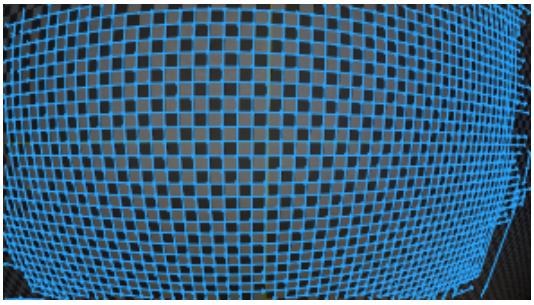
Feature Detection

The **Features** in a solve form the building blocks for accurate **Links**, so it's important to get the maximum coverage possible, while still maintaining quality.

Increasing the maximum **Number of Features** can improve coverage relatively easily. You can use the **Detection Threshold** control to reject bad features automatically. If you enter a low detection threshold value, features are detected evenly on all parts of the image, even if they would otherwise have been rejected.

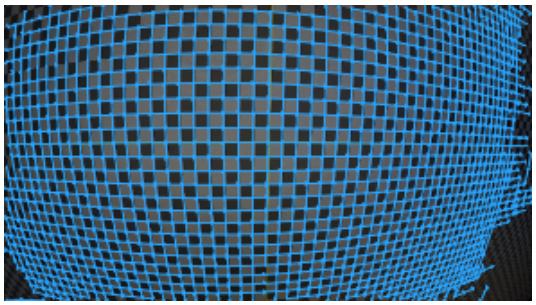


Low Number of Features

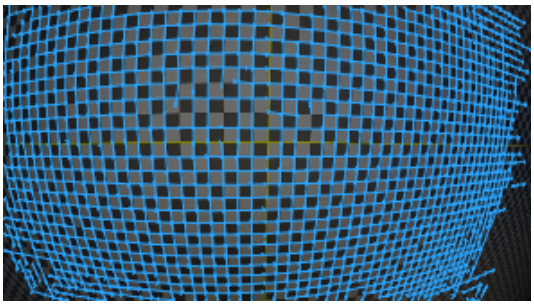


High Number of Features

The **Patch Size** control determines how much bias is placed on detecting features on saddle points. High values force features towards saddle points, which might not be desirable.



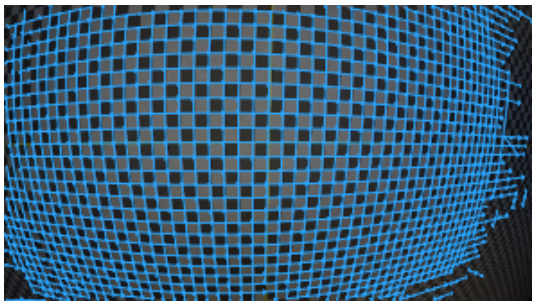
Low Patch Size



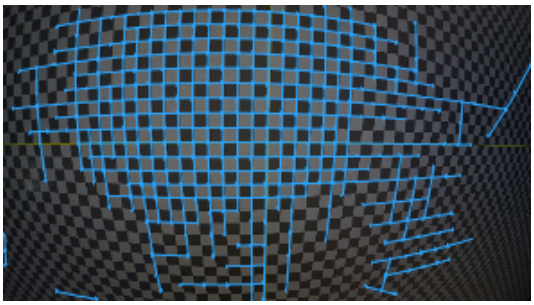
High Patch Size

The **Feature Separation** control sets the distribution of features in relation to each other. High values spread features at even distances over the image. It is important that the features do not cluster together, if this is the case, try increasing this value.

Increasing the separation too far can reduce the number of features detected dramatically, so use caution.



Low Feature Separation



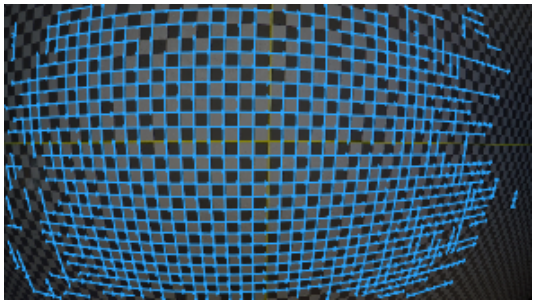
High Feature Separation

Feature Linking

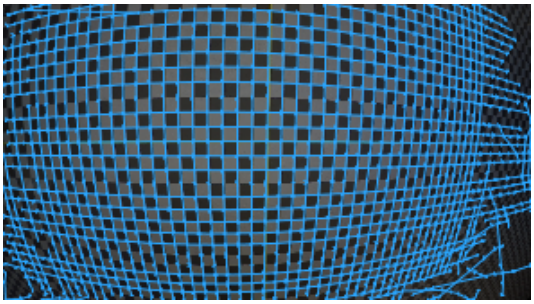
The **Links** in a solve depend on solid **Features** detection, so it's important to make sure your features coverage is good before linking. **Links** provide the curvature data from the lens so that the solve can be as accurate as possible.

The **Angle Threshold** control sets how much offset tolerance is allowed between potential features before they can be linked. Try increasing the **Angle Threshold** if there are missing links between features, but higher values can introduce links between features that may be incorrect.

The image on the right shows some potentially poor links on the right-hand side.

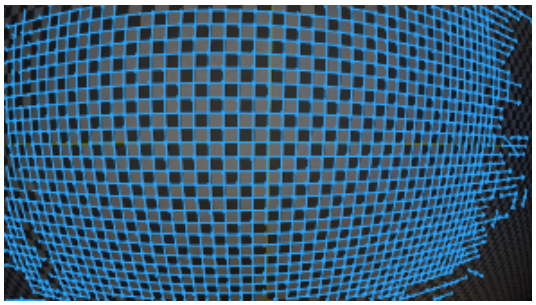


Low Angle Threshold

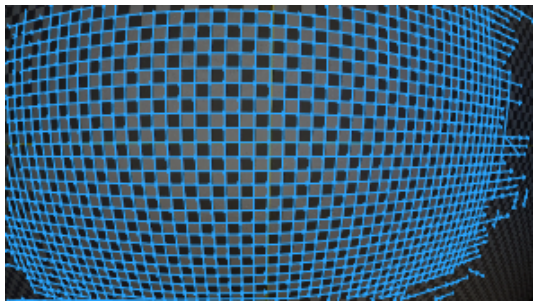


High Angle Threshold

The **Distance Threshold** control determines how far apart **Links** can be before they are merged into a single, averaged link. Try increasing the **Distance Threshold** if features are removed incorrectly.

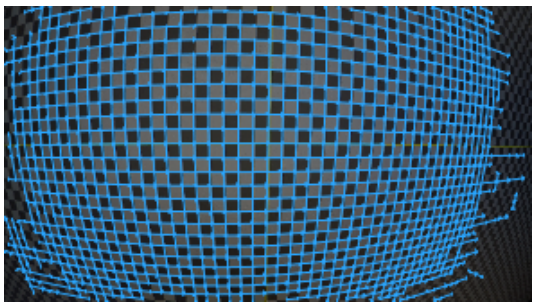


Low Distance Threshold

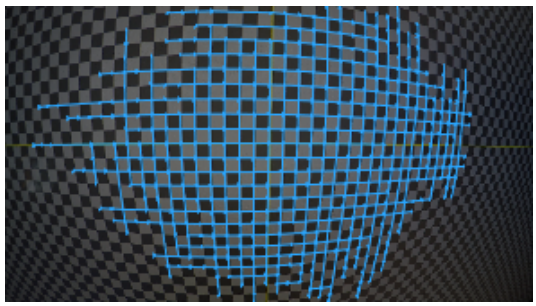


High Distance Threshold

The **Peak Threshold** control determines how much directional tolerance is allowed between potential features before they can be linked. Reducing the **Peak Threshold** can increase viable links with low contrast lens reference grids.



Low Peak Threshold



High Peak Threshold

Proceed to Removing Lens Distortion from an Image when you're satisfied with your grid.

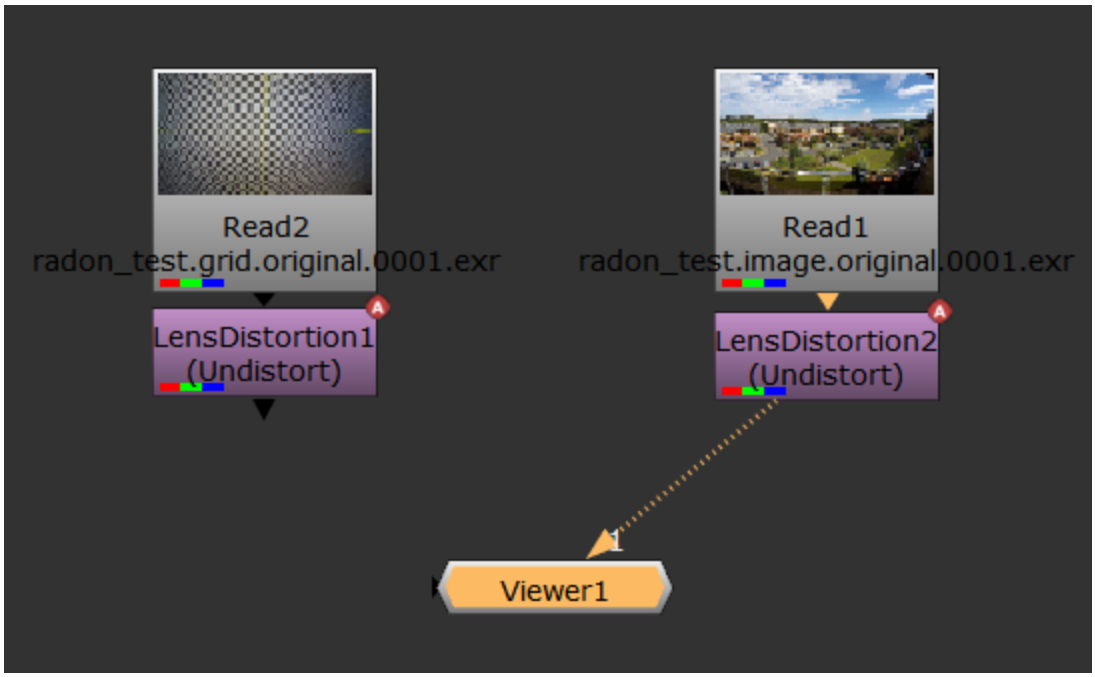
Removing Lens Distortion from an Image

After estimating the lens distortion from a reference grid, you can use the warp to remove the distortion from your sequence before performing comp work. The easiest way to do this is to copy and paste the LensDistortion node that was used to generate the warp and connect it to your sequence.

Note: You can also use STMaps to remove and apply distortion. See Working with STMaps for more information.

1. Read in the sequence from which the reference grid was taken.

2. Copy and paste the LensDistortion node containing the grid analysis (described in Estimating Lens Distortion Using a Grid), then connect it to the sequence and a Viewer.



The distortion is removed from the sequence in the Viewer.



The original plate

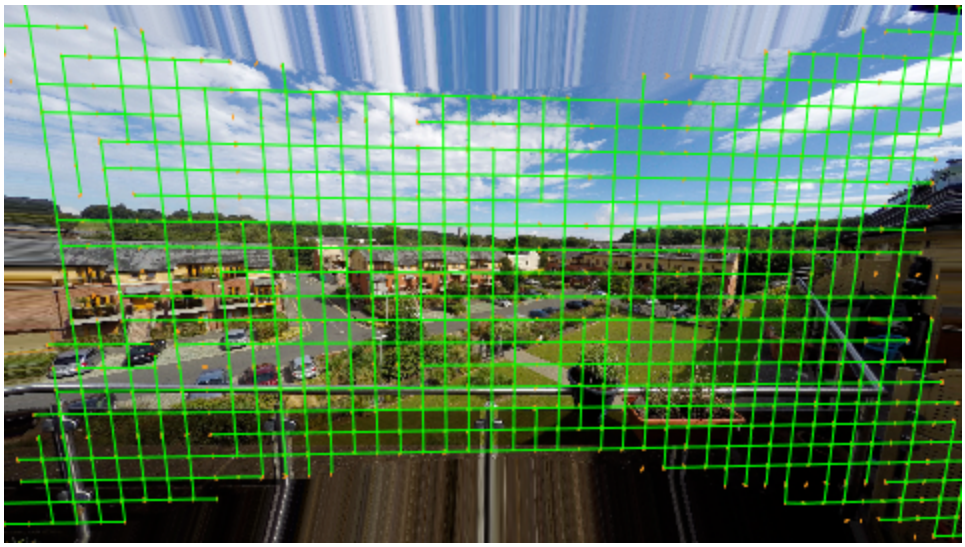


The undistorted plate

You may notice that the lines expand beyond the bounding box of the footage, which can mean you're losing image data.

3. You can control the bounding box manually using the **BBox** and **Output Format** controls on the **LensDistortion** tab. Any areas of the image where there is no data are overscanned, meaning the last pixel available is duplicated to the edge of the format.

   The areas at the top and bottom of the image in the center show overscan.



You can now apply your VFX work before redistorting the plate back into its original state.

Tip: You can also preserve overscan data by increasing the bounding box. The **BBox** > **Manual** control allows you to edit the bounding box manually by enabling **Output Format** > **BBox** control.

4. You can now track, matchmove, composite, and so on in the undistorted space before redistorting the image. See Applying Lens Distortion to an Image for more information.

Estimating and Removing Lens Distortion Using Lines

Line analysis estimates the distortion from lines drawn manually along features in the input that are known to be straight. This can be useful if there is no grid available or if you have a grid for your sequence but the grid analysis failed, for instance due to bad lighting.

The LensDistortion node's **Analysis** tab has a number of tools for selecting and drawing features and lines in the Viewer:

- **Select** - a multi-purpose tool used to select, move, and delete features and lines.
- **Select Feature** - used to select, move, and delete features. You cannot affect lines using this tool.
- **Select Line** - used to select, move, and delete lines. You cannot affect features using this tool.

Add Feature - used to add, select, and move features on existing lines. You cannot affect lines using this tool.

Remove Feature - used to delete features from existing lines. You cannot affect lines using this tool.

Add Line - used to add and delete lines and features.

To estimate lens distortion using lines, do the following:

1. Examine the source footage for distorted lines that are known to be straight. If you have a grid, this is simple, but real life examples include walls, railings, roads, and so on.
2. Select the **Add Line** tool and start to add points along the distorted lines by clicking in the Viewer.



3. You can add features to existing lines using the **Add Feature** tool. This give you finer control of lines in the Viewer.

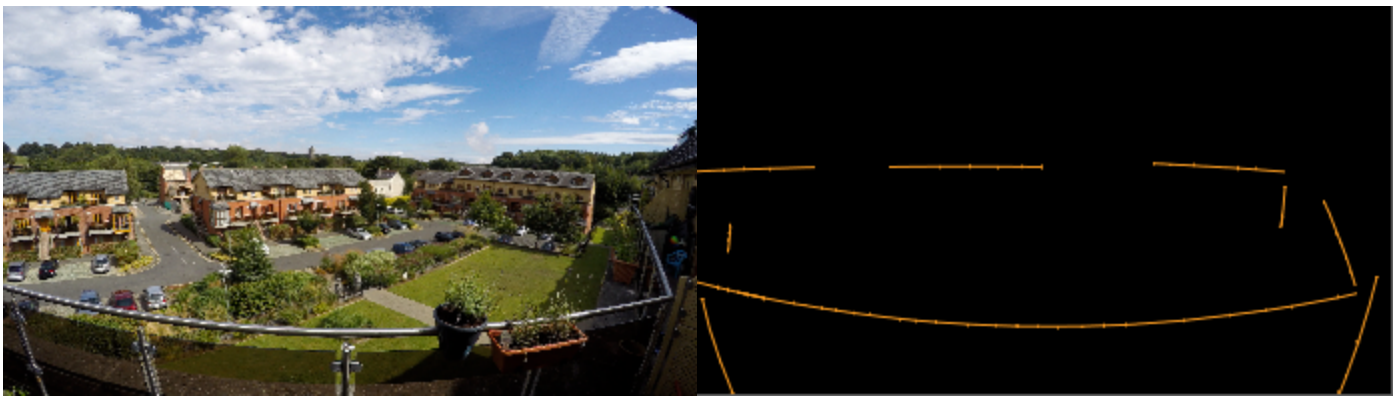
Tip: When you've finished drawing a line, either select a different tool or press **Enter** to finish the current line before starting another.



Tip: You can adjust the position of features and lines using the selection tools in the Viewer.

4. The solve requires at least as many lines as there are **Distortion Parameters** to calculate the distortion. In the case of **NukeX Classic**, two vertical and two horizontal to cover the **Denominator** and **Centre** parameters. Good line drawing practices include:
   - **Drawing lines with three or more features** - lines with only two points are ignored.
   - **Drawing longer lines** - they contain more useful information on the curvature resulting from lens distortion.
   - **Distributing lines evenly** - avoid biasing the solve by covering as much of the image as possible.

   The following example shows a typical set of lines.



The original distorted shot

Example lines in the alpha channel

5. Click **Solve** to calculate the distortion, there's no need to run the **Detect** step with manual features and lines.

Note: The **Solve** button is disabled if there are not enough lines for the detection computation.

Green lines represent links that fall within the **Distortion Threshold** value and red lines those that fall outside the threshold. You can hover over links in **Select** mode to display their solve error.



Note: Clicking **Solve** again can improve the result in some cases.

Tip: You can also output the distortion as an STMap for use in other images. STMaps contain the pre-computed warp data in the **motion** channel, allowing you apply the warp quickly and easily. See Working with STMaps for more information.

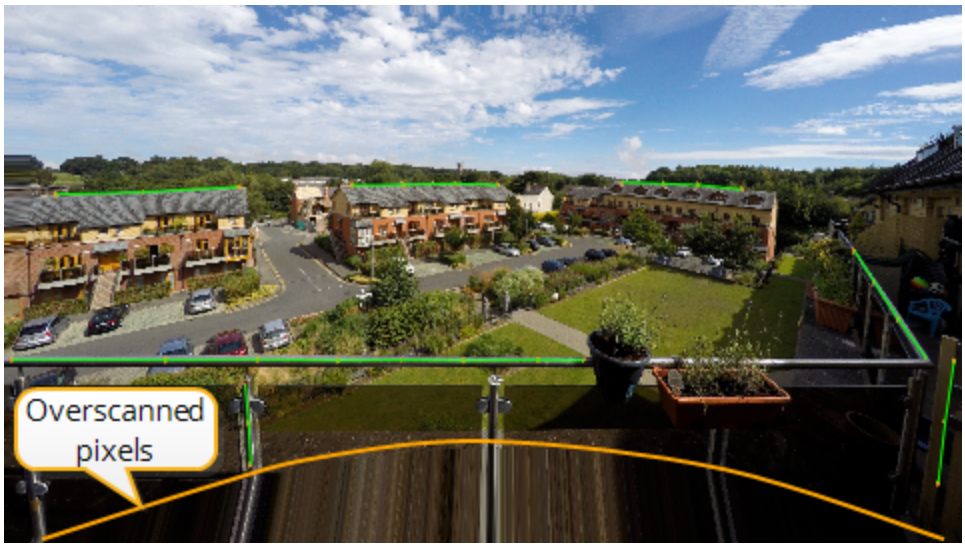
Nuke takes the estimated distortion and uses the result to 'straighten' the feature links.



You may notice that the lines expand beyond the bounding box of the footage, which can mean you're losing image data.

6. Set the **Output Format** control to **Format** and select a larger format to include the grid. For example, if your original grid is UHD_4K, you might reformat to 8K_LatLong.

   Reformatting the feature lines deforms it into a characteristic bow tie shape. Any areas of the image where there is no data are overscanned, meaning the last pixel available is duplicated to the edge of the format. The area at the bottom of the image in the center shows overscan.



7. You can now track, matchmove, composite, and so on in the undistorted space before redistorting the image. See Applying Lens Distortion to an Image for more information.

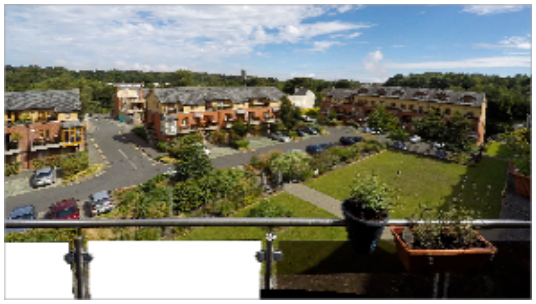
Applying Lens Distortion to an Image

The LensDistortion node's **Redistort** mode allows you to warp your VFX work to match the original plate, which avoids the increased processing time the filter hit on the entire image would incur.

Note: You can also use STMaps to remove and apply distortion. See Working with STMaps for more information.

1. Copy and paste the LensDistortion node containing the grid analysis (described in Estimating Lens Distortion Using a Grid), then connect it downstream of your VFX work.

2. Change the **Output** > **Mode** control to **Redistort** to reverse the warp that was applied during the undistortion of the plate.
3. Merge the redistorted VFX work back over the plate to complete the comp.



The undistorted plate and comp



The comp work after redistortion

Working with STMaps

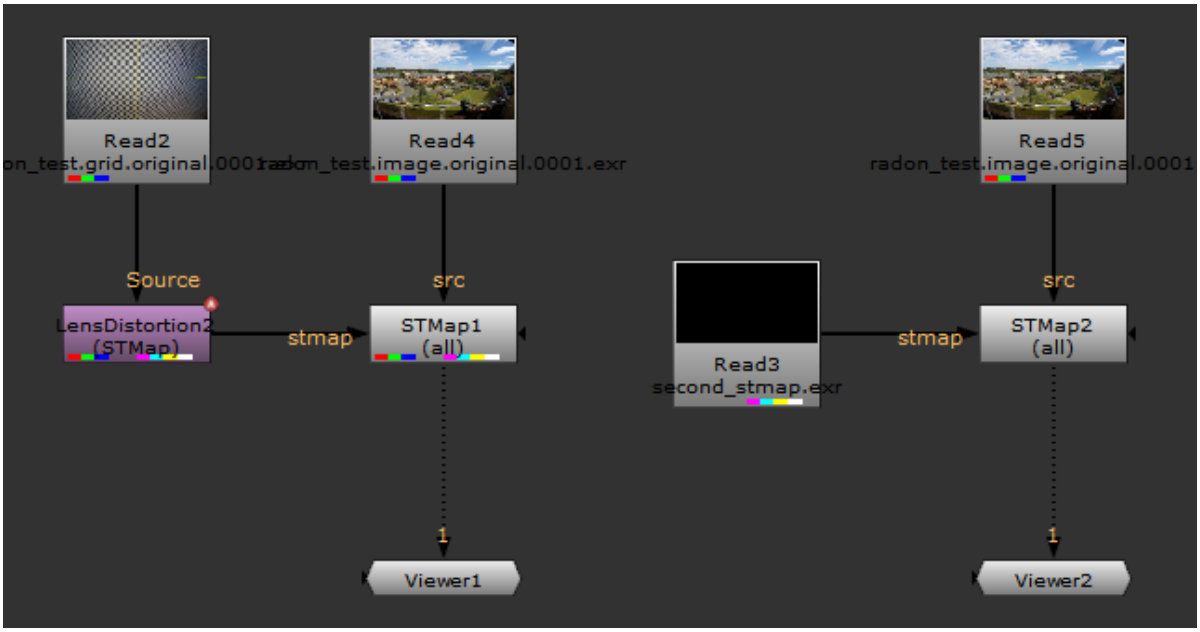
STMaps allow you to warp an image or sequence according to the pre-calculated distortion from a LensDistortion node, stored in the **motion** layer. The **motion** channels represent the absolute pixel positions of an image normalized between 0 and 1, which can then be used on another image to remove or add distortion without the warp estimation step. See Estimating Lens Distortion Using a Grid and Estimating and Removing Lens Distortion Using Lines for more information on warp estimation.

To remove distortion using the STMap node, do the following:

1. Read in your sequence and add an STMap node downstream. Make sure the **src** input is connected to the sequence.
2. Copy and paste the LensDistortion node containing the grid analysis with the **Output** > **Mode** control set to **STMap**,

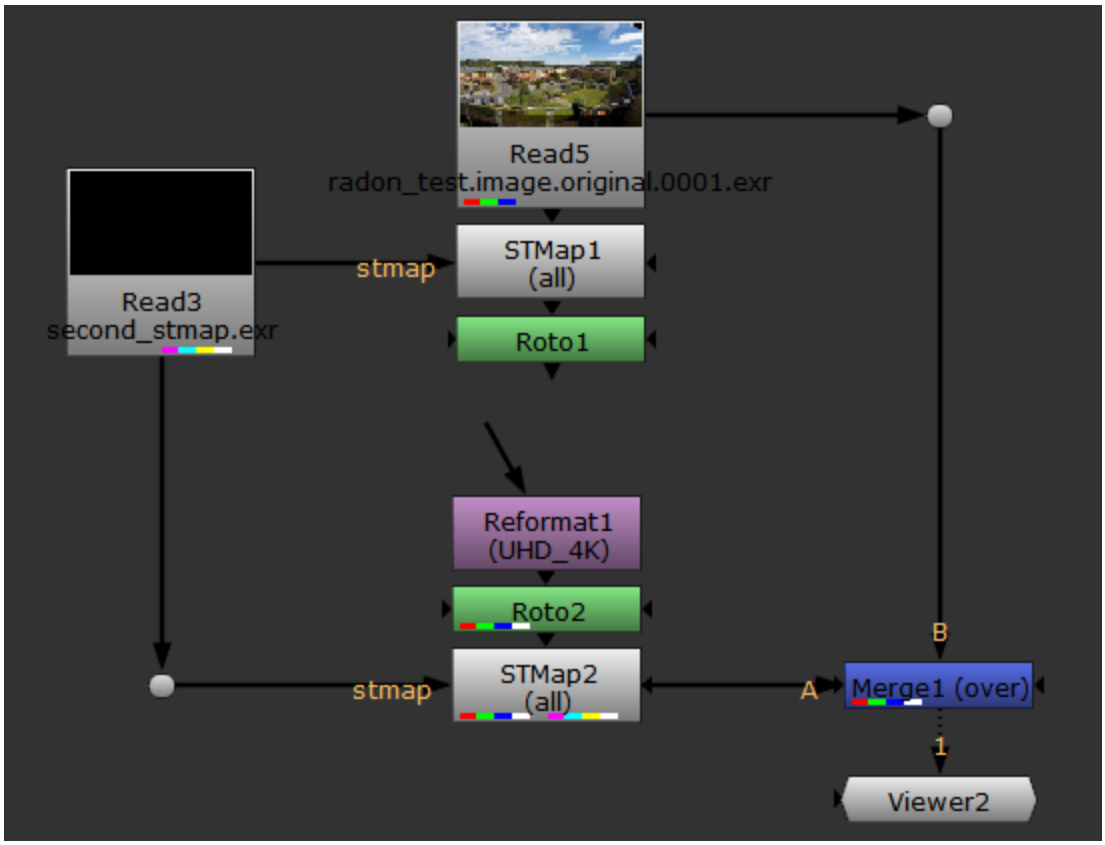
   OR

   Read in the file containing the **motion** layer, if you wrote it out separately.
3. Connect the **stmap** input to the source of the **motion** layer.

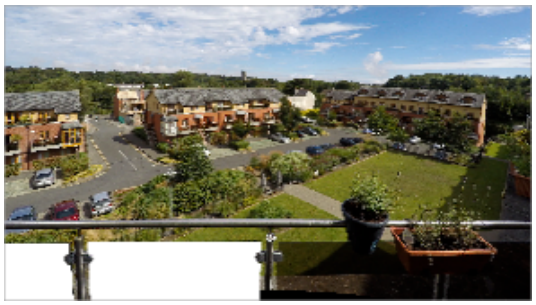


4. Set the STMap **UV channels** control to the **forward** layer to remove the distortion.

   You can now track, matchmove, composite, and so on in the undistorted space before redistorting the image.
5. After the comp work is complete, you can warp the VFX work by applying the **backward** layer in the **UV channels** control using another STMap node.
6. The final step is merging the warped VFX work back over the original image to create the comp.



The output from nodes placed before the second STMap node are distorted by the same warp as the sequence.



The undistorted plate and comp



The comp after redistortion

Tracking with PlanarTracker

PlanarTracker is a powerful tool for tracking surfaces that lie on a plane in your source footage. Planar tracking is often better than tracking individual points (with the Tracker node for instance) as it takes a great deal more information into account and gives you a more accurate result. A rigid surface, like a wall or a side of an object, are good planes to track. You can use your tracking results to replace the tracked plane with another image for instance. You can define the region to track by creating and animating roto shapes.

Tracking a Plane

Before you can track a plane, you need to draw one using the Roto node.

Drawing a Plane to Track

You can use the PlanarTracker to track rigid objects and objects that deform slightly throughout the track. As the PlanarTracker tries to fit a plane to the object to be tracked, rigid objects obtain better track results than objects that deform.

For instance, a wall or a flat side of an object are good planes, but you can also get good results tracking faces or people. It's also important that the plane you're tracking has some texture and that the plane isn't completely obscured at any point of the tracking. Tracking surfaces without texture and few features to track is not likely to produce good results.

1. You can do one of the following:
   - Insert a PlanarTracker node by either selecting **Transform** > **PlanarTracker**, or by pressing tab, typing PlanarTracker , and pressing **Return**. This inserts a Roto node that is already in PlanarTracker mode. You can use this to draw a Bezier shape, which is automatically added as a track object. The shape's boundary appears in purple, denoting this, and the shape is automatically added to a layer called **PlanarTrackLayer1** in the stroke/shape list.

     OR

- Create a Roto or RotoPaint node and use it to draw a Bezier shape around the plane you want to track. Your new shape's boundaries appear in red in the Viewer, and a Bezier shape item appears in the stroke/shape list. The shape remains as a normal roto shape until it is converted into a track object.



2. If you're drawing more than one shape, you can arrange them in the stroke/shape list to tell PlanarTracker that they are different layers. Order your shapes from closest to camera (top of the list) to furthest away (bottom of the list), and PlanarTracker automatically holds out any track layers above the current one.

Note: It is not currently possible to create a holdout mask using a key matte from another node, such as Keylight.

3. Make sure you're still on the same frame as you used to draw the Bezier shape.

Note: When you select a frame to draw a roto shape on, that frame becomes your reference frame. When you proceed with tracking your plane it's important that you're always starting on the same reference frame. Move to your reference frame using the **Go to Reference Frame** button . In the course of the process you may decide to change your reference frame. You can do this by clicking **Set Reference Frame**  in the Viewer.

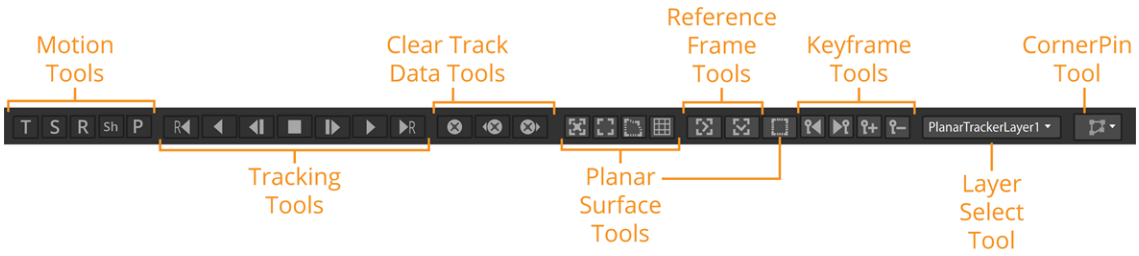
Tracking the Footage

You can now proceed to track the plane you've drawn:

Depending on whether you have chosen to insert a PlanarTracker node, or a Roto (or RotoPaint) node, do one of the following:

If you have chosen to insert a PlanarTracker node:

1. Use the tracking tools above the Viewer to track the Roto shape.



2. See Tracker Menu Options in the Viewer for more information.

If you have chosen to insert a Roto or a RotoPaint node:

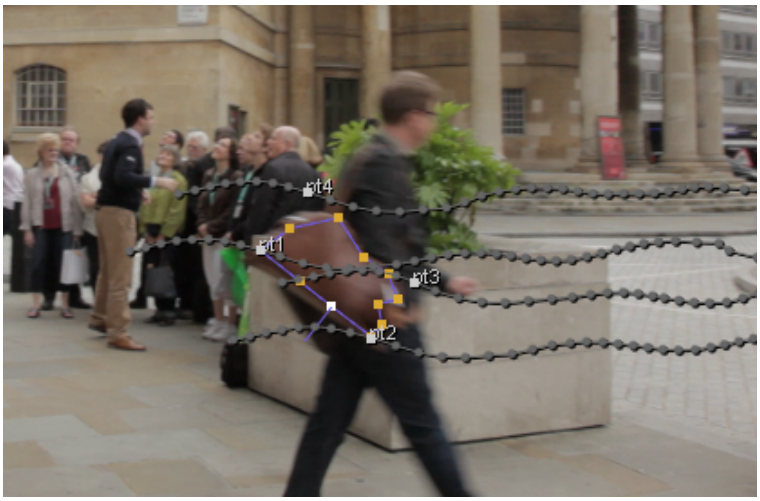
1. Right-click the Roto shape and select one of the following options:
   - **planar-track this shape** - Select this to convert the roto shape into a track object that doesn't contain any tracking data (if it's not already).
   - **planar-track this shape (fwd)** - Select this to track your shape while the track is played forwards.
   - **planar-track this shape (bkwd)** - Select this to track your shape while the track is played backwards.
2. If you selected one of the latter two options, your shape is now tracked throughout the timeline. A progress dialog is displayed while the shape is tracked.

   If you selected the first option to convert your roto shape into a track object without any tracking data, you can use the tracking tools above the Viewer to track forwards or backwards. See Tracker Menu Options in the Viewer for more information. When you select any of the options above the tracker menu tools are displayed above the Viewer.

   The shape is now automatically added to a layer called **PlanarTrackLayer1** in the stroke/shape list and is displayed in purple in the Viewer, signifying that it has been converted into a track object.
3. In the stroke/shape list, a purple rectangle appears in the **PT** column next to your shape(s). You can toggle the purple rectangle on and off to revert the shape to a normal roto shape instead of a track object.

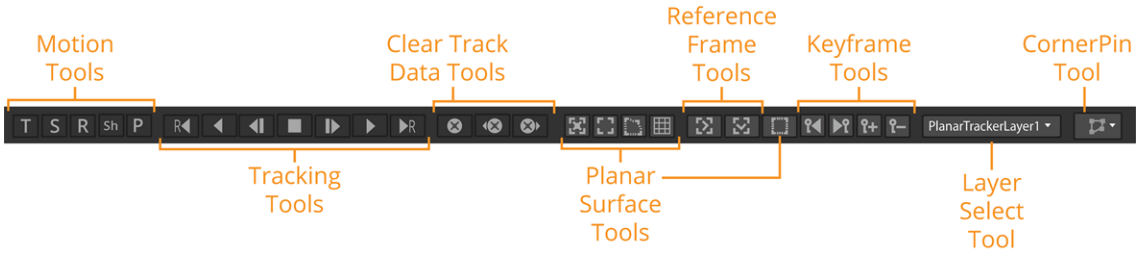
> **Note:** You can only toggle a shape between being a Roto shape or a track object, after converting the Roto shape into a track object.

4. Select the **Tracking** tab in the Roto node properties to display the tracking data, as shown below.



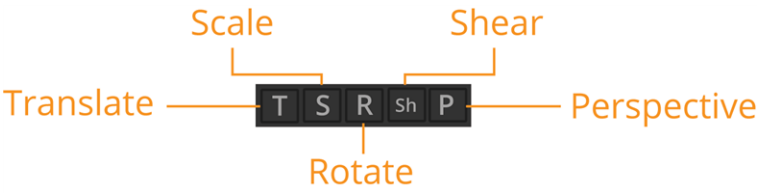
Tracker Menu Options in the Viewer

You can use the tracker menu options to perform more advanced tracking. These options are only visible in the Viewer after you have converted a shape into a track object.



Track Motion Controls

Before tracking an object, you can use the track motion controls to specify the type of movement that the PlanarTracker can expect. The following options are available:

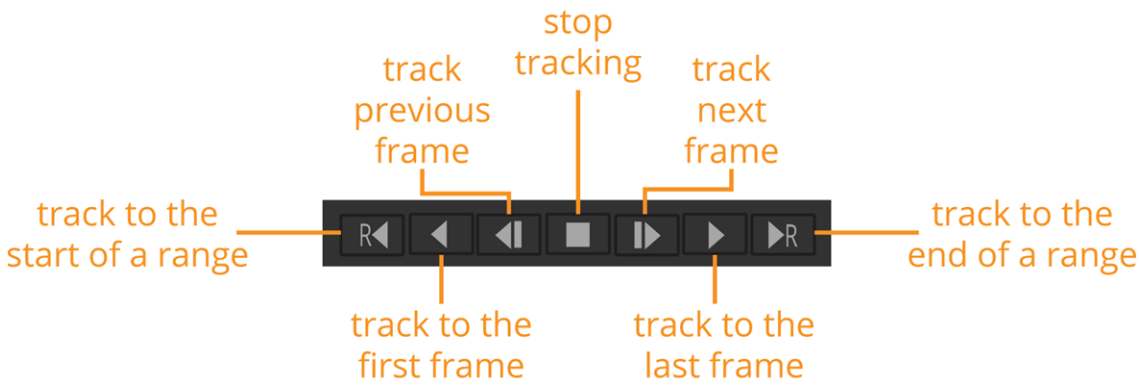


- **Translation** - Select this so the tracker expects translation.
- **Rotate** - Select this so the tracker expects rotation.
- **Scale** - Select this so the tracker expects scaling.
- **Shear** - Select this so the tracker expects shearing.

• **Perspective** - Select this so the tracker expects changing perspective.

Tracking Controls

Use the **Tracking buttons** to track backwards or forwards throughout the whole footage, or if you want, select a specified frame range.

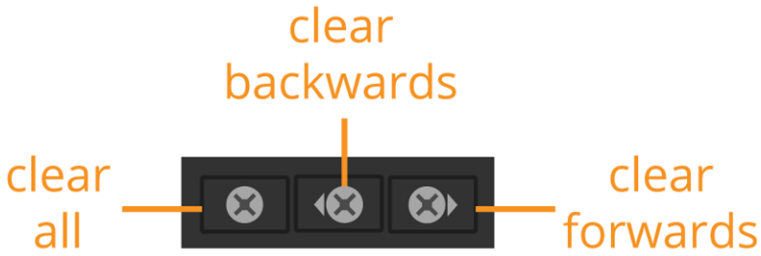


You can also track on a frame by frame basis. For example, if the plane you're tracking is visible throughout the footage, forward tracking might be all you need. If the plane is only partially visible over a part of the footage though, it might be a good idea to first track the part with the whole plane visible, and then track the rest of the footage separately.

With the Tracking controls you can track and re-track your footage. For more information see Tracking and Stabilizing .

Clear Tracking Data Controls

You can clear tracking information that you've already created with the clear buttons.



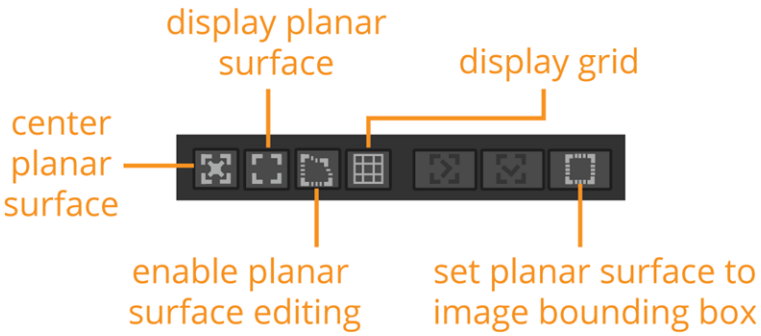
• **clear all** - clear all tracking information created by PlanarTracker.

• **clear backwards** - clear all tracking information backwards from the current frame.

• **clear forwards** - clear all tracking information forwards from the current frame.

PlanarTracker Surface Controls

After you have tracked your shape or shapes, you can review your results by scrubbing back and forth on the timeline. You can adjust the planar surface shape if, for example, your roto shape drifts in the course of the tracking, or you simply want to change it. To do this, you can use the planar surface controls.



- **center planar surface** - Select this to center the planar surface in the Viewer during playback.
- **display planar surface** - Select this to display the boundary of the planar surface.
- **enable planar surface editing** - Select this so that you can edit the planar surface by dragging the corner points in the Viewer.
- **display grid** - Select this to display a grid over the planar surface.
- **set planar surface to image bounding box** - Select this to change the planar surface to be the same as the image bounding box.

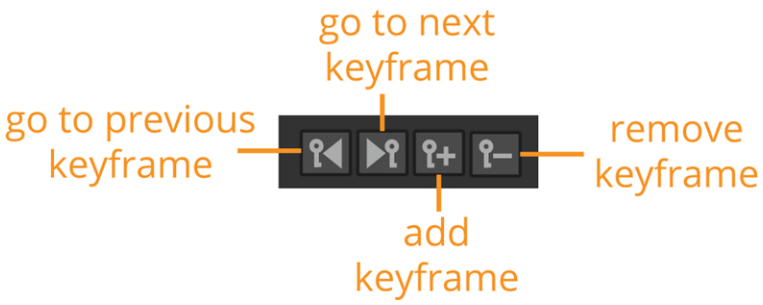
Reference Frame Controls

You can use the reference controls above the Viewer to either go to the current reference frame, or change the reference frame to the current frame.



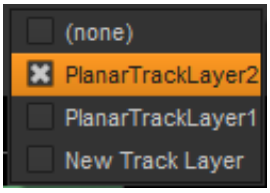
Keyframe Controls

You can set and remove keyframes, and go to previous or next keyframes using the keyframe controls above the Viewer.



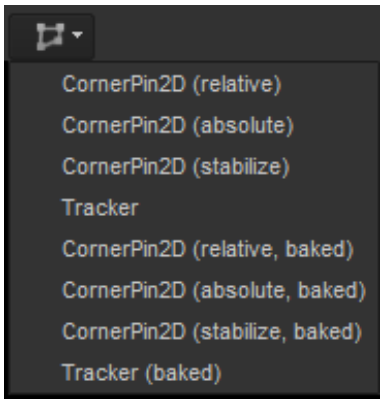
Layer Dropdown

You can use the layer dropdown to quickly select various PlanarTracker layers, or to add a new layer.



CornerPin Dropdown

You can use the **CornerPin** dropdown to insert a CornerPin node. The CornerPin2D tool is designed to map the four corners of an image sequence to positions derived from tracking data. In practice, this allows you to replace any four-cornered feature with another image sequence.



The **CornerPin** dropdown provides you with several different kinds of CornerPin you can choose from:

- **relative** - to warp the image according to the relative transform between the current frame and the reference frame. The image remains unchanged in the reference frame. You can also pick the baked version of the relative CornerPin node. A baked node has the keyframe values copied from PlanarTracker, rather than having them expression linked.

- **absolute** - to use the **to** and **from** controls to place the image exactly within the selected plane. This may skew the image throughout the footage. This attaches and automatically sets its format to the dimensions of any currently selected node. You can also pick the baked version of the absolute CornerPin node. A baked node has the keyframe values copied from PlanarTracker, rather than having them expression linked.
- **stabilize** - to apply an inverse transform on the image, which effectively locks the image in its place and stabilizes it in the footage. This type of corner pinning is most useful for drift corrections and making sure your tracking results are reliable. You can also select **Tracker** to create a baked Tracker node for further reducing jitter. You can also pick the baked version of the stabilize CornerPin node. A baked node has the keyframe values copied from PlanarTracker, rather than having them expression linked.
- **Tracker** - creates a Tracker node with four tracks set to each of the tracked corner points. These tracked points give you access to the Tracker's comprehensive functions, such as track averaging and enhanced smoothing controls. See Tracking and Stabilizing for more information.

Tracking Tab Menu Options

Select the **Tracking** tab in the Roto node's properties to access more tracking options.

Settings

track channels	Use this dropdown to select the channel set on which you want to track.
image channels	Use this to select an individual channel from the channel set. You do not need to select an individual channel, and instead can leave it as **none**, the default.
pre-track filter	Before image patches are compared, the selected filter is applied. You can select one of the following options: **none** - This disables all pre-filtering, which allows you to have full control of tuning the input image for tracking. **adjust contrast** - This stretches the image contrast to better suit the tracking algorithm. This option is recommended. **grayscale** - This converts any input RGB channels to a grayscale image for faster processing.
adjust for luminance changes	If your footage changes brightness over time, either gradually or suddenly, enabling this option performs extra pre-filtering to help compensate for the

	changes. This may make the track slower and less accurate and therefore should only be selected when you need to handle changes in luminance.
clamp super-white, sub-zero footage	Select this to clamp the tracking patch pixel values to lie between 0 and 1. If you want to track using the full dynamic range of your super-white or sub-zero footage, disable this option.
hide progress bar	As it is possible to stop a track using the tools above the Viewer, you can choose to hide the progress dialog that appears when you track a shape.

Export

You can use the **CornerPin** dropdown to select the type of corner pin node you want to insert. See CornerPin Dropdown for more information about the different kinds of CornerPin. After you have selected the required type of CornerPin, you can press **Create** to insert it.

Select the **link output** checkbox to link to the PlanarTracker output so that exported nodes are updated with the track.

Correction

The CornerPin points are populated automatically when you track an object. When you draw a roto shape and convert it into a track object, Nuke automatically places 4 corner pins around the shape. These are the points that are tracked.

You can correct the four automatically placed points by offsetting any or all of the four points. To offset a point, simply click and drag it in the Viewer to the correct position.

Reference Frame

This is the frame used as a reference to compare all other frames containing tracking data. It defaults to the first frame used to track. You can change this by entering a new frame number in the **reference frame** input field.

Reusing a Track Result

You can reuse an entire tracked plane or a single track.

Reusing a Tracked Plane

You can use a plane you've already tracked to analyze a larger plane situated on the same plane.

1. Make sure you're in the same frame as you used to draw the previous shape.
2. Draw a shape on the same plane as your previously tracked shape. Make sure the new shape is in the same **PlanarTrackLayer** as the first one in the stroke/shape list.
3. PlanarTracker has now placed your new roto shape on the same plane as the old one. If you now scrub in the timeline, you should find that your new shape sticks with the camera movement.

Reusing a Single Track

You can also reuse your tracked points to verify that the overall results are sticking to your footage.

1. Track a plane as described under Tracking a Plane, and adjust the results if necessary.
2. When you're happy with the tracking results for that plane, you can drag a corner of your tracked shape and place it on a specific detail that may not be getting tracked correctly. Note that you're moving the roto shape locally, and not affecting the actual track results.
3. Scrub through the frames to see if the point stays on the detail as you expect.
4. If drifting occurs, drag the point to correct it.

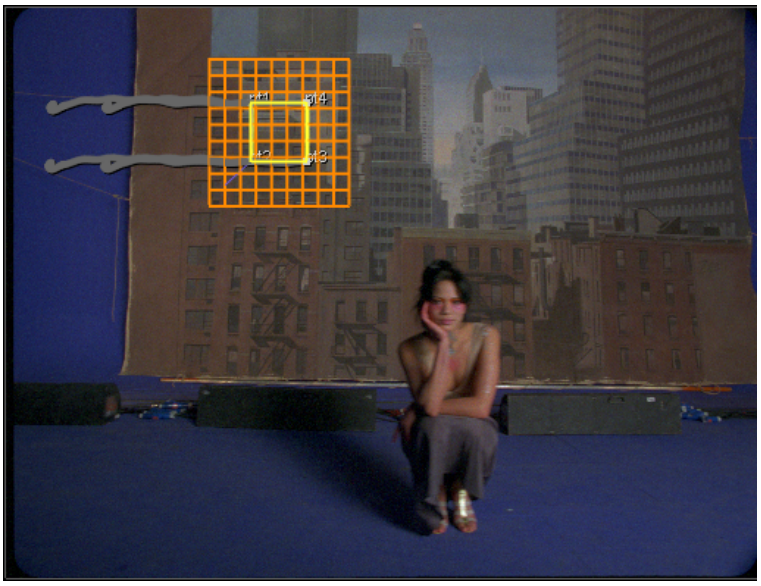
Placing an Image on the Planar Surface

When you've tracked a planar surface on your footage, you might want to place an image on it. To place an image on the tracked planar surface, do the following:

1. Ensure you're on the reference frame you've drawn the roto shape on, and select the PlanarTrackLayer you want to use in the **Create New Track** dropdown on top of the Viewer.

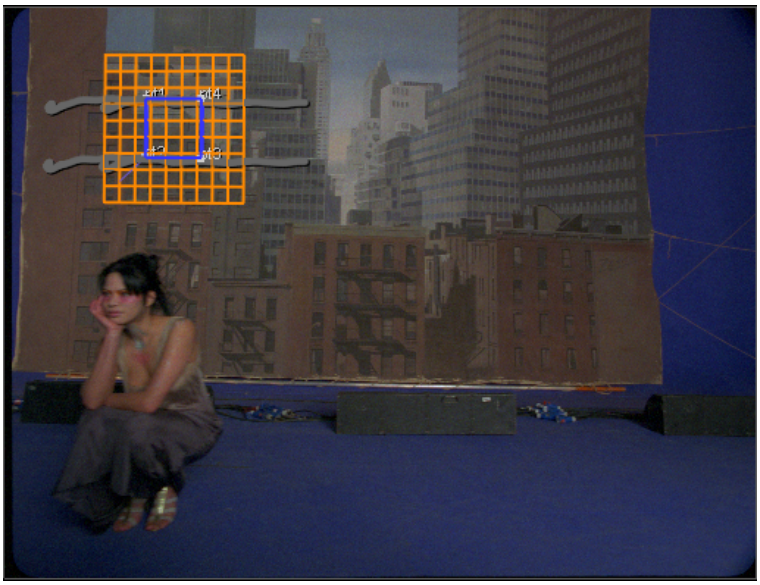


2. Check **show plane** and **correct plane** on the properties panel to make your planar surface visible and to enable modifying it. You can also click the corresponding buttons above the Viewer.
3. A rectangle indicates your planar surface in the Viewer. If the rectangle is very large, you can click the **Resize Planar Surface to Image** button in the Viewer.
4. Drag the corners of the rectangle to cover the area over which you want to place an image. Click the **Show grid lines** button in the Viewer to use a guide grid in positioning your rectangle.



This shows grid lines corresponding with the current plane and it helps with realigning your plane.

5. Scrub in the timeline to make sure the planar surface sticks to the area you want.



If you need to adjust it, you can do one of the following:

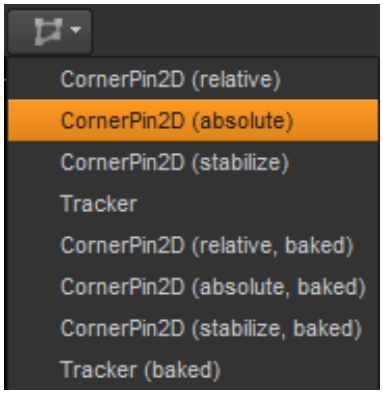
- Adjust the points in the reference frame to change the planar surface over the whole footage. This way you're adjusting the actual dimensions of the planar surface rectangle. The rectangle appears in yellow.
- Adjust the points in other frames to change the planar surface in the current frame and its adjacent frames. This way you're correcting small drifts in the planar surface rectangle without changing its real dimensions. The rectangle appears in blue.

6. The values in the **CornerPin points** show how your plane has warped from the reference frame to the current frame. Adjust them in the **Curve Editor** by the selecting **Curve Editor** tab above the Node Graph.



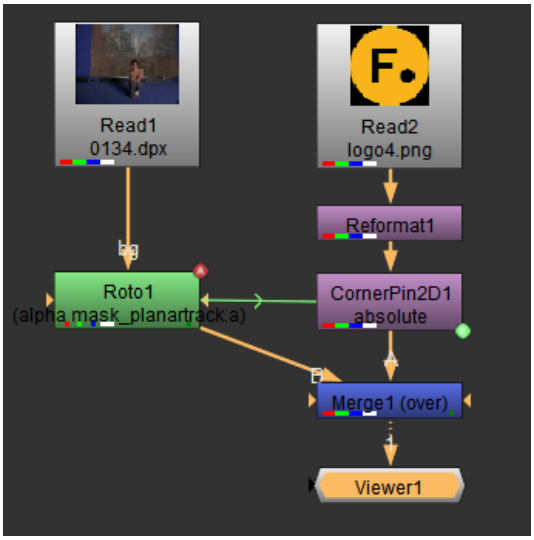
Note: While you can drag the **CornerPin points** data to another node or control by simply dragging selecting all the points and dragging on any of the animation buttons. If you press **Ctrl**/**Cmd**+drag, this creates an expression link to the **CornerPin points** data instead of placing them in the new location.

7. Read in the image you want to place on the tracker planar surface.
8. To convert the new image to the same format, insert a Reformat node and select the correct image format from the **output format** dropdown.
9. You can now add a CornerPin node to help you place an image on the plane. Click the **Create CornerPin2D Node** dropdown in the Viewer (or use the **Export** dropdown in the properties panel) and select the type of corner pinning you want to use.



See the 'Tracker Menu Options in the Viewer' section in Tracking a Plane for more information about the different types of CornerPin.

10. Insert a Merge node and connect the CornerPin node to the **A** input, and the Roto node to the **B** input. Your node tree should now look similar the one shown below:



11. You can now make any other required changes to your image. For example, you can adjust the **mix** control in the Merge node's properties, to change the opacity of the image.
12. When you're happy, close all the node property panels to see the result clearly in the Viewer. You can scrub through the timeline and make sure the position of the image in sorrect throughout.



Camera Tracking

Nuke's CameraTracker node is designed to provide an integrated camera tracking or match-moving tool, which allows you to create a virtual camera whose movement matches that of your original camera. Tracking camera movement in a 2D footage enables you to add virtual 3D objects to your 2D footage.

Introduction

With the CameraTracker node, you can track the camera motion in 2D sequences or stills to create an animated 3D camera or a point cloud and scene linked to the solve. You can automatically track features, add User Tracks or tracks from a Tracker node, mask out moving objects using a Bezier or B-spline shape, and edit your tracks manually. CameraTracker can solve the position of several types of cameras as well as solve stereo sequences.

Quick Start

The tracking process is outlined below, whether you intend to track a sequence or a set of stills:

1. Connect the CameraTracker node to the sequence you want to track. See Connecting the CameraTracker Node.
2. Mask out any areas of the image that may cause CameraTracker problems, such as movement within the scene or burn-in. See Masking Out Regions of the Image.
3. If you're tracking stereoscopic or multi-view images, set the **Principal View** on the **CameraTracker** or **Settings** tabs. See Working with Multi-View Scripts for more information.
4. Set the camera parameters, such as **Focal Length** and **Film Back Size**, if they are known. These are described under Setting Camera Parameters.
5. Set the **Source** dropdown to **Sequence** or **Stills**, and then:
   - If you intend to track a continuous **Sequence** of frames, set the tracking preferences using the **Settings** tab **Features** and **Tracking** controls. See Tracking in Sequence Mode for more information.
   - If you're using **Stills**, you can track all frames in the same way as sequence tracking, or a subset of **Reference Frames** using the **+/-** keyframe buttons above the Viewer or in the properties panel. See Tracking in Stills Mode for more information.

6. You can place **User Tracks** to improve difficult solves, use an entirely manual tracking approach, or set 3D survey points. You can use 3D survey points to tie your sequence to a known 3D world, such as those created using stills. See Working with User Tracks for more information.

Tip: 3D survey points have replaced the ProjectionSolver workflow, but you can still add ProjectionSolver nodes by pressing **X** in the Node Graph and entering **ProjectionSolver** as a Tcl command.

7. Click **Track** to begin tracking the sequence.
8. Solve the Camera position by clicking **Solve** and refine it, if necessary. For more information, see Solving the Camera Position
9. Set the ground plane, if required, and adjust your scene. See Adjusting the Scene.
10. Select what to export from the solve using the **Export** dropdown and click **Create**.

    You can export an animated camera, a stereoscopic or multi-view rig, a 3D scene and point cloud, lens distortion, or cards. See Using Solve Data.
11. If you have multiple footage sources of the same scene or content available, you can also use survey points to solve each of your sources and then register them all in the same world. See Combining Solves.
12. Add your 3D virtual objects to the footage. See Placing Objects in the Scene.
13. By default, any 3D objects you added to your footage do not have lens distortion applied to them. As a result, they can look like they weren't shot with the same camera. To fix this, see Accounting for Lens Distortion.

Connecting the CameraTracker Node

To connect the CameraTracker node:

1. Read in and select the clip you want to track.
2. Click **3D** > **CameraTracker**.
3. If you want to omit a part of the scene from being tracked, connect a matte to the **Mask** input. Note that, unlike the **Source** input, this input is hidden and appears as a small triangle on the left hand side of the node. For more information about masking, see Masking Out Regions of the Image.
4. Click **Image** > **Viewer** to insert a Viewer node and connect it to the CameraTracker node.

Masking Out Regions of the Image

Tracking works best on fixed, rigid parts of the scene so that each track can create a single, fixed 3D point. The solver uses these 3D points to work out the camera path. Moving elements and burn-ins do not have a fixed 3D point in the world and should be masked out before tracking.

To mask regions of your sequence, attach a matte to the **Mask** input to define image regions that should not be tracked. You can also use the source input's alpha channel as a matte.

1. If you want to use a separate matte for masking, connect a Roto node to the CameraTracker **Mask** input.
2. Scrub through the sequence and keyframe the roto shapes to cover the areas you don't want to track.

   You don't have to be too accurate with the mask, it's only intended to cover areas that are likely to cause CameraTracker problems. For example, in the image shown, the actors and the copyright burn-in are masked.



3. In the Properties panel, set **Mask** to the component you want to use as a mask:
   - **None** - Track features in the whole footage.
   - **Source Alpha** - use the alpha channel of the source clip to define which areas to ignore.
   - **Source Inverted Alpha** - use the inverted alpha channel of the source clip to define which areas to ignore.
   - **Mask Luminance** - use the luminance of the mask input to define which areas to ignore.
   - **Mask Inverted Luminance** - use the inverted luminance of the mask input to define which areas to ignore.
   - **Mask Alpha** - use the mask input alpha channel to define which areas to ignore.

- **Mask Inverted Alpha** - use the inverted mask input alpha channel to define which areas to ignore.

4. Track as normal using the automated Analysis **Track** button. See Tracking in Sequence Mode.



Note: There is no need to mask areas of the image when tracking manually - you specify where User Tracks are placed.

Working with Multi-View Scripts

CameraTracker can track and solve stereoscopic or multi-view projects in much the same way as single view projects.

1. Connect the CameraTracker node as described in Connecting the CameraTracker Node.
2. Use the **CameraTracker** or **Settings** tab **Principal View** dropdown to select the view used to create the tracks. Tracks in any other views present in the script are calculated from the **Principal View**.
3. Follow the workflow described under Camera Tracking to track and solve the camera.
4. If you create any User Tracks, note that they currently only support at most two views. To select which views are to be used as your left and right views for user tracks, use the **User Track Views** control on the UserTracks tab of the CameraTracker properties.
5. To visualize both cameras for your views, create a multi-view rig with a camera for each of your views by selecting **Camera rig** from the **Export** section of the **CameraTracker** tab. See Creating Camera Nodes for more information.

Setting Camera Parameters

Camera settings relate to the physical aspects of the camera used on set. Accurate physical camera data produces a better camera track and solution.

1. Select the motion type of the on set camera from the **CameraTracker** tab **Camera Motion** dropdown menu. This is linked to a control of the same name on the **Settings** tab.
   - **Rotation Only** - select this option if the camera is static and rotating, for example, if you're using a tripod mounted camera for nodal pans.
   - **Free Camera** - select this option if the camera is both translating and rotating.
   - **Linear Motion** - select this if the camera has a straight, linear path.

- **Planar Motion** - select this if the camera has a flat path, moving in a two-dimensional plane only.

2. If you have already used a separate LensDistortion node (see Working with Lens Distortion on page 1) to remove lens distortion from your footage, you can leave the **Lens Distortion** control set to **No Lens Distortion**.

   Otherwise, set **Lens Distortion** to **Unknown Lens** before you solve the camera position to force CameraTracker to calculate the distortion.
3. Select the **Focal Length** type for the camera from the dropdown menu:
   - **Known** - select this option if the focal length is available and enter a value in the **Length** control.
   - **Approximate Varying** - select this option if an approximate focal length is available and enter keyframed focal length values in the **Length** control.
   - **Approximate Constant** - select this option if an approximate focal length is available and there is no zoom, and enter a focal length value in the **Length** control.



Note: CameraTracker attempts to refine the focal length during the solve if you select an **Approximate** option.

   - **Unknown Varying** - select this option if the focal length is unknown and changing.
   - **Unknown Constant** - this is the default option. Use this option if the focal length is unknown and there is no zoom.
4. Either choose a **Film Back Preset** from the dropdown to populate the **Film Back Size** controls, or if your camera isn't in the list, enter the **Film Back Size** manually.

Tip: You can add your own defaults by editing the **../NukeScripts/camerapresets.py** file in the Nuke installation package.

Tracking in Sequence Mode

In **Sequence** mode, CameraTracker tracks the footage attached to the **Source** input and defines a set of 2D feature tracks that correspond to fixed points in the scene.

Note: If you intend to remove lens distortion manually using a separate LensDistortion node, you should do that before you track the sequence. See Working with Lens Distortion. Otherwise, set **Lens Distortion** to **Unknown Lens** before you solve the camera position to force CameraTracker to calculate the distortion.

Before tracking, use CameraTracker's properties panel to control Viewer output and determine tracking behavior:

1. On the **CameraTracker** tab, ensure that **Source** is set to **Sequence**.
2. If you intend to mask out parts of your image, set the **Mask** control to the matte source.

   For example, if you're using the alpha channel from a Roto node attached to the **Mask** input, select **Mask Alpha**. For more information, see Masking Out Regions of the Image.

Note: For stereoscopic or multi-view footage, set the **Principal View** on the **CameraTracker** or **Settings** tab to the main tracking view. Any masking should be applied to this view, which is then used to calculate the secondary camera.

3. Use the **Range** dropdown to determine which frames are analyzed:
   - **Input** - the default value, sets the frame range to the length of the sequence attached to the **Source** input.
   - **Global** - sets the tracking frame range to the range set in the Project Settings **frame range** controls.

     If no frame range is defined, the frame range of the first image you read in is used as the **Global** frame range.
   - **Custom** - sets a tracking range of frames described by the **from** and **to** fields.
4. On the **Settings** tab, define the starting points for feature tracking:
   - **Number of Features** - define the number of features you want to track in each frame.

     Ideally, you should use more than 100 tracks per frame. In most cases, the default 150 should be sufficient, but in difficult sequences you may consider using a higher number.
   - **Detection Threshold** - set the distribution of features over the input image.

     If you enter a low detection threshold value, features are tracked evenly on all parts of the image and vice versa.



Low threshold distribution



High threshold distribution

- **Feature Separation** - set the distribution of features in relation to each other.

  To force feature separation and spread features evenly over the image at even distances, enter a high feature separation value.



Low separation value



High separation value

5. Check **Refine Feature Locations** to lock detected features to local corners. If you activate this, CameraTracker finds the closest corner point in your footage and locks the feature to it.
6. Check **Preview Features** to view the current distribution of tracking features.

   Preview comes in handy when you want to tweak the tracking parameters further before tracking - it updates dynamically when controls are adjusted.

   It's important to make sure that the previewed features are distributed evenly and not clustered together too densely before tracking using the distribution and separation controls.



Previewing features in a sequence

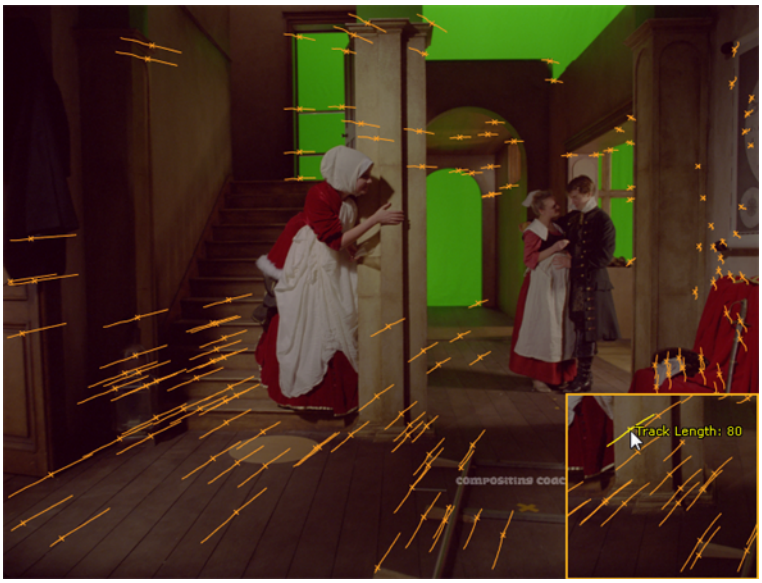
7. Once you're happy with the feature distribution in the preview, click **Track** to begin analyzing the sequence.
8. CameraTracker begins reading frames sequentially and tracking the features present. Tracks that don't meet the quality thresholds set in the **Settings** tab **Tracking** controls are reseeded so that the number of tracks remains constant.

   See Troubleshooting Sequence Tracks for more details on adjusting these controls.

   When the playhead reaches the end of the sequence, it begins a verifying pass by reading the frames sequentially backwards. Any tracks that were reseeded due to error are tracked back past the point where they were created, if they remain viable.

Viewing Track Data

Once tracking is complete, scrub through the timeline to examine the tracked features. The points represent the features and the vectors the track length calculated for the associated feature. Hover over a point to display its length, in frames.



Consider masking out areas where tracks are consistently seeded and rejected, such as highly reflective surfaces, and adjusting the **Settings** tab **Tracking** controls to increase track reliability.

See Masking Out Regions of the Image and Troubleshooting Sequence Tracks for more information.

Reviewing AutoTracks Curves

Detailed tracking information is displayed in curves in the properties panel on the **AutoTracks** tab. You can select all track curves at once or get a more detailed view of a single curve, such as **num tracks**.

Tip: Pressing **F** with focus on a curve maximizes the selected curve in the space available.



Selecting all track curves.



Framing a single curve.

Curves can indicate areas of the sequence where tracking encountered problems. For example, the number of tracks curve on the right shows a significant dip on the current frame, as indicated by the playhead position.

In this instance, you could try adjusting the **Number of Features** or **Minimum Length** and retracking. See Troubleshooting Sequence Tracks

Troubleshooting Sequence Tracks

Some sequences are inevitably going to cause problems. There are a number of pre-tracking checks and post-tracking refinement controls to assist CameraTracker.

Pre-Tracking Checks

- Play through the sequence before tracking and mask out any problem areas in the scene. Large moving objects can confuse the tracking and solving process as can appear to be fixed areas of the scene.

  See Masking Out Regions of the Image for more information.

Tip: If you're compositing on moving elements, such as faces, try tracking and solving on just the moving element using masks. You'll then get a camera that moves around the element as if it was fixed in a single position.

- Textureless areas of the sequence, such as greenscreens, can cause features to cluster together in other areas, affecting the track and solve.

  Turn on **Preview Features** and use the distribution and separation controls to even out the features. See Tracking in Sequence Mode
- Regular edges, such as striped patterns, in the scene can confuse CameraTracker.

  Again, turn on **Preview Features** and use the **Detection Threshold** to force CameraTracker to use corner-like images textures. See Tracking in Sequence Mode
- You can improve tracking data by adding User Tracks manually, see Working with User Tracksfor more information.

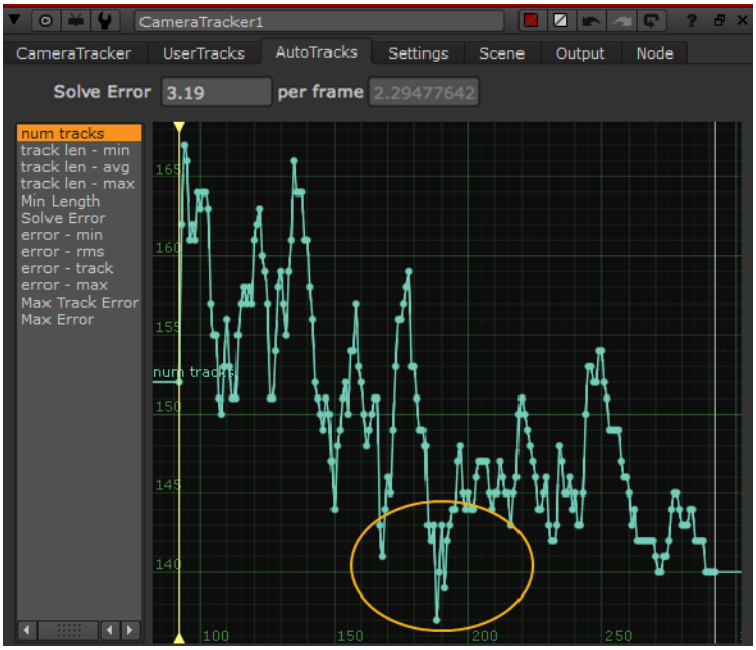
Reviewing and Refining Tracking Data

You can use the threshold controls on the **AutoTracks** tab to dynamically reject tracks and remove them, before solving, to improve accuracy. Using the **num tracks** curve as an example:

1. Click the **AutoTracks** tab in the properties and select the **num tracks** curve.



Tip: Press **F** in the curve display to fit the selected track(s) to the available screen space.



In the example, you can see that the number of tracks has dropped significantly around frames 180-200.

2. Move the Viewer playhead to the affected area and examine the tracked features in the Viewer.
3. If you increase the **Min Length** control in the properties panel, you'll start to see tracks turn red as they fall below the threshold.



Min Length set at the default value, 3 frames.



Min Length set to 50 frames.

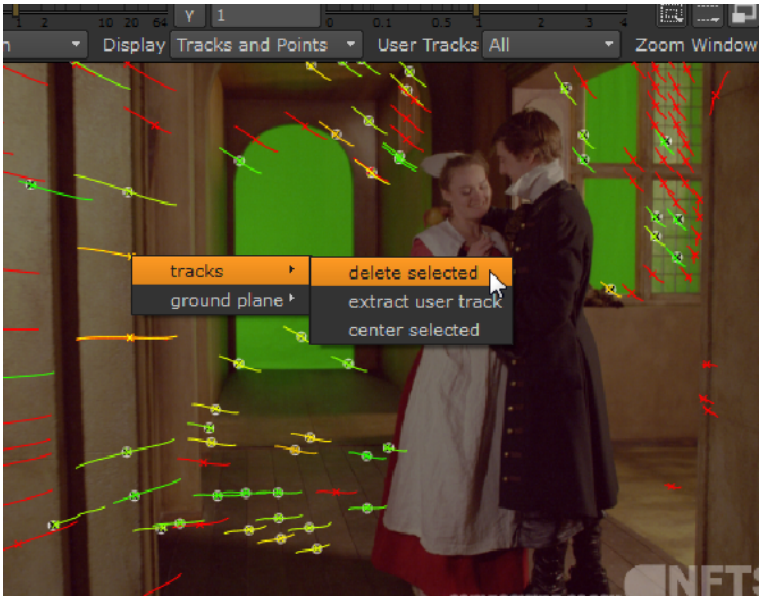
4. Click **Delete Rejected** to remove all the tracks that fall below the specified threshold.

> **Tip:** You can also remove tracks manually, by selecting them in the Viewer, right-clicking, and choosing **tracks** > **delete selected**.

5. You can then retrack the affected frame range by clicking **Update Track**. See Retracking Partial Frame Ranges for more information.

Re-tracking Using Tracking Settings

Once you have refined the tracking data, review the sequence by scrubbing the playhead to make sure you have tracks across the image on the rigid fixed background. If you see lots of tracks that drift or jump, select them manually, right-click, and go to **tracks** > **delete selected**.



You can also add User Tracks to improve tracking data before solving, see Working with User Tracks.

You can improve a set of feature tracks using the following controls and then retrack to improve your chances of getting a good solve:

- **Minimum Length** - sets a threshold value for the minimum acceptable track length. Tracks that fail to stick to the associated feature for this amount of frames are rejected.

  In long slow camera moves, it is best to solve from long tracks generated in the shot. If there are lots of short tracks, this can lead to noise on the calculated camera. Try removing the short tracks before solving.
- **Track Threshold** - CameraTracker's tolerance to change along the track is determined by this control.

  Reducing this threshold makes tracking more tolerant to image changes, potentially producing longer tracks.
- **Track Smoothness** - sets the threshold for smooth track generation. Adjusting this value can be useful in preventing poor tracks in complex sequences.

  Increasing the smoothness value removes tracks that fail over time.
- **Track Consistency** - sets the threshold for how inconsistent a feature track can be before CameraTracker discards it and reseeds in a different location.

  Higher values allow for less inconsistency and vice versa.

Click **Delete Rejected** to remove all the tracks that fall below the specified thresholds.

Extending Existing Camera Tracks

CameraTracker allows you to extend an existing set of tracking data by adding new frames, such as when more frames become available from a shoot.



Note: You can update tracking data as often as you like before solving, but once you've solved the camera position, updating tracking data should only be used to add a relatively small number of frames. See Updating Solves with Extended Tracking Data for more information.

To add new frames to your tracking data:

1. Read the new frames into Nuke using the Read node's **frame range** controls.
2. Open the CameraTracker properties panel and click **Update Track**.

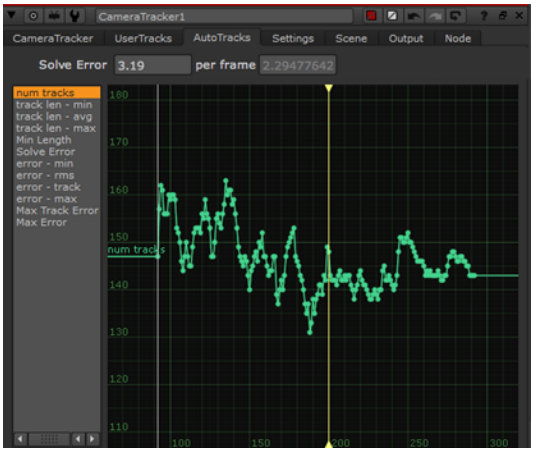
   A dialog displays allowing you to set the frame range to update.
3. Set the required frame range and click **OK**.

CameraTracker tracks the specified range and combines the tracking data with the existing tracks.

For example, if your original **AutoTracks** > **num tracks** curve appeared as shown on the left, the image on the right represents the new tracking data after the update is complete.



Original frame range track data.



Updated track including the extended frame range.

Tip: If the transition between the ranges seems abrupt, you can retrack the frames around the join using the method described in Retracking Partial Frame Ranges.

4. Proceed to Solving the Camera Position.

Retracking Partial Frame Ranges

After refining your feature points, you may not need to retrack the entire sequence. You can use **Update Track** to analyze a specific frame range.

1. Refine your features using the controls described previously.
2. Click **Update Track**.

   A dialog displays allowing you to set the frame range to update.
3. Set the required frame range and click **OK**.

   CameraTracker retracks the selected range and combines the tracking data with any existing tracks.

Note: You may notice that CameraTracker performs some analysis outside your selected frame range. This can be necessary to extend existing tracks into the update range, avoid creating duplicate tracks, and to combine the new tracks with existing tracking data.

Tracking in Stills Mode

In **Stills** mode, CameraTracker tracks the reference frames in the **Source** input and analyzes the input in two stages. Tracking defines the set of 2D feature tracks that correspond to fixed rigid points in the scene, then the solver calculates the camera path and projection that creates a 3D point for each reference frame.

Note: If you intend to remove lens distortion manually using a separate LensDistortion node, you should do that before you track the stills. See Working with Lens Distortion. Otherwise, set **Lens Distortion** to **Unknown Lens** before you solve the camera position to force CameraTracker to calculate the distortion.

Still Photography Guidelines

Still tracking relies on good input to produce good tracking output, so it's vital that you capture good still photographs that CameraTracker can interpret correctly.

CameraTracker has different requirements depending on the subject of the stills. For example, stills for a near-flat scenes are captured differently to those required for an interior set or model.

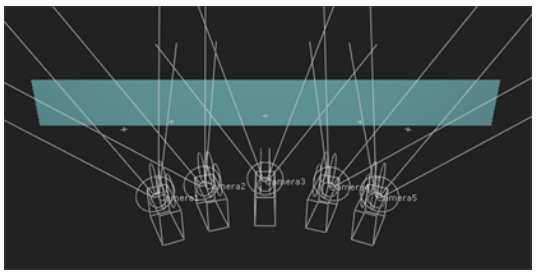
General Guidelines

- Don't crop your stills or transform them in anyway, such as rotation.
- Avoid dramatic changes in scale and angle between stills.
- To ensure you capture every part of the scene in 3 to 4 images, aim for a maximum change of 20-25% in content between frames.
- Avoid unnecessary redundancy where still frames contain large portions of the previous frame.

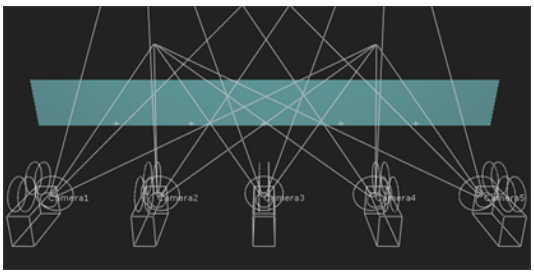
- Avoid tracking stills containing multiple occlusions - features with complex overlap can result in drastic changes in content between frames.

Shooting Near-flat Scenes

The best stills for near-flat scenes are captured head-on, facing the subject. In Nuke terms, that would mean taking your stills from the camera positions shown on the right.



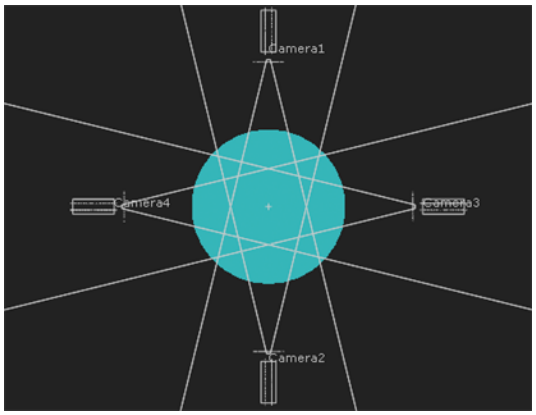
Incorrect camera positioning.



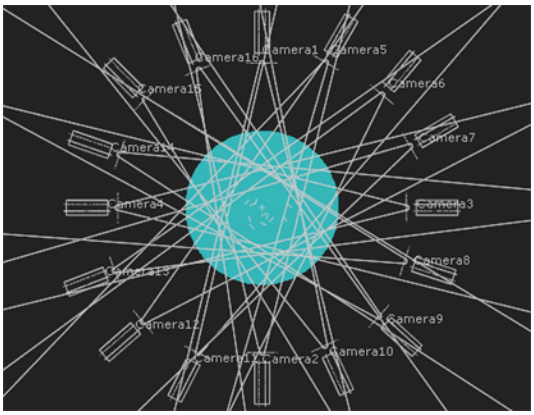
Correct camera positioning.

Shooting 3D Objects

When moving around an object, a photo every 15-25 degrees should be adequate, so a minimum of 16 shots for a full 360. Of course, taking more can improve the result.



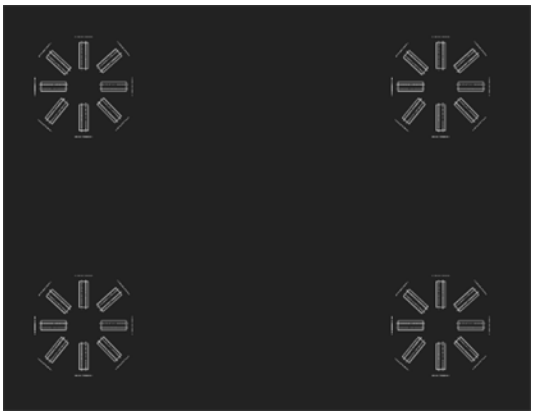
Too few stills.



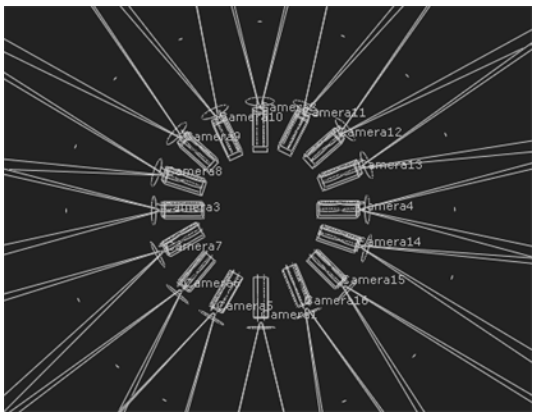
Minimum number of stills.

Shooting Interiors

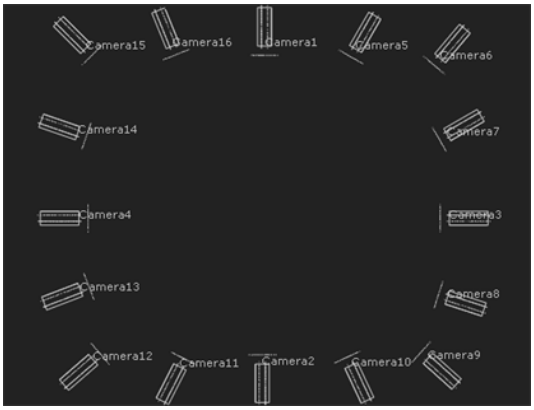
For enclosed spaces, such as interiors, capture stills from the center facing outward and then around the perimeter facing inward, rather than from different points within the space.



Incorrect camera positioning from different points.



Capturing stills facing outward.



Capturing stills facing inward.

Tracking Still Frames

Before you do anything else, you need to tell CameraTracker which still frames you want it to track and how it should distribute the tracking features on each frame. Then, you can track your still frames.

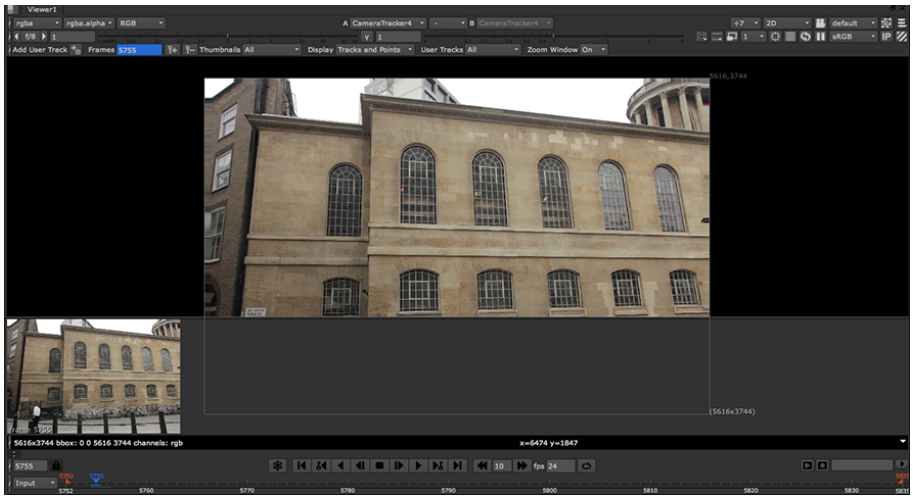
Selecting the Frames to Track

1. On the **CameraTracker** tab of the CameraTracker properties, ensure that **Source** is set to **Stills**.
2. Use the **Range** dropdown to determine which frames are analyzed:
   - **Input** - the default value, sets the frame range to the length of the sequence attached to the **Source** input.
   - **Global** - sets the tracking frame range to the range set in the Project Settings **frame range** controls.

If no frame range is defined, the frame range of the first image you read in is used as the **Global** frame range.

- **Custom** - sets a tracking range of frames described by the **from** and **to** fields.
- **Reference Frames** - allows you to manually set keyframes on the frames you want to analyze.

3. If you set **Range** to **Reference Frames**, do one of the following:
   - To set all frames in the input as reference frames, set the **Add** dropdown menu to **Add All** . Any missing frames on the input are skipped.
   - To set a specific frame range as reference frames, set the **Add** dropdown menu to **Add Range**. In the dialog that opens, enter the frame range in the **Frame range** field and click **OK**.
   - To set individual frames as reference frames, set the **Thumbnails** dropdown menu to **All** at the top of the Viewer. Then, scrub to a frame you want to set as a reference frame and click the add reference frame button . The current frame is set as a keyframe and displayed in the thumbnail gallery at the bottom of the Viewer.



Scrub to the next frame you want to add as a reference frame and repeat the process until you're happy with your set of reference frames.

Tip: If necessary, you can also use the delete reference frame button  or the **Delete** dropdown menu in the CameraTracker properties to remove frames from the set of frames to analyze.

4. Proceed to Adjusting the Features to Track below.

Adjusting the Features to Track

1. If you intend to mask out parts of your image (such as burn-ins), set the **Mask** control to the matte source.

For example, if you're using the alpha channel from a Roto node attached to the **Mask** input, select **Mask Alpha**. For more information, see Masking Out Regions of the Image.

Note: For stereoscopic or multi-view footage, set the **Principal View** on the **CameraTracker** or **Settings** tab to the main tracking view. Any masking should be applied to this view, which is then used to calculate the secondary camera.

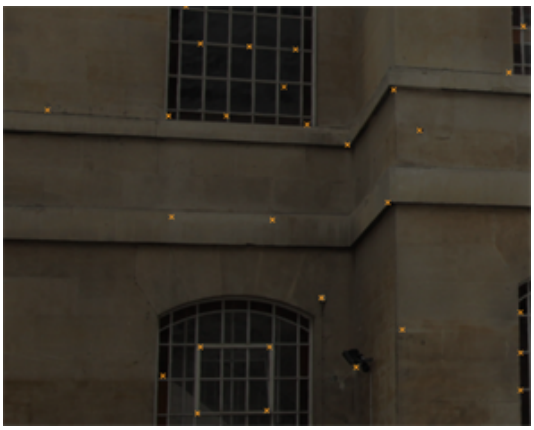
2. On the **Settings** tab, define the starting points for feature tracking:
   - **Number of Features** - define the number of features you want to track in each frame.

     Ideally, you should use more than 200 tracks per frame. In most cases, the default 250 should be sufficient, but in difficult solves you may consider using twice that number.
   - **Detection Threshold** - set the distribution of features over the input image.

     If you enter a low detection threshold value, features are tracked evenly on all parts of the image and vice versa.



Low threshold distribution



High threshold distribution

   - **Feature Separation** - set the distribution of features in relation to each other.

     To force feature separation and spread features evenly over the image at even distances, enter a high feature separation value.



Low separation value



High separation value

3. Check **Refine Feature Locations** to lock detected features to local corners. If you activate this, CameraTracker finds the closest corner point in your footage and locks the feature to it.
4. Check **Preview Features** to view the current distribution of tracking features.

   Preview comes in handy when you want to tweak the tracking parameters further before tracking - it updates dynamically when controls are adjusted.

   It's important to make sure that the previewed features are distributed evenly and not clustered together too densely before tracking using the distribution and separation controls.



Previewing features in a sequence

5. Once you're happy with the feature distribution in the preview, proceed to Tracking the Selected Frames below.

Tracking the Selected Frames

1. In the CameraTracker properties, click **Track** to begin analyzing the still frames.

   CameraTracker begins by detecting viable features, aligning the reference frames, and then tracking forwards. Tracks that don't meet the quality thresholds set in the **Settings** tab **Tracking** controls are reseeded so that the number of tracks remains constant.

   When the playhead reaches the end of the sequence, it begins a verifying pass by reading the frames sequentially backwards. Any tracks that were reseeded due to error are tracked back past the point where they were created, if they remain viable.
2. Proceed to Viewing Reference Frames and Track Data.

Viewing Reference Frames and Track Data

Once tracking is complete, you can review how CameraTracker connected the analyzed frames:

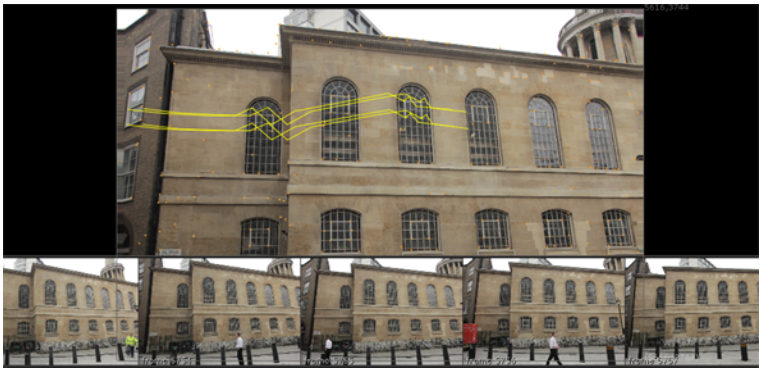
1. Do one of the following:
   - Switch the **Thumbnails** control above the Viewer to **Tracked**. CameraTracker displays the frames that are connected to the current frame in the thumbnail gallery at the bottom of the Viewer. This allows you to simply step through the sequence frame-by-frame to evaluate the connections.
   - Switch the **Thumbnails** control above the Viewer to **All** to have CameraTracker display all frames in the tracked range in the thumbnail gallery. This allows you to hover the cursor over a thumbnail to view its connections to adjacent reference frames highlighted in orange. Disconnected frames are highlighted in red. See Disconnected Frame Sets for more information.



Tip: You can click-and-drag to scroll the gallery left and right, or click a thumbnail to move the playhead to the corresponding reference frame. To adjust the size of the thumbnail gallery, drag the top of the gallery up and down.

2. To display track information, hover over a feature or marquee select several features.

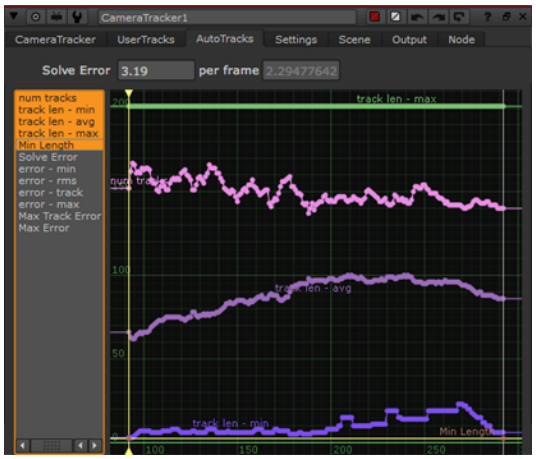
Note: If you make multiple selections, you won't see the track lengths in frames, only the tracks themselves.



Reviewing AutoTracks Curves

Detailed tracking information is displayed in curves in the properties panel on the **AutoTracks** tab. You can select all track curves at once or get a more detailed view of a single curve, such as **num tracks**.

Tip: Pressing **F** with focus on a curve maximizes the selected curve in the space available.



Selecting all track curves.



Framing a single curve.

Curves can indicate areas of the sequence where tracking encountered problems. For example, the number of tracks curve on the right shows a significant dip on the current frame, as indicated by the playhead position.

In this instance, you could try adjusting the **Number of Features** or **Minimum Length** and retracking. See Troubleshooting Still Tracks

Disconnected Frame Sets

Auto-tracking stills is not perfect and may throw out some disconnected frames, most likely due to a large change in viewpoint. After tracking, set the **Thumbnails** control above the Viewer to **All** and hover over images in the thumbnail strip. Thumbnails highlighted in red were not matched with the reference frames around them and are flagged with an error message in the Viewer.



CameraTracker can deal with disconnected frames in two ways:

- work with the connected set by deleting the unconnected reference frames using the  button or **Delete** dropdown and tracking again, or
- if all the frames in the sequence are needed, create User Tracks that span the sets to connect them. Make sure to define the tracks in as many frames as possible, and then track again based on the User Tracks. See Linking Still Reference Frames.

Tip: When a camera track starts and ends in the same place, CameraTracker may not automatically connect the end frame with the starting frame. You can set the **Thumbnail** control to **Tracked** to review the connections created between stills, and then add User Tracks between the two frames and click **Update Track**. See Working with User Tracks for more information.

Troubleshooting Still Tracks

Some stills sequences are inevitably going to cause problems. There are a number of pre-tracking checks and post-tracking refinement controls to assist CameraTracker.

You can improve tracking data by adding User Tracks. See Working with User Tracks for more information.

Pre-tracking Checks

- For standard sequences, avoid too much or too little redundancy between still frames, that is, frames that contain large portions of the previous frame. As a guide, try to capture every part of the scene in 3 to 4 images, aiming for a maximum change of 20-25% in content between frames.
- Try to capture the same object in at least 3, but preferably 4 photos.

  For example, when moving around an object, a photo every 15-25 degrees would be adequate (approximately 16 shots for a full 360).

Post-Tracking Refinements

You can refine a set of feature tracks using the following controls and then retrack to improve your chances of getting a good solve:

- **Minimum Length** - sets a threshold value for the minimum acceptable track length. Tracks that fail to stick to the associated feature for this amount of frames are rejected.

  In long slow camera moves, it is best to solve from long tracks generated in the shot. If there are lots of short tracks, this can lead to noise on the calculated camera. Try removing the short tracks before solving.
- **Track Threshold** - CameraTracker's tolerance to change along the track is determined by this control.

  Reducing this threshold makes tracking more tolerant to image changes, potentially producing longer tracks.



Tip: Press **F** in the curve display to fit the selected track(s) to the available screen space.

Retracking Partial Frame Ranges

After refining your feature points, you may not need to retrack the entire sequence. You can use **Update Track** to analyze a specific frame range.

1. Refine your features using the controls described above.
2. Click **Update Track**.

   A dialog displays allowing you to set the frame range to update.
3. Set the required frame range and click **OK**.

   CameraTracker retracks the selected range and combines the tracking data with any existing tracks.



Note: You may notice that CameraTracker re-detects features on existing frames before re-tracking. This can happen when the cached feature data has been purged and needs to be rebuilt.

Working with User Tracks

User Tracks are placed manually, rather then being automatically seeded by CameraTracker, and can be used to improve or even replace auto tracking data. They can also be used to link unmatched reference frames together when tracking stills.

You can create User Tracks before tracking and solving to lock the camera to a particular part of the shot or afterwards to help improve the results.

These pages cover the following topics:

- Adding and positioning User Tracks in the Viewer. See Adding and Positioning User Tracks.
- Track User Tracks in several different ways. See User Tracking Methods.
- Using User Tracks to improve auto-tracking data. See Tracking Assists.
- Tracking a scene manually using only User Tracks. See Tracking a Scene Manually.
- Using User Tracks to link reference frames when tracking stills. See Linking Still Reference Frames.
- Assigning User Tracks as 3D survey points in order to tie your sequence to a known 3D world, such as those created using stills. See Assigning 3D Survey Points.

Adding and Positioning User Tracks

You can add as many User Tracks as required, depending on what you intend to accomplish. For example, auto-track assist may only require one or two User Tracks, whereas manual tracking would require at least eight User Tracks.

1. Enable the fast-add button  at the top of the Viewer and click in the Viewer to add User Tracks or click **Add Track** on the **UserTracks** tab in the properties panel.



Note: Clicking **Add Track** places the new track in the center of the current Viewer.



Tip: You can quickly add User Tracks by holding **Ctrl**/**Cmd**+**Alt** and clicking in the Viewer.

Track zoom windows display in the top left of the Viewer.



There are three zoom windows initially:

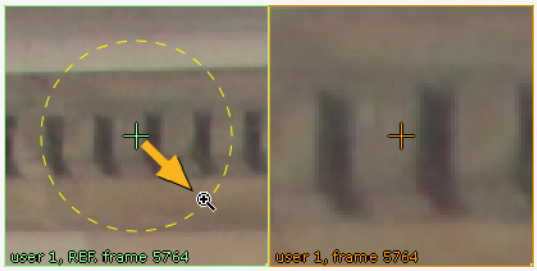
- **Reference frame** (green) - the first frame in the track, which remains constant, allowing you to locate the feature more accurately in subsequent frames.
- **Current frame** (orange) - the playhead position. Use this window to adjust the tracking anchor on the current frame by comparing it to the reference frame.
- **Keyframe** (blue) - the first keyframe in the sequence. Adding more keyframes adds a zoom window per keyframe.

Tip: You can click on a keyframe zoom window to instantly jump to that frame.

Tip: You can hide the zoom windows by setting the **Zoom Window** Viewer control to **Off**.

2. Drag the crosshair in the center of the anchor over the feature to be tracked or manually adjust the track **x** and **y** fields in the UserTracks list.
3. Refine the track position by clicking and dragging in the current frame zoom window.

Tip: You can change the magnification of zoom windows by holding **Shift** and dragging the magnifying glass cursor away from the center of the window.



To resize a zoom window, drag its lower right corner to a new location.

4. Once you're happy with the track's position, proceed to User Tracking Methods.

User Tracking Methods

User Tracks can be tracked automatically or manually, extracted from auto-tracking data, and imported from a Tracker node.

Auto User Tracks

The easiest way to get started with User Tracks is using the **Autotrack** feature, but you can create tracks totally independent of auto-tracking. After placing your tracks on the required features in the sequence:

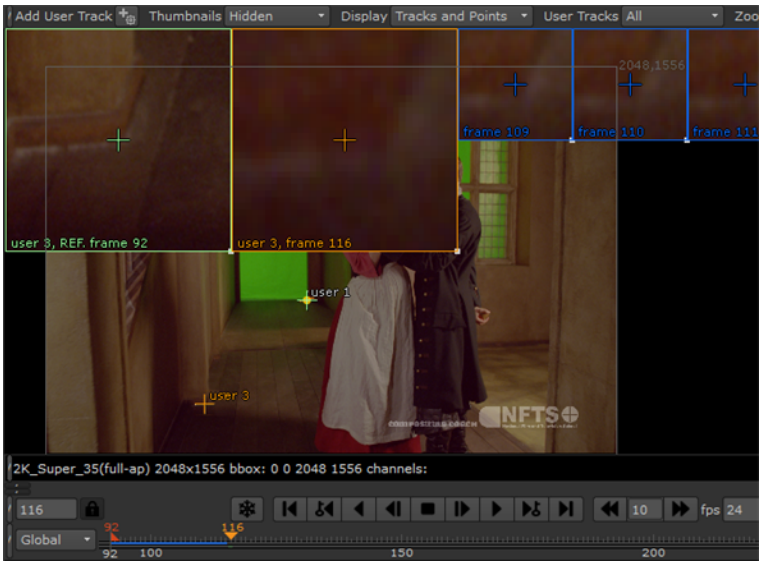
1. Click on the **UserTracks** tab in the properties panel to display the track table.
2. Select all the tracks you want to calculate in the table and then click **Autotrack**.

   CameraTracker attempts to track the selected features, in a similar way to Auto Tracks, except that when a track fails it is not reseeded.
3. You can examine features during and after tracking to determine if they failed:

   • During tracking - the anchors are colored yellow until they fail.

   • After tracking - any failed tracks don't have keyframes all the way through the sequence.

   The image shows a good track, **user 1**, highlighted in yellow and a failed track, **user 3**, which has started to drift on frame 116.



4. If a track fails or drifts, adjust the position of the tracking anchor by dragging the anchor in the zoom window.
5. Scrub through the sequence a few frames and repeat. Continue on through to the end of the sequence.
6. At each frame, a new keyframe window is added to the right of the zoom window.

Manual User Tracks

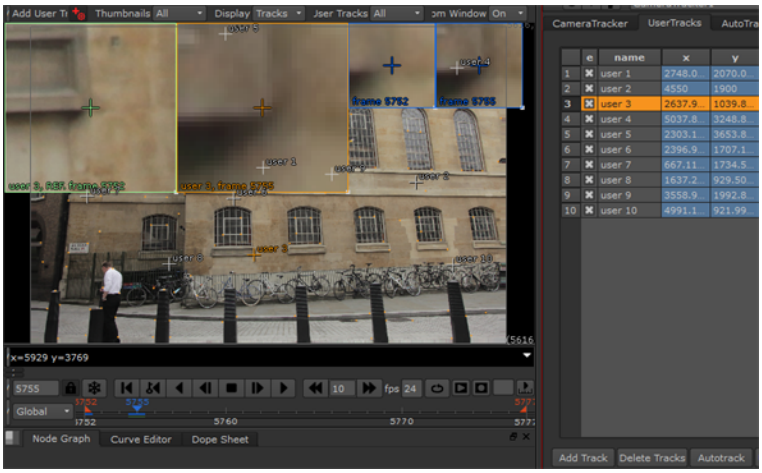
Manual tracking can produce the best results, but is time consuming because all keyframes are tracked manually by you, rather than CameraTracker.

Tip: Generally, you would only manually create User Tracks in stills. Nuke's Tracker node can create reliable tracks for continuous sequences which you can then import as User Tracks.

After placing your tracks on the required features in the sequence:

1. Click on the **UserTracks** tab in the properties panel to display the track table.
2. Select the required track from the tracks list.
3. Scrub the playhead forward a few frames and move the tracking anchor to the correct position in the new frame using the current zoom window.

   A new keyframe zoom window is added to the Viewer.



4. At the top of the Viewer, set **Thumbnails** to **User Tracks**.

   All keyframes for the currently selected user track are displayed in the thumbnail gallery.
5. To see a User Track's reference frame, hover over the track in the Viewer. This can help you quickly decide if the object you're tracking is in the current frame or not.

Tip: The keyframe windows at the top of the Viewer show you all of the keyframes for the currently selected User Track. You can click on a keyframe window to jump directly to that frame.

6. When you've finally keyframed your User Track in all of the desired frames, the tick marks on the Viewer's timeline are set on all of those keyframes. By using **Alt**+left or right arrow, you can quickly jump to the previous or next keyframe. Doing this while looking at the zoom window helps show you what keyframes your User Track is slightly off in, making this a very useful trick for getting perfectly placed User Tracks.

Extracting User Tracks

After auto-tracking, you can use the data to extract User Tracks. CameraTracker considers User Tracks to be important and tries to lock down on them over auto-tracks. As a result, extracting User Tracks from an auto-track can be useful if you have an auto-track that appears to be reliable.

You might also extract auto-tracks to User Tracks if you want to export some auto-tracks to be used in a Tracker node for 2D work.

1. Right-click on an auto-track in the Viewer and select **tracks** > **extract user track**.

   The selected track is prefixed with **extr.** and placed in the User Tracks table.
2. If a track fails or drifts, adjust the position of the tracking anchor by dragging the anchor using the zoom window.
3. Scrub through the sequence a few frames and repeat. Continue on through to the end of the sequence.
4. At each frame, a new keyframe window is added to the right of the zoom window.

Importing a Tracker

You can quickly create User Tracks by importing keyframes and tracking data from a Tracker node within the same script.

1. Click on the **UserTracks** tab in the properties panel to display the track table.
2. Click **Import Tracker**.
3. The **Select Tracker to Import From** dialog displays.
4. Select the required Tracker node from the **tracker node** dropdown.

   A new User Track is added to the tracks list.

Tracking Assists

Auto-tracking data can be improved by seeding User Tracks in a sequence, as CameraTracker assumes manually placed tracks are superior to auto-tracking. Re-tracking after adding User Tracks forces CameraTracker to recalculate auto-tracks using more accurate data.

1. Track your sequence using the steps described in Tracking in Sequence Mode or Tracking in Stills Mode
2. Track one or two User Tracks using the methods described in User Tracking Methods.

Tip: Adding tracks is time consuming, but the more User Tracks you add, the better the result is likely to be.

3. Either re-track the entire sequence by clicking **Track**,

   OR

   If you only added a few keyframes to a particularly troublesome area of the sequence, click **Update Track** to re-track just that range of frames.
4. If the tracking data requires more work, try adjusting some of the controls described in Troubleshooting Sequence Tracks orTroubleshooting Still Tracks.
5. Proceed to Solving the Camera Position.

Tracking a Scene Manually

Sometimes, a scene can be difficult to track automatically. For example, you might be tracking a set of still photos of a scene with a very wide baseline between images. Alternatively, you might already have 2D track data from an earlier process that you want to get a solve from. In cases like this, it can be easier to only set up and work with User Tracks as your track data.

The key thing to remember is that you typically need at least 8 to 16 User Tracks per frame to get a reliable solve.

1. Use the steps described in Manual User Tracks to create the required number of manual tracks.
2. Proceed to Solving the Camera Position.

Linking Still Reference Frames

During stills tracking, the links between reference frames can fail. This can occur for a number of reasons, such as a lack of overlap between stills. You can help CameraTracker by adding User Tracks on features common to the frames before auto-tracking.

1. Use the steps described in Working with User Tracks to create at least four User Tracks in as many frames as possible.

   For example, in the stills shown, the tops of the pillars and the stone work on the corner of the building.



Reference frame A



Reference frame B

2. Move to the next frame and reposition the User Tracks to the corresponding features in the still. Continue throughout the sequence of stills.
3. Once all your frames have been linked in this way, click **Track**.

CameraTracker uses the reliable information from the User Tracks to create a set of Auto Tracks.

4. Proceed to Solving the Camera Position.



Note: The above workflow describes how to use User Tracks to help connect frames that weren't automatically tracked. Sometimes, there are so many poorly tracked frames that instead of hoping that linking frames and clicking **Track** works, it might be easier to start from scratch and perform an entirely manual track. See Tracking a Scene Manually for more details.

Assigning 3D Survey Points

If you create a User Track, you can assign it as a known 3D survey point. This tells CameraTracker which points in your 2D footage go with their counterparts on your 3D model, allowing it to solve the camera to match the known 3D points and achieve the best results.



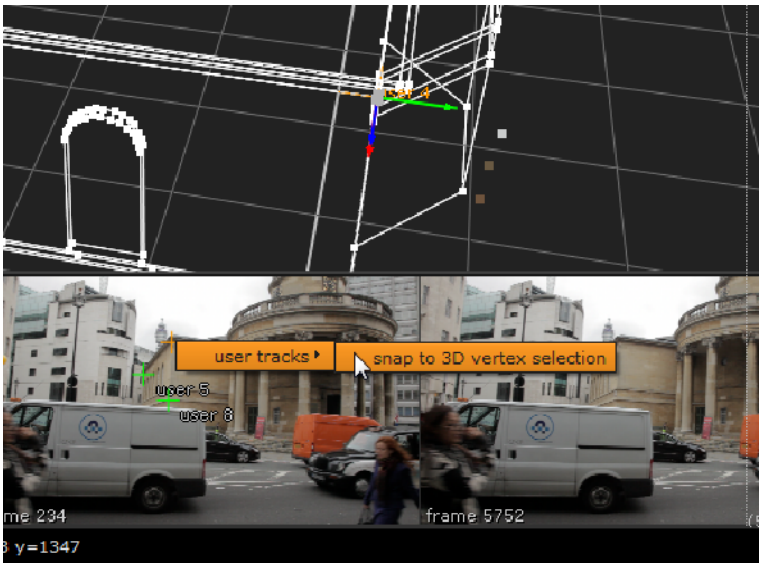
Note: 3D survey points have replaced the ProjectionSolver workflow, but you can still add ProjectionSolver nodes by pressing **X** in the Node Graph and entering **ProjectionSolver** as a Tcl command.



Tip: You can create a 3D model for your footage using ModelBuilder or an external application.

To create 3D survey points in your scene:

1. To define the full rotation, translation, and scale between the scene and your survey points, create and track at least three User Tracks. If you track more than three, you'll get the best fit result. See Working with User Tracks for more information.
2. Switch the Viewer to 3D mode, either by pressing **Tab** in the Viewer or using the **View selection** dropdown.
3. Set the **Thumbnail** control above the Viewer to **All** to display all reference frame thumbnails as an overlay in the Viewer.
4. Use the selection mode button above the Viewer to swap to **Vertex selection** mode.
5. Select a vertex on the 3D model and the User Track you want to be a survey point in the thumbnail gallery.
6. Right-click on the User Track and select **user tracks** > **snap to 3D vertex selection**.



7. In the CameraTracker properties panel, click the **UserTracks** tab to display a list of tracks.

   The user track table displays the pixel error when matching the 3D vertex to the User Track. You can use this to review how well the camera solve fits the 3D survey points.

   If the solver could not match a point, there will be a high error value. If this happens, try looking at the 2D feature positions to check they correspond to the same point and also double check the 3D vertex on the model.

8. Repeat the process for the required User Tracks in the scene, and then enable the **s** checkbox for each track in the User Tracks table.

   The **s** designates the points as known 3D survey positions during the solve, allowing CameraTracker to create a camera positioned correctly for the geometry.

Tip: If you have multiple footage sources of the same scene or content available, you can also use survey points to solve each of your sources and then register them all in the same world. See Combining Solves.

Solving the Camera Position

When you're happy with the features that you've tracked, you can solve the camera position. CameraTracker uses the tracking information to calculate the camera position and add position information to the tracked feature points in the Viewer.

In the CameraTracker properties panel, click **Solve**. The solver selects a subset of keyframes positioned relatively far apart so that there is sufficient parallax in the images to define the 3D points for the tracks.

When the solve is complete, the Viewer displays the tracked features with solve data applied.



Viewing Solve Data

Once the solve data is placed in the Viewer, you can zoom in to display the points more clearly. You can control what appears in the Viewer using the **Display** controls on the **Settings** tab:

- **Show tracks** - show or hide the 2D tracking information.
- **Show projected 3D points** - show or hide the 2D position of the 3D points.
- **Show key tracks only** - only show the longest tracks used to calculate the solve.
- **Show 3D marker** - show or hide 3D markers at each point in the 3D Viewer.

A traffic light scheme applies to the 2D tracks to help find good thresholds for track rejection in the **AutoTracks** tab - amber by default, green for good tracks, and red for poor tracks.

The circles and crosses are reprojected solved auto-tracks. The closer the 3D point circles are to the feature points, the better the solve.

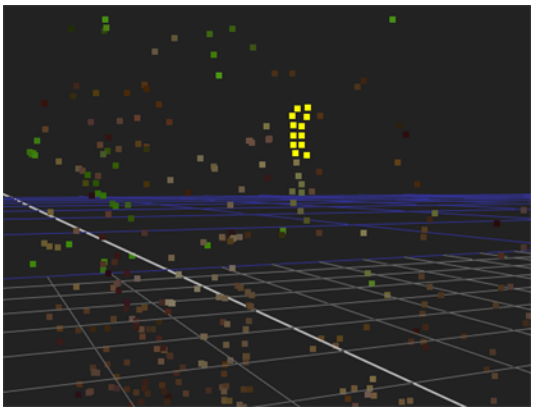
Hover over a point to display its solve information.



CameraTracker also creates a point cloud in the 3D Viewer allowing you to cross-check the relative positions of the 2D and solved 3D points. Select some points in 2D and then press **Tab** in the Viewer to switch between 2D and 3D space to check the point positions.



Several points selected in the 2D sequence.



The same points shown in the 3D Viewer.

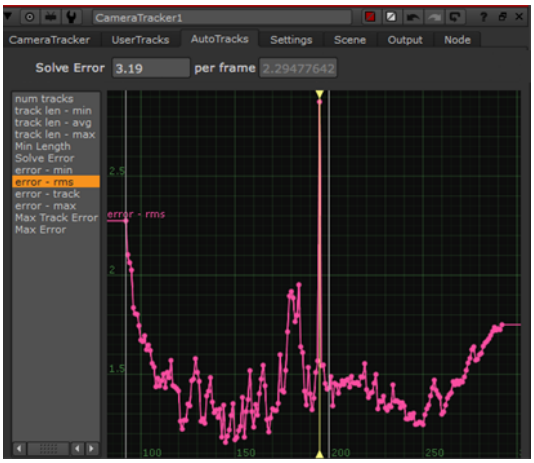
Reviewing AutoTracks Curves

Detailed solve information is displayed in curves in the properties panel on the **AutoTracks** tab. You can select all solve curves at once or get a more detailed view of a single curve, such as **error - rms** (root mean square).

Tip: Pressing **F** with focus on a curve maximizes the selected curve in the space available.



Selecting all solve curves.



Framing a single curve.

Curves can indicate areas of the sequence where solving encountered problems. For example, the root mean square error curve on the right shows a significant blip on the current frame, as indicated by the playhead position. In this instance, you could try adjusting the **Keyframe Spacing** or **Smoothness** and updating the solve. See Troubleshooting Solves

The curves in the graph show the following solve information:

- **Solve Error** - displays the constant **SolveError** parameter.
- **error min** - the minimum reprojection error at each frame (in pixels).
- **error rms** - the root mean reprojection error at each frame (in pixels).
- **error track** - the maximum root mean reprojection error calculated over the track lifetime at each frame (in pixels).
- **error max** - the maximum reprojection error at each frame (in pixels).
- **Max Track Error** - displays the constant Max RMS Error parameter.
- **Max Error** - displays the Max Error threshold parameter.

Reviewing Solved User Tracks

When a User Track is solved, you can review its error in the User Track table on the **UserTracks** tab. If the error is high, try reviewing the 2D x,y feature track. Alternatively, you can uncheck the User Track's **e** (enable) column to remove it from the scene before clicking **Solve** again.

The User Track's 3D position (X,Y,Z) is also shown in the table, as well as in the 3D Viewer.

Tip: You can create a User Track after solving the camera in order to extract a specific 3D position in the shot. To produce an accurate 3D point, the User Track must be defined in 3 or more frames with good parallax. When you've created the User Track, select it in the User Tracks table and click the **Update XYZ** button to triangulate the 3D position from the current solve.

Tip: After auto-tracking, you might want to select an auto-track and extract it as a User Track. This hints to the solver that this is an important track and should be locked down on. This is particularly useful if a particular part of the scene is not solving properly, despite the auto-tracks appearing to have tracked well in that region.

For more information, see Extracting User Tracks.

Previewing Matchmove Quality

After calculating the solve, you may want to review its quality by previewing how well objects added to the 3D scene are going to stick to your footage. If you set **Lens Distortion** to **Unknown Lens** before creating the solve, there are two ways to do this:

Method 1

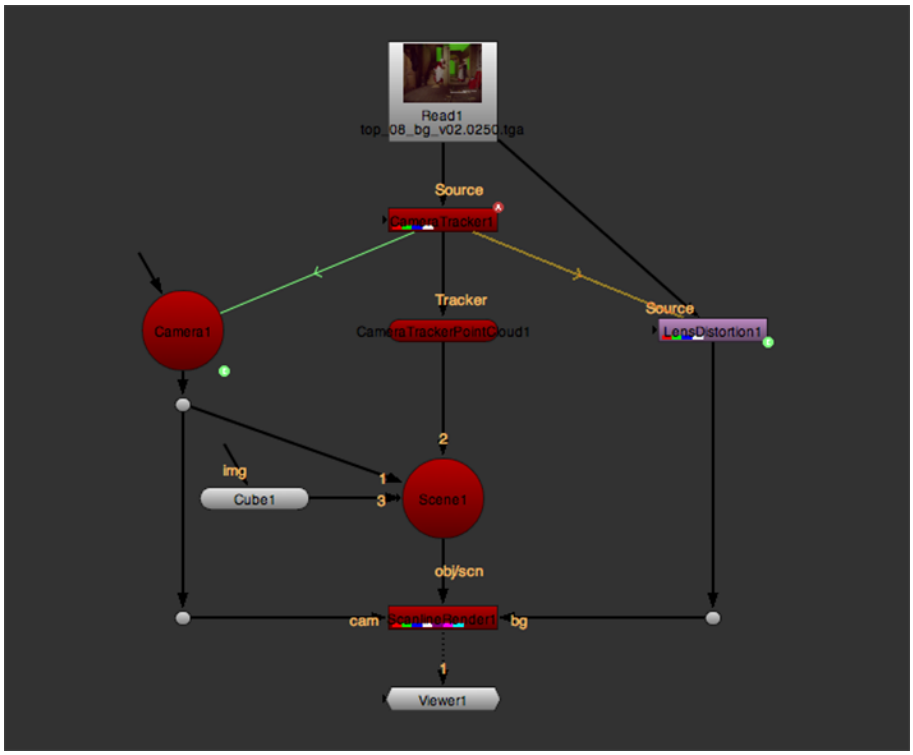
1. In the 2D Viewer, right-click on a track point (or several selected points, ideally on the same plane) and select **create** > **cube**, for example.

   CameraTracker creates a Cube node and places it in the average position of the selected points. You are going to use the cube to test how well it sticks to the input footage.

Tip: If necessary, you can adjust the size of the cube using the **uniform scale** control in the Cube properties.

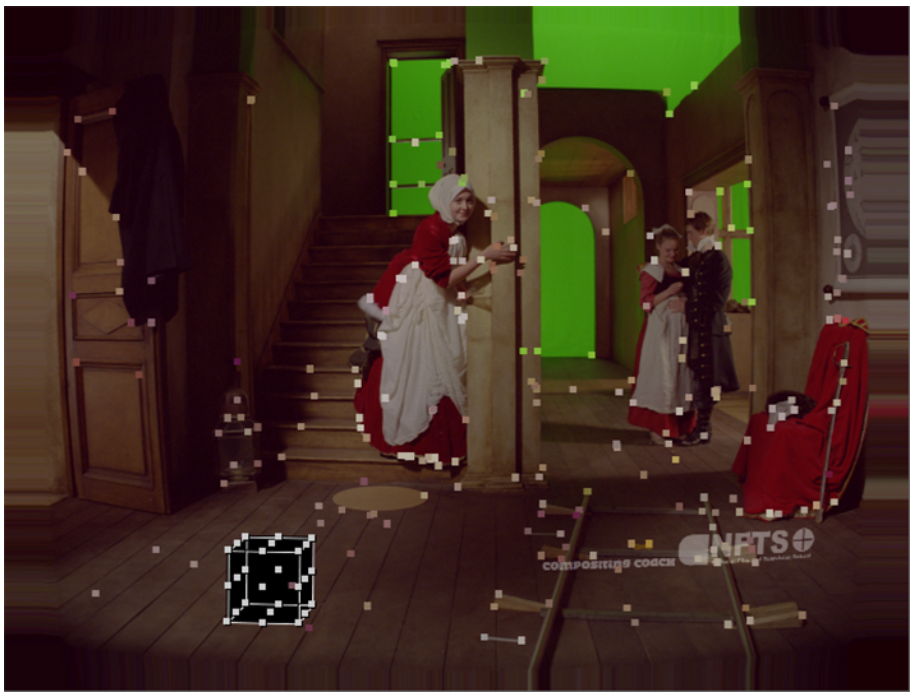
2. In the CameraTracker properties, set the **Export** menu to **Scene+** and click **Create**.

   CameraTracker creates a 3D Scene with a Camera, PointCloud, ScanlineRender, and a LensDistortion node set to undistort the input.
3. Connect your Cube node to one of the Scene node's inputs.



4. View the output from ScanlineRender.

   The cube you created in step 1 and the point cloud CameraTracker generated are displayed on top of the undistorted footage in the 2D Viewer.



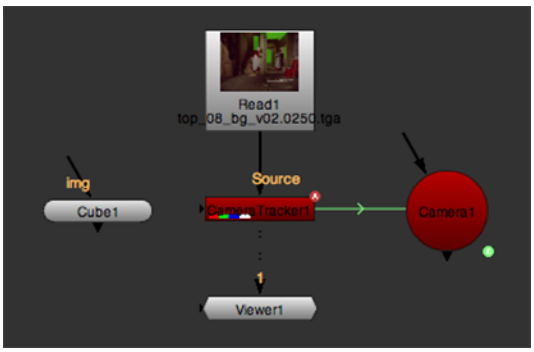
5. Scrub through the timeline to see whether the cube and the point cloud stick to the footage. If they do, delete any nodes you no longer need (such as the Cube node) and proceed to Adjusting the Scene. If they don't, proceed to Troubleshooting Solves.

Method 2

1. In the CameraTracker properties, set the **Export** menu to **Camera** and click **Create**.

   CameraTracker creates a Camera node that emulates the camera used on set.
2. In the 2D Viewer, right-click on a track point (or several selected points, ideally on the same plane) and select **create** > **cube**, for example.

   CameraTracker creates a Cube node and places it in the average position of the selected points. You are going to use the cube to test how well it sticks to the input footage.

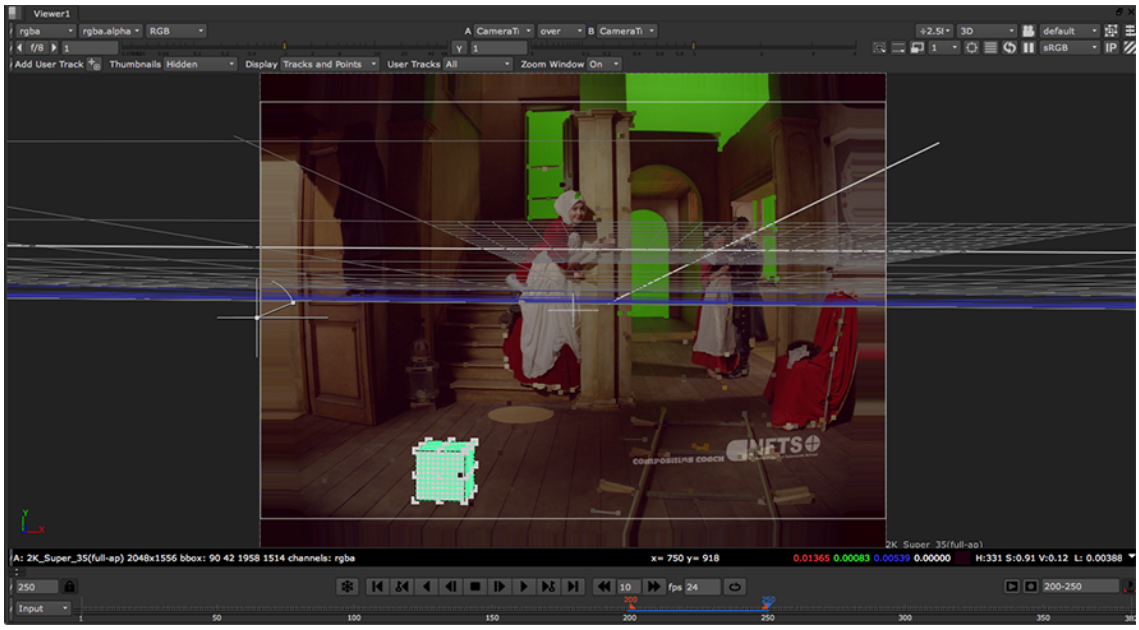


Tip: If necessary, you can adjust the size of the cube using the **uniform scale** control in the Cube properties.

3. Enable **Undistort Input**.

   CameraTracker removes lens distortion from the input footage.
4. Make sure that both the CameraTracker and Camera properties panels are open.
5. Press **Tab** on the Viewer to switch to the 3D view.
6. Set the camera look through menu in the top right corner of the Viewer to the camera you created in step 1 and click the button to lock the 3D view to that camera.
7. At the top of the Viewer, set the Viewer composite dropdown menu to **over**, so that you are viewing your CameraTracker node over the same CameraTracker node.

   This overlays the input image currently undistorted by CameraTracker over the point cloud in the 3D Viewer.



8. Scrub through the timeline to see whether the cube you created in step 2 and the point cloud CameraTracker generated stick to the footage. If they do, delete any nodes you no longer need (such as the Cube node) and proceed to Adjusting the Scene. If they don't, proceed to Troubleshooting Solves.

Troubleshooting Solves

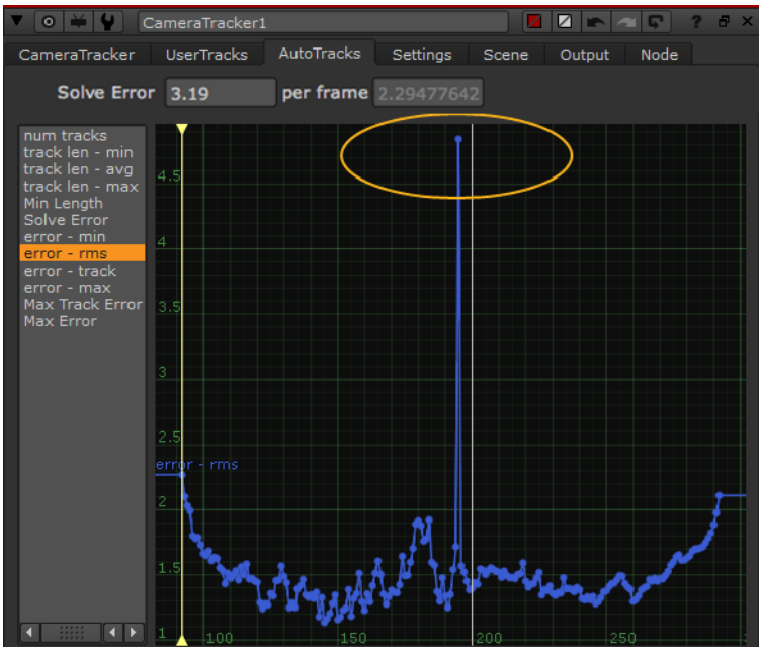
CameraTracker has several troubleshooting workflows available to improve solve accuracy, but ultimately, good solves rely on good tracking data.

Using Curve Thresholds to Delete Tracks

You can use the threshold controls on the **AutoTracks** tab to dynamically reject tracks and remove them to improve accuracy. Using the **error - rms** curve as an example:

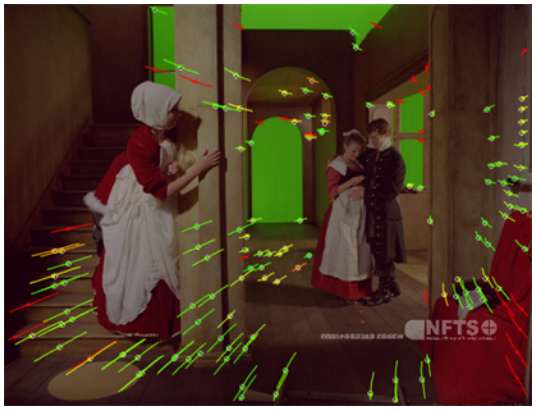
1. Click the **AutoTracks** tab in the properties and select the **error - rms** curve.

Tip: Press **F** in the curve display to fit the selected track(s) to the available screen space.



In the example, you can see that the error curve has increased significantly around frame 195.

2. Move the Viewer playhead to the affected area and examine the tracked features in the Viewer.
3. If you decrease the **Max Track Error** control in the properties panel, you'll start to see tracks turn red as they fall below the threshold.



Max Track Error set to 6.



Max Track Error set to 2.

4. Click **Delete Rejected** to remove all the tracks that fall below the specified threshold.

> **Tip:** You can also remove tracks manually, by selecting them in the Viewer, right-clicking, and choosing **tracks** > **delete selected**.

5. You can then resolve the affected frame range by clicking **Update Solve**.
6. Improve the solve further using the **Settings** tab **Solving** controls:

- **Camera motion** - set the camera movement that CameraTracker should account for during the solve calculation.
- **Keyframe Separation** - adjusts the separation value between keyframes.

  High separation values generate fewer keyframes with a greater spread and are typically used for slower camera motion.

  Low values generate more keyframes with a tighter spread and are typically used for rapid camera motion.



Note: When enabled, the **Reference Frame** control determines where the first keyframe is placed.

- **Smoothness** - higher values can help smooth a camera path by adding 'weight' to iron out the path.
- **Reference Frame** - allows you to specify the start frame for updating camera solves, as well as determining where the first keyframe is placed.



Note: Check **Set reference frame** to enable this control.

Refining a Solve

On the **AutoTracks** tab, in the **Refinement** section, there are three refinement controls to help improve your solve. If the **Error** and **per frame** controls on the **CameraTracker** tab show a relatively high value, try refining the solve using the inlier tracks. The inliers are defined by the curve thresholds and can refine the focal length, camera position or camera rotation (or a combination of these). You can manually edit the camera solve first on the **Output** tab, then select:

- **Focal Length** - Check to refine the camera's focal length.
- **Position** - Check to refine the camera position.
- **Rotation** - Check to refine the camera rotation.

Finally, click **Refine Solve**.

Updating Solves with Extended Tracking Data

CameraTracker allows you to add more frames to an existing camera solve using updated tracking data. It is usually not possible to extend the range very far (about 10-15% of the original range), because the existing solve locks the 3D points in place, so matching new 2D tracking data to 3D points can produce high tracking errors, which CameraTracker rejects.

For example, if you've calculated a solve from frames 1-10, the 3D point for a given track is based on the data from only those frames. Updating that track by extending it to frame 20 attempts to fit the 2D point from frames 11-20 to the calculated 3D point. This generally doesn't produce a good fit, and the solve error for the extended range is increased.

To update a solve with updated tracking data:

1. Track the additional frames as described in Extending Existing Camera Tracks.

   If there are a lot of rejected tracks, highlighted in red, try using fewer frames in the tracking update.
2. Click **Update Solve** to calculate the new tracking data.

   If there are a lot of rejected tracks, you may find it easier to retrack the entire range and solve the camera position from the new, complete data. Re-solving from scratch uses all of the parallax information over all frames (rather than the initially solved frames), and so you often get a lower overall solve error.

Adjusting the Scene

When the tracking and solving processes are complete, you can use the solve data to align and scale the scene before adding cameras, point clouds, objects, and so on.

You can:

- Set the ground plane, axes, or origin for the scene. See Setting the Ground Plane and Axes.
- Manually adjust the scene using the **Scene** tab's **Scene Transform** controls. See Transforming the Scene Manually.
- Use known distances, measured on set, to scale a scene more accurately. See Using Scene Constraints.

Setting the Ground Plane and Axes

A solved camera has no notion of where the ground plane is in the scene, which can produce an unexpected offset and angle relative to where you'd expect the ground to be in 3D space.

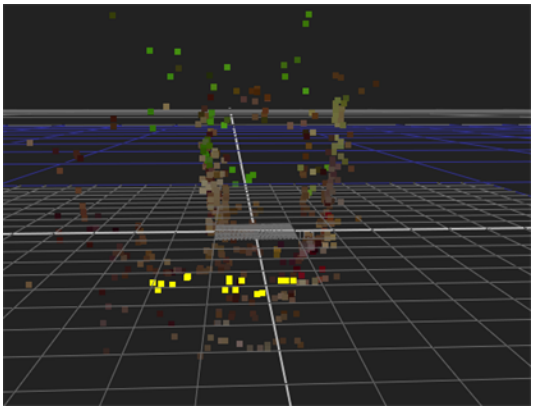
Setting the ground plane or set of axes is designed to provide CameraTracker with a sensible frame of reference. While doing so is not strictly necessary, it can simplify working in the 3D environment.

Before placing the ground plane, scan through your sequence to find a frame with good track and solve data in an area that you know to be at ground level.

1. Hover over likely points to display the track and solve data, typically points with RMS error less than 1.
2. Hold Shift and select your chosen points, from multiple frames if necessary.
3. Switch between the 2D and 3D Viewers to confirm the position of the selected points.



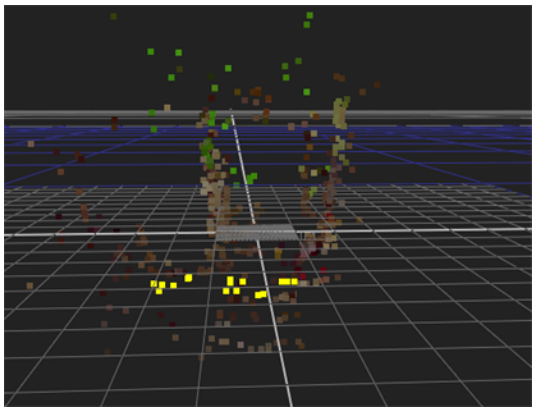
Ground plane points selected in the 2D Viewer.



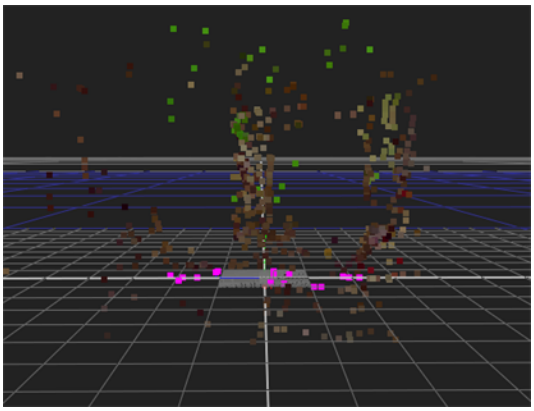
The same points viewed in the 3D Viewer.

4. Switch back to the 2D Viewer and right-click a selected point to display the track options menu.
5. Select **ground plane** > **set to selected** to orient the camera in relation to the selected points.

   Points designated as the ground plane are highlighted in magenta and the camera is reoriented in relation to the new 'ground'.



Camera orientation before the ground plane shift.



Camera orientation after the ground plane shift.

Setting Axes Individually

In some situations, you may not have a clearly defined ground plane to reference. Using CameraTracker, you can set the X, Y, and Z axes separately instead of grouping them on the ground plane.

1. Select two or more points that you know to be on the required axis using the 3D point cloud to assist you.
2. Right-click a point and select **ground plane** > **set X**, **Y**, or **Z** as required.

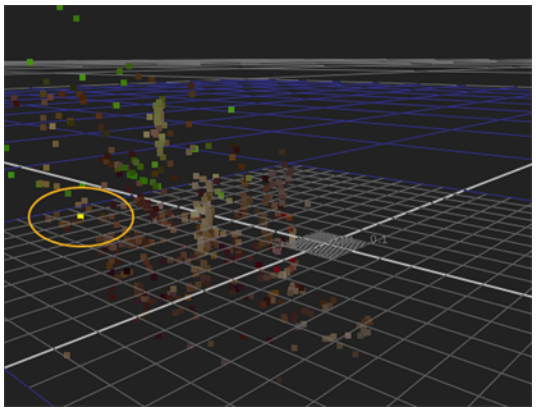
   Points designated as an axis are highlighted in yellow and the camera is reoriented in relation to the new axis.

Setting the Origin Within a Scene

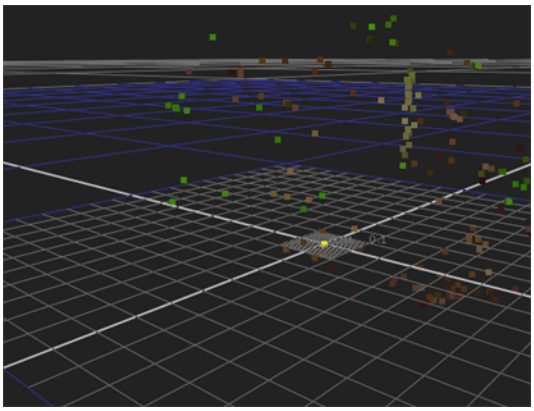
Defining the origin within a layer can assist you when placing objects in your composition - if you know where 0, 0, 0 is located, you can easily place objects precisely where you want them on any axis.

1. Select a single track as an origin point from your solve data.
2. Right-click the point and select **ground plane** > **set origin**.

   The point designated as the origin is highlighted in yellow and the point cloud is translated in relation to the new origin.



Point before the origin shift.



Point after the origin shift.

Transforming the Scene Manually

In addition to defining the ground plane, axes, and origin automatically, you can manually adjust the scene using the **Scene** tab's **Scene Transform** controls. These controls operate in exactly the same way as other transform tools in Nuke.

To transform a scene, use the following controls:

- **rotation order** - set the operation order for rotations from the dropdown menu which displays all the possible axial combinations.
- **translate** - set translate values for the scene. You can adjust translate values by clicking and dragging the axis in the 3D viewer.
- **rotate** - set values to rotate your camera or use **Ctrl/Cmd** to rotate the camera in the Viewer.
- **scale** - set scale values on individual axes, rather than uniformly or use **Ctrl/Cmd**+**Shift** to scale the camera in the Viewer.

Note: This control is only displayed if you've scaled the scene on individual axes.

- **uniform scale** - set a scale for your scene. You can also scale the scene using actual measurements taken on set, see Using Scene Constraints for more information.

Alternatively, you can also use the **Local** matrix to perform transforms on your scene.

Note: The **World** matrix is read-only and cannot be edited.

Transforming Using Channel Files

Channel files contain a set of cartesian coordinates for every frame of animation in a given shot. You can create and export them using Nuke or 3D tracking software, such as 3D-Equalizer, Maya, or Boujou.

You can transform the scene using channel files by clicking the  button:

- **Import chan file** - import a channel file and transform the input object according to the transformation data in the channel file.
- **Export chan file** - export the translation parameters that you've applied to the input object as a channel file. This is a useful method of sharing setups between artists.

Transforming Using Snap To Position

The Snap To menu allows you to match the position, orientation, and scale of existing points.

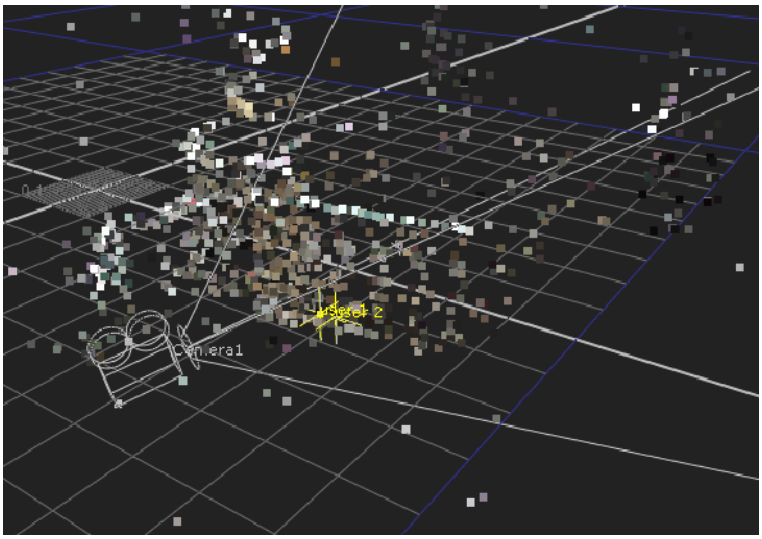
Select the required points and then click the  button:

- **Match selection position** - the scene is snapped to a new position depending on the points selected.
- **Match selection position, orientation** - the scene is snapped to a new position and orientation depending on the points selected.
- **Match selection position, orientation, size** - the scene is snapped to a new position, orientation, and size depending on the points selected.

Using Scene Constraints

Scene constraints enable you to use known distances, measured on set, to scale a scene more accurately. They are applicable to both Auto Tracks and User Tracks.

The unit of measure used is up to you, because Nuke equates any value as being equal to one unit in 3D space. The image shows a point cloud and camera with two User Tracks marked out. The blue square represents one unit of measure, so the smaller squares represent one tenth of that unit. So, if you measured on set in meters, one small square could be equal to a meter, centimeter, or millimeter.



To create scene constraints:

1. Select two solved points in the Viewer.
2. Right-click a highlighted point and select **scene** > **add scale distance**.

   The image shows User Tracks, but the same workflow applies to Auto Tracks. For example, the Auto Tracks located on the window panes could easily be measured.



   A connecting line, labled **dist. 1**, is drawn in the Viewer.
3. In the properties panel, click the **Scene** tab to display the **Scale Constraints** table.
4. Click the distance field for **dist. 1** and enter the measured distance.

   The measured distance is now taken into account when you adjust the **Scene Transform** controls.

Using Solve Data

Once your camera solve is correctly aligned, you can create animated or baked cameras, multi-view rigs, 3D scenes and point cloud, lens distortion, or cards from the camera data. See:

- Creating Camera Nodes.
- Creating Scenes.
- Creating Point Clouds.
- Creating Cards.

Creating Camera Nodes

Using the tracking and solve data, CameraTracker can create linked or baked cameras that emulate the original camera track on set. You can create a single camera or a camera rig for multi-view projects, depending on your script.

1. Select **Camera** from the **Export** dropdown menu.
2. Enable or disable the **Link output** control to determine whether the Camera node is expression linked or baked:
   - When enabled, CameraTracker creates an expression linked camera so that any adjustments made in the properties panel **Output** > **Camera** controls update the camera.
   - When disabled, any adjustments made are ignored by the camera.
3. Click **Create**.

   A Camera node is added to the Node Graph. Expression linked cameras are linked to the CameraTracker node with a green expression arrow.



Camera node data can be used in various Nuke workflows, such as Creating Dense Point Clouds and Generating Depth Maps.

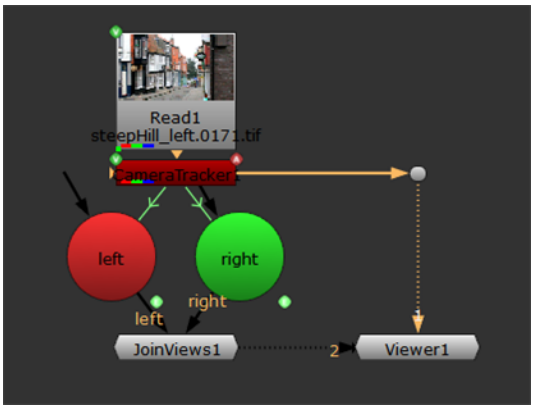
Creating a Camera Rig

CameraTracker creates as many cameras as you have views in you script, but the most common use of rigs is for stereoscopic scripts with left and right views.

1. Select **Camera rig** from the **Export** dropdown menu.
2. Enable or disable the **Link output** control to determine whether the Camera nodes are expression linked or baked:
   - When enabled, CameraTracker creates expression linked cameras so that any adjustments made in the properties panel **Output** > **Camera** controls update the cameras.
   - When disabled, any adjustments made are ignored by the cameras.
3. Click **Create**.

Note: If you try to create a rig with only one view, an error is displayed.

A Camera node is added to the Node Graph for each view present in the script. Expression linked cameras are linked to the CameraTracker node with a green expression arrow.



Camera node data can be used in various Nuke workflows, such as Creating Dense Point Clouds and Generating Depth Maps.

Creating a Camera Set

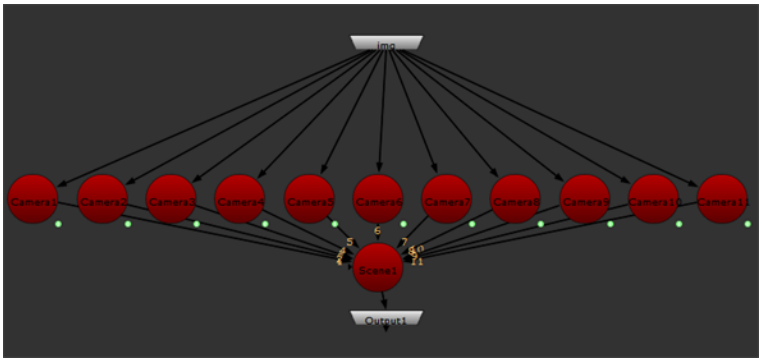
CameraTracker can automatically create a camera for each solved frame.

1. Select **Camera set** from the **Export** dropdown menu.

2. Enable or disable the **Link output** control to determine whether the cameras are expression linked or baked:
   - When enabled, CameraTracker creates expression linked cameras so that any adjustments made in the properties panel **Output** > **Camera** controls update the cameras.
   - When disabled, any adjustments made are ignored by the cameras.
3. Click **Create**.

   If you're processing a large amount of frames, a confirmation dialog displays.
4. Click **Yes** to continue or **No** to cancel the export.

   CameraTracker adds a Group node to the Node Graph containing Camera nodes for every frame specified connected to a Scene node.
5. Double click the Group to open the properties panel, then click the **S** above the panel to open the group in the Node Graph.



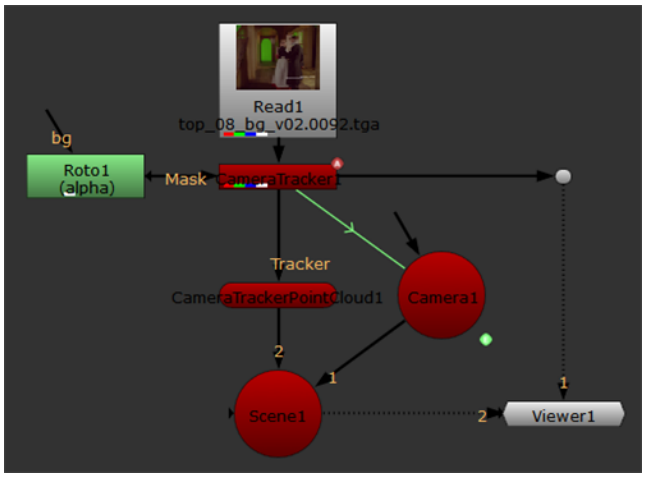
You could then write the camera data to an **.fbx** file ready for import into Modo or Maya for projection painting work.

Creating Scenes

CameraTracker can create a ready-to-use 3D scene containing a point cloud, camera, and Scene node from the track and solve data. The **Scene+** option adds LensDistortion and ScanlineRender nodes in addition to the standard scene nodes.

1. Select **Scene** from the **Export** dropdown menu.
2. Enable or disable the **Link output** control to determine whether the scene's Camera node is expression linked or baked:
   - When enabled, CameraTracker creates an expression-linked camera so that any adjustments made in the properties panel's **Output** > **Camera** controls update the camera.
   - When disabled, any adjustments made are ignored by the camera.
3. Click **Create**.

CameraTracker adds Camera, CameraTrackerPointCloud, and Scene nodes to the Node Graph. Expression linked cameras are linked to the CameraTracker node with a green expression arrow.

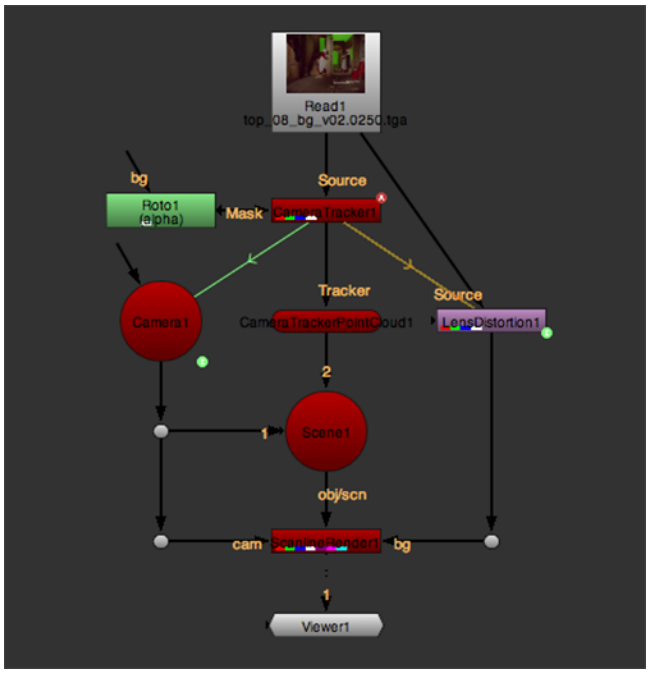


Creating a 3D Scene with ScanlineRender and LensDistortion

1. Select **Scene+** from the **Export** dropdown menu.
2. Enable or disable the **Link output** control to determine whether the scene's Camera node is expression linked or baked:
   - When enabled, CameraTracker creates an expression-linked camera so that any adjustments made in the properties panel's **Output** > **Camera** controls update the camera.
   - When disabled, any adjustments made are ignored by the camera.
3. Click **Create**.

   CameraTracker adds Camera, CameraTrackerPointCloud, Scene, ScanlineRender, and LensDistortion nodes to the Node Graph. The LensDistortion node is set to undistort the 2D footage in CameraTracker's **Source** input and use the result as the background for the 3D scene.

   Expression-linked cameras are linked to the CameraTracker node with a green expression arrow.



Creating Point Clouds

CameraTracker can create a baked point cloud, that is, points that don't update when you make changes to the CameraTracker properties panel controls.

1. Select **Point cloud** from the **Export** dropdown menu.
2. Click **Create**.

   CameraTracker adds a CameraTrackerPointCloud node to the Node Graph.
3. Double-click the CameraTrackerPointCloud node to open the properties panel.
4. Set how the points should appear, both in the Viewer and when rendered out, using the display and render dropdowns:
   - **off** - hides the 3D points.
   - **wireframe** - displays only the points.
   - **solid** - displays all points using the color of the pixel they represent.
   - **solid+wireframe** - displays all points using the color of the pixel they represent and the points themselves.
   - **textured** - displays only the surface texture.
   - **textured+wireframe** - displays the the surface texture and the points.
5. Enable or disable shadow casting and receiving when the point cloud is rendered to 2D using the checkboxes.

See Casting Shadows for more information on lighting and casting shadows.

6. Set the size of the rendered points using the **Point Size** control.

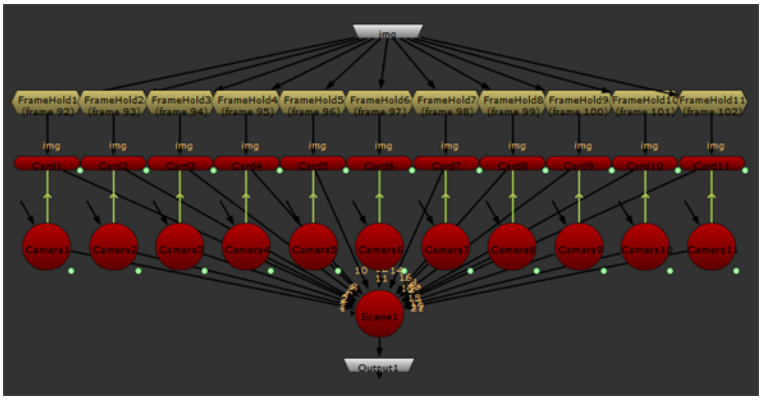
Creating Cards

CameraTracker can automatically create a 3D card for each solved frame using the camera to project the image at that frame onto the card.

1. Select **Cards** from the **Export** dropdown menu.
2. Enable or disable the **Link output** control to determine whether the cameras are expression linked or baked:
   - When enabled, CameraTracker creates expression linked cameras so that any adjustments made in the properties panel **Output** > **Camera** controls update the cameras.
   - When disabled, any adjustments made are ignored by the cameras.
3. Click **Create**.

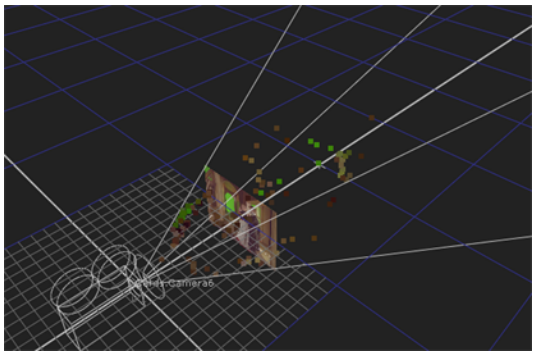
   If you're processing a large amount of frames, a confirmation dialog displays.
4. Click **Yes** to continue or **No** to cancel the export.

   CameraTracker adds a Group node to the Node Graph containing Card, FrameHold, and Camera nodes for every frame specified connected to a Scene node.
5. Double click the Group to open the properties panel, then click the **S** above the panel to open the group in the Node Graph.

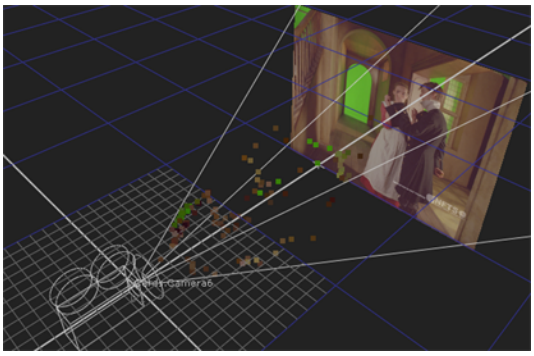


6. Use the Group node's **z** slider to control the card distance from the camera. You can use this setting to create a pan and tile dome at this distance from the camera.

   The Cards are scaled automatically using the camera settings.



The **z** depth set to 1, the default.



The **z** depth set to 3.

Combining Solves

Sometimes, you might have multiple footage sources of the same scene or content available. For example, you might have footage from witness cameras, or someone has taken detailed still photos of the scene. CameraTracker provides a way to solve each of your sources and then register them all in the same world. This allows you to leverage a high-quality camera track from a secondary source that is good to solve from, and use it in another source that's difficult to solve. You can also use this technique to tie detailed close-up stills to wide-angle shots.

The idea is that you pick one of your sources as the 'reference' solve. This should be footage that has good parallax, for example a set of wide-angle stills taken on set. You also need some geometry created from the solve data. The geometry is to establish 3D survey points, which are then used to tie together the reference and the satellite camera solves.

The workflow is something like this:

1. Pick one of your sources as the reference solve and use CameraTracker to track and solve the camera.
2. Create a linked camera using CameraTracker's **Export** menu. See Creating Camera Nodes.
3. Use the camera data to create some reference geometry. Either:
   - connect the Camera straight into the ModelBuilder node to create the geometry, or
   - export the camera data, create the geometry in an external application, then import the geometry back into Nuke.

   See Using ModelBuilder or Exporting Geometry, Cameras, Lights, Axes, or Point Clouds .
4. Set some 3D survey points on the geometry in the reference solve. See Assigning 3D Survey Points.
5. Create a separate CameraTracker for each of your satellite cameras, and track the footage as you see fit.
6. Create User Tracks in the satellite cameras, corresponding to the 3D Survey points in the reference solve, making sure to also mark them as survey points. See Assigning 3D Survey Points.

7. Solve the satellite cameras to align them consistently with the reference solve.

Tip: You can copy the X, Y, Z position from a track in one CameraTracker node to another by right-clicking on a track and selecting **copy** > **translate**.

Placing Objects in the Scene

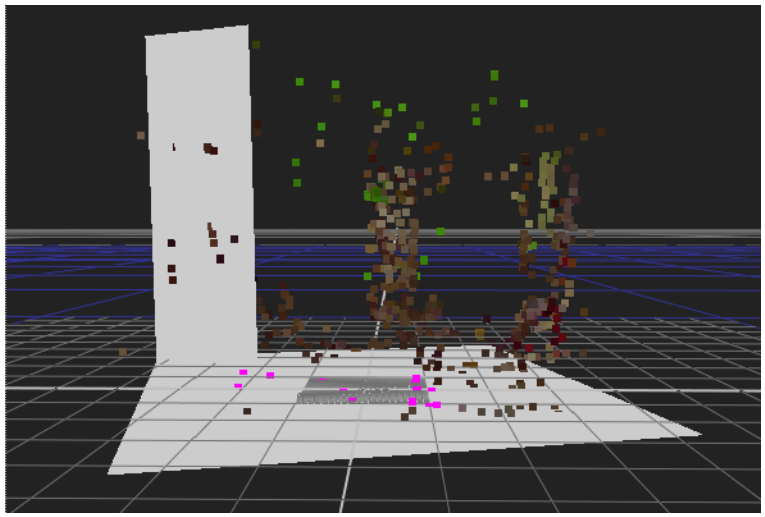
You can use the camera and point cloud to add geometry to the scene. You can add objects manually, but placing them inside the camera's field of vision at the desired position can be time consuming. CameraTracker provides an automatic creation function to help you achieve the results you need.

1. In the 2D view, select points on the plane to hold the geometry. For example, you might place a card on a vertical or horizontal plane.

Tip: You might find that swapping between the 2D plate and 3D point cloud helps to locate potential points.

2. Drag a marquee over the required points or hold down the **Shift** key and click individual points.
3. Right-click a selected point in the 2D Viewer and choose the **create** menu to display the available geometry.
4. Select the required shape to place it in the scene using the average position of all selected points.

   The following example shows two cards placed in the scene using points from the vertical and horizontal planes.



Accounting for Lens Distortion

By default, any CG elements you add to your 3D scene do not have lens distortion applied to them. As a result, when you combine them with your 2D footage, they can look like they weren't shot with the same camera. To fix this, you can:

- Use CameraTracker to calculate the lens distortion on your 2D footage and generate a LensDistortion node that applies the same distortion to your CG elements. See Distorting Your CG Elements to Match Your 2D Footage.
- Use CameraTracker to calculate the lens distortion on your 2D footage and generate a LensDistortion node that undistorts the 2D footage to match your CG elements. See Undistorting Your 2D Footage to Match Your CG Elements (Using LensDistortion).
- Use CameraTracker to both calculate the lens distortion on your 2D footage and undistort the footage to match your CG elements. See Undistorting Your 2D Footage to Match Your CG Elements (Using CameraTracker).



Note: The above assumes you set **LensDistortion** to **Unknown Lens** before creating the solve. If you set **Lens Distortion** to **No Lens Distortion** and used a separate LensDistortion node to remove lens distortion from your footage before camera tracking, you can use the same LensDistortion node to either apply the distortion to your CG elements or to undistort your 2D footage. See Working with Lens Distortion for more information on using the LensDistortion node in Nuke.

Distorting Your CG Elements to Match Your 2D Footage

1. In the CameraTracker properties, set the **Export** menu to **Distortion** and click **Create**.

   CameraTracker creates a LensDistortion node that is set to apply the lens distortion that exists in your 2D footage.
2. Connect the LensDistortion node to the output of your ScanlineRender node.

   The LensDistortion node distorts the CG elements in your 3D scene to match your 2D footage.
3. Composite your CG elements and 2D footage together as necessary.

Undistorting Your 2D Footage to Match Your CG Elements (Using LensDistortion)

1. In the CameraTracker properties, set the **Export** menu to **Undistortion** and click **Create**.
   CameraTracker creates a LensDistortion node that is set to remove the lens distortion that exists in your 2D footage.
2. Connect the LensDistortion node to the output of your 2D footage.
   The LensDistortion node removes the lens distortion in your 2D footage to match your CG elements.
3. Connect the LensDistortion node to the **bg** input of ScanlineRender.
   The undistorted 2D footage is used as a background image for the 3D scene.

Undistorting Your 2D Footage to Match Your CG Elements (Using CameraTracker)

1. In the CameraTracker properties, enable **Undistort Input**.
   CameraTracker removes the lens distortion that exists in its input footage.
2. Connect the CameraTracker node to the **bg** input of ScanlineRender.
   The undistorted 2D footage is used as a background image for the 3D scene.

About the Lens Model

Normally, you do not need to set the **Lens** parameters on the **Output** tab, as CameraTracker does this for you, but it may be helpful to know a little about the equations CameraTracker uses to account for lens distortion. There are two modes, depending on the lens type detected:

- **Spherical** - compensates for different spherical lenses. This is the simpler of the two lens corrections and uses the following equation:

$$X_u = \frac{X_d}{1 + d_1 r^2 + d_2 r^4}$$

In normalized coordinates from the distortion center, **Xu** and **Xd** are equal to the same point in an undistorted plate and a distorted plate respectively.

d1 is equal to Radial Distortion 1.

d2 is equal to Radial Distortion 2.

r2 is equal to the distance of the point from the Distortion Center.

r4 is equal to the square of r2.

- **Anamorphic** - compensates for anamorphic lenses. Anamorphic correction uses three additional parameters (**Ax**, **Ay**, and **Asq**) and requires two equations because, unlike with spherical lenses, the amount of distortion parallel to the x-axis is not the same as that parallel to the y-axis.

$$x_u = \frac{x_d}{1 + \left(\frac{d_1}{A_{sq}}\right) r^2 + \left(\frac{d_2}{A_{sq}}\right) r^4 + \left(\frac{A_x}{A_{sq}}\right) y_d^2}$$

$$y_u = \frac{y_d}{1 + d_1 r^2 + d_2 r^4 + A_y x_d^2}$$

In normalized coordinates from the distortion center, (**xu**, **yu**) and (**xd**, **yd**) are equal to the same point in an undistorted plate and a distorted plate respectively.

d1 is equal to Radial Distortion 1.

d2 is equal to Radial Distortion 2.

r2 is equal to the distance of the point from the Distortion Center.

r4 is equal to the square of r2.

Ax is equal to Asymmetric Distortion X.

Ay is equal to Asymmetric Distortion Y.

Asq is equal to Anamorphic Squeeze.

Match Grades between Shots with MatchGrade

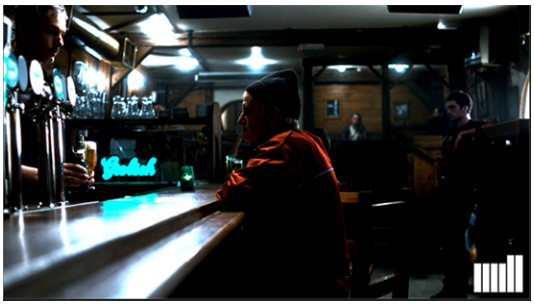
This chapter teaches you how to use MatchGrade to automatically calculate a grade to match the colors in the **Source** input to the colors in the **Target** input. You can use MatchGrade to:

- extract a baked-in grade if the **Target** clip that you want to match is simply a graded version of the **Source** clip. See Extracting a Baked-In Grade.
- match the grade between two different clips to create the same look. See Matching a Grade Between Two Different Clips.

In both cases, you can also mask the grade to only match certain elements between the **Source** and **Target** clips, and export LUT or CDL files to re-use the calculated grade elsewhere.



The **Source** image.



The **Target** image.

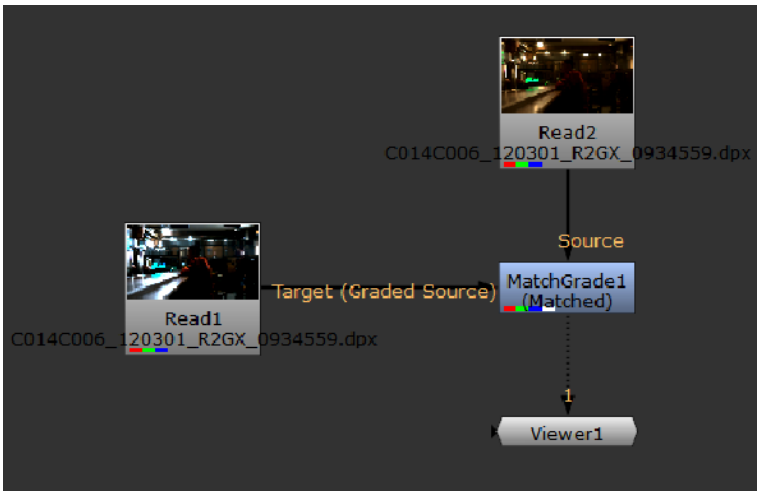


The color-matched result.

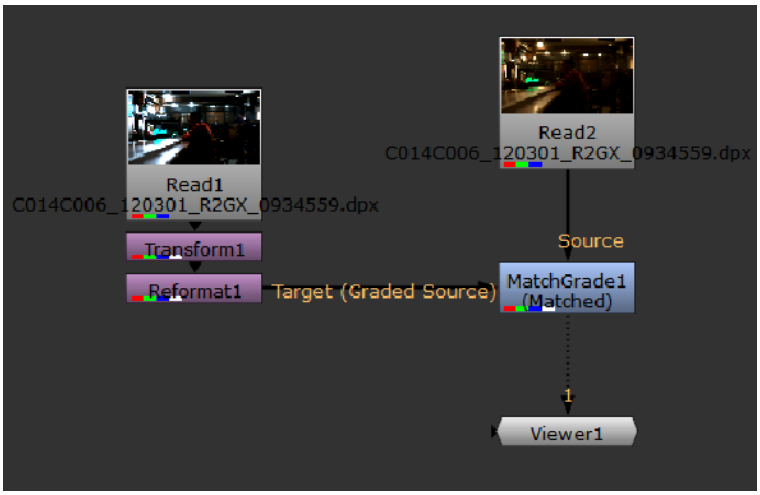
Special thanks to The Mill for use of the above footage, used throughout this chapter.

Extracting a Baked-In Grade

1. Select **Color** > **MatchGrade** to insert a MatchGrade node into your script.
2. Do the following:
   - Connect MatchGrade's **Source** input to the clip to which you want to apply the color transform.
   - Connect the **Target** input to the clip you want to match. This should be a graded version of the **Source**.
   - Connect a Viewer to the output of MatchGrade.



3. In the MatchGrade properties, make sure **Task** is set to **Match Graded Source**.
4. To extract a baked-in grade, MatchGrade requires pixel-to-pixel correspondence between the **Source** and **Target** clips. If the clips are not aligned in time and space, such as when the target clip has been reformatted, click **Align Target to Source** to add a Transform node and a Reformat node upstream of the MatchGrade node.



Tip: If the target contains a region that isn't present in the source, for example a black border, you can enable **Crop Target**. Clicking **Align Target to Source** when **Crop Target** is enabled generates a rectangular crop for the **Target** input.

5. If you want MatchGrade to calculate a single, global grade from selected reference frames that cover the characteristic colors in the sequence and apply that grade to every frame of the **Source** sequence, proceed to Analyzing Reference Frames. In this mode, you can export the grade to an OCIOFileTransform or OCIOCDLTransform node to re-use it elsewhere in the script.

   If you'd rather have MatchGrade calculate a local grade frame-by-frame, so that the color transform updates according to the current frame, proceed to Auto-Analyzing Per Frame. In this mode, you cannot export the grade to re-use it elsewhere.

Analyzing Reference Frames

1. In the MatchGrade properties, make sure **Analysis** is set to **Analyze Reference Frames**.
2. If you only want to color match certain areas of the image, make sure there is a mask in the **Source** input's alpha channel. Then, set **Mask** to either:
   - **Alpha** - to use the alpha channel as a mask. The grade is limited to the non-black areas of the alpha channel.
   - **Inverted Alpha** - to invert the alpha channel and use that as a mask. The grade is limited to the non-white areas of the alpha channel.

Note: When the **Mask** control is set to anything other than **None**, an additional **Apply Grade to Masked Region** control is displayed.

This control is enabled by default, but you can disable it to apply the grade to the whole image instead, allowing you to compute the grade from a selected region and apply it to the whole image without having to export the LUT.

3. Set **Output** to **Target**.

   MatchGrade displays the **Target** footage in the Viewer.
4. To select reference frames for MatchGrade to use in the analysis, play through the sequence. When you find frames that cover the characteristic colors in the sequence, click on the  button to set the current frame as a reference frame. If necessary, you can also use the  button to remove a frame from the set of reference frames.

   Typically, the more reference frames you set, the better MatchGrade is able to match the colors.

Tip: You can also set **Target** to **Matched** and **Analysis** to **Auto-analyze Per Frame** to preview where to set reference frames.

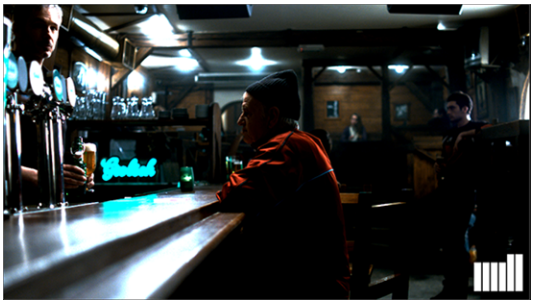
5. When you're done setting reference frames, use the **Transform** dropdown menu to choose how to calculate the grade:
   - **3D LUT** - Calculate the grade as a 3D look-up table (LUT). This allows you to export the grade to a **.csp** format, which you can use with the OCIOFileTransform node.
   - **CDL** - Calculate the grade as a color decision list (CDL). This allows you to export the grade to an OCIOCDLTransform node.

Note: The **CDL** transform is limited and cannot model all types of color transformation. In most cases, selecting **3D LUT** gives the best results.

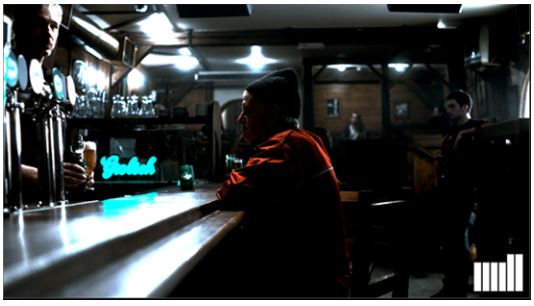
6. Set **Output** back to **Matched** and click **Analyze Reference Frames**.
   MatchGrade calculates the transform needed to match the **Source** image to the **Target** image.
7. To evaluate the results, select the MatchGrade node and press **D** repeatedly to toggle between the original **Source** image and the graded output.



The **Source** image.



The **Target** image.



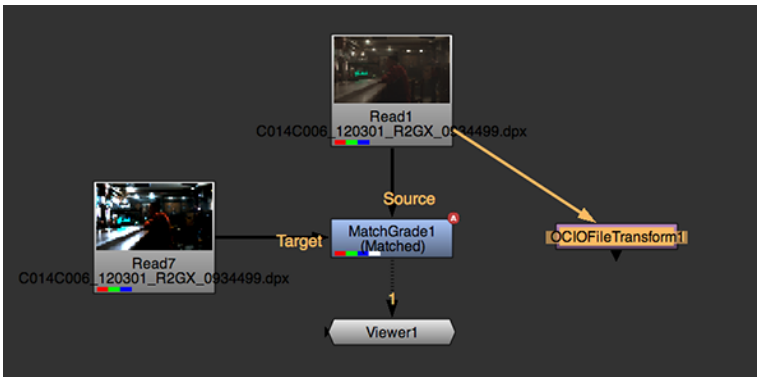
The color-matched result.

8. If you're not happy with the results, try one or more of the following:
   - Set more reference frames.

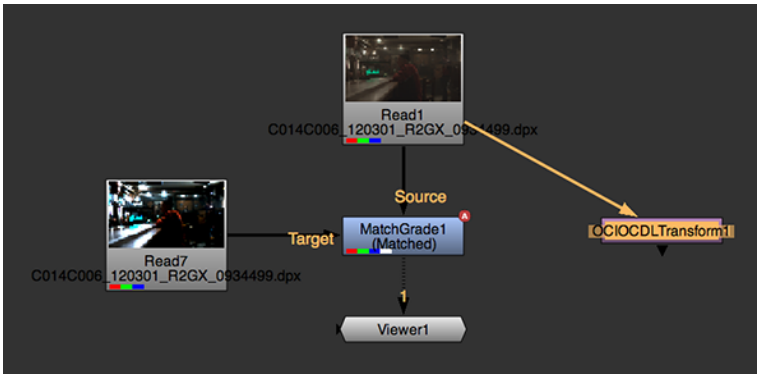
- Use the **Pre LUT** dropdown menu to specify a 1D shaper LUT to use for the analysis: **Linear** or **Logarithmic**.
- Increase the **LUT Resolution** value. This is the resolution of the look-up table (LUT) in which MatchGrade stores the color transformation. Higher values can improve the results, but also increase processing time. The maximum value is 64.

To update the results, click **Analyze Reference Frames** again.

9. If you set **Transform** to **CDL** in step 5, you can fine-tune the grade manually by adjusting **slope**, **offset**, **power**, and **saturation**. The **lock** controls on the right prevent the values from being recalculated when you click **Recalculate CDL**, allowing you to set some values manually and estimate others automatically. For example, you can adjust and lock **slope** and **offset** and then click **Recalculate CDL** to automatically recalculate **power** and **saturation**.
10. If necessary, you can export the calculated grade and apply it to other nodes in your script. How you do this depends on whether you chose to calculate the grade as a 3D LUT or a CDL:
    - If you set **Transform** to **3D LUT**, you can write the LUT into a **.csp** file and create an OCIOFileTransform node that you can use elsewhere in the script to apply the same grade. To do so, enter a file path and name for the **.csp** file in the **LUT output file** field and click **Write** to export the LUT. Then, click **Create OCIOFileTransform** to create an OCIOFileTransform node that applies the color transform from the **.csp** file.



    - If you set **Transform** to **CDL**, you can click **Create OCIOCDLTransform** to create an OCIOCDLTransform node that uses the values in the **CDL Output** section of the MatchGrade properties.



Auto-Analyzing Per Frame

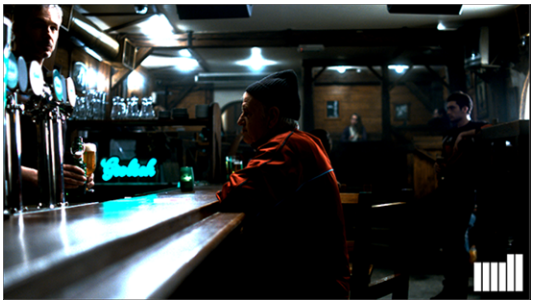
1. In the MatchGrade properties, set **Analysis** to **Auto-analyze Per Frame**.

   MatchGrade calculates the transform needed to match the **Source** image to the **Target** image.
2. If you only want to color match certain areas of the image, make sure there is a mask in the **Source** input's alpha channel. Then, set **Mask** to either:
   - **Alpha** - to use the alpha channel as a mask. The grade is limited to the non-black areas of the alpha channel.
   - **Inverted Alpha** - to invert the alpha channel and use that as a mask. The grade is limited to the non-white areas of the alpha channel.
3. To evaluate the results, select the MatchGrade node and press **D** repeatedly to toggle between the original **Source** image and the graded output.

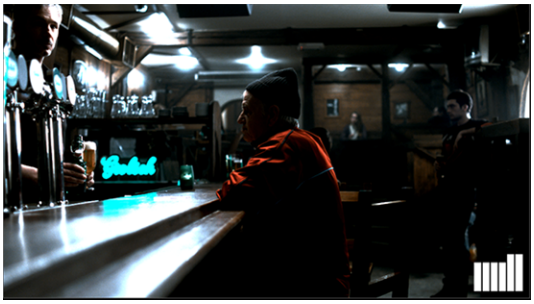
   Alternatively, you can use the **Output** dropdown menu in the MatchGrade properties to choose what to display in the Viewer:
   - **Matched** - View the color matched result.
   - **Source** - View the **Source** input.
   - **Target** - View the **Target** input.



The **Source** image.



The **Target** image.



The color-matched result.

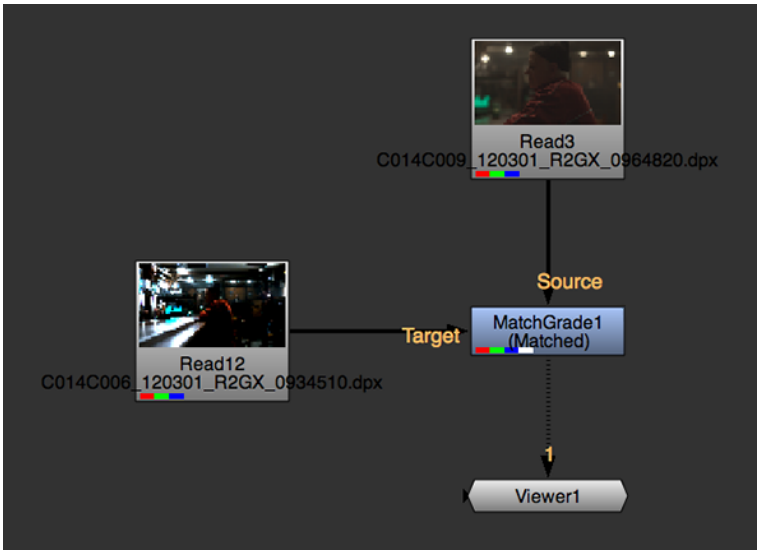
4. If you're not happy with the results, try increasing the **LUT Resolution** value. This is the resolution of the look-up table (LUT) in which MatchGrade stores the color transformation.

   Higher values can improve the results, but also increase processing time. The maximum value is 64.

5. If necessary, you can also use the **Pre LUT** dropdown menu to specify a 1D shaper LUT to use for the analysis:
   - **Auto Detect** - Automatically detect the best pre-LUT to use.
   - **Linear** - Use a linear pre-LUT.
   - **Logarithmic** - Use a logarithmic pre-LUT.

Matching a Grade Between Two Different Clips

1. Select **Color** > **MatchGrade** to insert a MatchGrade node into your script.
2. Do the following:
   - Connect MatchGrade's **Source** input to the clip to which you want to apply the color transform.
   - Connect the **Target** input to the clip you want to match. This should be a completely different clip, a stereo pair for the **Source** clip, or the **Source** clip in a different format.
   - Connect a Viewer to the output of MatchGrade.



3. In the MatchGrade properties, set **Task** to **Match Different Clip**.
4. Set **Output** to **Source** and then **Target** to compare the **Source** and **Target** images.
5. When you find frames that cover similar content in the **Source** and **Target** clips, click on the  buttons next to **Source** and **Target** to set the current frame as a reference frame. Only frames set as reference frames are used in the MatchGrade analysis.

For example, if you want grass in the **Source** clip to have the same color as grass in the **Target** clip, you need to choose frames where the relative amount of grass pixels is roughly the same. The more similar the content, the better MatchGrade is able to match the colors.

Note: You do not need to use the same number of **Source** and **Target** reference frames or set the same frames as reference frames for both.

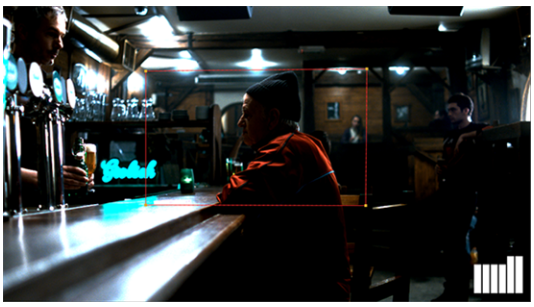
If necessary, you can also use the  button to remove a frame from the set of reference frames.

6. If you only want to match certain elements between the **Source** and **Target** clips, supply a mask in both the **Source** and **Target** inputs' alpha channels and set **Mask** to either:
   - **Alpha** - to use the alpha channel from both inputs as a mask. Only the non-black areas of the **Source** and **Target** inputs' alpha channels are used in the MatchGrade analysis, and the grade is limited to the non-black areas of the **Source** alpha.
   - **Inverted Alpha** - to use the inverted alpha channel from both inputs as a mask. Only the non-white areas of the **Source** and **Target** inputs' alpha channels are used in the MatchGrade analysis, and the grade is limited to the non-white areas of the **Source** alpha.

   Masks can also be useful if you cannot find frames that cover approximately the same amount of similar content. For example, if the **Source** and **Target** inputs have a different amount of grass and sky, you can use a Roto node in both inputs to create masks that cover the same amount of grass and sky. Bear in mind that if you do so, the grade is only applied to the areas indicated by the mask in the **Source** input. To apply it to the entire frame, you need to export the grade first. For more information, see step 12.



The **Source** image. In this case, you can set this image to have a fully opaque alpha...



...and draw a mask like this on the **Target** image to ensure both inputs cover similar content.

7. When you're done setting reference frames, use the **Transform** dropdown menu to choose how to calculate the grade:
   - **3D LUT** - Calculate the grade as a 3D look-up table (LUT). This allows you to export the grade to a **.csp** format, which you can use with the OCIOFileTransform node.
   - **CDL** - Calculate the grade as a color decision list (CDL). This allows you to export the grade to an OCIOCDLTransform node.

Note: The **CDL** transform is limited and cannot model all types of color transformation. In most cases, selecting **3D LUT** gives the best results.

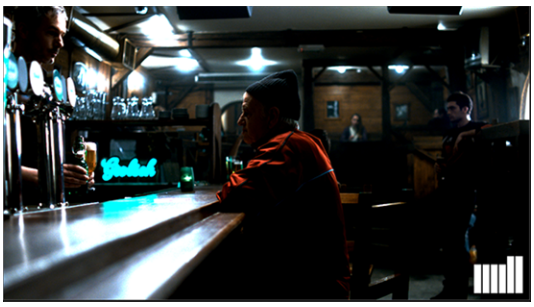
8. Set **Output** back to **Matched** and click **Analyze Reference Frames**.

   MatchGrade calculates the transform needed to match the **Source** image to the **Target** image.

9. To evaluate the results, select the MatchGrade node and press **D** repeatedly to toggle between the original **Source** image and the graded output.



The **Source** image.



The **Target** image.

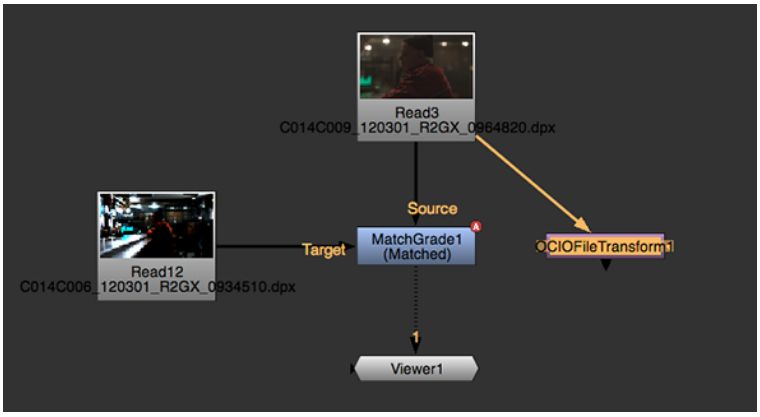


The color-matched result.

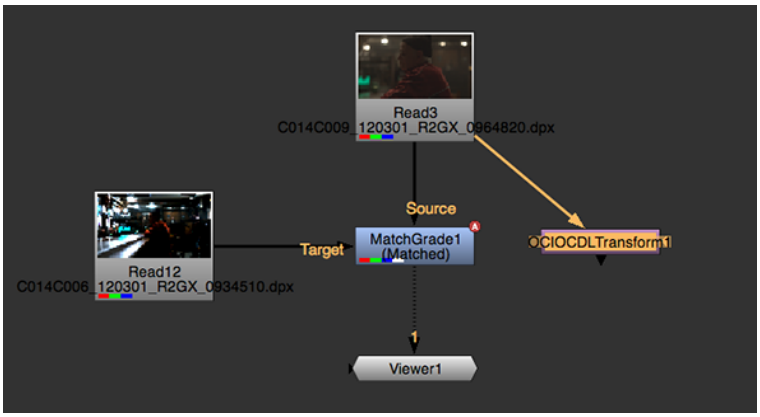
10. If you're not happy with the results, try one or more of the following:
    - Look for better reference frames or use a mask in the **Target** input to make sure both inputs cover approximately the same amount of similar content.
    - Use the **Pre LUT** dropdown menu to specify a 1D shaper LUT to use for the analysis: **Linear** or **Logarithmic**.
    - Increase the **LUT Resolution** value. This is the resolution of the look-up table (LUT) in which MatchGrade stores the color transformation. Higher values can improve the results, but also increase processing time. The maximum value is 64.
    - Set **Colorspace** to a different value. The correct setting to use depends on the nature of the transformation. Try each option to see which works best with your footage.
    - Increase the **Iterations** value. This determines the number of refinement passes. Higher values can produce a better color match, but also take longer to process.

    To update the results, click **Analyze Reference Frames** again.

11. If you set **Transform** to **CDL** in step 8, you can fine-tune the grade manually by adjusting **slope**, **offset**, **power**, and **saturation**. The **lock** controls on the right prevent the values from being recalculated when you click **Recalculate CDL**, allowing you to set some values manually and estimate others automatically. For example, you can adjust and lock **slope** and **offset** and then click **Recalculate CDL** to automatically recalculate **power** and **saturation**.
12. If necessary, you can export the calculated grade and apply it to other nodes in your script. How you do this depends on whether you chose to calculate the grade as a 3D LUT or a CDL:
    - If you set **Transform** to **3D LUT**, you can write the LUT into a **.csp** file and create an OCIOFileTransform node that you can use elsewhere in the script to apply the same grade. To do so, enter a file path and name for the **.csp** file in the **LUT output file** field and click **Write** to export the LUT. Then, click **Create OCIOFileTransform** to create an OCIOFileTransform node that applies the color transform from the **.csp** file.



    - If you set **Transform** to **CDL**, you can click **Create OCIOCDLTransform** to create an OCIOCDLTransform node that uses the values in the **CDL Output** section of the MatchGrade properties.



Tip: You can also apply the grade to a different source by simply connecting a new node to MatchGrade's **Source** input.

Using the Smart Vector Toolset

The Smart Vector toolset allows you to work on one frame in a sequence and then use motion vector information to accurately propagate work throughout the rest of the sequence. The vectors are generated in the SmartVector node and then piped into the VectorDistort node to warp the work. That way, you only need to render the motion vectors once.



Note: All images in this chapter are courtesy of Hollywood Camera Work. See www.hollywoodcamerawork.com for more information.

Quick Start

The toolset workflow is outlined below:

1. Connect the SmartVector node to the source sequence to generate the required motion vectors. You can also write them to file using the **.exr** format. See Generating Motion Vectors.
2. Find a suitable reference frame and add the required paint corrections or an image to the frame. See Adding Paint to the Source or Adding an Image to the Source.
3. Connect the VectorDistort node to the SmartVector node (or a Read node referencing the **.exr** files rendered from the SmartVector node) and the source sequence.
4. Premultiply the VectorDistort output and merge it over the source sequence to complete the comp. See Applying Motion Vectors to the Source.

Generating Motion Vectors

The SmartVector node generates motion vectors for use in the VectorDistort node. You can connect a SmartVector node directly to the VectorDistort or VectorCornerPin nodes or write motion vectors to the **.exr** format, which are then used to drive the warp nodes to reduce overheads later on.

To generate motion vectors:

1. Read in your source sequence and then connect a SmartVector node to the Read node.

2. Double-click the SmartVector node to open its **Properties** panel, if it's not already open.
3. Set the **Vector Detail** control to the required detail level. The default value of **0.3** is sufficient for sequences with low detail and movement, but you may want to increase the detail to improve the vector quality in some cases.

   For example, if the area you're working on is relatively small, the default value of **0.3** might not capture movement properly. Try increasing the control to **1.0** to capture more detail.



Note: High detail vectors take longer to render, but can improve the results you get from the VectorDistort node.

4. Set the **Strength** control to force pixel matching between frames. Higher values allow you to accurately match similar pixels in one image to another, concentrating on detail matching even if the resulting motion field is jagged. Lower values may miss local detail, but are less likely to provide you with the odd spurious vector, producing smoother results.



Note: The default value works well for most sequences.

5. If there is a lot of movement in the foreground, you might want to add a mask. See Masking Foreground and Background Areas for more information.

Tip: You can examine the vectors produced by connecting a Viewer to the SmartVector node and switching the **channels** control above the Viewer to **smartvector_<value>**.

6. If you want to write the vectors to disk, click **Export Write** to automatically add a Write node to the script. The Write node's controls are set to **channels** > **all** and **.exr** output automatically.

   You can only write motion vectors to the **.exr** format. Don't forget to add frame padding in the form of hashes or printf notation, depending on your **Preferences** > **Behaviors** > **File Handling** settings.
7. Enter a **file** path in the Write node's controls and then click **Render**.



Note: SmartVector does not currently output **motion**, **forward**, and **backward** channels by default. If you require these channels, add a VectorToMotion node after the SmartVector node. VectorToMotion converts the vectors to motion that can be used with VectorBlur to create motion blur, without using a VectorGenerator.

8. Enable **Advanced** > **Flicker Compensation** to reduce variations in luminance and overall flickering, such as light reflecting from shiny surfaces, which can cause problems with your vector output.

9. **Tolerances** in the **Advanced** section allow you to tune the weight of each color channel when calculating the image luminance.

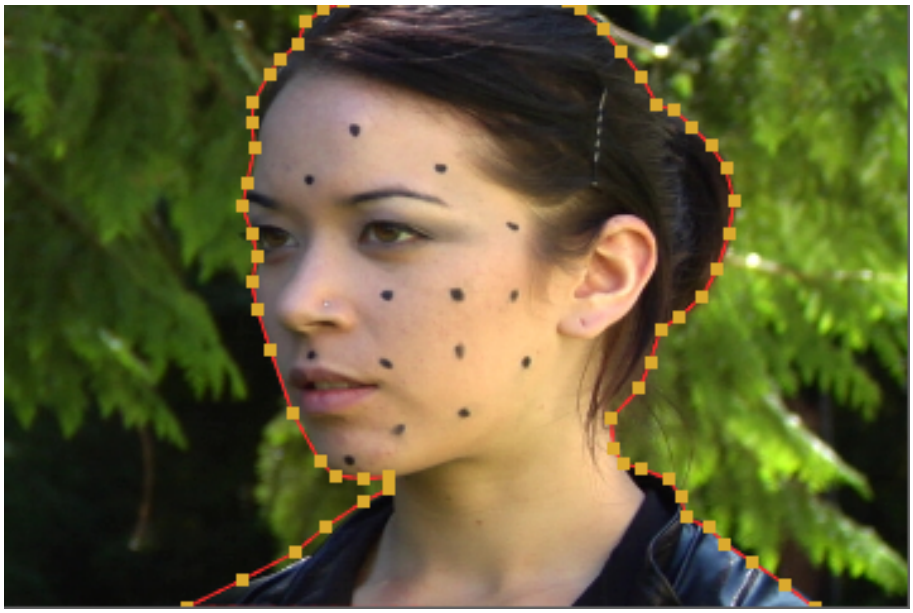
   These parameters rarely need tuning, but you may wish to increase the weighting on a particular color to allow the algorithm to concentrate on getting the motion for a certain object correct, at the cost of the rest of the items in a shot.
10. Proceed to Adding Paint to the Source or Adding an Image to the Source.

Masking Foreground and Background Areas

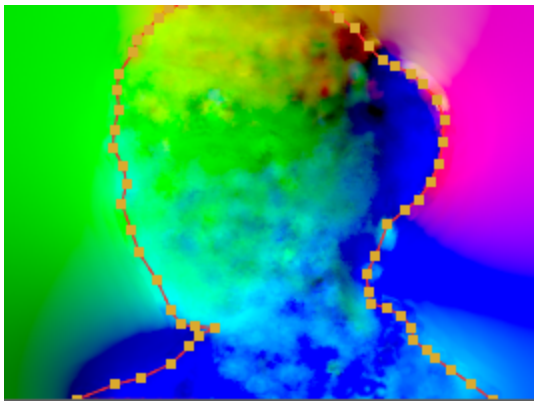
If your sequence is composed of a foreground object moving over a background, the motion estimation is likely to get confused at the edge between the two. To reduce artifacts, you can add an alpha mask to the **Matte** input.

To apply a mask to the sequence:

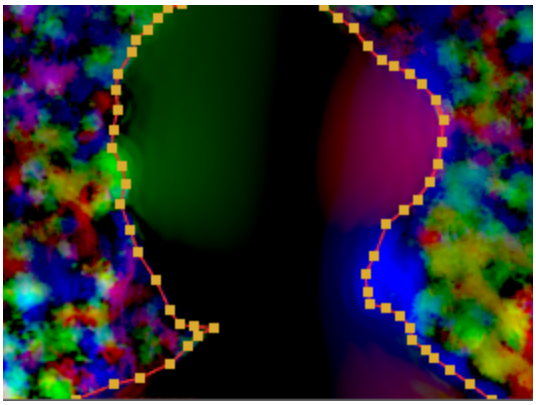
1. Add a Roto node downstream of the source.
2. Draw a matte around the area you want to mark as foreground or background.



3. Set the **Matte Channel** to **Matte Alpha** and the **Output** control to either **Foreground** or **Background**.

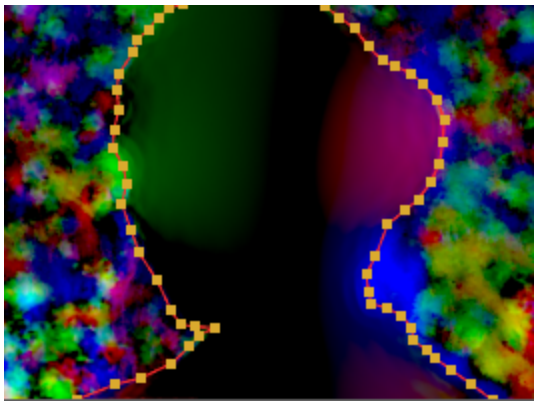


Foreground vectors

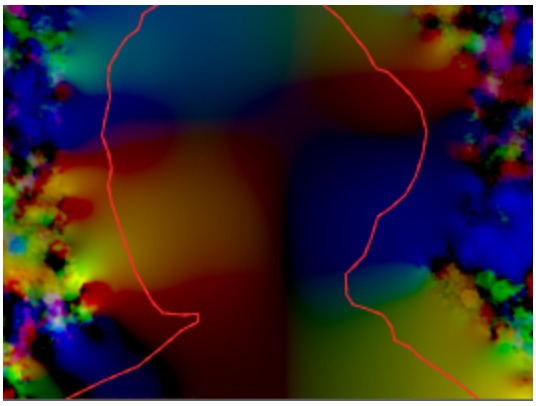


Background vectors

4. If you're working with **Background** vectors, you can enable **Inpaint Matte Region** to automatically fill in the missing foreground vectors using the nearest available background vectors. The **Matte Dilation** slider controls amount of dilation applied to the matte before inpainting is applied.



Background vectors



Inpainted foreground

You can use inpainting to effectively ignore the foreground, allowing you to work with background vectors in the masked area.

5. Generate the motion vectors as described in Generating Motion Vectors.

Adding Paint to the Source

The Smart Vector toolset minimizes the paint work needed to create your final comp by propagating paint through a sequence from a single source frame.

To add paint to your sequence:

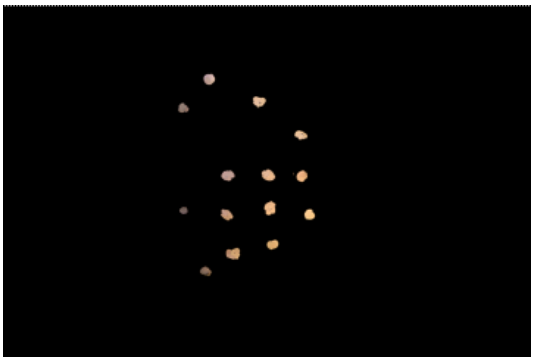
1. Scrub through your sequence to find a good reference frame. Good reference frames should:
   • be at the point of least motion, also known as the motion apex,

- contain minimal motion blur, and
- avoid areas of occlusion.

2. Add a RotoPaint node and apply your paint corrections to the selected reference frame. Ensure that all your paint is sourced from the background image by navigating to **Stroke** > **source** and selecting **background** once a stroke has been added.



Original image



Premultiplied background paint



Merged result

3. Proceed to Applying Motion Vectors to the Source.

Adding an Image to the Source

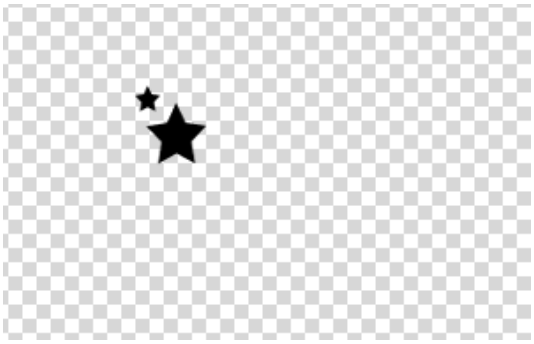
The Smart Vector toolset can also propagate an image through a sequence from a single source frame.

To add an image to your sequence:

1. Scrub through your sequence to find a good reference frame. Good reference frames should:
   - be at the point of least motion, also known as the motion apex,
   - contain minimal motion blur, and
   - avoid areas of occlusion.
2. Merge your image into the node tree at the selected reference frame.



Original image



Premultiplied image



Merged result

3. Proceed to Applying Motion Vectors to the Source.

Applying Motion Vectors to the Source

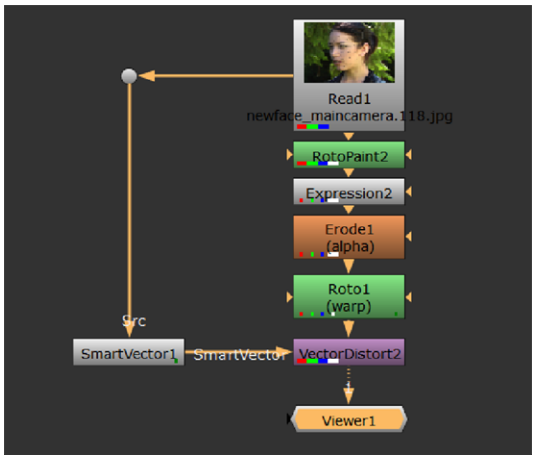
Once you've generated your motion vectors and added your paint to the source sequence, the VectorDistort node takes the paint from the reference frame and propagates it through the rest of the sequence using the motion vectors from the SmartVector node.

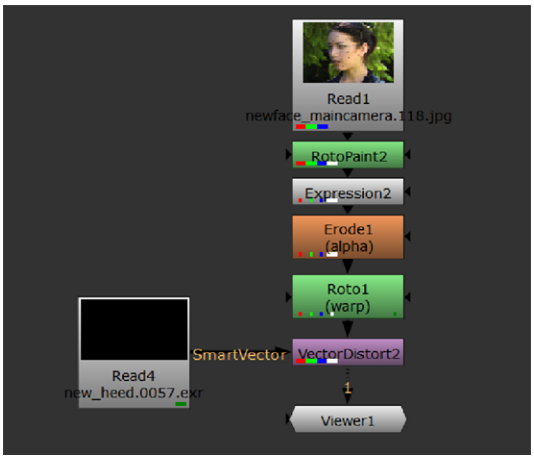
Warping the Vectors

After generating your vectors, applying your corrections, and masking out areas of the sequence you don't want to process, use VectorDistort to push your work through the rest of the sequence.

1. Add a VectorDistort node to the Node Graph and connect the **Source** input to your sequence, downstream of the paint or image you intend to propagate.

2. Connect the **SmartVector** input to either the SmartVector node or a Read node referencing the **.exr** files containing the motion vector data.





3. Double-click the VectorDistort node to open its **Properties** panel, if it's not already open.
4. Scrub to the frame containing your paint or image and click **set to current frame** to set the VectorDistort reference frame.

Tip: If you know what frame your paint or image is applied to, you can type the frame number into the **Reference Frame** control manually.

Scrubbing in the Viewer at this stage introduces undesirable warping. As you move farther from the reference frame, the warping increases.



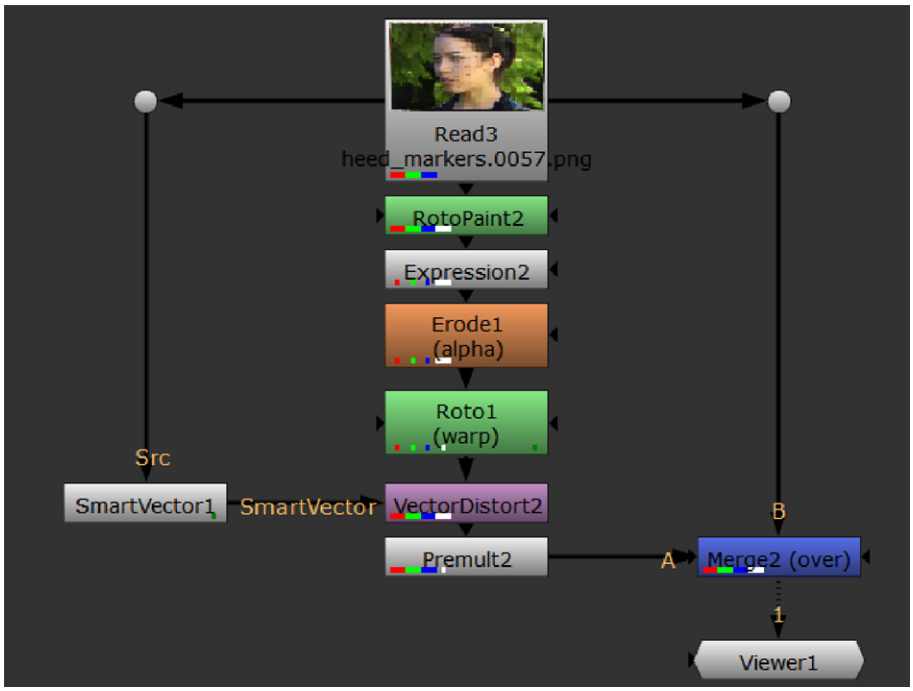
Warp close to the reference frame.



Warp far from the reference frame.

5. Set the maximum **Frame Distance** over which vectors are calculated. Sequences with rapid motion typically require values closer to **1 frame**, whereas sequences with slower motion typically require values closer to **64 frames**.

   See Improving Warps for more information on the **Frame Distance** control.
6. To apply the warped paint or image to the original sequence, premultiply the output from the VectorDistort nodes and then merge the result over the source.



For short sequences, or sequences with minimal movement and detail, your corrections are propagated nicely as shown.





If you see that your corrections warp incorrectly over time, you can use VectorDistort's properties to massage the results (see Improving Warps) or use the VectorCornerPin node or multiple VectorDistort nodes to apply warps for several reference frames (see Warping Images Using VectorCornerPin and Warping Multiple Reference Frames).

Improving Warps

For short sequences, or sequences with minimal movement and detail, your corrections are propagated nicely. If you see that your corrections warp incorrectly over time, you can use VectorDistort's **Frame**

Distance control to massage the results.

Tip: Warping relies on good vectors from SmartVector and a good reference frame in VectorDistort. Before doing anything else, try increasing the **Vector Detail** control in the SmartVector node and re-rendering the vectors. You can also try selecting a different **Reference Frame** in the VectorDistort node's controls.

See Generating Motion Vectors or Applying Motion Vectors to the Source for more information.

At **Frame Distance** > **1 frame**, VectorDistort calculates warp for every frame in the sequence, that is, frames 1-2, 2-3, 3-4, 4-5, and so on. For frames close to the **Reference Frame** with lots of movement between frames, this is a good thing as the warp needs to change significantly from frame to frame. However, for frames farther away from the **Reference Frame** with little movement between frames, this isn't required as the warp doesn't change significantly from frame to frame.

Increasing the **Frame Distance** essentially reduces the number of warps calculated between the **Reference Frame** and the current frame. In the example sequence, the **Reference Frame** is set to 57, and the warp at frame 230 has started to slip on the subject's forehead with **Frame Distance** set to **1 frame**. Increasing the **Frame Distance** works well up to around **2 frames**, but increasing the distance distorts the warp too much.



The warp with frame distance **1 frame**.



The warp with frame distance **2 frames**.



The warp with frame distance **4 frames**.

For longer sequences with local distortion, you can try increasing the **Blur Size** control to blur the internally calculated STMap. Increasing the **Blur Size** can remove local distortions in the warped result, particularly in longer sequences.

In some cases, no amount of adjustment is going to improve the warp over a large number of frames - particularly if there's a lot of movement or detail in the sequence. See Warping Images Using VectorCornerPin and Warping Multiple Reference Frames for more information on how to use VectorCornerPin and multiple VectorDistort nodes to minimize manual paint work.

Warping Images Using VectorCornerPin

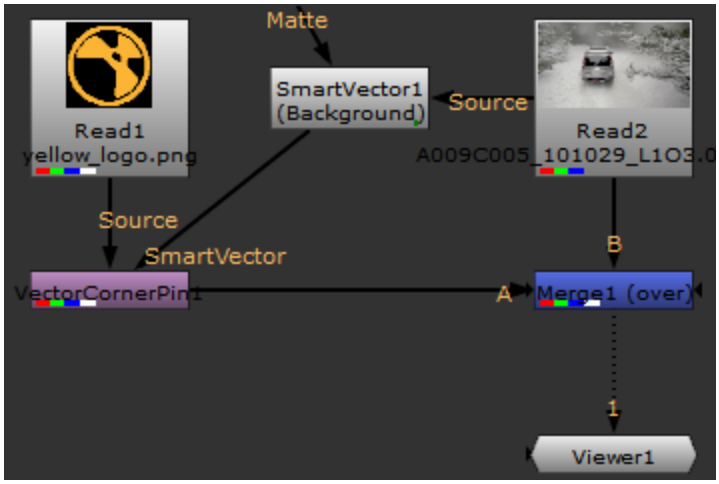
Some sequences involve movement and detail that can't be propagated from a single frame correction. In these cases, you can use the VectorCornerPin node to minimize the amount of correction work needed. The **to1-4** Viewer widgets allow you to add keyframes, which then use vector information from SmartVector to drive the warp.



The final result should be an image with minimal slip and distortion.



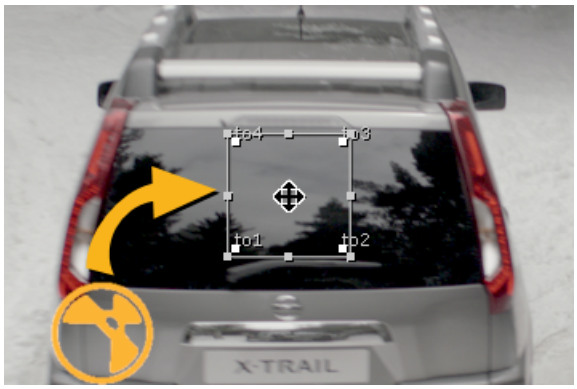
1. Set up your image as described in Adding an Image to the Source.
2. Connect the VectorCornerPin's **SmartVector** input to either a SmartVector node or a Read node referencing the **.exr** files containing the motion vector data.





Tip: If the image you're warping displays at the wrong resolution, switch to the VectorCornerPin's **From** tab, click **Set To Input**, and then switch back to the **VectorCornerPin** tab and click **Copy 'from'** to resize it automatically.

3. Scrub to the reference frame in your sequence and place the image as required using the **to1-4** Viewer widgets.



4. In the VectorCornerPin **Properties** panel, click add keyframe to set a keyframe at the current frame.

Note: You can remove the keyframe at the current frame using or remove all keyframes from the sequence using .

5. Play through the sequence until the image begins to slip or warp.

Tip: After adding a keyframe, you can click **Bake Corners** to calculate the positions of the pins at each frame in a range using the vectors from the **SmartVector** input.





6. Adjust the image using the **to1-4** Viewer widgets. A new keyframe is added at the current frame.

   Frames on which no keyframe is present are interpolated automatically, so you may not need to adjust each frame individually.
7. You can smooth distortion on interpolated frames using the **Frame Distance** and **Blur Size** controls on the **SmartVector** tab.



Frame Distance **1 frame** and Blur Size **0**.

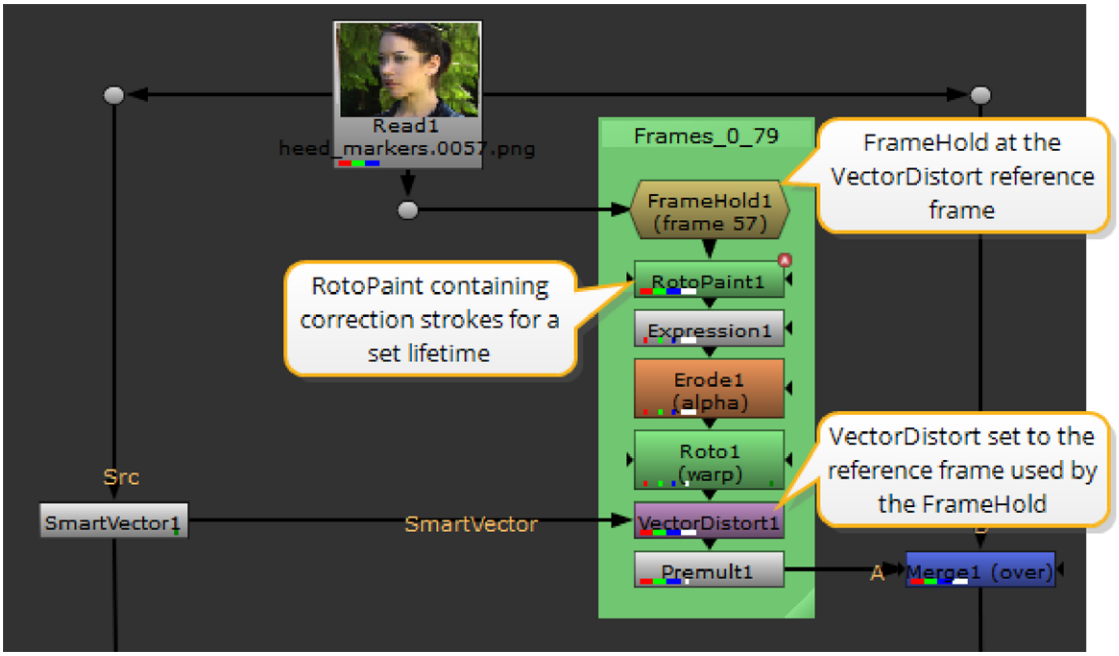


Frame Distance **4 frames** and Blur Size **10**.

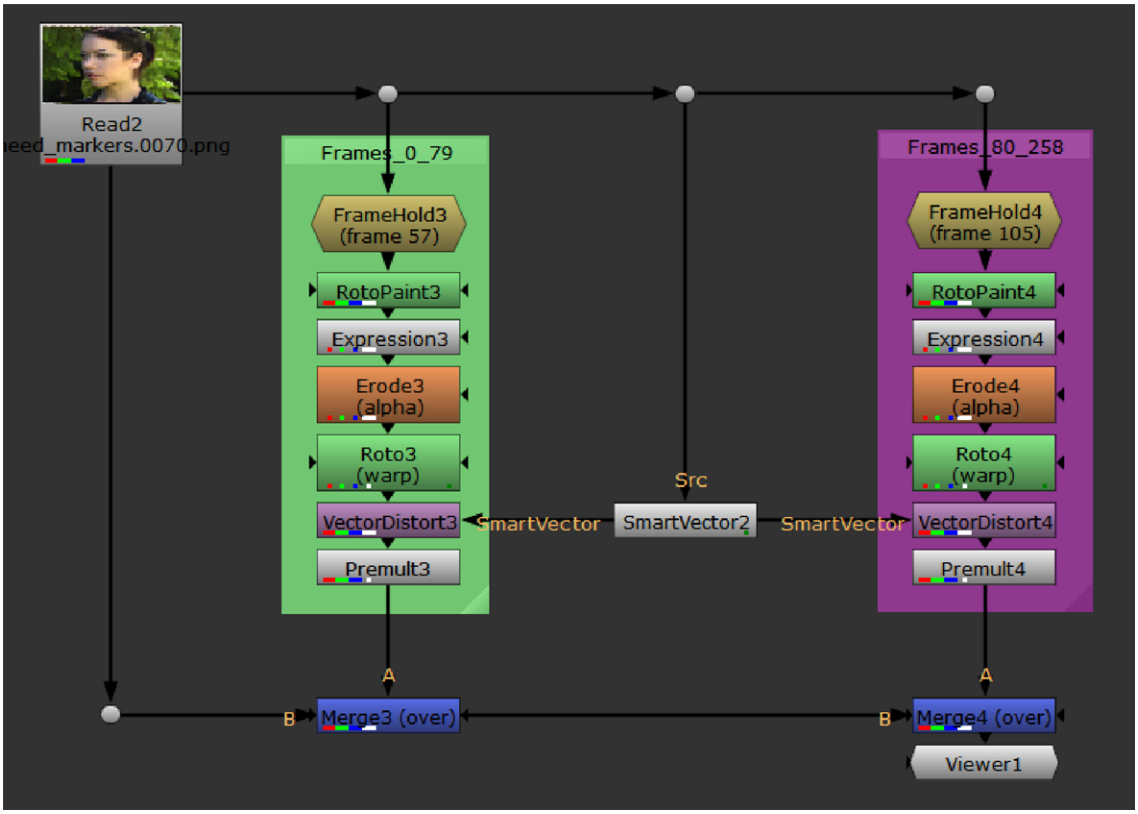
Warping Multiple Reference Frames

Some sequences involve movement and detail that can't be propagated from a single frame correction. In these cases, you can use multiple VectorDistort nodes with different reference frames to minimize the amount of correction work needed.

1. Set up your first paint correction as described in Adding Paint to the Source and Applying Motion Vectors to the Source.
2. In the VectorDistort node's properties, disable the **Hold Frame** control.
3. Add a FrameHold node before the RotoPaint node containing your corrections.
4. Set the **first frame** control to the **reference frame** specified in the VectorDistort node.
5. Set the **lifetime** of your paint strokes in the RotoPaint node **Lifetime** tab to the same frame range as the VectorDistort node.



6. Repeat the process for the required number of reference frames to complete your corrections. An example node tree might appear as follows:



Generating Depth Maps

You can use the DepthGenerator node in NukeX to generate a depth map from your footage. The node uses information from a tracked camera to create a channel that displays variations in depth.A depth map is an image that uses the brightness of each pixel to specify the distance between the 3D scene point and the virtual camera used to capture the scene.



A source image.



The generated depth map.

DepthGenerator also allows you to output depth as normals and position passes, and create a Card node positioned in 3D space and displaced according to the depth channel.

Note that DepthGenerator can only create a depth map from a sequence, so you can't use a still image.

Quick Start

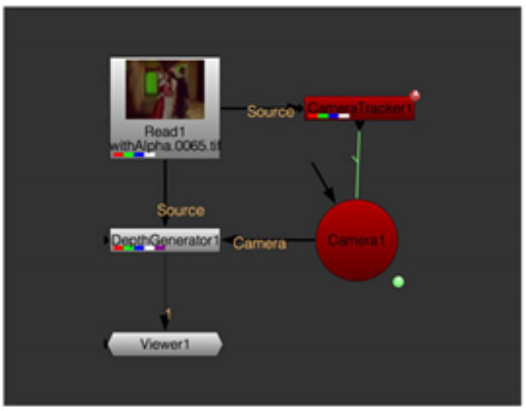
Here's a quick overview of the workflow:

1. Add DepthGenerator to your NukeX script. See Connecting DepthGenerator.
2. Select the type of depth map you want to output, and whether to also output normals and position passes. See Selecting What to Output.
3. Analyze the depth values for your footage. See Analyzing Depth.
4. Review and refine the results. See Refining the Results.
5. Use the results with other Nuke and NukeX nodes. See Using the Results.

Connecting DepthGenerator

To connect DepthGenerator:

1. To use the DepthGenerator node, you need a tracked camera that matches your footage. If you don't already have one, you can create one with the CameraTracker node (see Camera Tracking).
2. Create a DepthGenerator node by clicking **3D** > **DepthGenerator.**
3. Attach a Viewer to the output of DepthGenerator.
4. Connect the **Source** input to your footage and the **Camera** input to your Camera node.



A simple DepthGenerator node tree.

5. Depth can only be calculated where the real world 3D position of objects doesn't change. If there are moving objects in your **Source** footage, DepthGenerator is likely to struggle to create an accurate depth map for those regions. To prevent this, you can exclude moving foreground regions from the depth calculation by connecting a matte to the **Mask** input and setting **Ignore Mask** in the DepthGenerator properties to the channel that contains the matte. Note that DepthGenerator expects values of either 0 (for regions to use) or 1 (for regions to ignore).

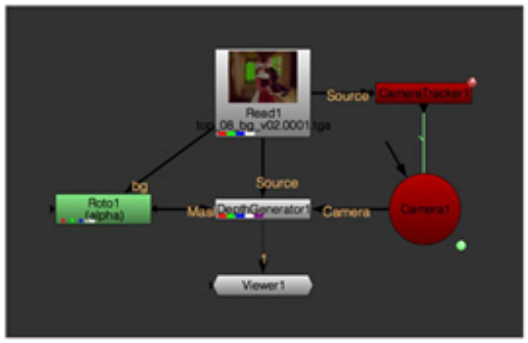


A Source image.



An ignore mask.

The **Mask** input only appears once you've connected the other two inputs.



Using the **Mask** input.

6. Proceed to Selecting What to Output.

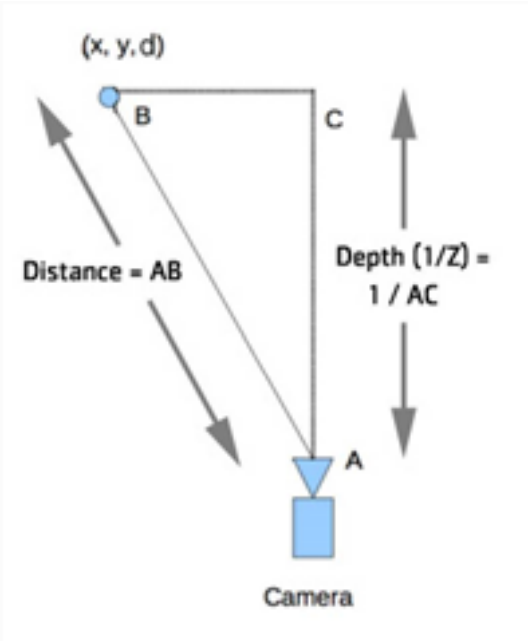
Selecting What to Output

To select what to output:

1. In the DepthGenerator properties, use **Depth Output** to select the type of depth map you want to generate:
   - **Depth (1/Z)** - output 1/Z where Z is the distance along the Z axis for the camera. This matches the depth output of the ScanlineRender node.

     This mode also allows you to later create a Group node with a Card positioned in 3D space and displaced according to the depth channel.
   - **Distance** - output the distance along the ray from the camera center to the 3D surface point.

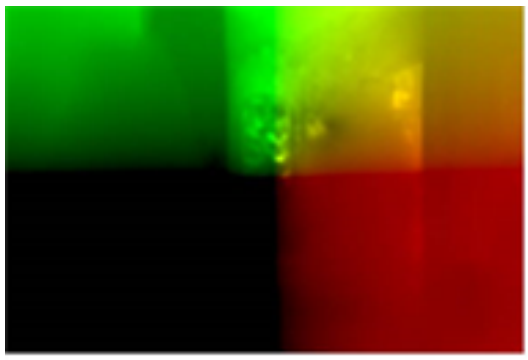
   In the figure below, each pixel forms a ray, and AB measures the physical distance from the camera to the 3D point, whereas AC measures the distance along the Z axis for the camera.



Calculating the depth.

2. If you also want to output depth as a position pass, set **Surface Point** to the channels where you want to store this (for example, you could create a new layer called **ppass**, which contains the channels **X**, **Y**, and **Z**).

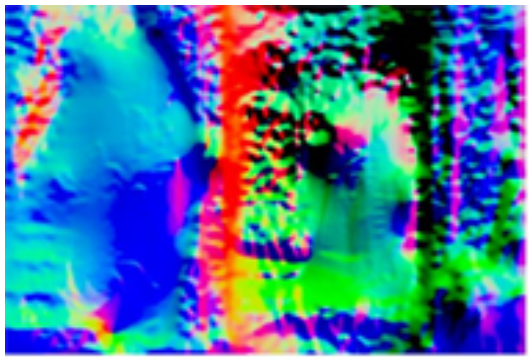
   The position pass includes the X, Y, and Z coordinates for each pixel in the image. You can use it with the Relight node, or with the PositionToPoints node to visualize depth as a point cloud.



A position pass.

3. Similarly, to output depth as a normals pass, set **Surface Normal** to the channels where you want to store the normals pass (for example, create a new layer called **npass**, which contains the channels **X**, **Y**, and **Z**).

   The normals pass contains three vectors of information for each pixel in the image: X direction, Y direction, and Z direction. In other words, it stores the direction in which each point in the image is facing. You can use a normals pass with the Relight node.



A normals pass.

4. Proceed to Analyzing Depth below.

Analyzing Depth

To analyze the depth:

1. Do one of the following:
   - Use the **Frame Separation** control to select the offset between the current frame and the frame against which to calculate depth for your footage. For example, if your current frame is 100, and your frame separation is 2, DepthGenerator uses frame 98 and frame 102 to generate the depth map.

     Ideally, you want frames close together so that the images are similar and include the same bits of the world. However, you get more accurate depth when the frames are further apart and the baseline between the cameras is bigger. So, for fast camera moves, you need a small frame separation, and for slow camera moves, you can use a larger frame separation. To change the separation for fast and slow movements, you can animate the **Frame Separation** control.

     Note that this control does NOT affect the number of frames used in the calculation, as that is always 2.
   - To have DepthGenerator attempt to calculate the best **Frame Separation** automatically, click **Analyze Sequence**. For this to work, the camera in the **Camera** input must be defined for all frames within the frame range.

     DepthGenerator goes through the entire sequence, attempts to work out the correct frame separation, and animates the **Frame Separation** value.
   - Alternatively, you can also click **Analyze Frame** to have DepthGenerator automatically calculate the frame separation for the current frame. This gives you more control than **Analyze Sequence**, as you can work through the timeline, analyze particular frames, and if necessary tweak the **Frame Separation** value manually.
2. Scrub through the timeline and keep an eye on the **Calculated Accuracy** value. This displays the depth accuracy calculated when analyzing frame separation. Values closer to 1 are considered accurate, and values closer to 0 inaccurate.

You can use the frames that produce accurate values later when placing elements in 3D (for example, by clicking **Create Card** or using a PositionToPoints node).

If you're not getting accurate depth values, try adjusting **Frame Separation** or using an **Ignore Mask**.

3. View the depth channel in 2D by selecting it from the list of channels in the Viewer. To see more details in the depth map, it can be a good idea to temporarily adjust the gain and gamma controls in the Viewer.

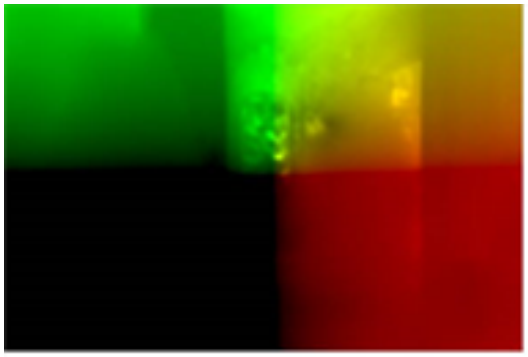


A depth map.

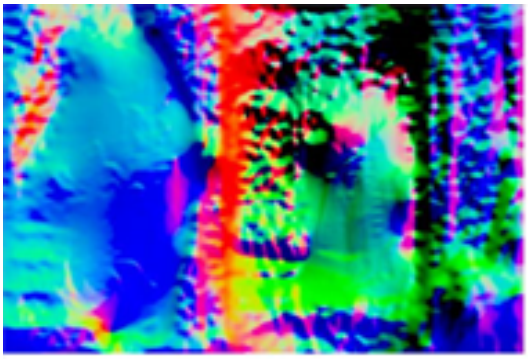


The same depth map after adjusting Viewer gain.

4. If you have set **Surface Point** or **Surface Normal** to generate position and normals passes, you can also view those by selecting them from the list of channels in the Viewer.



A position pass.



A normals pass.

5. Proceed to Refining the Results.

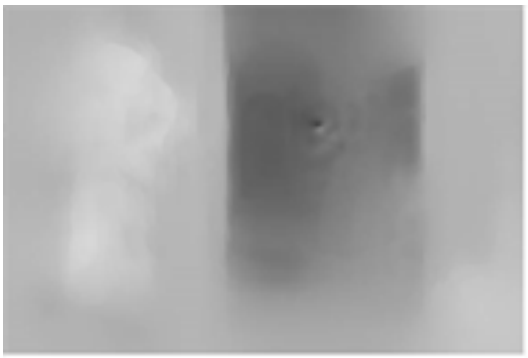
Refining the Results

To refine the results:

1. Adjust **Depth Detail** to vary the resolution of the images used to calculate the depth map. The default value of 0.5 equals half the image resolution. Lower values speed up processing and deliver a smoother result. Higher values pick up finer details, but also increase processing time.

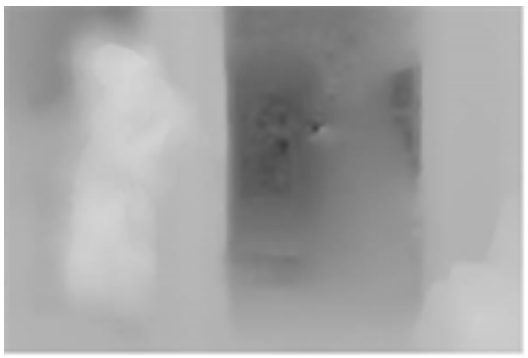


Depth Detail = 0.15



Depth Detail = 0.85

2. If you find that the depth map is noisy in flat image regions, try increasing the **Noise** value in the **Depth Generation** parameter group. This sets the amount of noise DepthGenerator should ignore in the input footage when calculating the depth map. The higher the value, the smoother the depth map.



Noise = 0



Noise = 0.15

3. If you think the depth map should still be smoother, you can also increase **Smoothness**. Rather than affecting the depth calculation, this applies an intelligent blur on the result.

   A high smoothness can miss lots of local detail, but is less likely to produce noisy depth values. A low smoothness concentrates on detail matching, even if the resulting depth map is jagged.

4. The **Strength** parameter defines the strength of pixel matching between frames. You can usually leave it at the default value. However, if the depth map does not appear to fit some edges of the image, you can increase **Strength** to force matches where fine details are missed. You can also reduce **Strength** to smooth the depth map, but generally it's a good idea to leave it at a relatively high value to make sure the algorithm is matching the images and calculating the depth correctly.

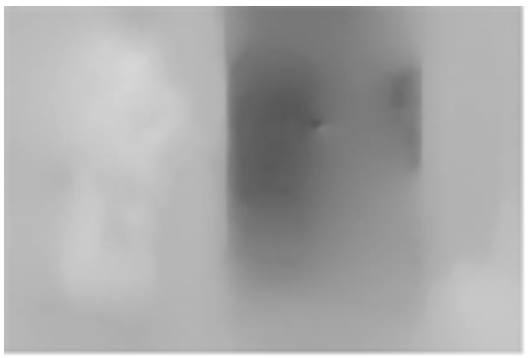


Strength = 0.5



Strength = 1.5

5. If you aren't happy with the object boundaries in your depth map, adjust **Sharpness**. Use a high value to produce distinct boundaries between the objects, or a low value to smooth the boundaries.



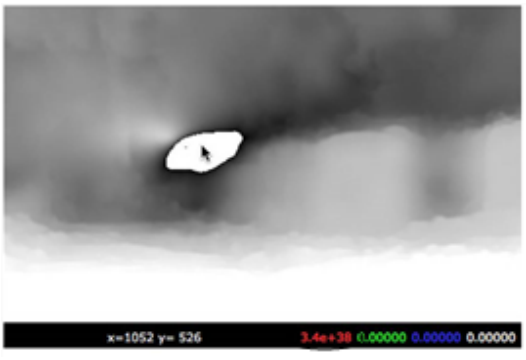
Sharpness = 0.05



Sharpness = 0.95

6. If necessary, use **Near Clip Plane** and **Far Clip Plane** to set the minimum and maximum values allowed in the depth map. All depth values that are outside this range are clipped to these values.
7. To see if there are any regions where the depth calculation is ambiguous, check **Mark Bad Regions**.

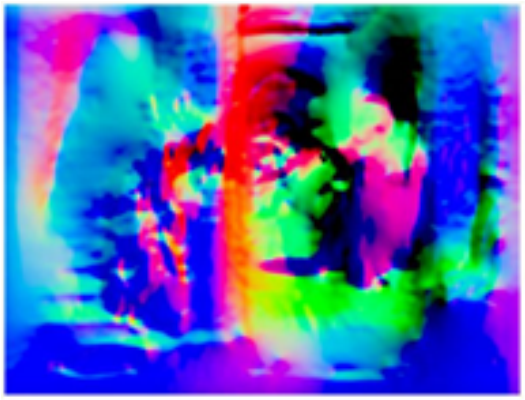
   Any ambiguous regions are marked as very large values in the depth map. Such regions might occur if CameraTracker hasn't been able to calculate the depth of all pixels using the camera data, due to certain camera movements for instance.



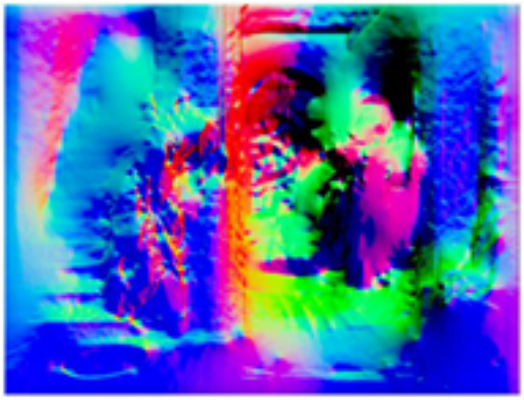
Mark Bad Regions enabled.

8. Once you've set the parameters so that you're happy with the depth map, view the normals pass (if you decided to generate one earlier). Normals are calculated from depth, and the normal direction is defined by the change in depth between pixels. The normals pass can be noisy, as small changes in depth lead to large changes in the normals. To smooth out the normal map by calculating normals using a lower resolution depth map, reduce **Normal Detail**. This integrates the change in depth across a wider range of pixels, reducing noise in the normals pass.

   The default value of 0.25 causes the normals pass to be calculated at a quarter resolution.



Normal Detail = 0.2



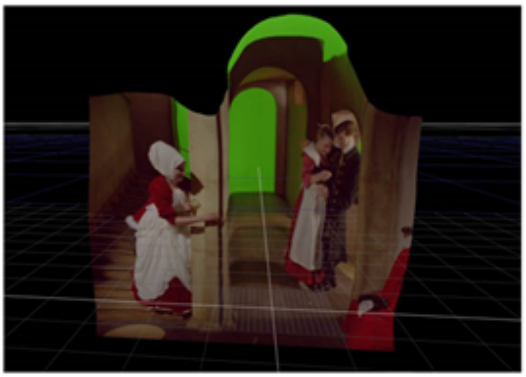
Normal Detail = 0.8

9. Proceed to Using the Results.

Using the Results

Once you are happy with the results from DepthGenerator, there are several ways you can use them in Nuke and NukeX:

- If you set **Depth Output** to **Depth 1/Z**, you can click **Create Card** in the DepthGenerator controls to create a Group node with a Card positioned in 3D space and displaced according to the depth channel. This allows you to visualize depth as a surface in 3D space, making it easier to place elements (such as Cards) at the correct depth in a live-action shot.



A Card node displaced according to the depth channel.

- If you set DepthGenerator to create a position pass, you can supply the position pass to the PositionToPoints node (**3D** > **Geometry** > **PositionToPoints**) to create a 3D point cloud. This is another good way to visualize depth in 3D space.

  For more information, see Creating a Dense Point Cloud Using the PositionToPoints Node.
- If you used DepthGenerator to create both a position pass and a normals pass, you can supply those to the Relight node (**3D** > **Lights** > **Relight**). This node applies a shader to its input image based on the position and normals passes, creating realistic lighting.

  For more information, see Relighting a 2D Image Using 3D Lights.
- You may also be able to use the depth map if you want to introduce fog and depth-of-field effects into a shot. In Nuke, the ZDefocus node requires a depth map in its input.

  For more information, see Simulating Depth-of-Field Blurring.

Creating Dense Point Clouds

Dense point clouds are a useful starting point for 3D modeling and can be helpful in positioning 3D objects into a scene. Using the PointCloudGenerator node, you can create a dense point cloud based on the information generated by CameraTracker and use the points to create a 3D mesh of your 2D footage.

Quick Start

Here's a quick overview of the workflow:

1. Make sure your PointCloudGenerator node is connected to the appropriate footage and Camera node. For more information, see Connecting the PointCloudGenerator Node.
2. Set keyframes in the footage to determine which frames are tracked to create the point cloud. For more information, see Setting Keyframes in a Sequence.
3. Track the footage to create a dense point cloud. For more information, see Tracking a Dense Point Cloud.



A 2D source image.



The resulting 3D point cloud.

4. Filter points from the result to adjust it and eliminate bad points. For more information, see Filtering Your Point Cloud.
5. Create groups within the cloud to aid visualization or bake out grouped sections. For more information, see Grouping, Labeling, and Baking Points.
6. Then if you need to, you can move on to creating meshes. For more information, see Creating a Mesh from a Point Cloud.
7. Alternatively, use the PoissonMesh node to create a mesh. For more information, see Using the PoissonMesh Node.

Connecting the PointCloudGenerator Node

To create a dense point cloud, PointCloudGenerator needs a Camera node with a solved camera path, either from CameraTracker or from a third party 3D application, and the tracked 2D footage.

To connect the node, do the following:

1. Click **3D** > **Geometry** > **PointCloudGenerator**.
2. Connect the necessary nodes to the inputs:
   - **Source** - the 2D footage from which the camera track was solved.
   - **Camera** - the solved Camera node.
3. If you want to mask part of the scene to avoid tracking moving objects or reflections, connect a matte to the **Mask** input. For more information about masking, see Masking Out Regions of the Image.
4. Click **Image** > **Viewer** to insert a Viewer node and connect it to the PointCloudGenerator node.
5. Proceed to Setting Keyframes in a Sequence.

Masking Out Regions of the Image

If you don't want to track all regions of your image, for example, if there are moving objects or reflections in the image, you can attach a matte to the **Mask** input to define image regions that should not be tracked. You can also use the source input's alpha channel as a matte.

1. If you want to use a separate matte for masking, connect it to the **Mask** input in the Node Graph.
2. In the properties panel, set **Ignore Mask** to the component you want to use as a mask:
   - **None** - Track features in the whole sequence.
   - **Source Alpha** - Use the alpha channel of the source clip to define which areas to ignore.
   - **Source Inverted Alpha** - Use the inverted alpha channel of the source clip to define which areas to ignore.
   - **Mask Luminance** - Use the luminance of the mask input to define which areas to ignore.
   - **Mask Inverted Luminance** - Use the inverted luminance of the mask input to define which areas to ignore.
   - **Mask Alpha** - Use the mask input alpha channel to define which areas to ignore.

- **Mask Inverted Alpha** - Use the inverted mask input alpha channel to define which areas to ignore.



Note: PointCloudGenerator can track multiple ranges, adding the new points to the existing cloud, so feel free to mask and re-track as much as necessary.

Setting Keyframes in a Sequence

Keyframes represent the best frames in a sequence to perform a particular function, in this case tracking. PointCloudGenerator can analyze the sequence for you, automatically picking frames with good parallax, or you can set keyframes manually by marking the frames yourself.

Setting Keyframes Automatically

1. Click **Analyze Sequence**.

   PointCloudGenerator walks the sequence, adding keyframes in suitable areas.
2. Once the process is complete, scrub the playhead and use the **Calculated Accuracy** field to evaluate the keyframes - values closer to 1 indicate a high accuracy.



Note: You can also add keyframes to the sequence manually. See below for more information.

3. Proceed to Tracking a Dense Point Cloud.

Setting Keyframes Manually

1. Scrub the playhead to the frame you want to mark as a keyframe and click **Add**,

   OR

   Set the **Frame Spacing** field to the number of frames between keyframes throughout the sequence and click **Add All**.

   For example, if you set this value to 2, keyframes are set at every other frame throughout your sequence.
2. If necessary, click **Delete** to remove the keyframe at the current playhead position or click **Delete All** to remove all keyframes.



Note: The keyframes generated automatically using **Analyze Sequence** are not deleted.

3. Proceed to Tracking a Dense Point Cloud.

Tracking a Dense Point Cloud

The next step towards creating a dense point cloud is to track your footage for more 3D feature points using the information from keyframes in the sequence and the solved Camera.

1. Click **Track Points** to track the sequence and create a dense point cloud.



An example 3D point cloud.

The default settings work well on most sequences, but you can adjust the **Dense Tracking** controls to alter the appearance of the cloud:

2. Adjust the **Point Separation** control to set the separation value, in pixels, between points in the point cloud.

   As you can see, lower values produce denser clouds, but at the expense of processing time.



Point Separation 0.

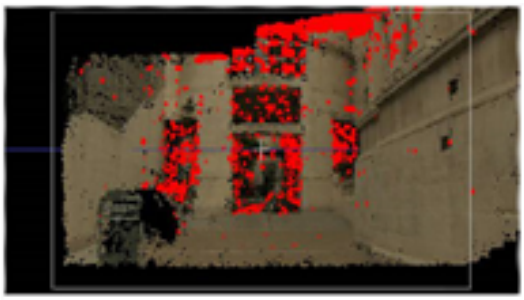


Point Separation 10.

3. Set the **Track Threshold** to compare the similarity of features over a number of frames and reject them if they exceed the threshold. You can adjust this value to test whether a track is reliable.



Low threshold.



High threshold.

Note: If the threshold is set too high, you may find that you don't get any reliable tracks at all. The red pixels in the above image are rejected tracks.

4. Proceed to Filtering Your Point Cloud.

Filtering Your Point Cloud

You can filter your point cloud by adjusting the number and quality of points it includes. NukeX filters the tracked point cloud directly, without recalculating, streamlining your workflow.

1. Ensure that **Display rejected points** is enabled and adjust the threshold controls to update the 3D Viewer dynamically. Rejected tracks are highlighted in red.
2. Adjust the **Angle Threshold** to set the minimum acceptable angle to triangulate 3D points (in degrees). Points with a large triangulation angle tend to be more accurate.

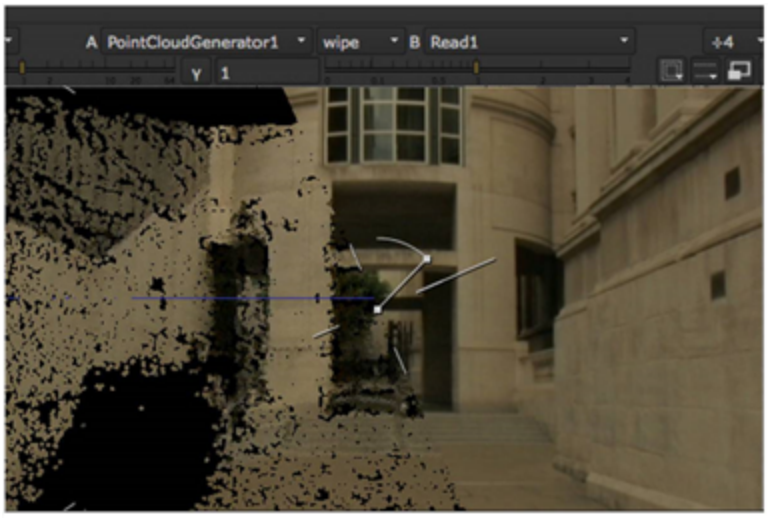
   Set a threshold of **0** to triangulate all points or increase the threshold to highlight the least accurate points.

Tip: As a rule of thumb, anything below 5 degrees is likely to be incorrect.

3. Adjust the **Density Threshold** to set the minimum acceptable point density. Isolated points tend to be less accurate.

   Set a threshold of **0** to output all points or increase the threshold to highlight the most isolated, less accurate points.
4. The Viewer **wipe** tool can help you locate isolated points by comparing the 2D source footage to the point cloud.

   Set Input A to the PointCloudGenerator node and Input B to the Read node for the source image.



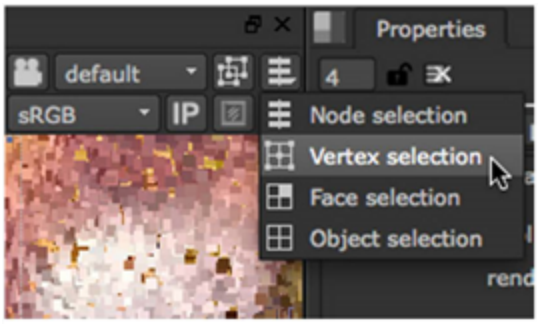
See Using the Viewer Controls for more information.

5. Proceed to Removing Rejected Points.

Removing Rejected Points

Once you've identified and highlighted the less accurate points:

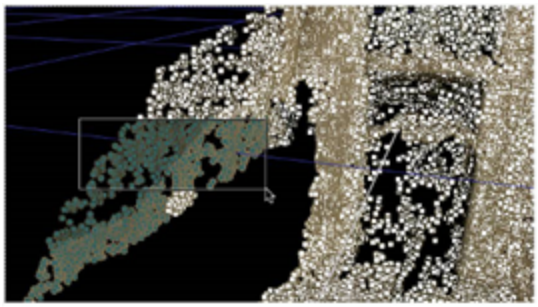
1. Click **Delete Rejected Points** to automatically remove all highlighted points.
2. You can then be a little more selective by switching the Viewer to **Vertex selection** mode.



3. Use the **Output** controls to assist you when visualizing your point cloud:
   - **Point Size** - set the pixel size of points in your cloud.
   - **Output points per frame** - by default, all tracked points in your sequence are displayed in the Viewer. Enabling this control allows you to view only the points generated at the current frame.

Note: If you're having trouble viewing points individually, try reducing the **3D handle size** in the **Preferences** > **Panels** > **Viewer Handles** tab.

4. Manually select points in the cloud using the 3D Viewer by:
   - Dragging a marquee over the required points in the Viewer, or
   - Holding **Shift**, and dragging a marquee over several selection areas in the Viewer to select all points at once.



Note: You can remove points from a selection by holding **Shift**+**Alt** and re-selecting highlighted points.

5. Press **Delete**/**Backspace** or right-click in the Viewer and select **delete selected** to remove the points.
6. Proceed to Grouping, Labeling, and Baking Points.

Grouping, Labeling, and Baking Points

Point clouds can be awkward to interpret, especially if the **Point Separation** control is set to a relatively high value. Grouping points in the cloud can help to visualize the scene, particularly when sensibly labeled and colored in the Viewer.

Groups can also be baked out as separate point clouds or converted into meshes. See Creating a Mesh from a Point Cloud for more information.

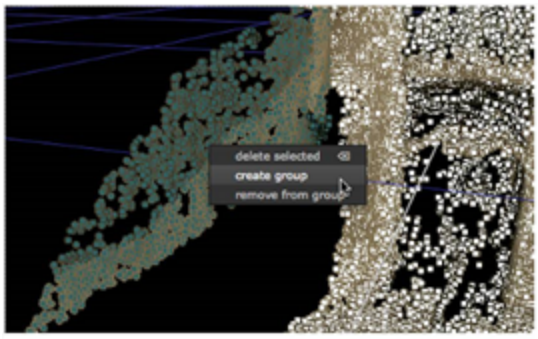
To create a group from a point cloud:

1. Manually select group points in the 3D Viewer by:

- Dragging a marquee over the required points in the Viewer, or
- Holding **Shift**, and dragging a marquee over several selection areas in the Viewer to select all points at once.

Note: You can remove points from a selection by holding **Alt** + **Shift** and re-selecting highlighted points.

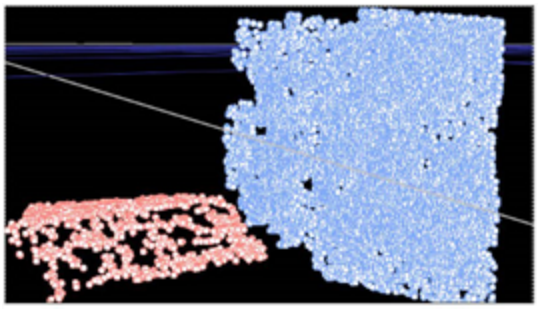
2. Right-click on a highlighted point and select **create group**.



The new group is added to the **Groups** tab.

Note: If groups already exist, the option to add the selected points to a group is enabled.

3. Add more groups as required to build up the visualization of the scene.
4. In the properties panel, select the **Groups** tab to display the list of existing groups. Use the **Groups** controls to determine the appearance of your groups:
   - **Display groups in overlay** - when enabled, groups are highlighted in the Viewer using their associated RGB color.
   - **Create Group** - click to add the selected points in the Viewer to a new group.
   - **Delete Selected Groups** - click to delete all the selected groups in the list.
5. Double-click the table columns to edit group names or colors, and toggle visibility.
6. Enable **Output visible groups only** to display only selected groups in the Viewer.



7. After creating the required groups, you can split them off into individual point clouds, using **Bake Selected Groups**, or create group meshes.
8. Proceed to Creating a Mesh from a Point Cloud

Creating a Mesh from a Point Cloud

PointCloudGenerator can create meshes from grouped points in the point cloud that you can use as stand-alone 3D objects, for example, in 3D modeling. You can also use these meshes to quickly project the 2D sequence onto the mesh using the Project3D node.

PointCloudGenerator meshes are based on the Poisson Surface Reconstruction calculation method. The original source code and paper were created by Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe (see http://www.cs.jhu.edu/~misha/Code/PoissonRecon/ for more information).

To create a mesh from a group:

1. On the **Groups** tab, select the required group(s) from the **Groups** list.

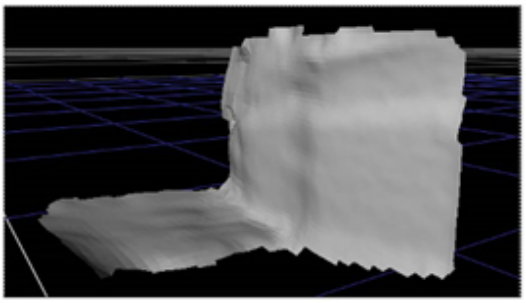
   You can mesh a single group or all the groups available in the list.
2. Click **Bake Selected Groups to Mesh**.

   A mesh is created automatically using the information from the PointCloudGenerator node.

Note: You can use the resulting group mesh node in Nuke as well, though you can't edit the geometry.

3. Proceed with Adding Texture to a Mesh.

Many factors influence the quality of any mesh, chief among which are the contents of the scene and the camera track obtained from the sequence. If you consider a simple scene, such as that shown in the example images, the resulting mesh can be quite accurate.



A mesh from a simple sequence.

Adding Texture to a Mesh

You can quickly add texture to your point cloud mesh using the Project3D node. Do the following:

1. Click **3D** > **Shader** > **Project3D**.
2. Connect the solved Camera node to the Project3D **cam** input.
3. Connect the other input of the Project3D node into the sequence.
4. Connect the Group1_Mesh node **img** input to the Project 3D node.



The mesh is now textured with your original 2D sequence.

Tip: You can export your camera, point cloud, and meshes as Alembic (**.abc**) objects to other applications. For more information about exporting **.abc** files, see Exporting Geometry, Cameras, Lights, Axes, or Point Clouds .

Using the PoissonMesh Node

The PoissonMesh node uses information from a dense point cloud to generate a mesh that you can further use as a 3D object, in 3D modeling for instance. The PoissonMesh node is based on the Poisson Surface Reconstruction calculation method. The original source code and paper were created by Michael Kazhdan, Matthew Bolitho and Hugues Hoppe (see http://www.cs.jhu.edu/~misha/Code/PoissonRecon/ for more information).

Before using the PoissonMesh node, you need a dense point cloud with normals such as the one created by the PointCloudGenerator node. For more information on creating a dense point cloud, see Tracking a Dense Point Cloud. When you've got one, you're ready to create a mesh:

1. Connect the PoissonMesh node to a PointCloudGenerator node.
2. A mesh is created automatically using the information from the PointCloudGenerator node and you can view it in the 3D view.
3. If you want to, you can tell PoissonMesh to use the selected points only in the point cloud to create a mesh. To do this, check the **Use Selection** box in the properties panel. You can also use the **Filtering** box to use the PointCloudGenerator filtering to refine your mesh.
4. You can also adjust other properties for your mesh on the **PoissonMesh** tab:
   - **Depth** - set the maximum depth (in integers) for the calculation tree used to create your mesh. You can enter values up to 10, but increasing this value increases the solver's memory usage, so it's a good idea to keep this as low as possible.
   - **Scale** - set the ratio between the 3D cube used to create the mesh and the bounding box of the footage it's created from.
   - **Solver Divide** - set the depth at which the solver equation is used in the creation of your mesh. This can help you reduce your memory usage when generating complex meshes.
   - **Iso Divide** - set the depth at which the iso-surface extractor is used in extraction. This can help you reduce your memory usage when generating complex meshes.
   - **Samples per Node** - set the minimum number of sample points used to create your mesh. Use larger values if your original footage is very noisy.
   - **Confidence** - check to use the size of the normals as confidence information when generating the mesh. This may take longer, but gives you better results since your point cloud point creation gets double-checked.



Note: You can only use the PoissonMesh node in NukeX as the information used to create the mesh is not stored in the node and therefore cannot be recreated for viewing in Nuke.

Adding Texture to the PoissonMesh

You can add texture to your point cloud mesh using the Project3D and ApplyMaterial nodes. Do the following:

1. Connect your texture node to the Project3D node.
2. Then connect the Project3D node to the **mat** input of the ApplyMaterial.
3. Connect the other input of the ApplyMaterial node into the PoissonMesh node. The mesh is now textured with your texture.

Tip: You can export both your point cloud and the mesh as FBX objects to other applications, but don't forget to export the Camera node too. For more information about exporting FBX files, see Exporting Geometry, Cameras, Lights, Axes, or Point Clouds .

Using ModelBuilder

The ModelBuilder node provides an easy way to create 3D models for 2D shots. You can build a model by creating shapes and then editing them, and align models over your 2D footage by dragging vertices to their corresponding 2D location.

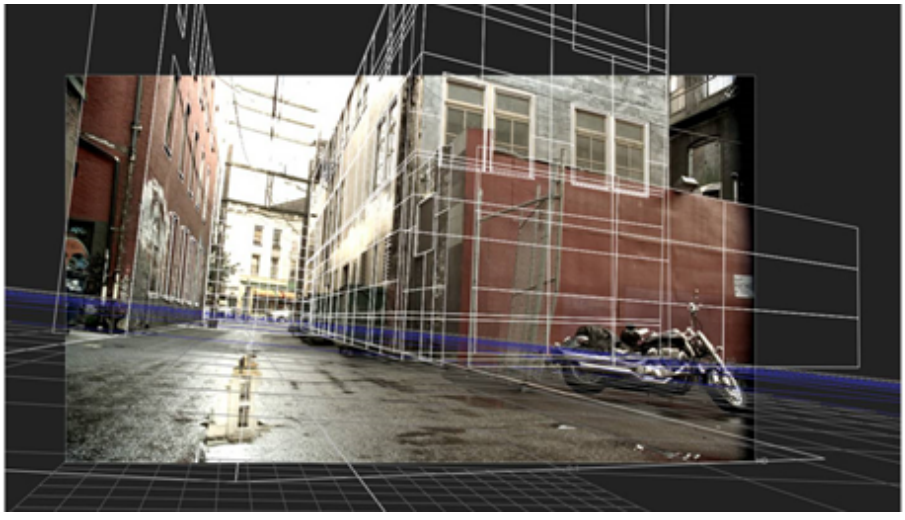
Prerequisites

To be able to align models, ModelBuilder needs a tracked camera and an input image for visual reference. You can also use other 3D geometry and point clouds as a reference if you already have these for your scene.



Note: You can use ModelBuilder without a camera, image sequence, or reference geometry. In this case, all the Edit mode features work just fine; it just means you aren't able to do anything in Align mode.

Creating and editing models using ModelBuilder requires a NukeX license, but the resulting geometry can also be used in Nuke.



A 2D source image with shapes created using ModelBuilder.

Special thanks to belowtheradar TV for use of the above footage, used throughout this chapter.

Quick Start

Here's a quick overview of the workflow:

1. Connect the ModelBuilder node to your footage and a changing Camera. For more information, see Connecting the ModelBuilder Node.
2. As a starting point for your model, create one or more of the built-in 3D shapes using the toolbar on the left hand side of the Viewer.

   See Creating Shapes.
3. If necessary, adjust the display characteristics of your shapes so you can better see them in the Viewer. You can do this at any point during the process.

   See Editing Shapes' Display Characteristics.
4. Position the shape using your 2D footage or existing 3D geometry as a reference.

   See Positioning Shapes.
5. Edit the shape until you're happy with your model.

   See Editing Shapes.
6. Once you're happy with your model, you're ready to texture it.

   See Applying Textures.
7. If necessary, you can also create a separate geometry node for the selected shapes and use that as you would use any other geometry node in Nuke.

   See Exporting Shapes to Separate Geometry Nodes.

Connecting the ModelBuilder Node

To connect the ModelBuilder node:

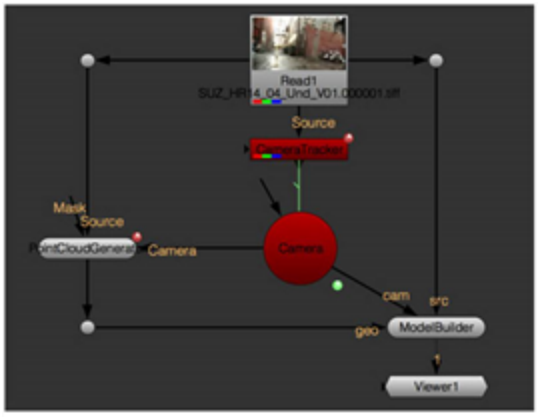
1. If you don't already have a tracked camera that matches your footage, create one using the CameraTracker node (see Camera Tracking).

   ModelBuilder needs a changing Camera (with a change of more than 5 degrees) in order to calculate 3D points from your 2D footage.
2. Select **3D** > **Geometry** > **ModelBuilder** to create a ModelBuilder node.
3. Connect your Camera node to ModelBuilder's **cam** input, and your 2D footage to its **src** input.

   Note that the objects in your **src** footage that you want to model should not be moving, as ModelBuilder works best with static objects.

4. If you have 3D geometry (such as a shape or a point cloud) that you want to use as a reference when positioning new shapes, connect it to ModelBuilder's **geo** input.

   For example, you can use the PointCloudGenerator node (see Creating Dense Point Clouds) to create a 3D point cloud for your shot, and connect that to the **geo** input. When creating new shapes, you can then select vertices on the point cloud to automatically align the shapes with the point cloud. This gives you an approximate initial position for your shapes.
5. If you have a 2D texture image that you want to display as a background in the UV preview window and on the model in the 3D Viewer, connect it to ModelBuilder's **tex** input.
6. Attach a Viewer to the output of the ModelBuilder node.



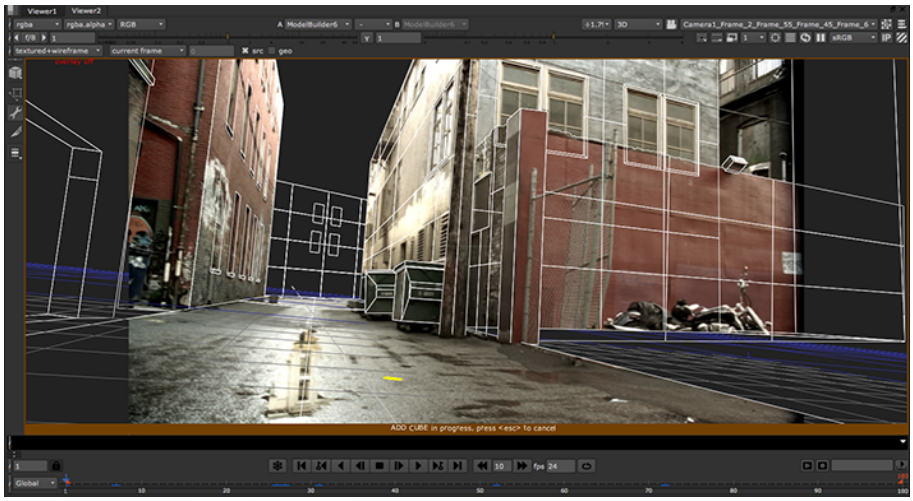
7. Proceed to Creating Shapes below.

Creating Shapes

To create shapes:

1. Double-click on the ModelBuilder node to open its properties.
2. Use the shape creation menu  in the ModelBuilder toolbar on the left hand side of the Viewer to select the basic 3D shape that best matches the object you are trying to model. You can select between **Point**, **Card**, **Cube**, **Sphere**, **Cone**, **Cylinder**, and **Polygon** (when creating a polygon, click on the Viewer to create vertices where you need them, and when you're done, press **Return**).

   A colored border appears around the Viewer window to indicate that the action of creating a shape is in progress.



Tip: If you want to model a simple wall, a Card might be a good starting point. For an entire building, you may want to use a Cube instead.

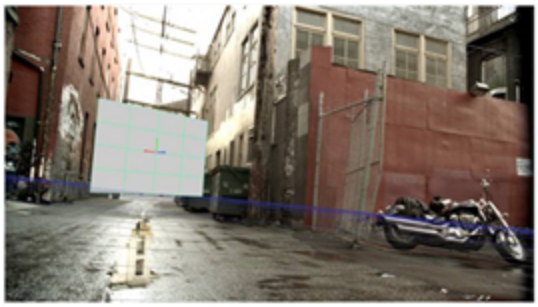


You can adjust the shapes (for example, the number of rows or columns in them) on the **Shape Defaults** tab. These controls only apply when creating a shape - you cannot adjust them after the fact.

3. Hover over the Viewer. You should see a yellow circle appear under the cursor. You can now:
   - click in the Viewer to set where you want the new shape created. If you click over empty space, the shape is created at a distance specified by the **Distance** control. There are no specific units for this value; low values position the shape close to the camera, and higher values mean further away.
   - hold down **Shift** and click to snap to the surface of any geometry under the cursor.
   - drag the mouse left or right to scale the shape.

   The shape you selected appears in the Viewer and in the Scene list in the ModelBuilder properties.

4. If the shape was created too close to the camera or too far away, Activate **Edit Mode** in the ModelBuilder toolbar and make sure the selection mode menu is set to **Select objects**. Then, select the shape in the Viewer, and drag the transform handles to reposition it.

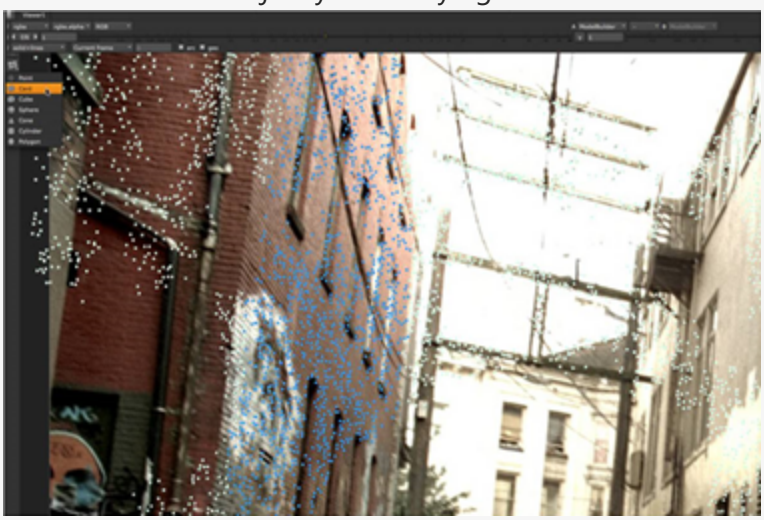


5. If you have more than one object that you want to model in your **src** footage, you can repeat the above steps to create several shapes using a single ModelBuilder node.
6. Proceed to Editing Shapes' Display Characteristics below.

Tip: You can also use the **Create** buttons on the **Shape Defaults** tab to create shapes. This is exactly the same as using the shape creation menu in the Viewer.

Tip: If you want to use 3D geometry as a visual reference and connected it to the **geo** input, before creating a shape, you need to:

1. Activate **Edit Mode** in the ModelBuilder toolbar on the left hand side of the Viewer and set the selection mode menu to **Select vertices**.
2. Drag in the Viewer to select vertices on the object that you want to model, bearing in mind that when you create a new shape, it's automatically aligned with the selected vertices.
If you cannot see the object in the geo input, make sure **Pass Through Geo** is enabled in the ModelBuilder properties.
3. Use the shape creation menu in the ModelBuilder toolbar to select the basic 3D shape that best matches the object you are trying to model.



ModelBuilder creates the shape to align with the selected vertices.

Editing Shapes' Display Characteristics

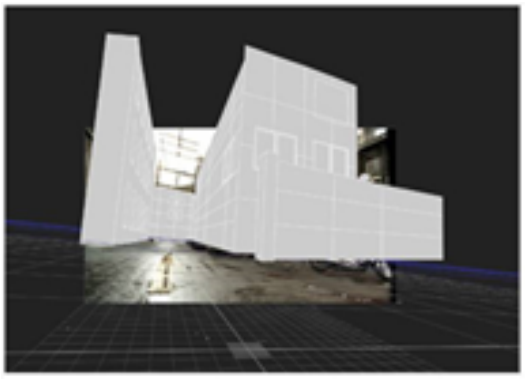
To edit shapes' display characteristics:

1. If you created a lot of shapes, you may want to organize them into groups. Click the **+** button in the ModelBuilder properties to create a Group item and drag shapes inside it to move them into the group.
2. To rename shapes or groups, click on them in the Scene list and enter a new name.

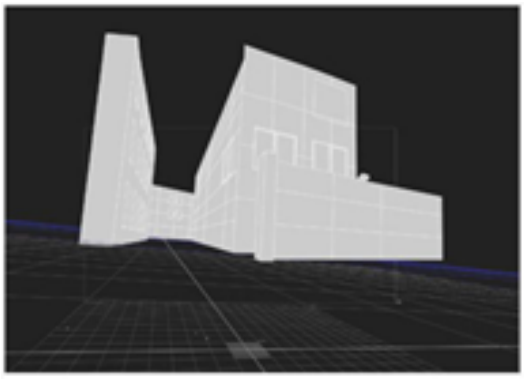
3. If the Viewer seems too cluttered, you can toggle the visibility of both shapes or groups by clicking  in the Scene list.

   Note that when an item is hidden, it doesn't appear in the Viewer or renders.
4. If you have connected geometry to the **geo** input but don't currently need it as a reference, you can also uncheck **Pass Through Geo**. This tells ModelBuilder to not output that geometry.

   Note that when **Pass Through Geo** is enabled, the geometry appears both in the Viewer and renders.
5. By default, the **src** image is displayed in the 3D Viewer whenever the Viewer is locked to the input camera. If you don't want this, uncheck **Show Source Image**.



Show Source Image enabled.



Show Source Image disabled.

6. Use the **display** and **render** controls to select how you want your shapes to appear in the Viewer and renders. These controls affect all the existing shapes.

   For example, you may want to set **display** to **wireframe** initially in order to better see what you are doing when positioning your shapes, but change it to **textured** to review the final result.

   For more information on the available options, see Object Display Properties.
7. If you set **display** to **textured** or **textured+wireframe**, you can also use the **Texture** control to select the frame used to texture the shapes:
   - **Current frame** - Project the current frame onto the shape.
   - **Locked frame** - Project the frame specified in the field on the right onto the shape. This can help you line up the shape against your source footage.
   - **Frame difference** - Subtract the frame specified in the field on the right from the current frame and project the resulting frame onto the shape. This can help you line up the shape against your source footage.
8. If you want to disable shape selection in the Viewer, uncheck **selectable**. This only has an effect when the selection mode menu is set to **Select nodes**  in the ModelBuilder toolbar.
9. If you don't want the shapes to cast or receive shadows, uncheck **cast shadow** or **receive shadow**.

   For more information on shadows, see Casting Shadows.

10. Proceed to Positioning Shapes below.

Positioning Shapes

To position shapes:

1. If the **Align Mode** isn't already active, click  in the ModelBuilder toolbar to activate it.
2. The Viewer should also automatically be locked to your input Camera node. If this isn't the case, select the Camera from the dropdown menu in the top right corner of the Viewer and click .

   The positioning handles only appear when the Viewer is locked to the input camera.
3. Use the **Align** dropdown menu on the top of the Viewer to choose whether you want to position the shape by transforming the entire shape or individual vertices:
   - **Object transform** - ModelBuilder tries to transform the whole object. This guarantees that flat polygons stay flat and the angle between polygons doesn't change, but it's less flexible because you're always changing the whole object at once.



The original card.



The card after transforming the entire object.

   - **Vertex position** - ModelBuilder transforms each vertex separately. While this can be a very convenient way to make the geometry match up exactly, it can also lead to non-flat polygons and other problems.

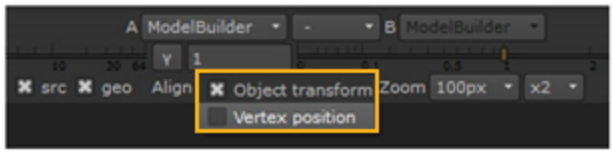


The original card.



The card after transforming

a vertex position.



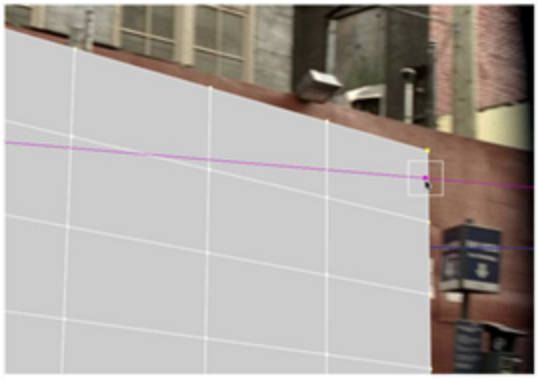
4. Move a few vertices so that they match the shape of the object you want to model. For example, if you are modeling a building using a cube, drag the corners of the cube so that they line up with the corners of the building.

   When you click on a vertex, you'll notice a zoom window in the top left of the Viewer. This allows you to accurately position vertices without zooming the entire Viewer. To adjust the size and magnification factor for the zoom window, use the **Zoom** dropdown menus on top of the Viewer.



   As you do this, ModelBuilder transforms the shape to fit your adjusted vertex positions. It also sets keys on the 2D positions of the vertices. Any vertices that have keys set on them turn purple in the Viewer.

5. Go to a different frame and move one of the blue vertices to match its new location in the **src** footage. When you do so, a purple line appears. The correct location for the vertex on the current frame should lie somewhere on this line; however, you can drag the vertex away from the line if necessary.



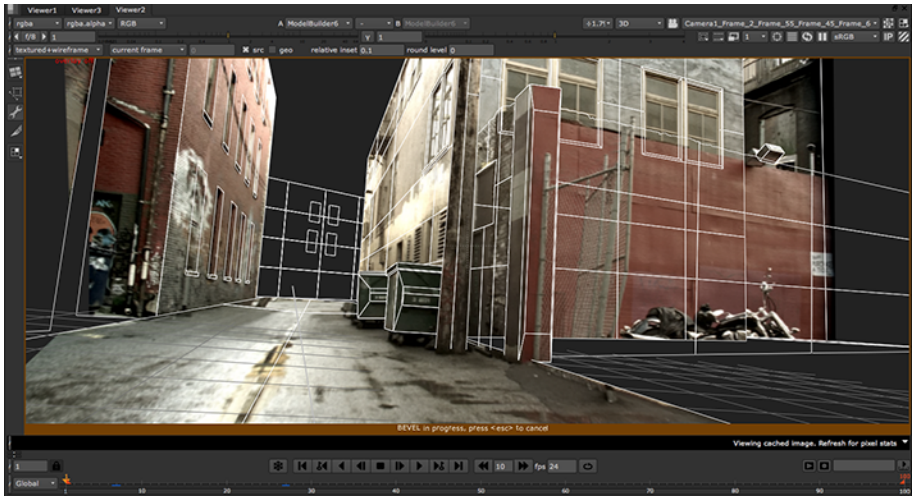
   Positioning just one vertex is usually enough to get the object position right, but not the rotation or scaling.

6. In order to lock off the rotation and scaling of the object, move a few more vertices to their correct location.
7. Play back through the sequence. The object should stay roughly in the right place relative to the **src** footage. If you see any vertices starting to drift, move them to their correct location.
8. Proceed to Editing Shapes.

Editing Shapes

When editing ModelBuilder shapes, there are two types of editing actions:

- **Single-step actions** that happen instantly, such as extruding and merging faces.
- **Multi-step actions** (such as beveling and subdiving faces) that have extra parameters you can set. All multi-step actions behave in the same way:
  - When you start the action, you can see a preview in the Viewer even though the action has not been applied yet. A colored border appears around the Viewer window to indicate that the action is in progress.



Tip: You can change the color of the border in the Preferences. Press **Shift**+**S** on the Node Graph to launch the **Preferences** dialog, go to the **Appearance** tab, and use the **Highlight** button to select a new color. Note that he brightness of the color is adjusted to prevent it from clashing with the Viewer background.

  - The parameters for the action appear at the top of the Viewer. As you adjust them, the preview in the Viewer is updated.
  - To cancel the action, you can press **Esc** on the Viewer.

- To accept the current parameter values and complete the action, you can press **Return** (or change the selection mode, start a new action, press the **+**, **-**, or **duplicate** buttons, add a new shape, change into Align mode, or close the ModelBuilder properties).

To edit your shapes:

1. You can edit vertices, edges, faces, and entire objects. For more information, see:
   - Editing Vertices
   - Editing Edges and Edge Loops
   - Editing Faces, and
   - Editing Objects.



Tip: See Viewer Selection Modes for more information on making selections.

2. When you're happy with your edits, proceed to Exporting Shapes to Separate Geometry Nodes.

Editing Vertices

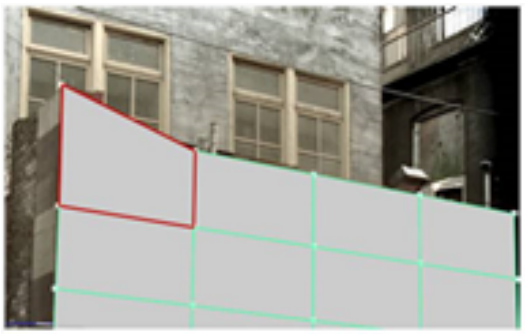
To edit vertices:

1. Activate **Edit Mode** by selecting  from the ModelBuilder toolbar.
2. Set the selection mode menu to **Select vertices**  in the ModelBuilder toolbar and select one or more vertices on the object.
3. Edit your selection as necessary:
   - To translate, rotate, or scale the selected vertices, drag the transform handles that appear on them. To move the pivot point for the transform handles, press **Ctrl**/**Cmd**+**Alt** while dragging on the center of the transformation overlay.

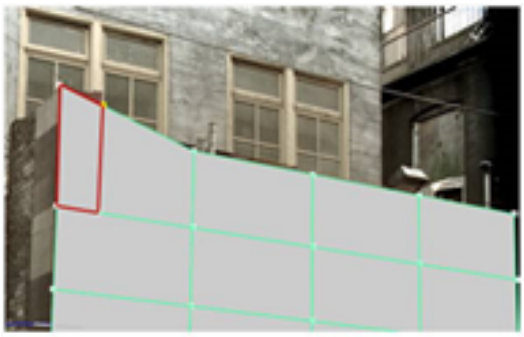
Tip: If necessary, you can also use the controls at the top of the Viewer to set the initial position and alignment of the transform handles (that is, the position used whenever you change the selection). For more information, see Setting the Initial Transform Action Center.

   - To carve the selected vertices, right-click on them and select **carve**. A colored border appears around the Viewer window to indicate that the action is in progress. Click on an edge or vertex to begin a carve. All of the surrounding faces are highlighted in red. Click anywhere inside a highlighted face, or

on an edge or vertex of a highlighted face, to carve an edge between the previous vertex and the place you just clicked.



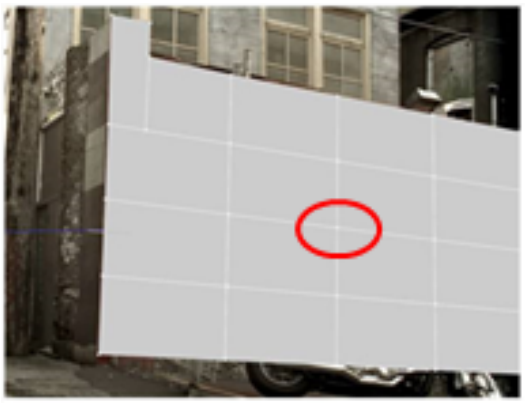
A highlighted face.



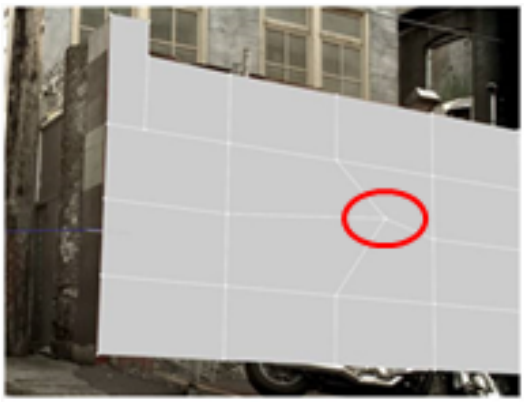
A new edge carved inside the highlighted face.

To carve out a freehand edge, hold down **Ctrl/Cmd**+**Shift** and drag.

- To extrude the selected vertices, select **extrude** from the right-click menu and drag. This stretches the selected vertices into three-dimensional polygons. ModelBuilder creates faces to complete the object, for example a rectangular pyramid from a vertex.



A selected vertex.



Extruding the selected vertex.

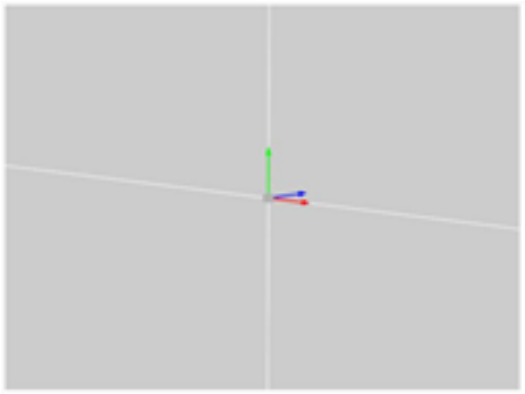
- To bevel the selected vertices, select **bevel** from the right-click menu.

  A colored border appears around the Viewer window to indicate that the action is in progress.

  At the top of the Viewer, use **relative inset** to define how far back along the surrounding edges to start the bevel from: **0.0** means no distance and **1.0** means at the opposite end of the edge. The default is **0.1**, meaning the beveling starts 1/10th of the way back along each edge.

  Set **round level** to the number of times to repeat the initial bevel, effectively rounding the edges. A value of **0** just does the initial bevel, a value of **1** bevels the output of the initial bevel; a value of **2** bevels the bevel of the initial bevel, and so on.

  When you're happy with the values you've entered, press **Return** to apply the bevel.



A selected vertex.



Beveling the selected vertex.

Beveling is like extruding the selected vertices, except that the resulting polygons have smooth edges and corners. You may want to use beveling to add realism to your model, as real-world objects rarely have perfectly sharp corners.

- To delete the selected vertices, select **delete vertices** from the right-click menu (or press **Delete**).

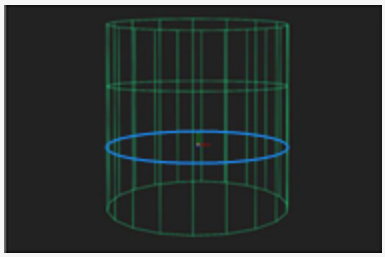
Note: You can only delete simple vertices that have no more than two edges connected to them.

Editing Edges and Edge Loops

To edit edges and edge loops:

1. Activate **Edit Mode** by selecting from the ModelBuilder toolbar.
2. Set the selection mode menu to either **Select edges** or **Select edge loops** in the ModelBuilder toolbar and select one or more edges or edge loops on the object.

Tip: An edge loop is a string of connected edges across the model. Often, the last edge meets again with the first edge, forming a loop. In ModelBuilder, the loop follows the middle edge in every intersection that has an even number of edges. The loop ends when it encounters an intersection with an odd number of edges.



Edge loops can help you produce more natural deformations for organic models that are animated. They work best when you've got a mesh that's mostly made up of four-sided polygons, known as *quads*.

3. Edit your selection as necessary:
   - To translate, rotate, or scale the selected edges or edge loops, drag the transform handles that appear on them. To move the pivot point for the transform handles, press **Ctrl**/**Cmd**+**Alt** while dragging on the center of the transformation overlay.



Selected edges.

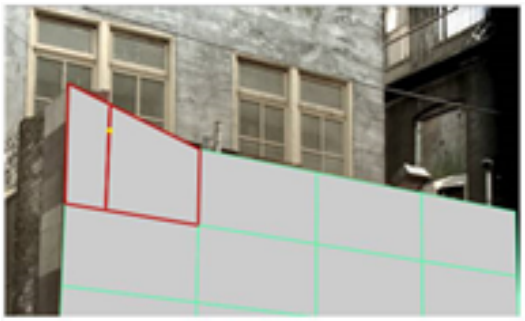


Translating the selected edges.

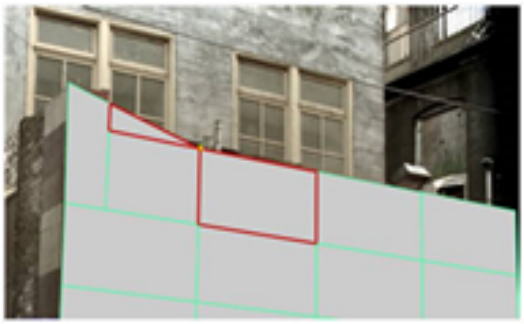
Tip: If necessary, you can also use the controls at the top of the Viewer to set the initial position and alignment of the transform handles (that is, the position used whenever you change the selection). For more information, see Setting the Initial Transform Action Center.

   - To carve the selected edges, right-click on them and select **carve**. A colored border appears around the Viewer window to indicate that the action is in progress. Click on an edge or vertex to begin a carve. All

of the surrounding faces are highlighted in red. Click anywhere inside a highlighted face, or on an edge or vertex of a highlighted face, to carve an edge between the previous vertex and the place you just clicked.



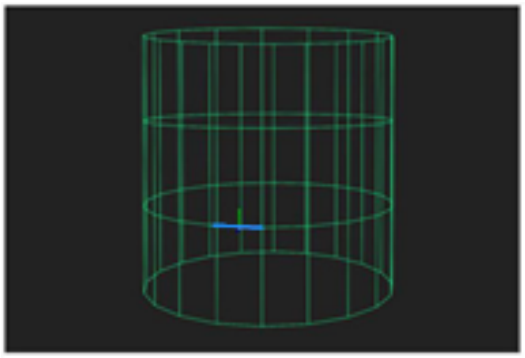
Highlighted faces.



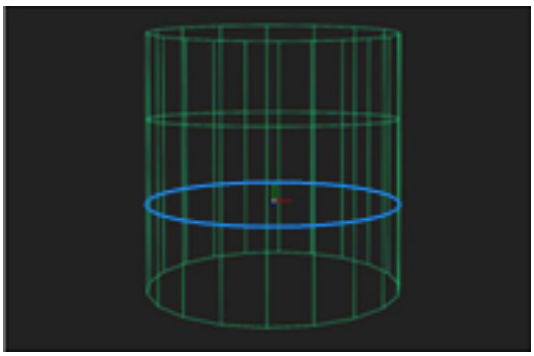
Carving a new edge inside the highlighted faces.

To carve out a freehand edge, hold down **Ctrl/Cmd**+**Shift** and drag.

- To extend your current edge selection along loops, select **select edge loop** from the right-click menu. This is the same as activating **Select edge loops** and clicking on an edge.

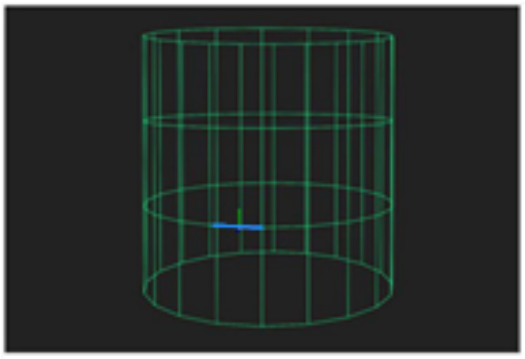


Original selection.

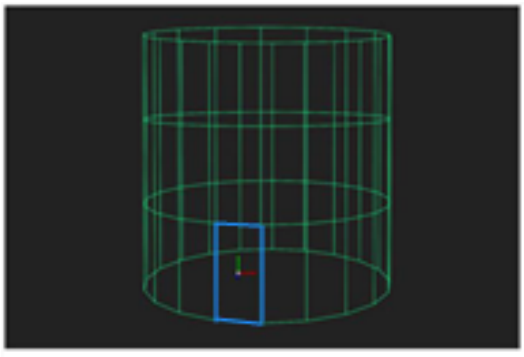


Select edge loop.

- To extend your current selection to form a ring around the face or the entire shape, choose select edge ring from the right-click menu.

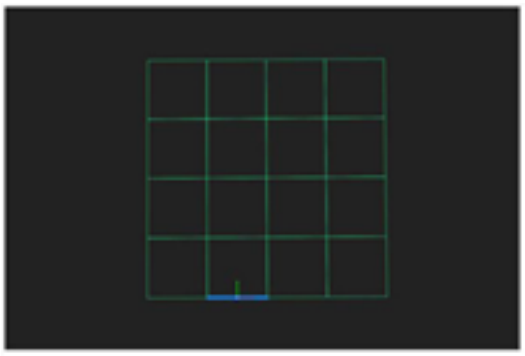


Original selection.

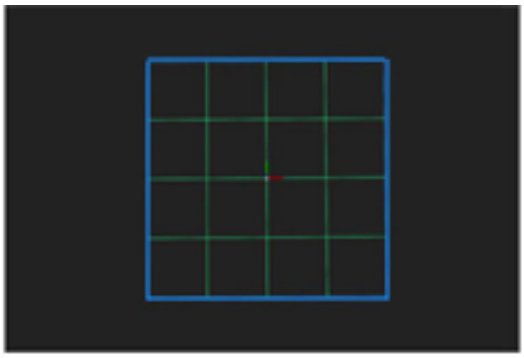


Select edge ring.

If the starting edge is part of the boundary for an entire shape, this selects all edges along that boundary.

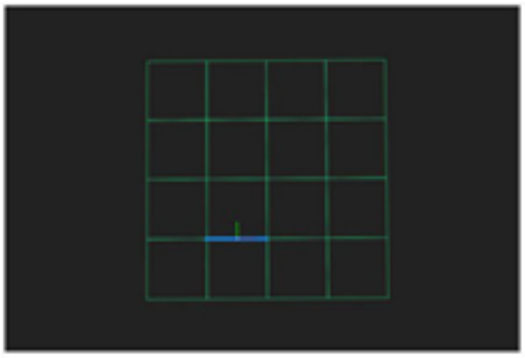


Original selection.

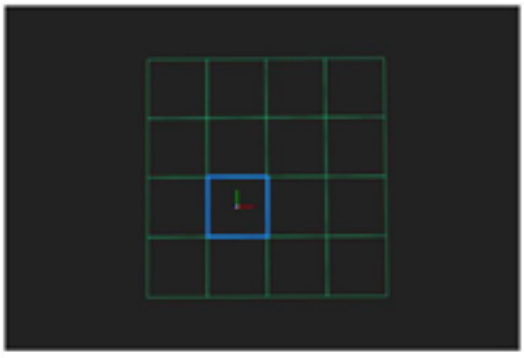


Select edge ring.

Otherwise, it selects all edges around the face that the starting edge belongs to.

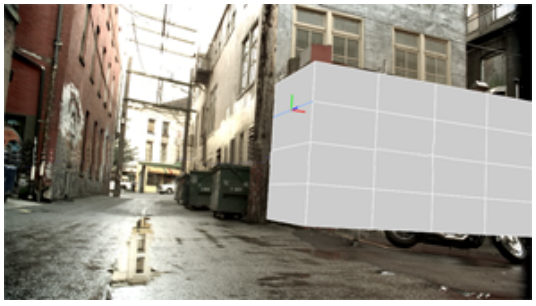


Original selection.



Select edge ring.

- To extrude the selected edges or edge loops, select **extrude** from the right-click menu and drag. This stretches the selected edges or edge loops into a three-dimensional polygon.



A selected edge.



Extruding the selected edge.

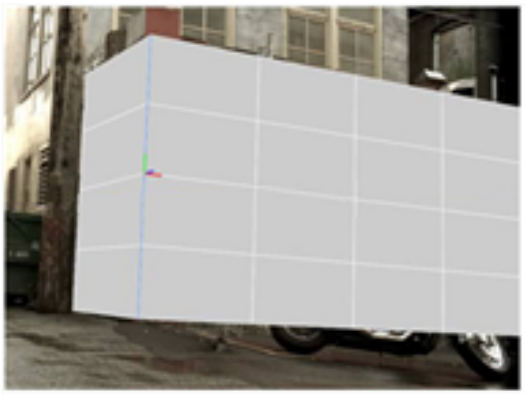
- To bevel the selected edges or edge loops, select **bevel** from the right-click menu.

  A colored border appears around the Viewer window to indicate that the action is in progress.

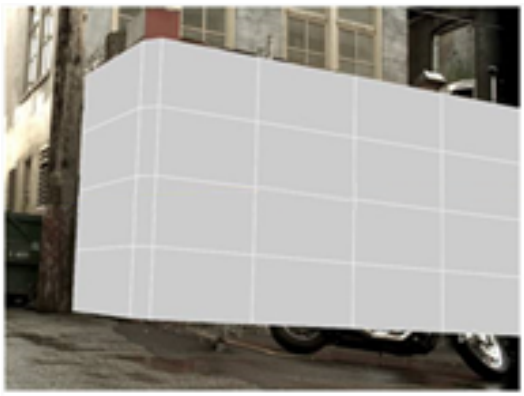
At the top of the Viewer, use **relative inset** to define how far back along the surrounding edges to start the bevel from: **0.0** means no distance and **1.0** means at the opposite end of the edge. The default is **0.1**, meaning the beveling starts 1/10th of the way back along each edge.

Set **round level** to the number of times to repeat the initial bevel, effectively rounding the edges. A value of **0** just does the initial bevel, a value of **1** bevels the output of the initial bevel; a value of **2** bevels the bevel of the initial bevel, and so on.

When you're happy with the values you've entered, press **Return** to apply the bevel.



Selected edges.



Beveling the selected edges.

Beveling is like extruding the selected edges, except that the resulting polygons have smooth edges and corners. You may want to use beveling to add realism to your model, as real-world objects rarely have perfectly sharp corners.

- To divide the selected edges or edge loops into segments of equal length, select **subdivide**. A colored border appears around the Viewer window to indicate that the action is in progress. Use the **segments** control on top of the Viewer to set the number of segments, and press **Return** to complete the action.
- To delete the selected edges or edge loops, select **delete edges** from the right-click menu (or press **Delete**).

Editing Faces

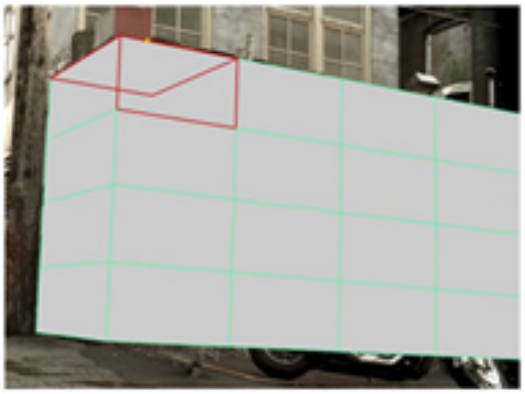
To edit faces:

1. Activate **Edit Mode** by selecting  from the ModelBuilder toolbar.
2. Set the selection mode menu to **Select faces**  in the ModelBuilder toolbar and select one or more faces on the object.
3. Edit your selection as necessary:

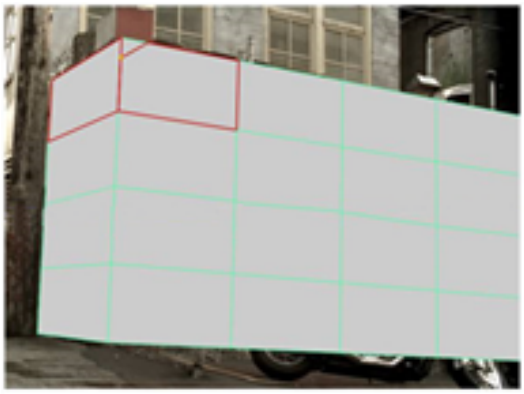
- To translate, rotate, or scale the selected faces, drag the transform handles that appear on them. To move the pivot point for the transform handles, press **Ctrl/Cmd**+**Alt** while dragging on the center of the transformation overlay.

Tip: If necessary, you can also use the controls at the top of the Viewer to set the initial position and alignment of the transform handles (that is, the position used whenever you change the selection). For more information, see Setting the Initial Transform Action Center.

- To carve the selected faces, right-click on them and select **carve**. A colored border appears around the Viewer window to indicate that the action is in progress. Click on an edge or vertex to begin a carve. All of the surrounding faces are highlighted in red. Click anywhere inside a highlighted face, or on an edge or vertex of a highlighted face, to carve an edge between the previous vertex and the place you just clicked.



Highlighted faces.

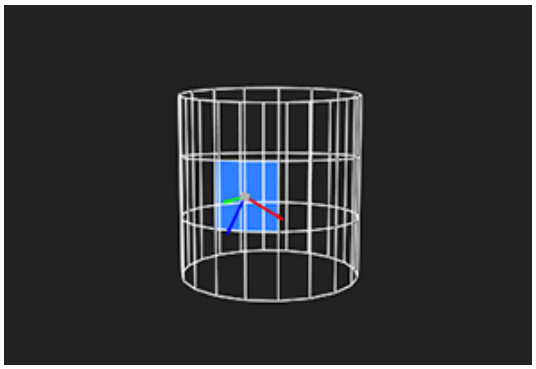


Carving a new edge inside the highlighted faces.

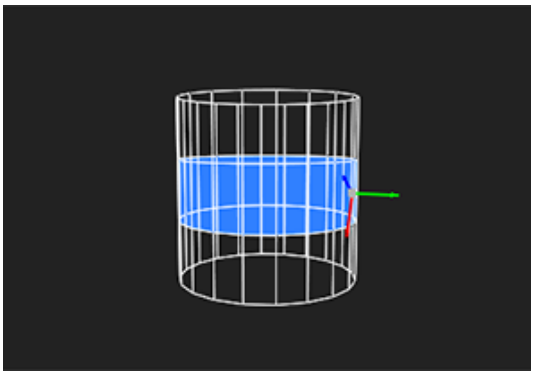
To carve out a freehand edge, hold down **Ctrl/Cmd**+**Shift** and drag.

- To extend your current selection along loops, choose **select face loop** from the right-click menu. A face loop is a string of connected faces across the model. Often, the last face meets again with the first face, forming a loop.

  To create a face loop, you need to select at least two neighboring faces, so that ModelBuilder knows which direction you want to expand the selection in. If the selected faces are above and below each other, ModelBuilder expands the selection vertically. If the selected faces are to the left and right of each other, ModelBuilder expands the selection horizontally. If you have multiple groups of selected faces, they all get expanded out.



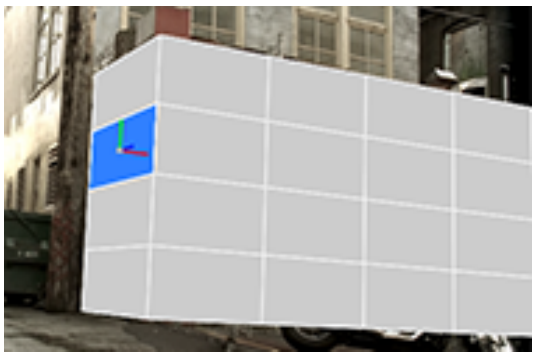
Original selection.



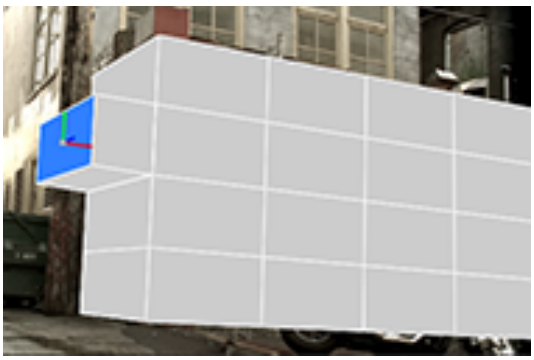
Select face loop.

Face loops only work with four-sided polygons, known as *quads*. If a face loop reaches a face with any other number of sides, the loop ends there.

- To extrude the selected faces, select **extrude** from the right-click menu (or press **Return**) and drag. This stretches the selected faces into a three-dimensional polygon. ModelBuilder creates faces to complete the object.



A selected face.



Extruding the selected face.

If you select **extrude** from the right-click menu again and extrude the same faces, ModelBuilder creates another three-dimensional polygon next to your first one.

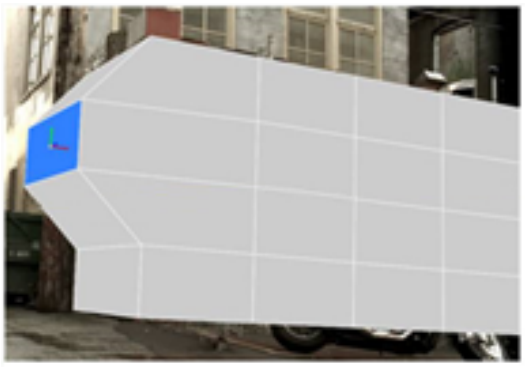
- To merge adjacent faces, select **merge** from the right-click menu.
- To bevel the selected faces, select **bevel** from the right-click menu.

  A colored border appears around the Viewer window to indicate that the action is in progress.

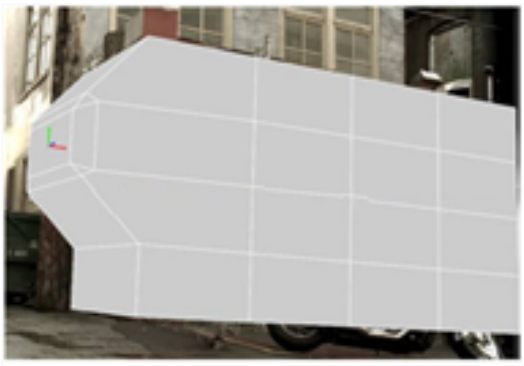
  At the top of the Viewer, use **relative inset** to define how far back along the surrounding edges to start the bevel from: **0.0** means no distance and **1.0** means at the opposite end of the edge. The default is **0.1**, meaning the beveling starts 1/10th of the way back along each edge.

  Set **round level** to the number of times to repeat the initial bevel, effectively rounding the edges. A value of **0** just does the initial bevel, a value of **1** bevels the output of the initial bevel; a value of **2** bevels the bevel of the initial bevel, and so on.

  When you're happy with the values you've entered, press **Return** to apply the bevel.



A selected face.

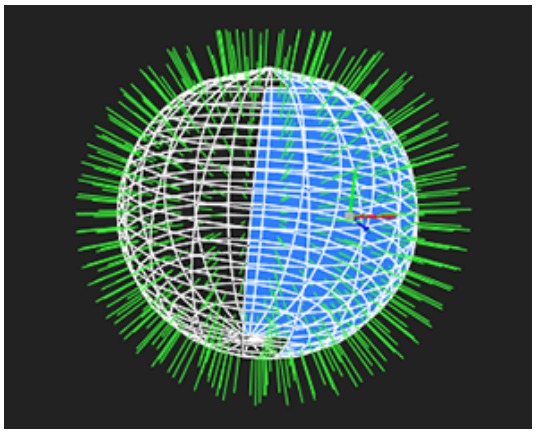


Beveling the selected face.

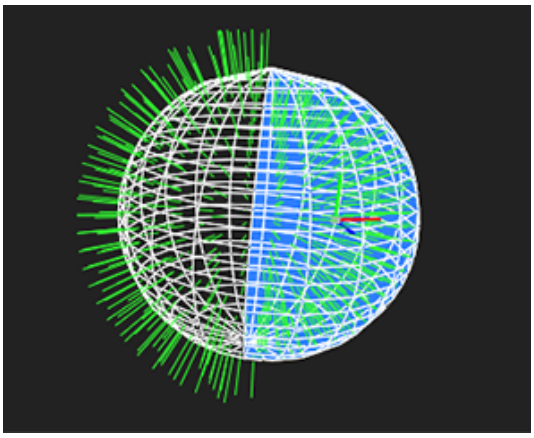
Beveling is like extruding the selected faces, except that the resulting polygons have smooth edges and corners. You may want to use beveling to add realism to your model, as real-world objects rarely have perfectly sharp corners.

- To invert the selected face's normals, select **flip face normals** from the right-click menu. This can be useful, for example, if editing operations (such as a sequence of extrusions) have resulted in normals that point the wrong way.

Tip: To see the normals, press **S** on the Viewer to display the Viewer Settings, go to the **3D** tab, and activate the **show primitive normals** button .

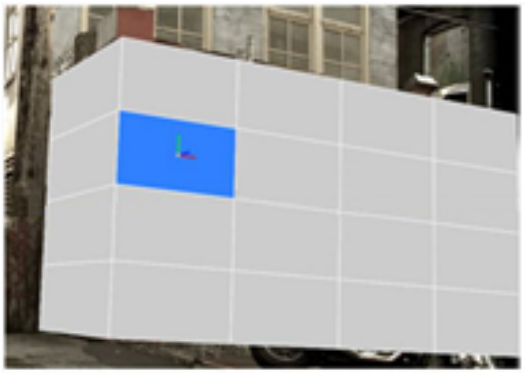


Before **flip face normals**.

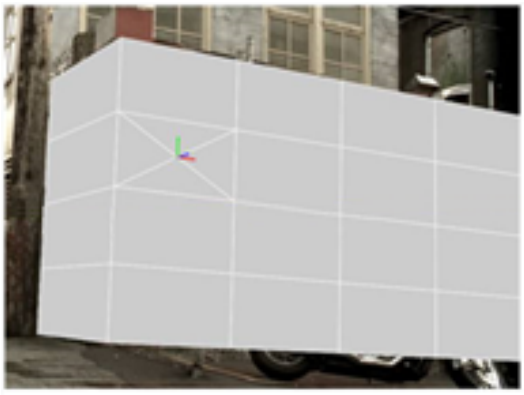


After **flip face normals**.

- To turn the selected face into a triangle fan, select **tessellate** > **triangle fan** from the right-click menu. This adds a new vertex at the center of the face and joins each of the vertices around the perimeter of the face to it. It can be useful if you want to quickly add more detail to your mesh.



A selected face.



The same face with
a tessellated triangle fan.

- To delete the selected faces, select **delete faces** from the right-click menu (or press **Delete**).

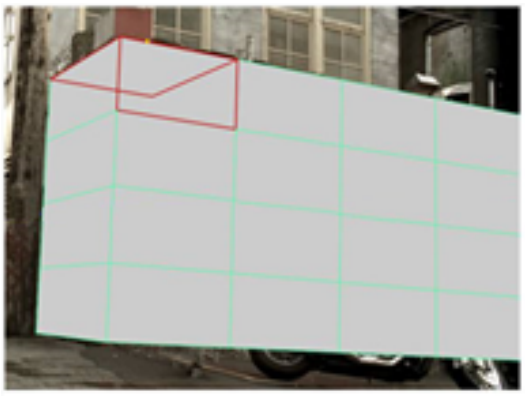
Editing Objects

To edit objects:

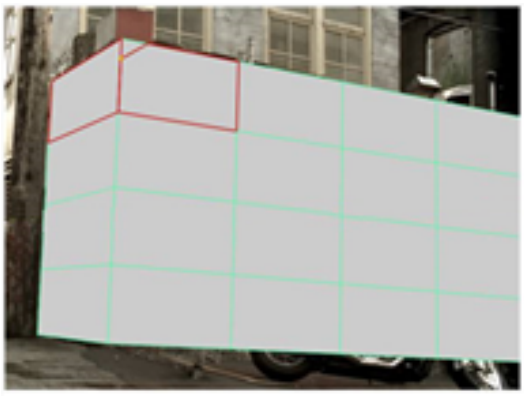
1. Activate **Edit Mode** by selecting  from the ModelBuilder toolbar.
2. Click the **Select object** button  in the ModelBuilder toolbar and select an object.
3. Edit the object as necessary:
   - To translate, rotate, or scale the selected object, drag the transform handles that appear on it. To move the pivot point for the transform handles, press **Ctrl**/**Cmd**+**Alt** while dragging on the center of the transformation overlay.

Tip: If necessary, you can also use the controls at the top of the Viewer to set the initial position and alignment of the transform handles (that is, the position used whenever you change the selection). For more information, see Setting the Initial Transform Action Center.

   - To carve the selected objects, right-click on them and select **carve**. A colored border appears around the Viewer window to indicate that the action is in progress. Click on an edge or vertex to begin a carve. All of the surrounding faces are highlighted in red. Click anywhere inside a highlighted face, or on an edge or vertex of a highlighted face, to carve an edge between the previous vertex and the place you just clicked.



Highlighted faces.

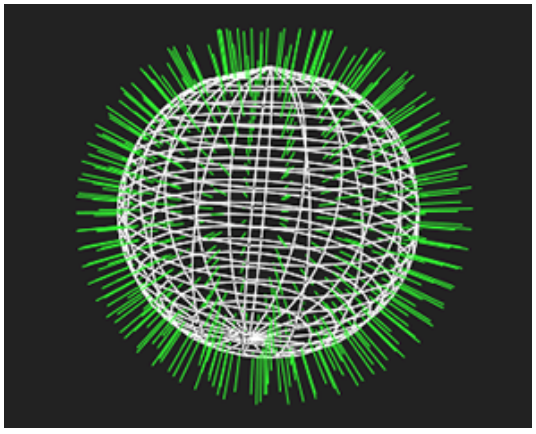


Carving a new edge inside the highlighted faces.

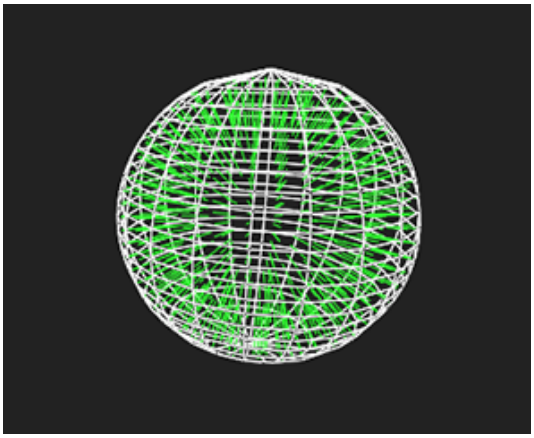
To carve out a freehand edge, hold down **Ctrl/Cmd**+**Shift** and drag.

- To mirror the selected object along its x axis, right-click on the object and select **mirror x**.
- To mirror the selected object along its y axis, right-click on the object and select **mirror y**.
- To mirror the selected object along its x axis, right-click on the object and select **mirror z**.
- To invert the selected object's normals, right-click on the object and select **flip face normals**. This can be useful, for example, if your camera is located inside the selected object.

Tip: To see the normals, press **S** on the Viewer to display the Viewer Settings, go to the **3D** tab, and activate the **show primitive normals** button .



Before **flip face normals**.



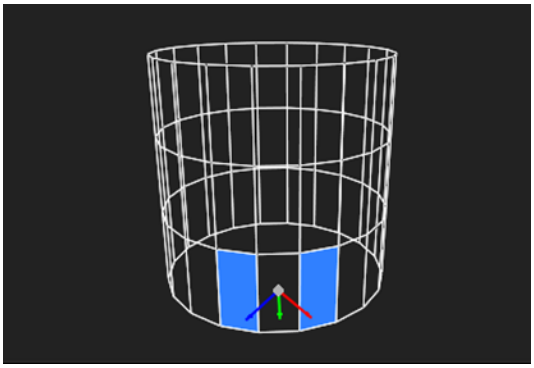
After **flip face normals**.

- To delete the selected object, select **delete objects** from the right-click menu (or press **Delete**).

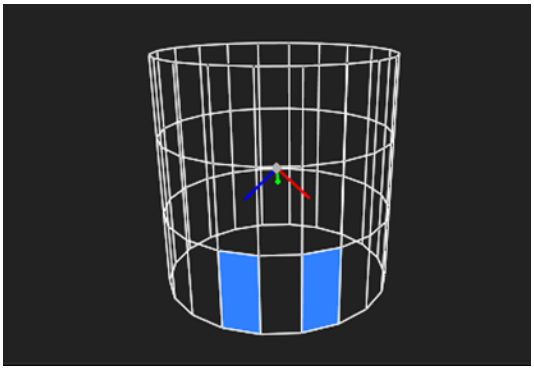
Setting the Initial Transform Action Center

When you translate, rotate, or scale vertices, edges, faces, or objects, you can change the initial position from where the action originates (that is, where the transform handles appear whenever you change the selection). To do so:

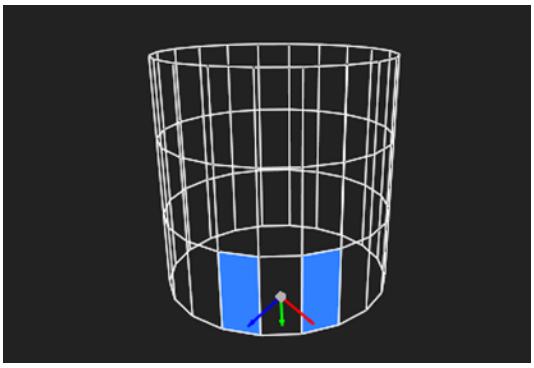
1. Make sure **Edit Mode**  is active in the ModelBuilder toolbar.
2. To change the default position of the transform handles relative to your current selection, set **handle pos** at the top of the Viewer to:
   - **selection average** - The transform handles appear at the average location of all selected items.



   - **object center** - The transform handles appear at the center of the bounding box for the object that the selected items are part of.

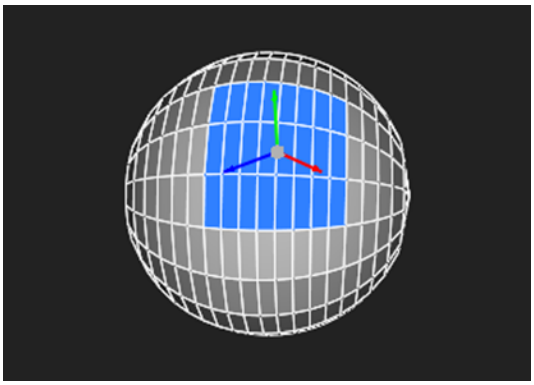


   - **object surface** - The transform handles are snapped to the surface of the object, as close as possible to the average position in the selection. This is the default behavior.

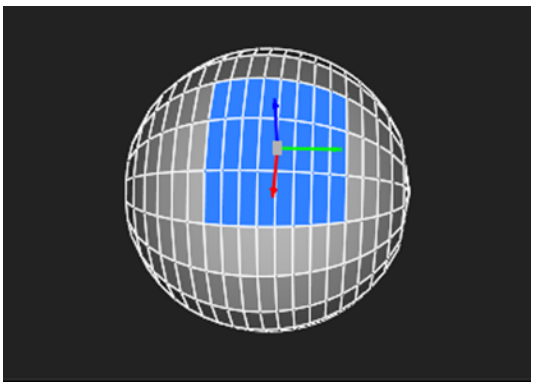


Note: The **object center** and **object surface** options only apply when you have selected items on a single object. If you have selected items on more than one object, **selection average** is always used.

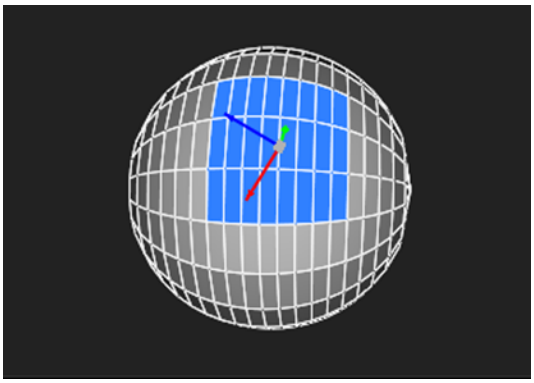
3. To change the orientation of the transform handles relative to your current selection, set **handle align** at the top of the Viewer to:
   - **world axes** - The transform handles are oriented so that the x, y, and z arrows align with the world x, y, and z axes.



   - **object axes** - The transform handles are oriented so that the x, y, and z arrows align with the object's local x, y, and z axes.



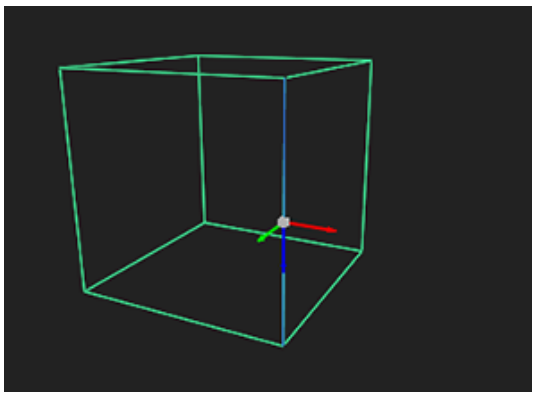
- **surface normal** - The transform handles are oriented so that the y arrow aligns with the object's surface normal and the x and z axes are at an arbitrary orientation. This is the default behavior.



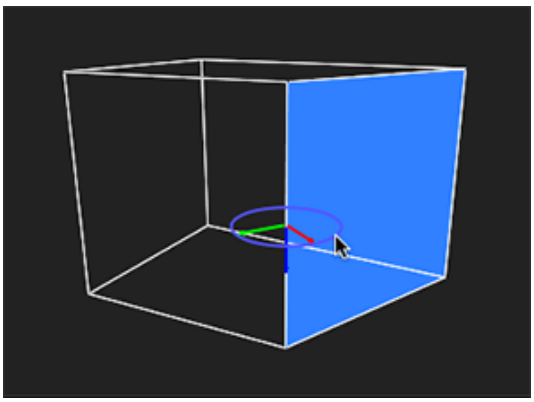
Note: The **object axes** and **surface normal** options only apply when you have selected items on a single object. If you have selected items on more than one object, **world axes** is always used.

4. To prevent the position and alignment of the transform handles from being changed when you change your selection, enable **locked** at the top of the Viewer.

   This makes it easier to rotate a face relative to one of its edges, for example: instead of having to **Ctrl**/**Cmd**+**Alt**+drag the transform handles into the right place by eye, you can set **handle pos** to **selection average**, select the edge, and enable **locked**. When you then select the face, the transform handles stay where they are.



Selecting an edge and enabling **locked**.

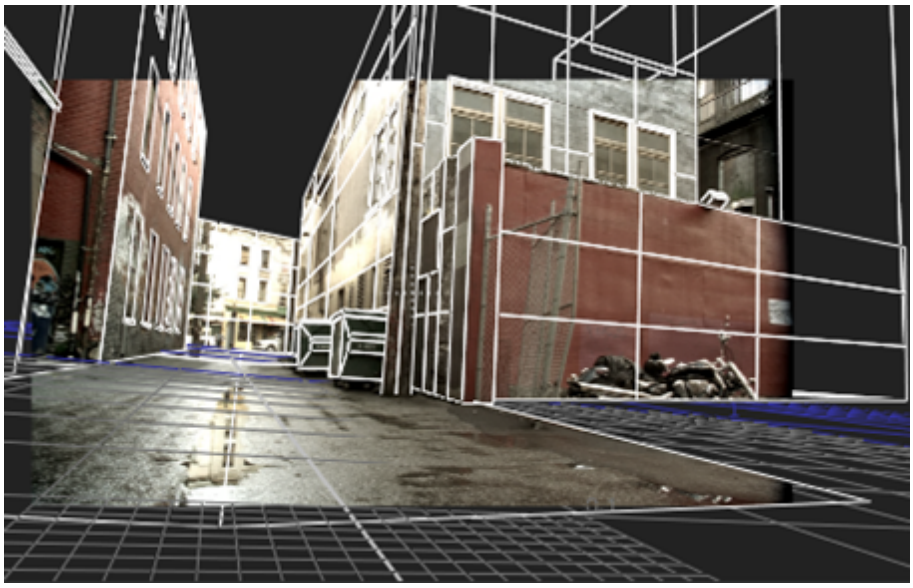


Rotating a face around the previously selected edge.

Applying Textures

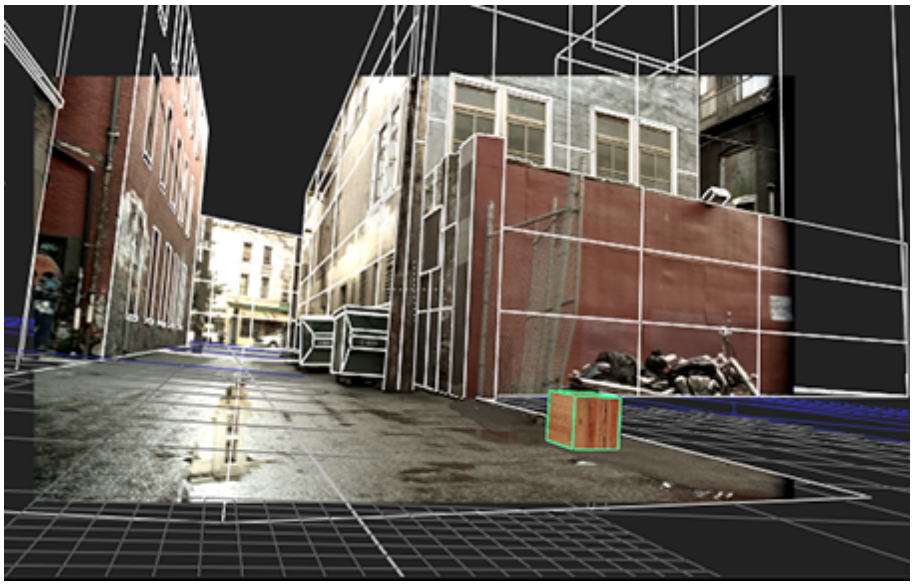
There are a couple of ways to texture your ModelBuilder models in Nuke:

- If your 3D model closely matches the original 2D footage, you can project the 2D footage onto the geometry. See Projecting Textures onto Your Shapes.



Projecting the original 2D footage onto the geometry.

- If you have added new objects over the top of your 2D footage (for example, added a new window to a building or extra bins alongside a street), you can't project your 2D footage over those objects because they were never in the original footage. Instead, you need to supply a texture from somewhere else and map it over the surface of your 3D object. See UV Unwrapping.



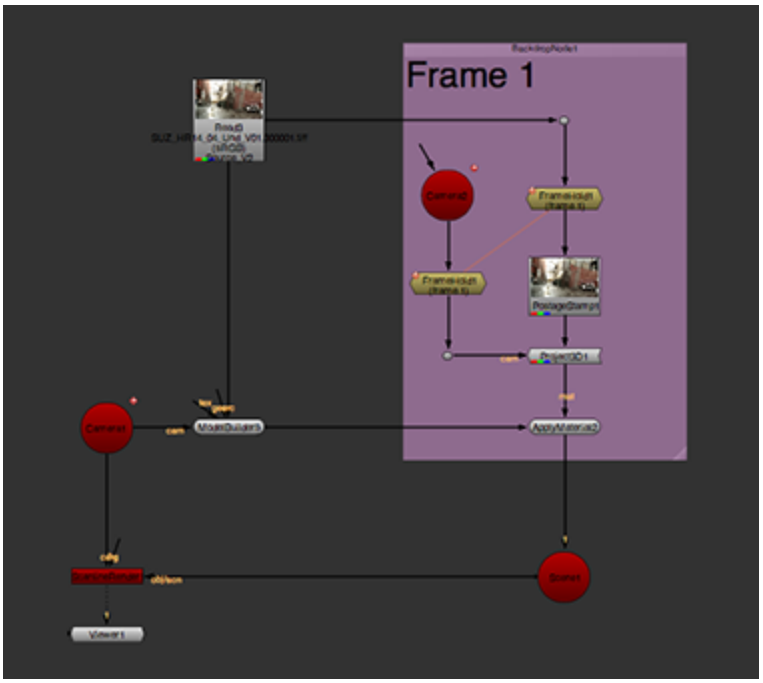
Texturing an added cube object separately.

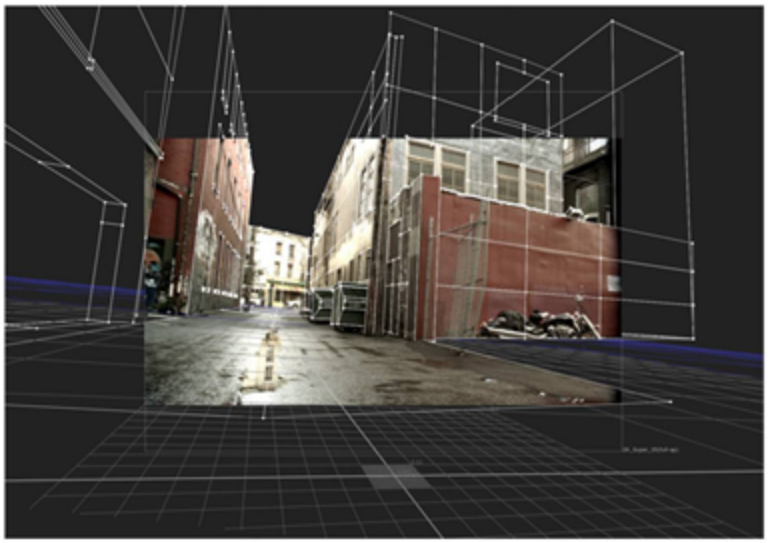
Projecting Textures onto Your Shapes

If your ModelBuilder model closely matches the original 2D footage, you can simply project the 2D footage onto the geometry. To do so:

1. At the bottom of the ModelBuilder properties panel, set the bake menu to **Projection** and click **Bake**.

   This creates a projection at the current texture frame: a Project3D node with a FrameHold node set to lock the input image and camera to the texture frame. The Project3D node is also connected to the **mat** input of an ApplyMaterial node.
2. To apply the projection to the geometry, connect the ModelBuilder node (or a geometry node created by baking **Selected geometry**) to the ApplyMaterial node's unnamed input and view the result.



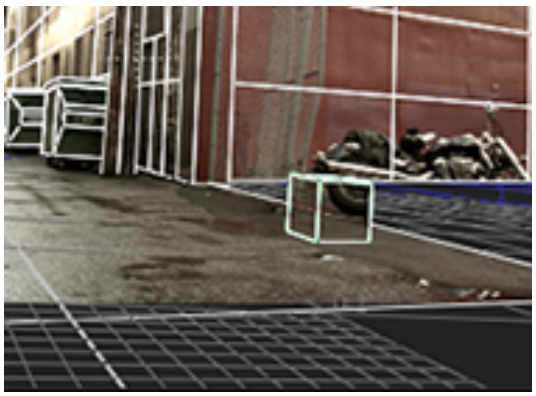


Tip: To texture your geometry on more than one frame, you can bake **Projection** on several frames and use Roto nodes to mask things out before combining the Project3D nodes using a MergeMat node. Then, connect the MergeMat node to the ApplyMaterial node's **mat** input and ModelBuilder to its unnamed input. Connect a Viewer to the output of ApplyMaterial and view the results.

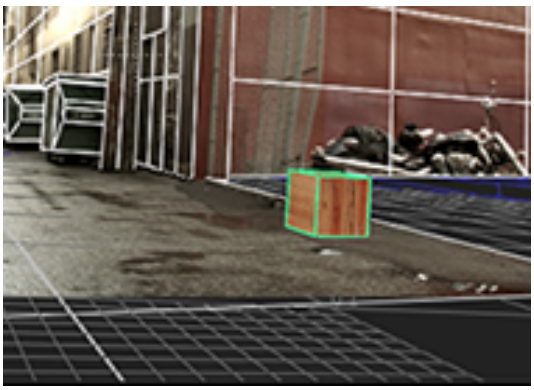
Tip: The options in the bake menu are implemented in Python, and you can also use Python to add your own entries to the menu. To see how the built-in options have been implemented, have a look at the **modelbuilder.py** module in the **nukescripts** folder of your installation (for more information on the location of this file, see Viewing More Examples). Then, to create your own menu options, edit the **modelbuilder.py** file and use the populateBakeMenu function to add entries where it says "# Add your own entries here, if desired".

UV Unwrapping

If you've used ModelBuilder to add new objects over the top of your 2D footage (for example, added a new window to a building or extra bins alongside a street), you can't project your 2D footage over those objects because they were never in the original footage. Instead, you need to supply a texture from somewhere else and map it over the surface of your 3D object.

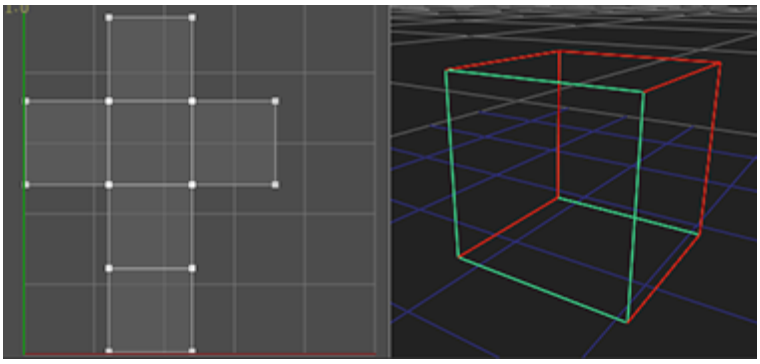


An extra cube has been added to the scene. It cannot be textured using the original footage.

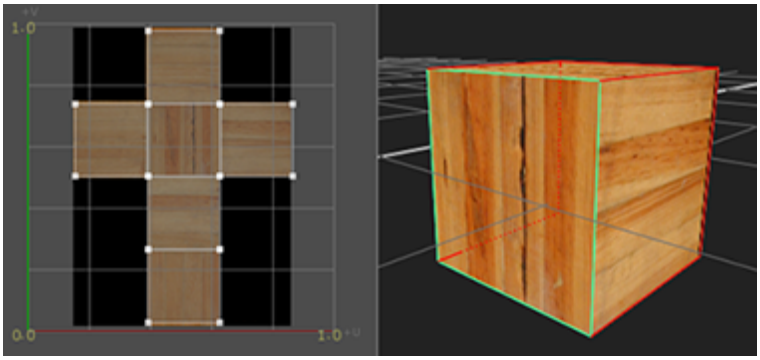


Here, the cube has been textured separately.

To map a texture image over the surface of your 3D object, you first need to flatten your 3D object out into 2D space through a process called UV unwrapping. This generates UV coordinates for each vertex in the object, allowing you to map points on the surface of your 3D object to pixels in your texture image.

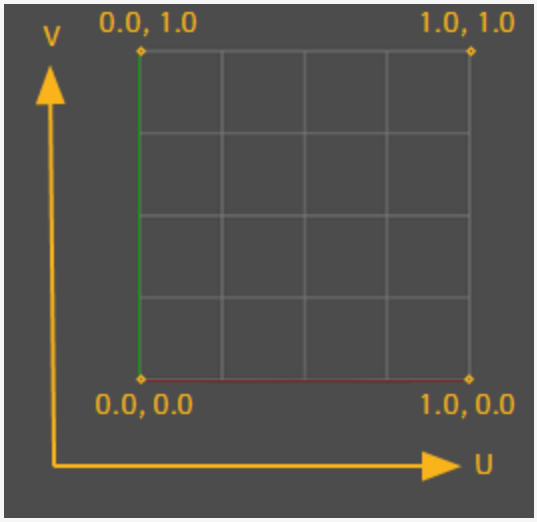


A 3D cube flattened out into 2D space.



Previewing a texture image while editing the cube's UV coordinates.

Tip: UVs are simply 2D coordinates that tell 3D applications how to apply a texture to a model. The letters U and V were chosen because X, Y, and Z were already used to denote the axes of objects in 3D space. The U coordinate represents the horizontal axis of the 2D texture, and the V coordinate the vertical axis.



Quick Start

The process of UV unwrapping is roughly the following:

1. Define where to cut the model in order to flatten it out into 2D space. See Creating Seams on Your Model.
2. Unwrap the model and preview the results. See Unwrapping Your Model and Previewing Its UVs.
3. Edit the generated UVs as necessary. See Editing UVs.
4. Apply a texture to your model. See Applying Textures.

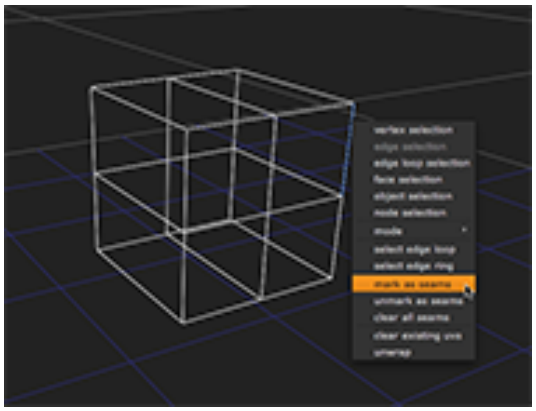
Creating Seams on Your Model

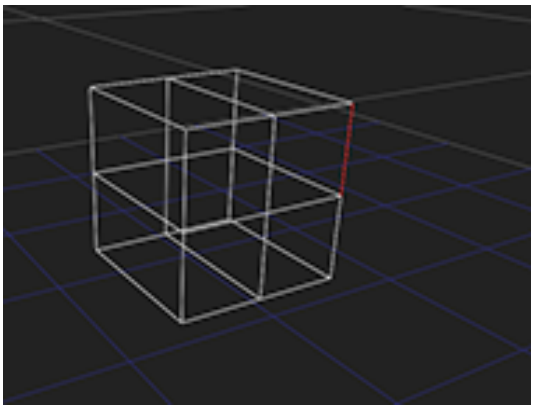
To prepare your model for UV unwrapping, you need to mark some seams. The seams tell ModelBuilder where it's allowed to cut the model in order to flatten it out into 2D space.

Tip: If you don't have enough seams marked, the unwrapping will be very poor with lots of overlapping faces as well as stretching and distortion. On the other hand, too many seams can make it difficult to texture the object, as the gaps in the seams can easily become noticeable on the textured object. If possible, it's a good idea to hide seams in areas where they will not be seen.

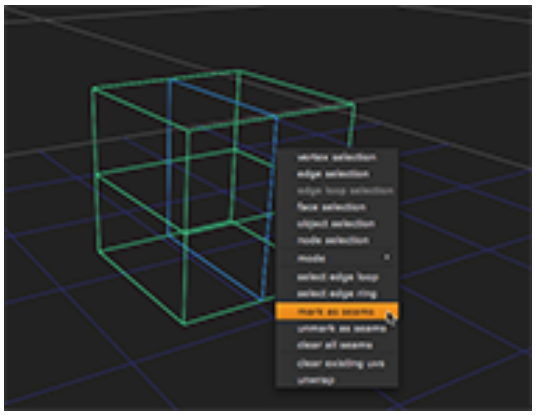
To Create Seams

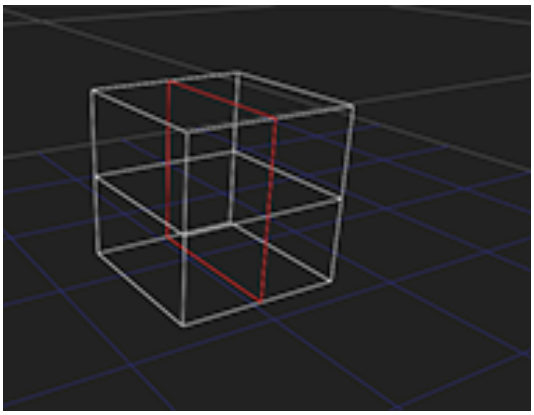
1. If you have created several objects using the same ModelBuilder node, click  in the node properties to hide all the other objects except the one you want to generate UVs for.
2. In the ModelBuilder toolbar, click  to activate UV mode.
3. Repeat one or more of the following until you've got a good set of seams:
   - To mark an edge as a seam, right-click on the Viewer and choose **edge selection**. Select an edge on your model, right-click, and choose **mark as seam**.



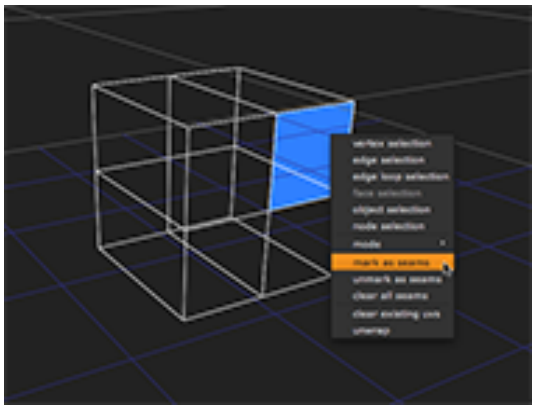


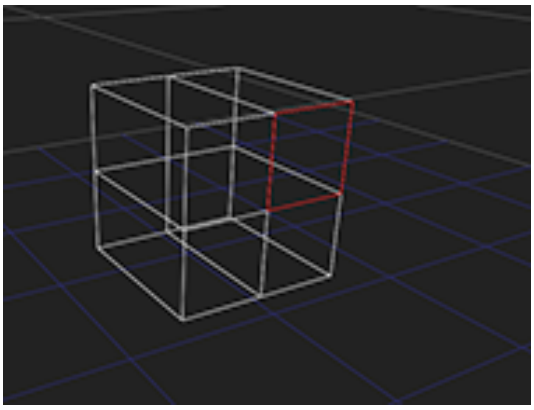
- To mark an edge loop as a seam, right-click on the Viewer and choose **edge loop selection**. Select an edge loop on your model, right-click, and choose **mark as seam**.





- To mark all edges around a face as seams, right-click on the Viewer and choose **face selection**. Select a face on your model, right-click, and choose **mark as seam**.





Tip: You can also select a face loop and mark all edges around it as seams. To do so, select at least two faces on your model, right-click, and choose **select face loop**. Then, right-click on the Viewer again and choose **mark as seams**.

- To remove an edge, edge loop, or face from the set of seams, select it in the Viewer, right-click, and choose **unmark as seam**.
- To clear all seams, right-click on the object and choose **clear all seams**.

Any edges that have been marked as seams are displayed in red in the Viewer.

4. Proceed to Unwrapping Your Model and Previewing Its UVs below.

Tip: The easiest way to create a good set of seams for buildings (or any other roughly cubic or cylindrical object) is to 1) select the top faces and mark them as seams, 2) select the bottom faces and mark them as seams, and 3) select edges along the side of the model that connect the top to the bottom and mark them as seams.

Unwrapping Your Model and Previewing Its UVs

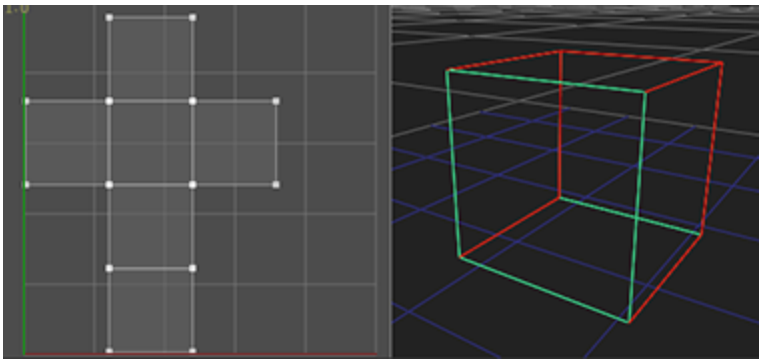
Once you're happy with your set of seams, you are ready to unwrap the model.

To Unwrap a Model and Preview Its UVs

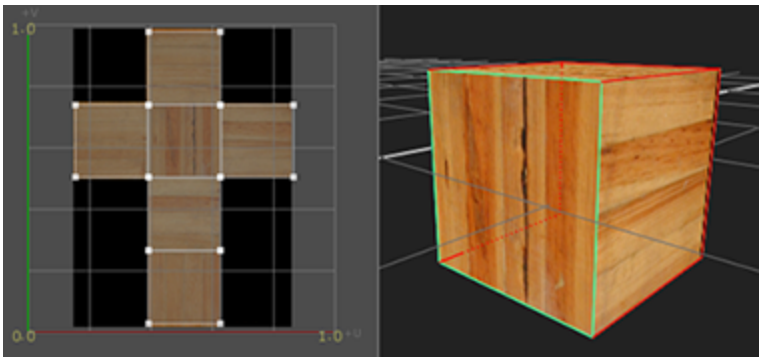
1. Right-click on the Viewer and choose **object selection**.
2. Right-click on your model and choose **unwrap**.

   A colored border appears around the Viewer window to indicate that the action is in progress.
3. Press **Return** to complete the unwrap.

   ModelBuilder generates UV values for each of the vertices on your model. The unwrapped model appears in the UV preview window in the top left of the Viewer.



4. To display a 2D texture image as a background in the UV preview window, connect the image to ModelBuilder's **tex** input and make sure the **Preview** dropdown in the ModelBuilder properties (or on top of the Viewer) is set to **tex input**. To also see the **tex** input on the model in the 3D Viewer, set **display** to either **textured** or **textured+wireframe** in the ModelBuilder properties.



5. To zoom in or out of the UV preview window, press **Ctrl**/**Cmd**+**Alt** and drag.
6. To pan the UV preview window, press **Ctrl**/**Cmd** and drag.
7. To change the size of the UV preview window, click and drag on its lower right corner.
8. To hide the UV preview window, disable **uv window** at the top of the Viewer.
9. Proceed to Editing UVs below.

Tip: If you are not happy with the generated UVs, you can right-click on your model and select **clear existing uvs**. If necessary, you can also change the seams and unwrap your model again. Each time you do, ModelBuilder replaces the UVs you had with the newly generated UVs.

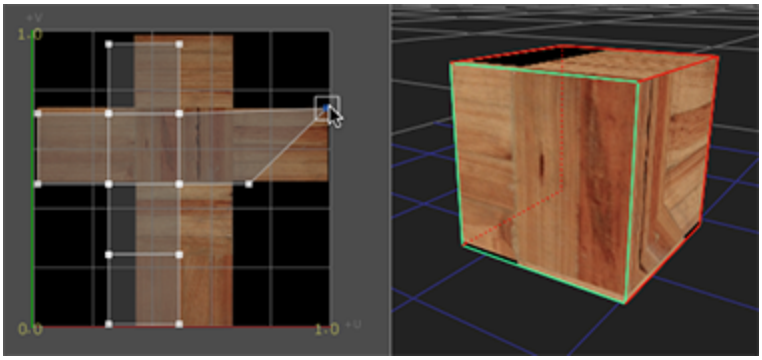
Editing UVs

You can tweak the controls at the top of the Viewer or edit the UVs manually in the UV preview window.

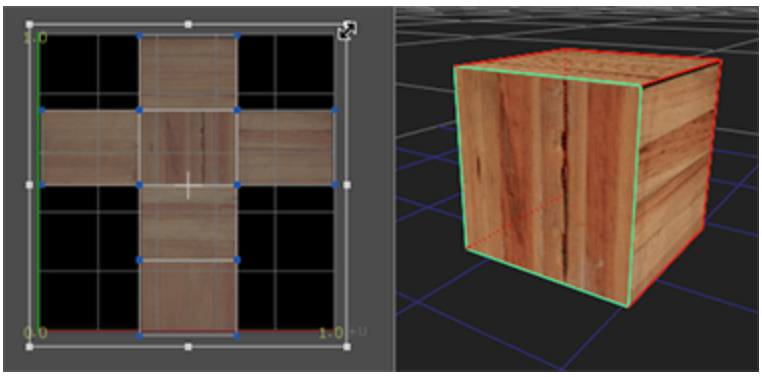
1. Adjust the following controls at the top of the Viewer:
   - **iterations** - The number of times the unwrapper applies its rules about how to move the UV coordinates towards good locations. This gives you a speed vs. quality trade-off: more iterations give a better result but take longer to unwrap. Note that the results can only be improved up to a point, and where that point is depends on the complexity of your model.
   - **threshold** - The unwrapper stops before reaching the maximum number of **iterations** if it sees that the improvement from iteration to iteration is negligible. The **threshold** control tells it how much change is considered negligible. If the amount of change between iterations is below the threshold, the unwrapper stops. This is another way to trade off quality for speed. You generally use this if you want a higher quality result: if you've increased **iterations** but the unwrapper is stopping before reaching the maximum, you can lower the threshold to make it keep running.
   - **separation** - The number of pixels to leave between each patch in the unwrapping. If you don't have enough pixels between your patches, you can get color bleeding through from neighboring patches when you use the texture. A higher value means more widely spaced patches, but also more wasted space in your texture image.

As you edit these controls, the UV preview window updates to allow you to see the effect of your changes.

2. If necessary, you can also edit the UVs manually in the UV preview window:
   • To edit a single vertex, drag it to a new location.



   • To edit several vertices together, select them and use the transform jack that appears to translate, rotate, or scale the selection. The transform jack is the same one used elsewhere in Nuke, so all the controls work the same way. For example, you can click and drag on the handles to scale out from the center, or **Ctrl**/**Cmd**+click and drag to scale out from the opposite edge.



3. Proceed to Applying Textures.

Applying Textures

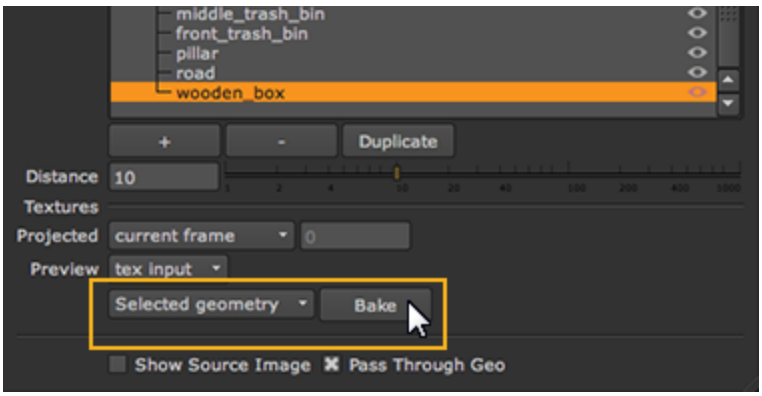
At this point, it may be that you've got several objects in the scene that you still want to project your camera footage onto (see Projecting Textures onto Your Shapes), but there's one object that you want to texture differently. There are two ways to do this:

• You can export your model as a separate geometry node and use an ApplyMaterial node to texture it. This can be a convenient way to work if your geometry is finished and locked off, so you never need to go back and change it. But what you lose with this is the "live" view: if you do need to go back and edit the object, you have to export it again. See Method 1.

• Alternatively, you can add an ApplyMaterial node directly after your ModelBuilder node and tell ApplyMaterial to ignore all the geometry that doesn't match the filter you give it. See Method 2.

Method 1

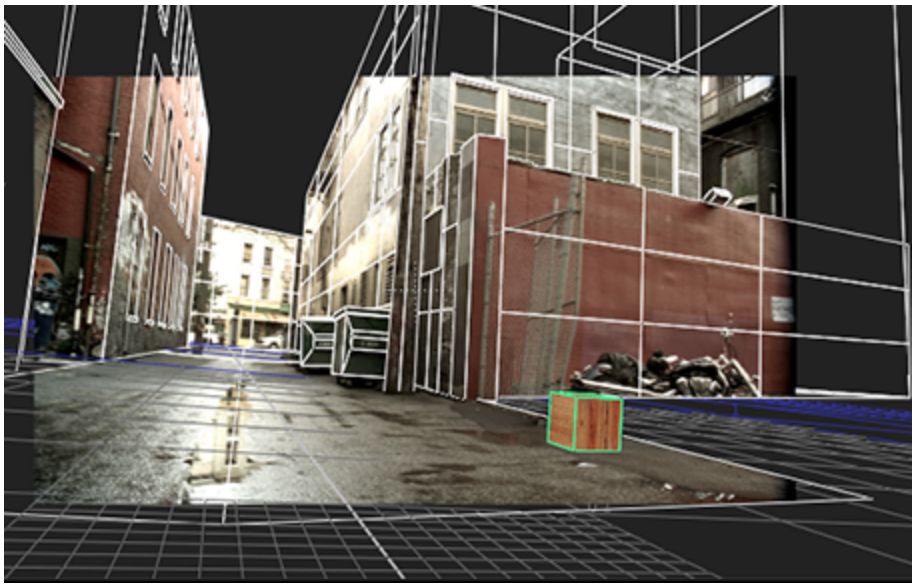
1. In the ModelBuilder properties, select the object that you want to texture differently.
2. At the bottom of the properties panel, set the bake menu to **Selected geometry** and click **Bake**.



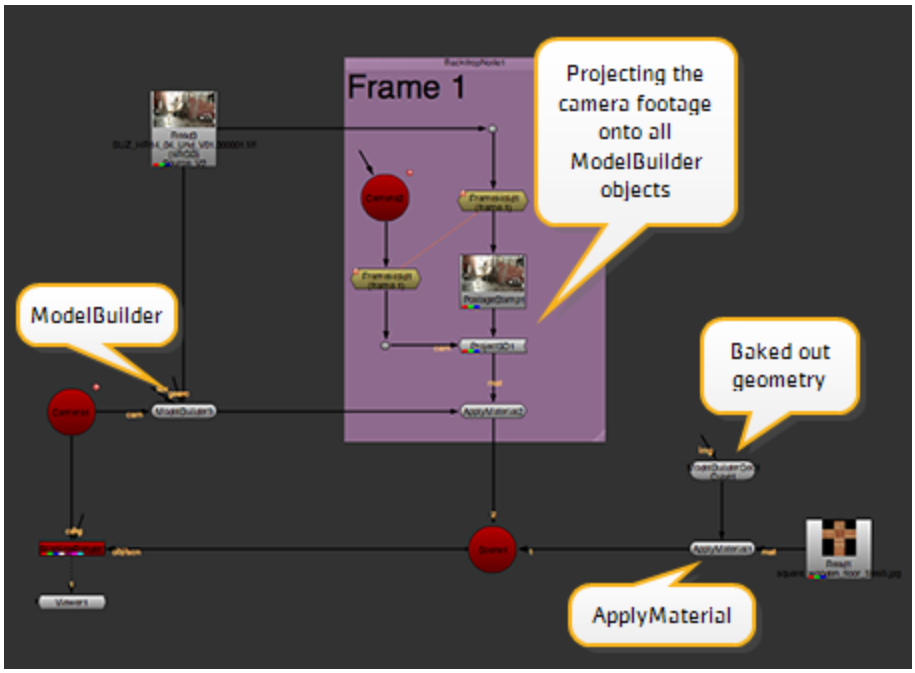
ModelBuilder creates a geometry node for the selected object.

3. From the toolbar, select **3D** > **Shader** > **ApplyMaterial** to create an ApplyMaterial node.
4. Connect the geometry node you created in step 2 to the unnamed input of the ApplyMaterial node.
5. Then, connect your 2D texture image to the **mat** input of the ApplyMaterial node.

   The ApplyMaterial node applies the texture from the **mat** input onto your 3D geometry object. (To be able to see this in the 3D Viewer, you may have to hide the object in the ModelBuilder properties.)



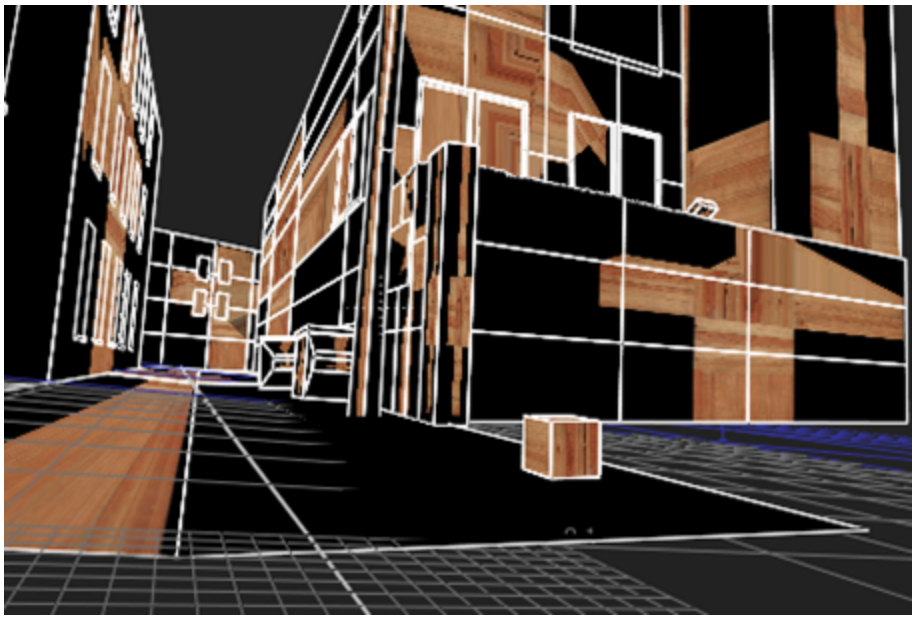
6. Connect the ApplyMaterial node to your Scene node and use a ScanlineRender node to render all the objects connected to that scene.



Method 2

1. From the toolbar, select **3D** > **Shader** > **ApplyMaterial** to create an ApplyMaterial node.
2. Connect the unnamed input of the ApplyMaterial node to your ModelBuilder node and the **mat** input to your 2D texture image.

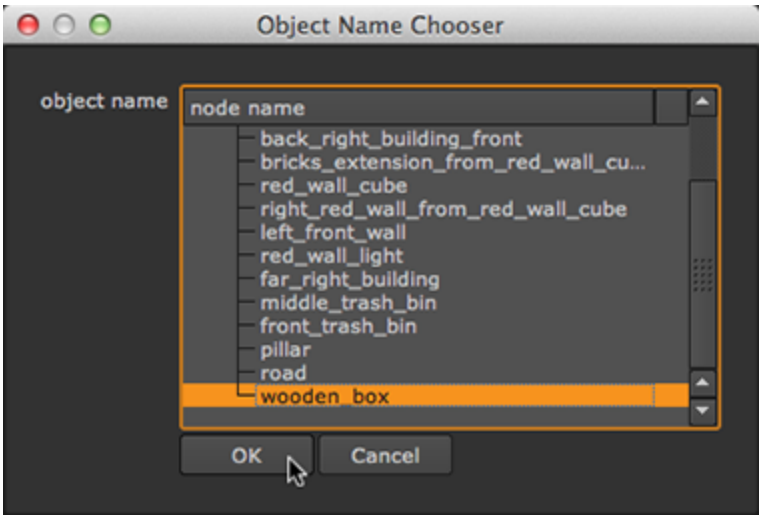
   By default, ApplyMaterial applies the texture from the **mat** input onto all objects in your ModelBuilder node.



3. To only apply the texture onto a particular object, open the ApplyMaterial properties and set **filter** to **name**. This allows you to tell ApplyMaterial to ignore any geometry that doesn't match the filter.
4. To set the filter, click the **choose** button. In the dialog that opens, select the object you want to apply the texture to and click **OK**.

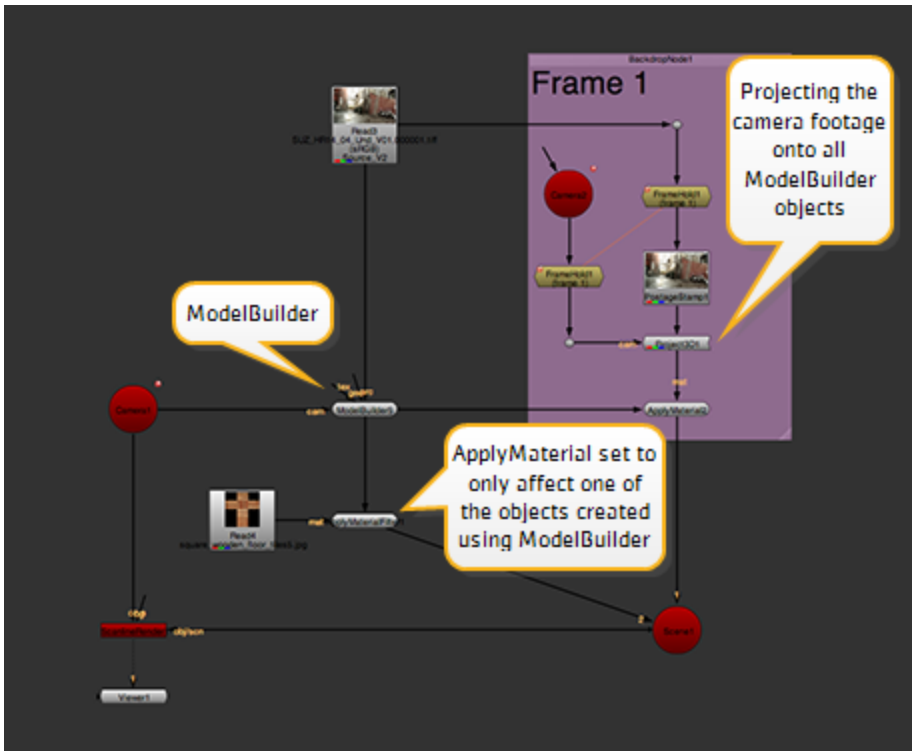


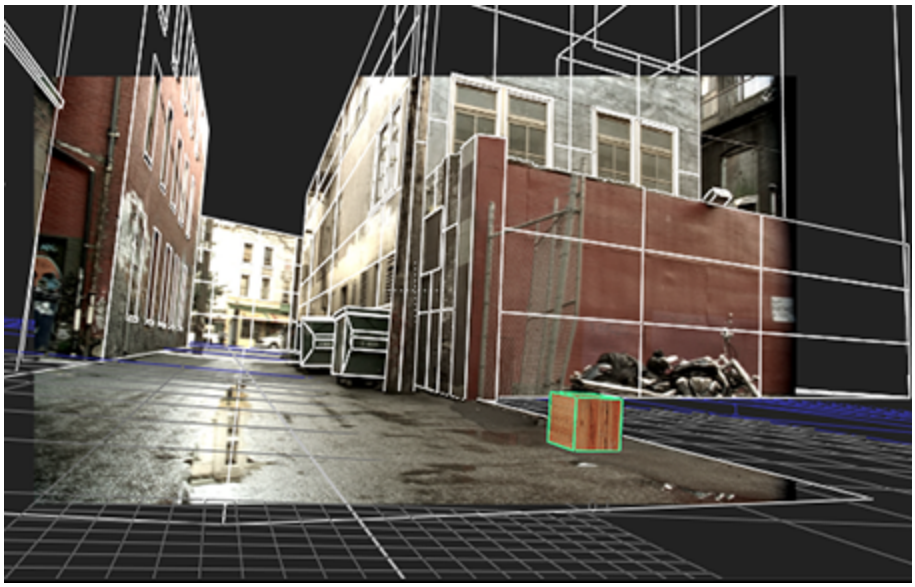
Tip: You can also **Ctrl/Cmd**+click or **Shift**+click to select multiple objects.



For more information on ApplyMaterial, see Applying a Material Using the ApplyMaterial Node.

5. Connect the ApplyMaterial node to your Scene node and use a ScanlineRender node to render all the objects connected to that scene.


Frame 1
Projecting the camera footage onto all ModelBuilder objects
ModelBuilder
ApplyMaterial set to only affect one of the objects created using ModelBuilder



Exporting Shapes to Separate Geometry Nodes

If necessary, you can export shapes on your ModelBuilder model to separate geometry nodes. This allows you to operate on one part of the scene separately from the rest. You can use the geometry nodes in the same way as any other geometry nodes in Nuke.

To Export Shapes to Separate Geometry Nodes:

1. In the ModelBuilder properties, select any shapes you want to export to a separate geometry node.
2. Make sure the bake menu under **Export** is set to **Selected geometry** and click **Bake**.
   ModelBuilder creates a geometry node for the selected items in the scene.



Tip: The options in the bake menu are implemented in Python, and you can also use Python to add your own entries to the menu. To see how the built-in options have been implemented, have a look at the **modelbuilder.py** module in the **nukescripts** folder of your installation (for more information on the location of this file, see Viewing More Examples). Then, to create your own menu options, edit the **modelbuilder.py** file and use the populateBakeMenu function to add entries where it says "# Add your own entries here, if desired".

Creating 3D Particles

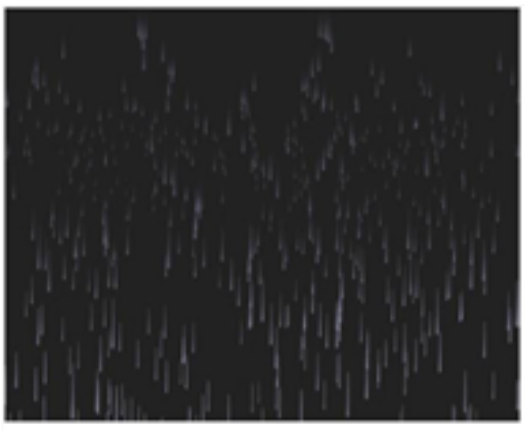
Nuke's Particle node set is a solution for creating particles in a 3D environment. You can create things like smoke, fog, falling snow, explosions, and bubbles - the possibilities are endless. You can use the various

Particle nodes for emitting, manipulating, and displaying limitless types of particles in your 3D scene.

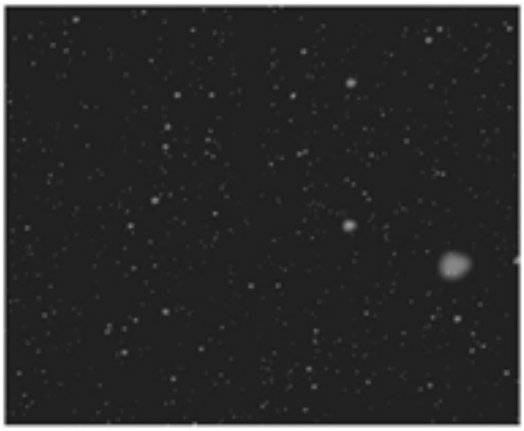
Quick Start

Here's a quick overview of the workflow:

1. Create a ParticleEmitter node, and connect it to a Viewer.
2. Connect a source of emission and a particle representation to the **emit** and **particle** inputs of the ParticleEmitter. For more information, see Connecting Particle Nodes.
3. Modify your particles' lifetime, velocity and other basic properties in the ParticleEmitter properties panel. For more information, Emitting Particles.
4. Connect other particle nodes to the ParticleEmitter's output. See Adjusting the Speed and Direction of Particles, Modifying the Particles' Movement and Adjusting Particle Simulation Settings.
5. If necessary, cache your particle simulation in order to read it back in without the need for recalculation. For more information, see Caching Particles.



Rain created using particles.

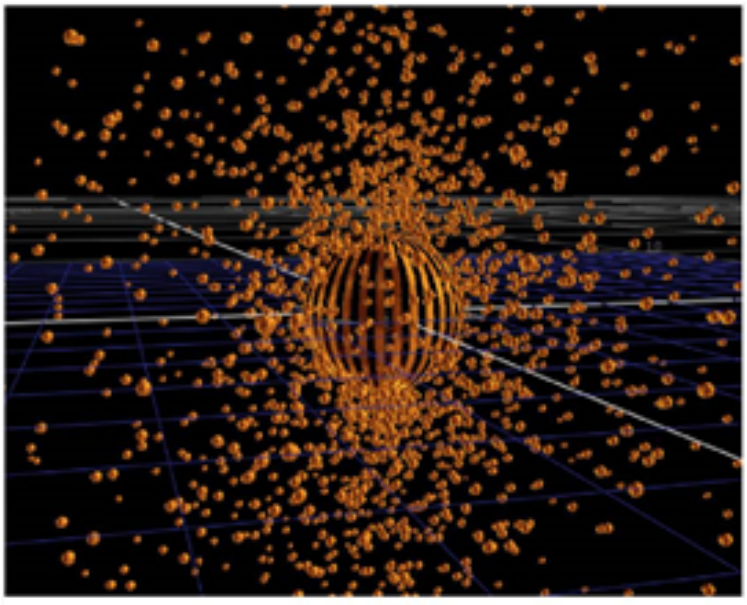


Snow created using particles

Connecting Particle Nodes

In order to create particles, the minimum setup you need is the ParticleEmitter node. To connect your Particle nodes:

1. Click the **Particles** menu in the Toolbar and select the ParticleEmitter node.
2. Connect it to a Viewer node.
3. Connect a 3D geometry object, or PositionToPoints point cloud, to the **emit** input of the ParticleEmitter. A 3D object from which to emit the particles is optional: if you don't use one, the particles are emitted along the y axis from a point of origin.
4. To specify the appearance of your particles, connect an image or a geometry in the **particle** input of the ParticleEmitter. This image or geometry is then multiplied and used as representations of each of your particles. If you want to use more than one representation, connect further image or geometries to the other numbered **particle** inputs. ParticleEmitter picks one of these at random for each particle.
5. If you have another particle system that you'd like to connect to your new ParticleEmitter, you can connect it to the **merge** input. You can also merge particle systems with the ParticleMerge node (see Merging Particle Streams).
6. Now you're ready to start modifying your particles to look the way you want. To do this, you can pick any of the various particle nodes in the Toolbar's **Particle** menu and connect them to the ParticleEmitter node's output, or to other particle nodes. Each of them has its own effect on the particles and a set of controls you can make adjustments with.

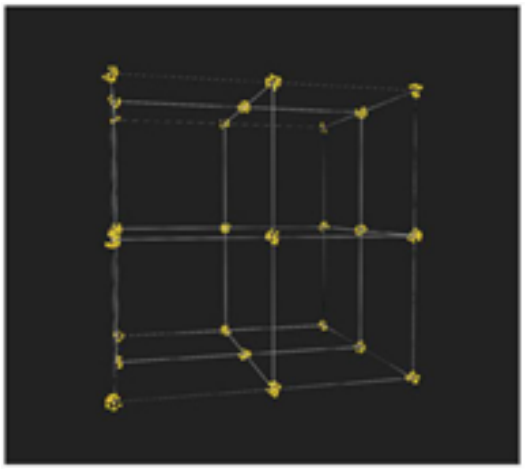


Particles emitted from Sphere geometry.

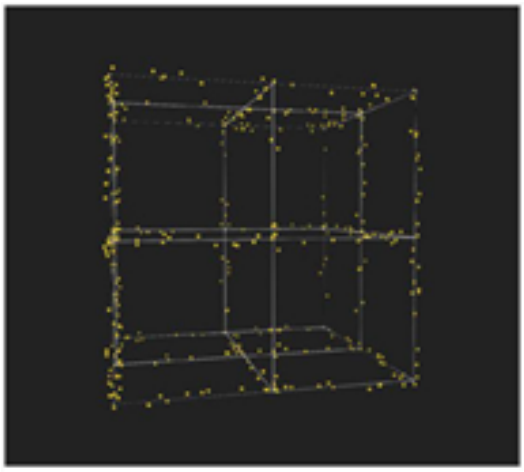
Emitting Particles

The ParticleEmitter node is the first and only required node in a minimum particle setup. Once you've created a ParticleEmitter, connected it to a Viewer and clicked **play** on the timeline, you'll see the default set of particles emitting (from a 3D geometry or point cloud, if you've connected one). You can then adjust the ParticleEmitter controls to change the way the particles appear:

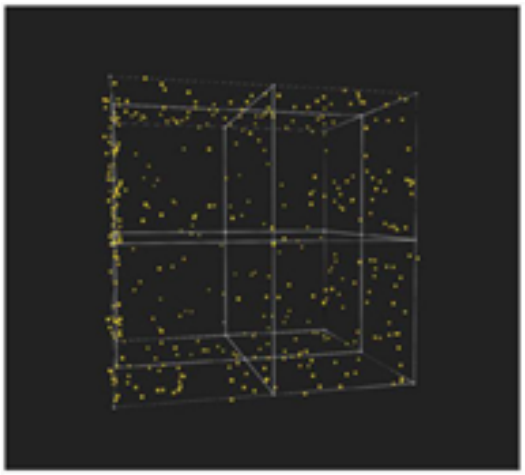
1. Set the **channels** in which particles are emitted. Channels **a** and **b** are arbitrary names for channels which are useful if you want different particle force nodes to have an effect on separate channels.
2. Use the **start at** field to pre-roll or delay the point at which the first particles are emitted. For example, particles could imitate snow that has fallen already, instead of displaying the first flakes falling down, by using a negative **start at** frame.
3. Select the emission order and rate for the particles. Set:
   - **emit from** - select **points**, **edges**, **faces** or **bbox** (bounding box) to specify from which part of the 3D object, or point cloud, particles are emitted.



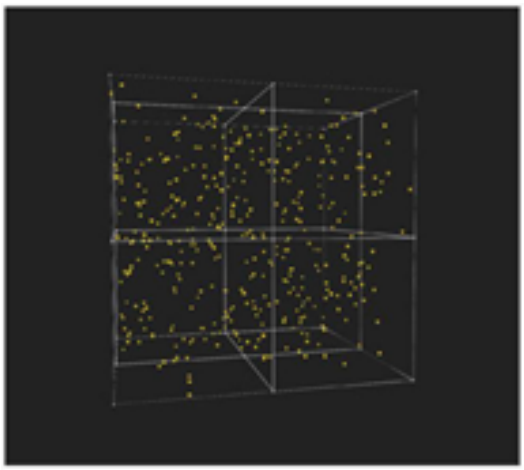
emit from > **points**



emit from > **edges**

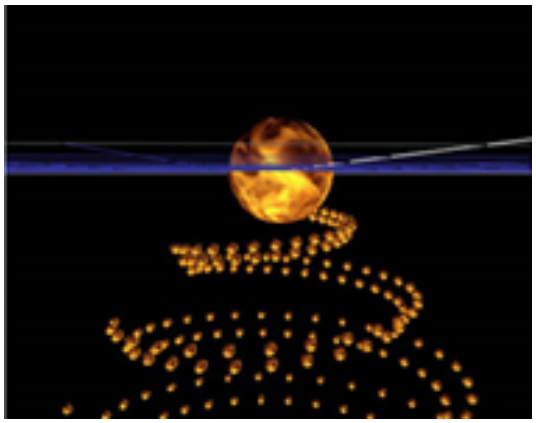


emit from > **faces**

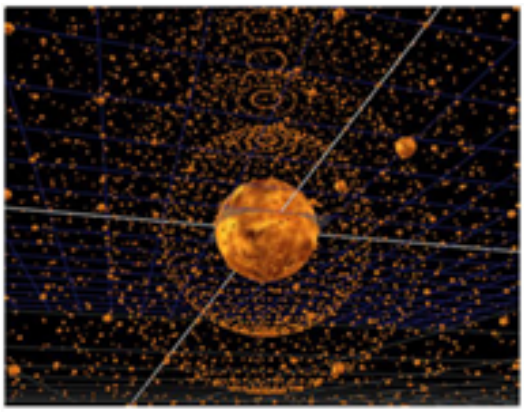


emit from > **bbox**

- **emit order** - select the order that the particles are emitted:
- **randomly** - produces a random emission order.
- **uniformly** - emits all particles at the same time.
- **in order** - emit particles as a multiple of the **emission rate**. For example, an **emission rate** of 2 could emit from two points, edges, or faces at a time.



Particles emitted in order.

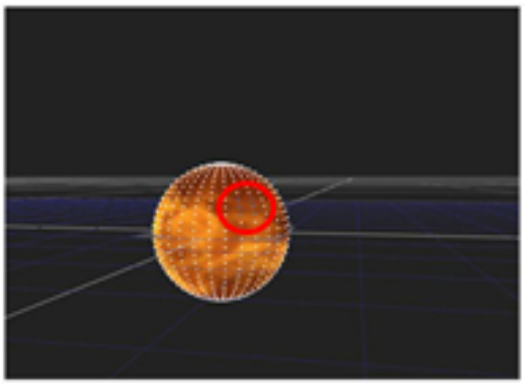


Particles emitted uniformly.

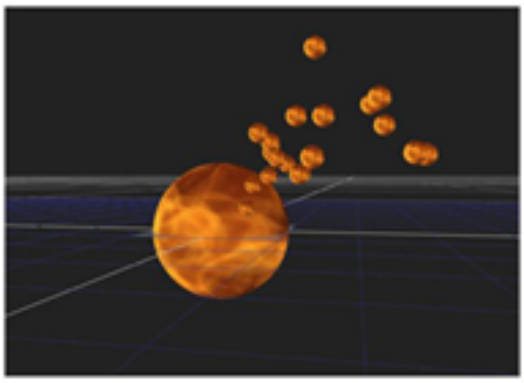
- **randomize type** - select whether or not particles are emitted randomly. Select **no random direction** for a specific direction based on the emit object's normals, **randomized direction** for a randomly selected initial direction, or **randomize outwards** for a randomly selected direction (dependent on the **emit from** selection):
- **bbox** - particles move randomly away from the center.
- **points**, **edges**, and **faces** - particles move randomly away from the origin, but at no more than 90 degrees from the nearest normal.

- **emission rate** - select the number of particles emitted per frame. This is an exact number, and it is affected by the **rate channel**. If the channel is at a varying value, the emission rate also increases or decreases.
- **only emit from selected points** - specify whether the particles are emitted from the object, or selected vertices of the object using a GeoSelect node. You can also emit from selected points in a PositionToPoints point cloud using any normals information present.

  See 3D Selection Tools for more information.



Vertices selected using an upstream GeoSelect node.



Emitting particles only from the selected vertices.

- **rate variation** - specify the range of variation for emitting particles. If you set this to zero, particles are emitted at a constant even level.
- **rate channel** - select a specific channel to which you want to emit particles. Unchecking this is the same as selecting none, and in that case particles are emitted to all the channels. For example, if you are emitting from a Card node that has a Ramp texture, it emits particles from the light parts of the ramp at a higher rate (values closer to 1) than from the dark parts (values closer to 0).

4. Set the particles' color and channels. Enter:
   - **color** - select a color for your particles. Use this if you're not using the **particle** input.
   - **color from texture** - check to tint the particles with the colors from the geometry's texture. If you leave this unchecked, the particles get color only from their own texture.
   - **channels** - select the channels you want to apply particles to. By default, particles are emitted to channel **a.** Channels **a** and **b** are arbitrary names for channels which are useful if you want different ParticleEmitter nodes or other particle force nodes to have an effect on separate channels. An example of this might be if you want two ParticleEmitter nodes emitting particles, one to **a** channel, the other to **b** channel, so further down in your particle stream you can apply an effect to a specific set of particles.
5. Select how long you want the particles to exist. Set:
   - **max lifetime** - specify the maximum life span for a particle in frames.
   - **max lifetime range** - adjust the range within which your particles' lifetime varies. If you set this to 0, all particles have the same lifetime.

- **lifetime channel** - select a channel of the input geometry texture that you want to use to modulate the lifetime. For example, if you are emitting particles from a Card node with a Ramp texture in the lifetime channel, particles emitted from the lighter parts of the ramp (values closer to 1) would have a lifetime value closer to that set in the **max lifetime** than particles emitted from the dark parts (values closer to 0).
- **halflife** - select the number of frames over which the number of the particles emitted is halved.

6. Adjust the velocity and rotation for your particles. Set:
   - **velocity** - specify the initial speed at which you want your particles to move.
   - **velocity range** - adjust the range within which you want your particles' velocity to vary. If you set this to 0, the velocity doesn't vary between particles.
   - **velocity channel** - select a channel of the input geometry texture that you want to use to modulate the velocity. For example, if you are emitting from a Card node that has a Ramp texture, it emits particles from the light parts of the ramp at a higher velocity (values closer to 1) than from the dark parts (values closer to 0).
   - **rotation velocity** - adjust the initial speed at which each particle rotates around its individual Y-axis. The Y-axis points to the direction the particles were initially emitted, and then stays unchanged (unless you use the ParticleMotionAlign node to change its direction). Rotational velocity is most useful when you're emitting particles from a geometry object.
   - **rotation velocity range** - adjust the scale of a random variation in the **rotation velocity** value. Value of 0 means the rotation velocity value is as set, value of 1 means it's very random.
   - **rotation velocity channel** - Select a channel of the input geometry texture that you want to use to modulate the speed of rotation. For example, if you are emitting from a Card which has a Ramp texture, the rotation velocity of the particles emitted from the light part of the Ramp (values closer to 1) is greater than that of those emitted from the dark parts (values closer to 0).
   - **transfer velocity** - adjust this to transfer the velocity of the initial emitter to the particles. If you set this to 0, no velocity is transferred to the spawned particles. At value 1, full velocity is transferred from the originating particle.
   - **transfer window** - adjust the time, in frames, to look forward or backward in order to determine the velocity that should be transferred to the particles.
7. Modify the size and mass of the particles. Set:
   - **size** - specify the initial size of each particle.
   - **size range** - specify the range within which your particle size varies.
   - **size channel** - select a channel of the geometry texture that you want to use to modulate the size of the particles. For example, if you are emitting particles from a Card node which has a Ramp texture in the size channel, the size of the particles emitted from the lighter part of the Ramp (values closer to 1) is greater than that of those emitted from the darker parts (values closer to 0).
   - **mass** - adjust the initial mass of each of your particles. The mass of the particles only becomes relevant when you apply a force to your particles, such as one generated by the ParticleDirectionalForce node.

- **mass range** - adjust to produce a random variation in the mass value. Simply put, 0 value means the particles' mass is the value specified by the **mass** control, whereas a value of 1 means it's very random.
- **mass channel** - select a channel of the input geometry texture that you want to use to modulate the mass. For example, if you are emitting from a Card which has a Ramp texture in the mass channel, the particles from the light part of the Ramp (values closer to 1) has a higher **mass** value than from the dark parts (values closer to 0). With this control, you can emit particles with different masses from different areas based on the input geometry's texture.
- **spread** - adjust the extent to which you want your particles to spread in different directions during their lifetime. By default, this forms a cone around the direction of emission. If you set this to zero, each particle has a straight trajectory.

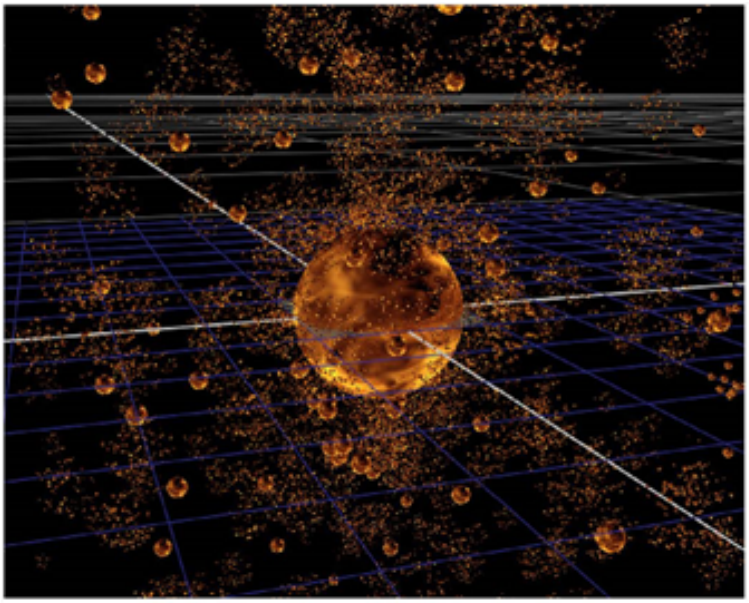
8. Modify the way your particles are affected by the ParticleEmitter's inputs. Adjust:
   - **input order** - if you're using more than one particle input, you can select which **particle** input Nuke should select when creating particles. Select **Randomly** to pick one of the inputs randomly, or **in order** to rotate the inputs in numerical order.
   - **start at** - select which frame of the **particle** input should be the representation of each new particle. Select **first** to pick the first frame of the particle input for each new particle. Select **in order** to pick consecutive frames from the input for each new particle. Select **current** to pick the frame where the particle was emitted. Select **random** to pick a random input frame for each new particle
   - **limit to range** - limit the particle output to the representation's frame rate, looping the frame range when **in order** or **current** is selected.
   - **advance** - use this to determine if a particle should animate after being emitted. Select **constant** to keep the same representation throughout the particle's lifetime. Select **in steps** - to animate the representation frame by frame. Select **randomly** to animate the representation one random frame after another.
9. Vary the results of your range controls with the **random seed** field. Enter the random number used in the range controls (such as **max lifetime range**) to achieve slightly different effects.

Tip: When you set large values for any of the particle controls it might take Nuke a few moments to calculate the result. In such a case a progress bar appears with a **Cancel** button you can use to cancel the calculation if necessary.

Spawning Particles with ParticleSpawn

If you're looking to have your existing particles emit even more particles, you should turn to ParticleSpawn.

1. Connect the ParticleSpawn node to your particle stream (the ParticleEmitter output, for example). All the particles emitted now start spawning more particles.
2. Adjust the ParticleSpawn controls. Most of the ParticleSpawn controls are identical to those in the ParticleEmitter node (see Emitting Particles), with only a few exceptions:
   - **transfer velocity** - adjust this to transfer the velocity of the initial emitter to the particles. If you set this to 0, no velocity is transferred to the spawned particles. At value 1, full velocity is transferred from the originating particle.
   - **conservation of mass** - check if you want the mass of spawned particles to be removed from the mass of the original particle. If the mass of a particle is zero at the end of a frame, it gets deleted.
   - **conservation of momentum** - check this to subtract the momentum of the spawned particles from the original particle, in correspondence with Newton's third law of motion.
   - **align velocity to direction of motion** - check to align velocity with the direction of the particles' motion.
   - **inherit color** - check to take the particle color from the originating particle. Otherwise the color is determined by the **color** control.



Spawned particles.

Adjusting the Speed and Direction of Particles

Applying Gravity to Particles

When applying gravity to particles, as opposed to our familiar gravity, Nuke doesn't restrict you to a certain direction but works in any or all of the **x**, **y** and **z** directions. You can add gravity either by:

- Using the ParticleDirectionalForce to apply a directional force. Just connect it to your particle stream, and adjust the strength of the force in the x, y, and z directions by entering **x**, **y,** and **z** values in the **strength** fields.

OR

- Using the ParticleGravity node. When you connect the ParticleGravity node to your particle stream, an arrow appears in the Viewer, which you can then use to determine the direction and velocity of the gravity. The bigger and longer the arrow, the stronger the effect. Instead of adjusting the arrow, you can also use the controls in the properties panel:
  - **from** - enter the point of origin for the gravity effect on the **x**, **y**, and **z** axis. This determines from which direction the force appears to come, indicated by the base of the arrow in the Viewer.
  - **to** - enter direction for the gravity effect on the **x**, **y**, and **z** axis. This is indicated by the point of the arrow in the Viewer.

Aligning Particles

To align your particles' motion, direction, and orientation, you can use two nodes:

- You can add the ParticleMotionAlign node in your particle stream to realign all the particles along their direction of motion. This is useful if your particles seem too rigid in their movement.
- Add the ParticleLookAt node to determine a 3D point that all the particles are looking toward. To specify this point, adjust the **position** control. The **x**, **y** and **z** coordinates specify the point that the particles are looking at.

Controlling Particle Speeds

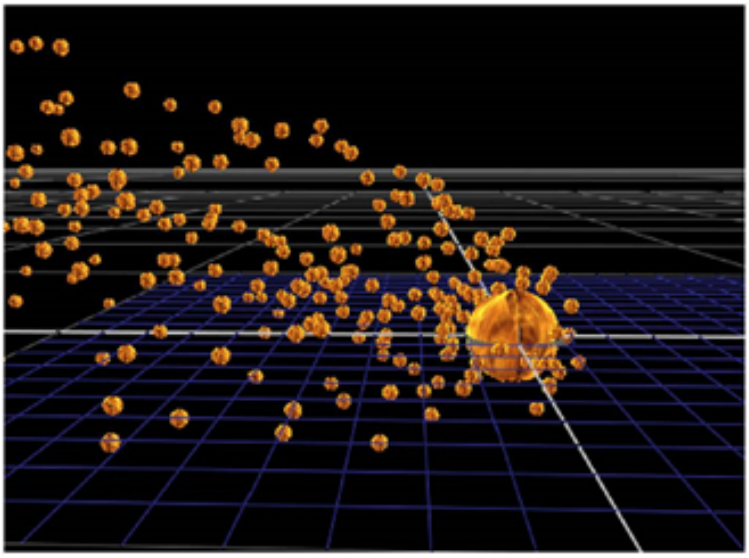
The ParticleSpeedLimit node limits the particles to a specified minimum and maximum speed.

1. Connect it to your particle stream.
2. In the properties panel, adjust:
   - **minimum** - the minimum speed at which each particle can travel.
   - **maximum** - the maximum speed at which each particle can travel.

Attracting Particles to a Specific Point

With the ParticlePointForce, you can attract particles to or repel them from a certain point in the 3D space.

1. Connect the node to your particle stream.
2. Adjust the ParticlePointForce controls:
   - **strength** - set the strength of the force attracting or repelling particles. Negative values cause attraction, positive values repulsion.
   - **falloff** - choose how quickly strength of the attraction falls off with respect to the distance by selecting the type of falloff, **none** for no falloff, **inverse** for inverse falloff, or **inverse square** for inverse falloff squared.
   - **radius** - set the radius of influence. Outside of this radius, no particles are affected by point force.
   - **position** - set the position of the point that attracts or repels particles. You can use an animated or a still Axis node expression-linked to these fields, or you can just enter a position value manually.



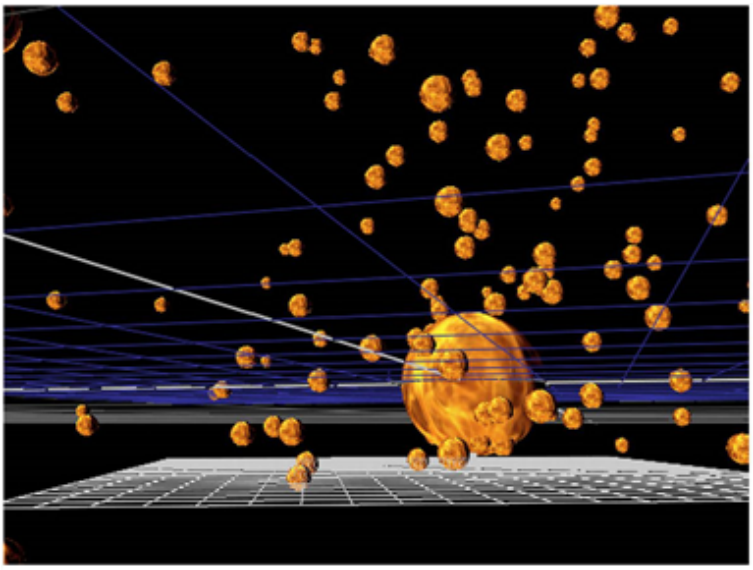
Particles attracted to a 3D point.

Modifying the Particles' Movement

Bouncing Particles off Objects

With the ParticleBounce node, you can make your particles bounce off the shape of a 3D object instead of traveling through it. Connect this node to your other particle nodes, and adjust the ParticleBounce controls:

1. To set your particles to bounce off the outside of the bounce object (specified by the **object** control), select a mode in the **external bounce mode** dropdown: **none** to apply no external bounce effect, **bounce** to bounce the particles, **kill** to end the life of the particles as they bounce.
2. Select a channel where a particle should be assigned to when a bounce is detected in the **new channels** dropdown. Setting this to **none** doesn't perform any channel assignment.
3. In the **bounce** field, specify the strength of the external bounce effect, and adjust **friction** to control the amount of friction for external bounce effect.
4. To set your particles to bounce off the inside of the bounce object (specified by the **object** control), use the **internal bounce mode** dropdown:
   - **none** - to apply no internal bounce effect.
   - **bounce** - to bounce the particles.
   - **kill** - to end the life of the particles as they bounce.
5. Select a channel where a particle should be assigned to when a bounce is detected in the **new channels** dropdown. Setting this to **none** doesn't perform any channel assignment.
6. In the **bounce** field, specify the strength of the internal bounce effect, and adjust **friction** to control the amount of friction for the internal bounce effect.
7. Select the object you want to use for the bounce effect in the **object** dropdown:
   - **plane**, **sphere** or **cylinder** - add a Nuke standard primitive to use as the bounce surface for the particle system. You can use these shapes to quickly test your particle effects.
   - **input** - add custom geometry attached to the **geometry** input.
8. Use the transform controls to fine-tune your geometry's position to achieve the desired bounce result.



Particles bouncing off a plane.

Adding Drag to Particles

With the ParticleDrag node you can apply drag on your particles. This gradually slows them down over time.

1. Connect the ParticleDrag node to your particle stream.
2. Adjust the ParticleDrag controls:
   - **drag** - increase to add a drag effect to your particles, slowing and stopping their movement as they are making distance from the particle center.
   - **rotational drag** - increase to add a drag effect to your particles' rotation, slowing and stopping their rotation.

Tip: You can also enter negative values to apply a reverse drag effect and speed the particles up.

Adding Turbulence Motion on Particles

The ParticleTurbulence node applies Perlin noise to the particle movement in the **x**, **y** and/or **z** directions.

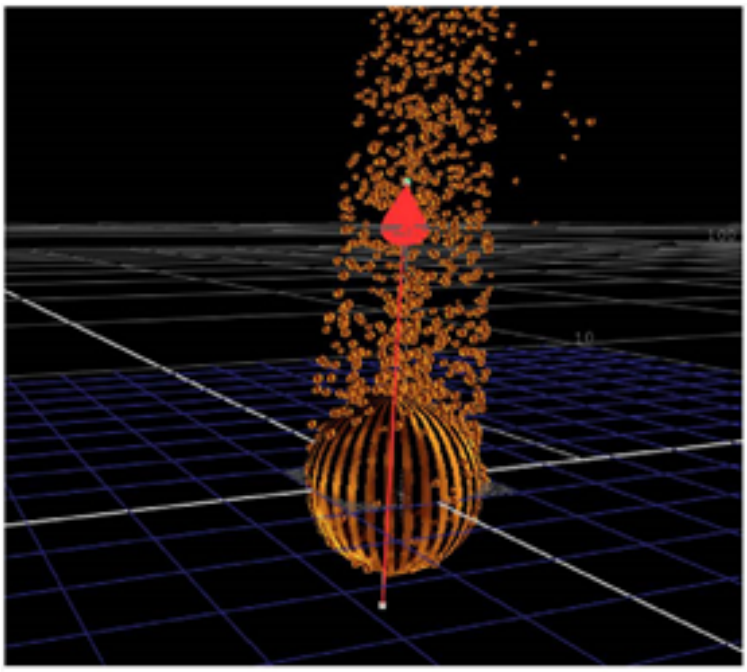
1. Connect a ParticleTurbulence node to your particle stream.
2. Adjust the ParticleTurbulence controls:
   - **strength** - set the strength for the turbulence effect on the **x**, **y** and **z** axes.
   - **scale** - set the scale of the effect, or the size of the area affected, on the **x**, **y** and **z** axes.
   - **offset** - set the offset applied to the effect, or the offset of the area affected, on the **x**, **y** and **z** axes.

Adding Spiral Motion to Particles

The ParticleVortex node applies a circular force to the particles and attracts them to an imaginary line, thus creating a whirlpool of particles.

1. Connect the ParticleVortex node to your particle stream.
2. An arrow appears in the Viewer, which you can drag to determine direction and velocity of the vortex effect. The bigger and longer the arrow, the stronger the effect. Alternatively, you can use the **from** and **to** controls in the properties panel.
3. In the ParticleVortex properties panel modify the parallel effect with the **parallel** control. This accelerates the particles in the direction of the imaginary vortex center line. If you set this to 0, no parallel force is applied, and positive and negative values determine the direction of the force. Adjust the **parallel falloff** to choose how quickly strength of the parallel force falls off with respect to the distance by selecting the type of falloff, **none** for no falloff, **inverse** for inverse falloff, or **inverse square** for inverse falloff squared.
4. With the **tangential** slider you can force the particles to circulate the vortex center line. Adjust **tangential falloff** to choose how quickly strength of the tangential force falls off with respect to the distance by selecting the type of falloff, **none** for no falloff, **inverse** for inverse falloff, or **inverse square** for inverse falloff squared.
5. Use the **radial** slider to adjust the force that attracts the particles to (positive values), or repels (negative values), them from the center line. Adjust **radial falloff** to choose how quickly strength of the radial force falls off with respect to the distance by selecting the type of falloff, **none** for no falloff, **inverse** for inverse falloff, or **inverse square** for inverse falloff squared.

Tip: If you want to create a helix of particles, you can turn up both the **parallel** and **tangential** values. This creates a particle vortex in a shape of a corkscrew.



ParticleVortex on a sphere.

Adding a Wind Effect to Particles

With the ParticleWind, you can simulate a wind blowing on your particles.

1. Connect the ParticleWind node to your particle stream. An arrow appears in the Viewer, which you can then use to determine direction and velocity of the wind. The bigger and longer the arrow, the stronger the wind effect. Alternatively, you can use the **from** and **to** controls in the properties panel.
2. In the ParticleWind controls, check **air resistance** to enable a drag effect on the wind.
3. Adjust the **drag** slider to increase or decrease the simulated air resistance.

Adjusting Controls Common to Several Particle Nodes

Many of the particle nodes share a number of controls, such as rendering and transform controls. The following covers the use of these.

Particle Rendering Controls

The first controls on the properties panels of all the particle nodes have to do with how the particles are output to the Viewer and rendered out. To read more about the display, selectable and render controls, see Object Display Properties.

Particle Transform Controls

Several of the particle nodes have a set of transformation controls in their properties panels. You can use these controls for example to translate, rotate and skew the force that the node applies on your particles. For more information on how transform controls work, see Transforming Geometry, Cameras, and Lights.

Condition and Region Controls

Various particle nodes have controls on the **Conditions** and **Region** tabs in the properties panel.

Conditions tab

On the **Conditions** tab, use the following controls to restrict the way in which the specific node affects your particles:

- **probability** - set the probability that this node affects your particles. If you set this to zero, the node won't affect any particles, and if the value is 1, the node affects every particle.
- **min age** - set this to limit the effect of this node only to particles above this minimum age. The age of the particle is its lifetime normalized between 0 and 1.
- **max age** - set this to limit the effect of this node only to particles below this maximum age. The age of the particle is its lifetime normalized between 0 and 1.
- **random seed** - enter an integer to change the results of generated randomness in your particles. You can achieve slightly different effects by changing this number.
- **channels** - specify which particle channels the effect of this node should be applied to. Channels **a** and **b** are arbitrary names for channels which are useful if you want different ParticleEmitter nodes or other particle force nodes to have an effect on separate channels.

Region tab

1. Using the **region** control you can select the region which you want to use to confine the particle effect to. For example, if you select a sphere, only particles inside that sphere-shaped region is affected by the

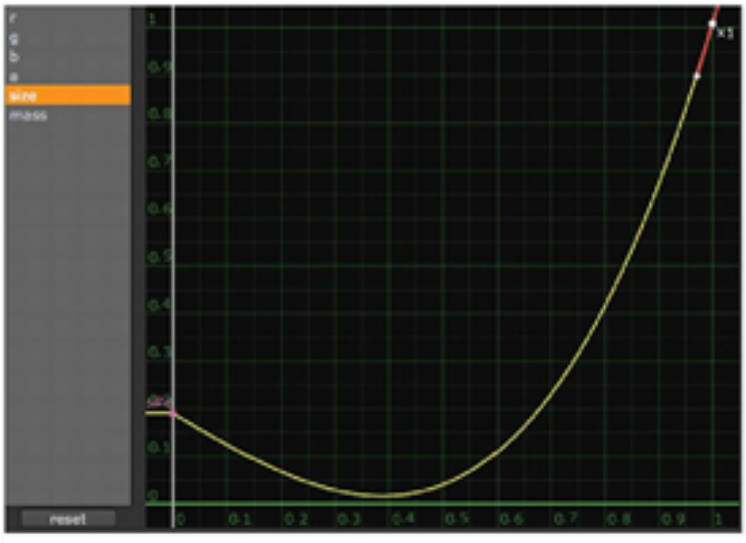
particle effect. Select **none** to apply no confining **region**, or the appropriate shape for a region between **sphere**, **box**, **half-space** and **cylinder.**

2. You can also check **invert region** to only affect the particles outside the region specified.
3. Then move on to adjusting your region with the transform controls. For more information, see Transforming Geometry, Cameras, and Lights.

Adjusting Particle Properties Using Curves

With the ParticleCurve, you can apply a curve to particle properties (such as size or mass) to change them over time.

1. Connect the node to your particle node stream.
2. Adjust the curves in the ParticleCurve properties panel. The x axis represents the lifetime of the particles.
   - **r** - adjust the curve for the red channel.
   - **g** - adjust the curve for the green channel.
   - **b** - adjust the curve for the blue channel.
   - **a** - adjust the curve for the alpha channel.
   - **size** - adjust the curve for the size of the particles.
   - **mass** - adjust the curve for the mass of the particles.



A size curve modifying the particles to grow in size toward the end of their lifetime.

Note: To use the ParticleCurve with RGBA values, you need to apply a BlendMat node to the texture, and set the surface blend knob to modulate. See BlendMat for more information.
This is because the ParticleCurve node edits the particle vertex color, which only works when there is no particle input. By adding a texture or shape as the particle input, this overrides the particle vertex color and uses the particle input instead. The **modulate** value in the BlendMat node multiples the particle's texture by the vertex color, allowing them to work together.

3. If you want, you can adjust the curve for your particles' alpha channel so that each particle fades to invisibility toward the end of its lifetime.

Adjusting Particles Using Expressions

With the ParticleExpression node, you can adjust your particles by setting expressions on their attributes. Using expressions gives you a vast variety of ways of adjusting how your particles behave. You can use a similar expression syntax as you would elsewhere in Nuke, with the exception that some functions that work in normal Nuke expressions aren't available in particle expressions and vice versa.

The main difference between Nuke's Expression node and ParticleExpression is that particle expressions can return a 3D vector instead of just a single floating point number. If a particle expression returns a single number N in a field that expects a vector (such as velocity or acceleration) it is converted into a vector with N for each of its components. For more information about the functions you can use with ParticleExpression, see Particle Expression Functions.

1. Connect the ParticleExpression node to your particle stream.
2. In the ParticleExpression controls you can use four temporary expression fields. With these, you can set up any expression on a particle attribute and then give it a temporary name. This temporary name can then be used in the following fields to refer to the corresponding temporary expression. This can be useful if you need to use a long expression in several fields. By default, the **per-particle** box is ticked to make the expressions affect each particle individually. Uncheck the box to apply the expression to all particles at once. For more information on expressions, see Expressions.
3. You can also set expressions on a set of attribute fields:
   - **color** - set an expression to edit the color of the particles.
   - **opacity** - set an expression to edit the opacity of the particles.
   - **size** - set an expression to edit the size of the particles.
   - **mass** - set an expression to edit the mass of the particles.

- **accel** - set an expression to edit the acceleration of the particles.
- **force** - set an expression to edit the force of the particles.
- **pos** - set an expression to edit the position of the particles. For example, enter **sin (age * 10) * 5** to emit particles randomly on a single static line.
- **vel** - set an expression to edit the velocity of the particles.
- **onlyonnew** - check this next to each attribute field to make the expression only affect new particles and ignore any existing ones.

Particle Expression Functions

Here are some functions you can use with the ParticleExpression node:

Function	Purpose	Related Functions
abs(f)	Returns the absolute value of f.	See also: fabs.
acos(f)	Returns the arc cosine of f. The result is the angle in radians whose cosine is f.	See also: cos, cosh, asin, atan2.
age	The age of the particle, in frames.	-
asin(f)	Returns the arc sine of an angle. The result is the angle in radians whose sine is f.	-
atan(f)	Returns the arc tangent of an angle. The result is the angle in radians whose tangent is f. Can be called with one or two arguments; if called with two arguments it's equivalent to atan2.	-
atan2(x,y)	Returns the principal value of the arc tangent of y/x, using the signs of the two arguments to determine the quadrant of the result.	See also: sin, cos, tan, asin, acos, atan, hypot.
ceil(f)	The ceiling of f, rounding any fractional part up.	-

Function	Purpose	Related Functions
color	The color of the particle. This is a 3D vector value, where x() is the red component, y() is green and z() is blue.	-
cos(f)	Returns the cosine of angle f. The angle is in radians.	See also: sin, tan, asin, acos, atan, hypot.
cosh(f)	Returns the hyperbolic cosine of f.	-
exp(x)	Returns the value of e (the base of natural logarithms) raised to the power of x.	-
fabs(f)	A synonym for abs(f).	-
floor(f)	The floor of a number, rounding any fractional part down.	-
fmod(x, y)	Floating point modulus function. fmod(x, y) returns the remainder after dividing x by y.	-
hypot(x,y)	The Euclidean distance function. hypot(x, y) returns the length of the hypotenuse of a right-angled triangle where the other sides have length x and y respectively.	-
id	The index number for each particle.	-
int(f)	Convert floating point number f to an integer, discarding any fractional part.	See also: trunc(f).
life	The maximum lifetime of a particle, in frames.	-
log(x)	Returns the natural logarithm of x.	-
log10(x)	Returns the base 10 logarithm of x.	-

Function	Purpose	Related Functions
mag(v)	Returns the magnitude (length) of the 3D vector v.	-
mass	The mass of the particle. Used when applying a force to a particle.	-
new	Returns 1 if the particle has just been created, 0 otherwise.	-
norm(v)	Normalise the 3D vector v to have a length of 1.0 while pointing in the same direction.	-
opacity	A number between 0.0 and 1.0, where 0.0 is fully transparent and 1.0 is fully opaque.	-
pos	The position of the particle. This is a 3D vector.	-
pow(x, y)	Returns x raised to the power of y.	-
pow2(f)	Returns the square of f, i.e. f raised to the power of 2.	-
random	Returns a random number.	-
randomv	Returns a vector with each of the components between 0 and 1 (as per a volumetric cube).	See also: uniformsamplesphere.
rint(f)	Round the floating point number f to an integer.	-
sin(f)	Returns the sine of the angle f. The angle is in radians.	-
sinh(f)	Returns the hyperbolic sine of f.	-
size	The size of the particle.	-
sqrt(f)	Returns the square root of f. f must	-

Function	Purpose	Related Functions
	be greater than or equal to zero.	
tan(f)	Returns the tangent of angle f. The angle is in radians.	-
tanh(f)	Returns the hyperbolic tangent of f.	-
trunc(f)	A synonym for int(f).	-
uniformsamplesphere	Similar to randomv, but generates a true random unit vector.	See also: randomv
v(x, y,z)	Create a vector from three separate numbers.	-
vel	The velocity of the particle. This is a 3D vector.	-
x(v)	Get the x component of the 3D vector v.	-
y(v)	Get the y component of the 3D vector v.	-
z(v)	Get the z component of the 3D vector v.	-

Adjusting Particle Simulation Settings

If you want to adjust how many steps of particle simulation take place per animation frame, you can use the **steps per frame** control in the ParticleSettings properties panel. Sometimes simulations cannot generate enough accuracy only calculating once per frame, and the resulting particle movement can appear jagged. **Steps per frame** forces the simulation to assess the movement of the particles multiple times per frame to enable a more analogue movement. You should enter the lowest value you can, as having an unnecessarily high value can slow down your particle calculations.

Merging Particle Streams

If you have more than one set of particle nodes, and you want to combine them into one stream, ParticleMerge is your node. You can merge as many particle streams as you need into a single ParticleMerge.

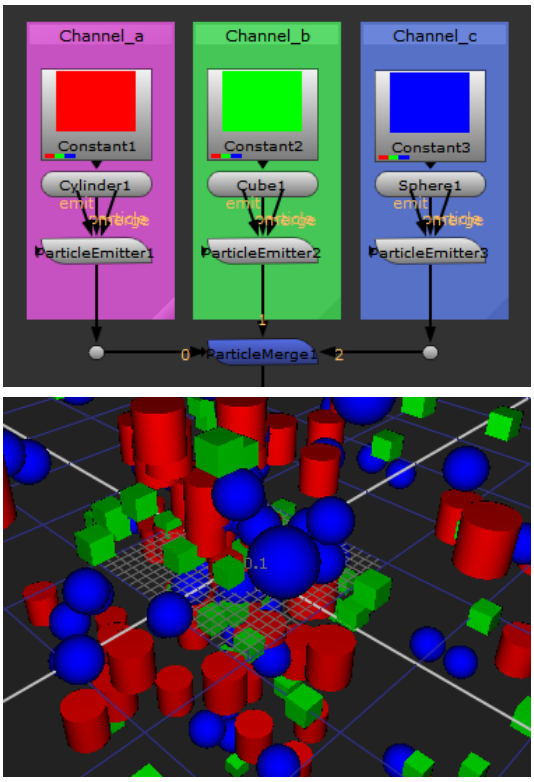
1. Connect a ParticleMerge node to other particle nodes at any point in the particle stream.
2. Attach another particle stream (or a ParticleEmitter node) to another one of ParticleMerge's numbered inputs and you're all set.

Controlling Particles by Channel

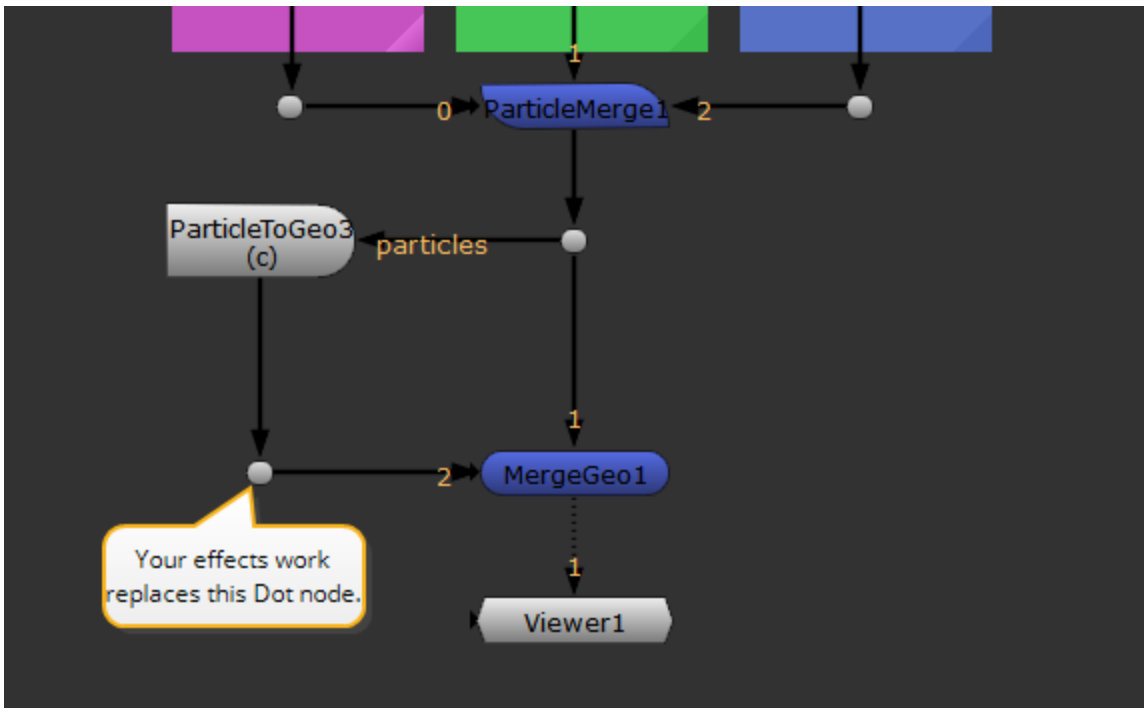
The ParticleToGeo node allows you to control particles in a simulation by channel, giving you the ability to isolate certain particles at any point in the Node Graph. For example, you might want to freeze certain particles within a simulation while allowing others to emit as normal or apply a particle effect to the particles in a single channel.

ParticleToGeo allows you to control geometry or sprite particles by channel, but in the case of sprite particles, you can also influence their alignment using the **align mode** control.

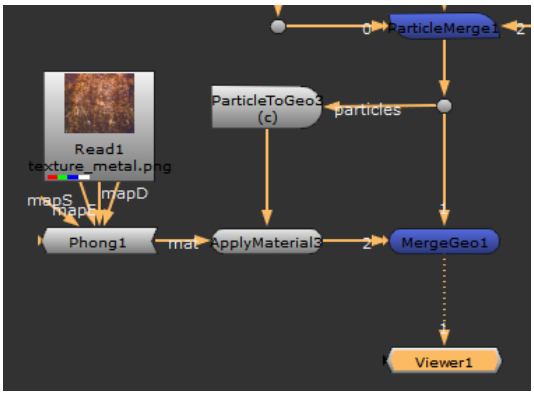
The following example uses a very simple particle system containing three colored particles in separate channels and adds effects to one channel using ParticlesToGeo. The other particles are unaffected.

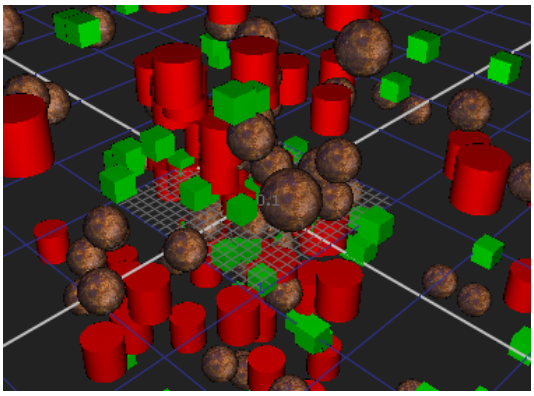


Split off the channel you're interested in using ParticleToGeo and MergeGeo nodes. Set the ParticleToGeo node's **channels** control to the channel you want to affect. In this example, channel **c** with the Sphere geometry is the affected particle.



To add a new texture to the geometry, you can use the AddMaterial node to apply a material to the affected channel.





Caching Particles

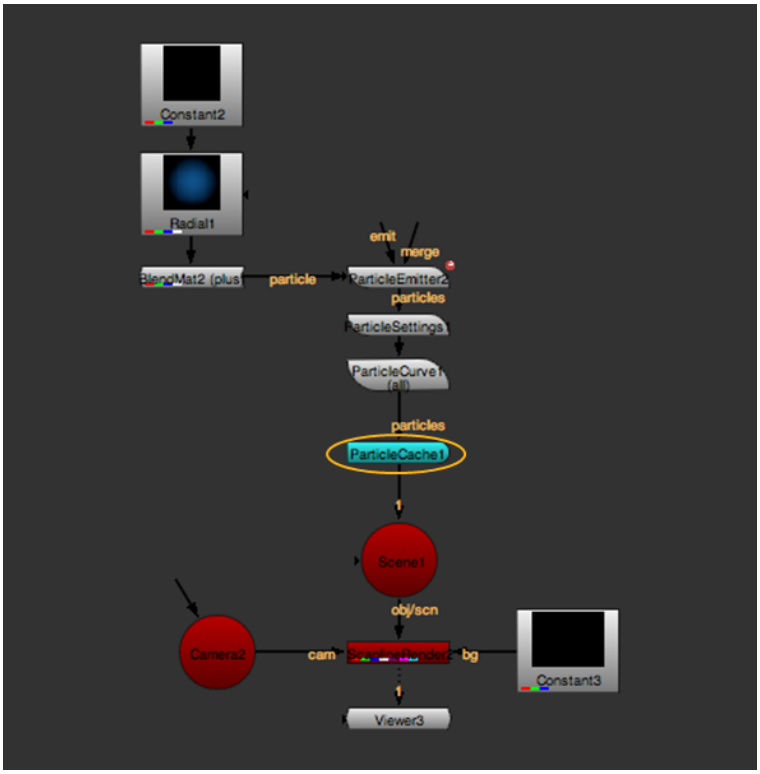
The ParticleCache node allows you to store the geometry simulation for a particle system to file. It can then be read back in different sessions of Nuke or on different machines without the need for recalculation.

This allows a particle system to be produced by an artist and then used by a render farm without recalculation, speeding up render times.

Note: The ParticleCache node doesn't replace the particle system. It just stores the simulation to disk and still relies on the particle system being connected in the same way with the same inputs. If anything in the particle system changes, the ParticleCache node detects this and shows an error to alert you to changes that have potentially been made upstream without your knowledge.

To cache your particle simulation:

1. Once you are happy with your particle simulation, select **Particles** > **ParticleCache** to create a ParticleCache node.
2. Place the ParticleCache node at the bottom of a single particle system or multiple merged particle systems.



Note: You cannot place a ParticleCache node in the middle of a string of particle nodes or beneath a Scene node connected to two separate particle streams.

3. In the ParticleCache properties, click the folder icon  next to the **file** field and navigate to the directory where you want to cache the particle simulation. After the directory path, enter a name for the cache files and include the frame number variable (for example, **####**) in the name. Click **Open**.

   ParticleCache uses the **.nkpc** file extension.

Note: ParticleCache may need to render up to 100 sub-frames. To account for this, it adds decimals to the file name's frame number. For example, if the file name in the **file** field is **particle_cache.####.nkpc**, ParticleCache may generate files called **particle_cache.0001.01.nkpc**, **particle_cache.0001.02.nkpc**, and so on.

4. If you have nodes downstream that generate motion blur, the particle system may need to request frames outside the normal frame range. If this is the case, increase the **padding** value in the ParticleCache properties to set the number of extra frames added to the start and end of the ParticleCache render.
5. Click **Render**.

   ParticleCache renders your particle simulation out to file frame by frame.
6. To use the cache data, enable **read from file**.

   If you get a "Particle cache data not found" error, return to step 4 and increase the **padding** value.

PrmanRender

PrmanRender is a render node that works together with Pixar's PhotoRealistic RenderMan® Pro Server software to give you an even better quality render result. PrmanRender is an alternative to Nuke's ScanlineRender with additional features for rendering 3D scenes.

Setting Up RenderMan Pro Server and PrmanRender

In order to use the PrmanRender node, you need to have Pixar's PhotoRealistic RenderMan Pro Server 20, or earlier, installed and licensed on your machine (for brevity, we call it RenderMan from now on). To do this:

1. Make sure X11 server and client libraries are installed on your system. X11 for macOS are available from the XQuartz project at www.xquartz.org. Download the latest version available.
2. Follow the instructions in the RenderMan installation guide to get RenderMan working on your machine. This is in most cases enough to get you going with using both RenderMan and the PrmanRender node in Nuke. Note that RenderMan specifically needs two environment variables set in order to work with Nuke:
   - RMANTREE - This needs to point to the location of your RenderMan distribution.
   - Depending on which operating system you're using either DYLD_LIBRARY_PATH (on Mac), LD_LIBRARY_PATH (on Linux) or PATH (on Windows). This environment variable needs to point to "%RMANTREE%/lib". For more information on setting environment variables, see Environment Variables.
3. To make sure your PrmanRender node is working in Nuke, start the application from the terminal (for more information, see the Installation and Licensing chapter).
4. Create a PrmanRender node in Nuke (**3D** > **RenderMan** > **PrmanRender**) and connect it to your nodes. You can try the following combination of nodes as a test: **Checkerboard** > **Cube** > **PrmanRender** > **Viewer**.

Using The PrmanRender Node

The PrmanRender node can render nearly everything that you previously used ScanlineRender for, but with PrmanRender, you have control over aspects like shadows and reflections in your render result.

Connect the PrmanRender node to your Scene node, Camera node and any optional inputs, in the same way you would connect ScanlineRender. For more information about connecting a ScanlineRender node, see Setting Up a Scene.

On the **PrmanRender** tab, you can select which aspects you'd like to render out by checking **shadows**, **reflections**, **refractions** or **dof** (depth of field). For more information, see Shadows, Reflections, Refractions and Depth of Field.

You can select your projection mode in the **projection mode** dropdown:

- **Perspective** - objects in front of the camera have the illusion of depth defined by the camera's focal-length and aperture.
- **Orthographic** - objects are viewed using a parallel projection.

If necessary, also adjust:

- **overscan** - to set how many pixels are rendered over the right/left and top/bottom of the frame, if a subsequent operation requests this.
- **ambient** - to add global ambient lighting.

Render Quality

On the **Sampling** tab, you can adjust controls that affect your render quality. Adjust:

- **ray trace max depth** - to set the maximum depth of the view rays PrmanRender uses to trace your scene and make render calculations.
- **pixel samples** - to set the number of samples to render per pixel. Having more samples increases your render quality, but also increases render time.
- **filter** - to select a texture sampling filtering algorithm. For more information on the different options, see Choosing a Filtering Algorithm.
- **antialiasing filter** - to select an antialiasing filter: **box**, **triangle**, **catmull-rom**, **sinc**, **gaussian**, **mitchell**, **separable-catmull-rom** or **blackman-harris**.
- **antialiasing filter size** - to select the size of the antialiasing filter.
- **shading rate** - to set the shading calculation for primitives. This value, along with pixel samples, directly affects your rendering time and the final quality of your results. A small shading rate value means your render takes more time, but the quality is very high. A large value on the other hand means your render is faster, but the final quality is not as good.

Shadows, Reflections, Refractions and Depth of Field

On the **PrmanRender** tab, you can select if you want to render shadows, reflections, refractions and depth of field, or all of them into your result. All these effects are calculated using a retracing method that is based on drawing rays from the camera to the object. Check the box for:

- **shadows** - to add shadows to your render. You can adjust the parameters for your shadows on your Light node's **Shadows** tab. For example, if you adjust the **sample width**, the shadows are softer. For more information, see Casting Shadows.
- **reflections** - to add reflections to your render. You can adjust the parameters for your reflections on your Reflection node's properties panel. See Using the Reflection Node.
- **refractions** - to add refractions to your render. You can adjust the parameters for your refractions on your Refraction node's properties panel. See Using the Refraction Node.
- **dof** - to add depth of field to your render.

Motion Blur Parameters

On the **Sampling** tab, you can adjust controls that affect motion blur. Adjust:

- **motion blur samples** - to set the number of samples to render per pixel when motion blurring.
- **shutter** - Enter the number of frames the shutter stays open when motion blurring. For example, a value of 0.5 would correspond to half a frame. Increasing the value produces more blur, and decreasing the value less.
- **shutter offset** - Select when the shutter opens and closes in relation to the current frame value when motion blurring:
    - **centered** - to center the shutter around the current frame. For example, if you set the shutter value to 1 and your current frame is 30, the shutter stays open from frame 29.5 to 30.5.
    - **start** - to open the shutter at the current frame. For example, if you set the shutter value to 1 and your current frame is 30, the shutter stays open from frame 30 to 31.
    - **end** - to close the shutter at the current frame. For example, if you set the shutter value to 1 and your current frame is 30, the shutter stays open from frame 29 to 30.
    - **custom** - to open the shutter at the time you specify. In the field next to the dropdown menu, enter a value (in frames) you want to add to the current frame. To open the shutter before the current frame,

enter a negative value. For example, a value of -0.5 would open the shutter half a frame before the current frame.

- **randomize time** - adjust this to add randomness to the distribution of samples in time so they don't produce regularly spaced images. The larger the value, the larger the time difference between the samples.
- **shutter opening** - to set the shutter opening behavior. Select:
  - **none** - to use the default shutter opening method, resulting in instantaneous timing to open and close.
  - **linear** - to set the shutter to open and close in linear intervals.
  - **bezier** - to set the shutter to open and close more gradually, according to a Bezier curve.

Shader Parameters

On the **Shader** tab you can select which channels are affected by motion vectors and output vectors.

- Click **motion vectors** and select the type of vectors you'd like to render:
  - **off** - no motion vector information is rendered.
  - **velocity** - store the velocity of every single pixel in the **motion vector channels**.
  - **distance** - for every pixel, store the distance (in pixels) between samples in the **motion vector channels**.
- Adjust **motion vector channels** to select which channels you want the motion vectors to be output to.
- Check **output vectors** if you want to render output vectors.
- Select which channels to apply surface points and surface normals from the **surface point** and **surface normal** dropdowns.

RIB Parameters

With the RIB (RenderMan Interface Bytestream) parameters, you can choose to filter the information Nuke generates for RenderMan, set your arguments to it, and output your own RIB file. On the **RIB** tab:

- filter - check this to filter the information on your scene, generated by Nuke for RenderMan. In order to do this, Nuke calls a Python function called nukescripts.renderman.filterRIB. Filtering can make the render startup slightly slower as a temporary RIB file is created for each render.
- arguments - specify your arguments for filtering. This string is passed by Nuke's Python filter function as extra arguments to RenderMan. If you want to use your own filter, you can also replace Nuke's Python function, and have your arguments passed directly to your own Python function. For example, you could set the filter arguments to "-rif myfilter.so" to load your own RenderMan Interface filter.



Tip: For further details on filtering, have a look at your RenderMan documentation.

A RIB file is a RenderMan-compatible ASCII file with information that Nuke generates when rendering your footage. To output a RIB file:

1. In the **file** field under **output**, specify the file name and location for your RIB file.
2. Click **Execute**.

Using the ModifyRIB Node

You can use the ModifyRIB node to insert RIB (RenderMan Interface Bytestream) statements into your script to modify a RIB stream before it's passed to the PrmanRender node. This can be useful in situations where you might want to adjust the shading on the surface of an object, replace it, or perform a variety of transformations and adjustments to multiple objects in a scene. For example, to replace an object in your script with basic geometry from RenderMan:

1. Add a Scene node to the basic script that you created at the beginning of the chapter and connect two Light nodes and a Camera.
2. Click **3D** > **Modify** > **RenderMan** > **ModifyRIB** and insert the node between the Cube and the Scene.
3. Next to the **archive** field, uncheck **use**. This activates the **statements** field.



Note: You can load a RIB archive by leaving **use** checked and clicking the **Select file** button to locate it.

4. In the ModifyRIB control panel select **replace** from the **operation** dropdown menu and enter the following RIB statement in the **statements** field:

   Sphere 0.25 -0.25 0.25 360

   You should now see a basic sphere in the Viewer where the cube was.
5. To change the color of the sphere and add a basic surface shader to the sphere, enter the following statements underneath:

   Color 0.4 0.6 0.1

   Surface "wood"

   This changes the color of the sphere to green and applies a wood-like finish to the surface.

Note: If you're copying statements directly from a RIB file, only copy statements from between **WorldBegin** and **WorldEnd** as copying the entire contents of the file may result in instability issues.

For more information on RIB statements, please refer to the documentation provided with RenderMan.

Using the Reflection Node

Reflection is the familiar physical phenomenon where an image of an object is cast back from a particular kind of surface, such as glass or water. Using PrmanRender node, you can replicate this effect in your render result of 3D objects, and using the Reflection node, you can adjust the controls for creating the reflection effect. PrmanRender uses raytracing to create this effect and you can use the Reflection node to adjust the result.

Note: Nuke's RayRender node can also use the Reflection node to control reflections.

1. Create a Reflection node by clicking **3D** > **Shader** > **RenderMan** > **Reflection.**
2. Connect the node to the PrmanRender node and set the controls to adjust your reflection:
   - **reflection color** - sets the color of the reflection.
   - **value** - sets the intensity of the reflection.

Using the Refraction Node

Refraction is the familiar physical phenomenon of light traveling differently through different materials and thus reflecting differently off objects behind that material. For example, if you have a glass of water with a straw in it, the part of the straw that's not in water appears to be in a different angle to the part which is in the water. This is due to water bending the light waves. PrmanRender uses raytracing to create this effect and you can use the Refraction node to adjust the result. Without the PrmanRender node and RenderMan software though, the Refraction node has no effect.

1. Create a **Refraction** node by clicking **3D** > **Shader** > **RenderMan** > **Refraction.**
2. Connect the node to the PrmanRender node and set the controls to adjust your refraction:
   - **refraction index** - slide to change the type of refraction.

- **value** - sets the intensity of the refraction.

FurnaceCore Nodes

FurnaceCore nodes contain the most popular Furnace plug-ins incorporated in NukeX and Nuke Studio.



Note: The following FurnaceCore nodes have been replaced by other nodes and can no longer be found in the **FurnaceCore** menu:
F_DeGrain and F_Denoise were replaced by Denoise (**Filter** > **Denoise**) in Nuke 6.3.
F_Kronos was replaced by Kronos (**Time** > **Kronos**) in Nuke 7.0.
F_MotionBlur was replaced by MotionBlur (**Filter** > **MotionBlur**) in Nuke 7.0.
F_VectorGenerator was replaced by VectorGenerator (**Time** > **VectorGenerator**) in Nuke 7.0.

Global Motion Estimation

FurnaceCore has three effects based on global motion estimation (GME).

Introduction

FurnaceCore effect based on global motion estimation calculate a four-corner pin, which finds the best fit of one image onto another, and then apply that pin. These effects differ in how that corner pin is calculated and to which image that pin is applied. This chapter describes the general idea behind these plug-ins; for more detailed information on each plug-in, please read their individual chapters.

These effects are:

- **F_Align** - which lines up two shots of the same scene, by finding a corner pin from each source clip frame to the corresponding reference clip frame. For example, you can use this to align two separate but similar steady cam shots of the same scene.
- **F_RigRemoval** - which removes unwanted objects (rigs) from image sequences without accurate rotoscoping or keying to produce a clean plate.
- **F_Steadiness** - which removes motion from a single clip, by calculating a pin that either locks all frames in the clip to a single reference frame, or smooths that motion out over a window of frames. For example, you can use this to remove camera shake.

What is Global Motion Estimation?

Global motion estimation (GME) is a technique that attempts to map one image onto another with a simple four-corner pin. This differs from local motion estimation (LME), which attempts to find where each individual pixel in the image is in the other image. GME is much cheaper to compute than LME, but gives you less information about the image. Nevertheless, it is still very powerful for a variety of applications.

Using the plug-ins' parameters, you can tell the GME engine what type of motion to expect. This can be a combination of any of:

1. **translation** - which allows the four corners to translate by the same amount,
2. **rotation** - which allows the corners to rotate about their center,

3. **scale** - which allows the size of the area defined by the corners to change,
4. **perspective** - which allows the angles at the corners to change, so that the area defined by them is no longer a rectangle.

The more types of motion you allow, the more expensive the motion estimation becomes. For many scenes, rotation and translation are sufficient.

The GME effects have an accuracy control, which controls the amount of work Foundry's GME engine does to calculate the global motion. Typically, the higher this is, the better the estimation, but the more expensive it is.

Limitations of GME

As stated above, global motion estimation simply calculates a four-corner pin to transform one image onto another. This means that GME can't be used to match two images where there is heavy parallax, very complicated foreground motion, changing objects, and so on.

The best way to think of what GME can do is that if you can do it with a four-corner pin, it can; if you can't, it can't. However, GME will take the pain out of hand matching pins frame by frame.

Global Motion Estimation Effects

F_Align and F_Steadiness work in a similar way, which is distinct from the way F_RigRemoval works. These two effects calculate a four-corner pin for each frame and save it into the corner pin parameters. These pins are then used during the render to move the source image.

Using Them

These effects analyze images over a range of frames to figure out their four-corner pins. This is done in response to the user pressing the **Analyse** button in the effects control panel. During analysis, the effect will run through a range of frames adding keys to the corner pin parameters. These corner pins are then applied to the source clip to render a frame.

F_Steadiness has a separate analysis pass that happens interactively; it then uses the previously computed and key-framed corner pins during render. This speeds up its operation, as the analysis step only has to be done once, and once it has been done this plug-in is very quick to render. However, F_Align, which only ever needs the two current frames from each clip, can compute the corner pin on the fly (but not keyframe it!). This leads to a slightly different mode of operation for the following effects:

- F_Steadiness
  - needs to have an analysis run before it renders useful output,
  - will always use the value in the corner pin parameters when rendering the output image.
- F_Align
  - no need to have the analysis run to render output, but doing so will give you a key-framed corner pin,
  - during render, it will use the value of the corner pin parameters only if there is a keyframe there, otherwise, it will analyse on the fly during render. This means that analysis will speed up later renders, as just rendering the corner pin is much cheaper than calculating it.

Some parameters to the effect control how the effect performs GME analysis, and some only affect the rendering. If you ever modify one of these parameters, then any analysis you may have performed will be out of date. To let you know, an overlay warning will be posted whenever this happens. You don't have to re-analyze and your renders will still look at the keyed corner pins.

If you have not modified a parameter that affects analysis (the warning overlay will let you know), pressing **Analyse** will only re-analyse a frame if there is no key on the corner pin at that time. This avoids redundant re-analysis if you have interrupted the analysis or extended the analysis range. However, if you want to force re-analysis, press **Clear Analysis** and all keys will be deleted.

F_Steadiness and F_Align have an analysis region rectangle parameter which is used to specify which area of the reference image should be analysed during GME. So, for example, with F_Steadiness set to **Lock Mode**, this is the area inside the lock frame that a match will be sought for. The documentation for each plug-in describes exactly how to use the analysis region.

Controls

The controls common to all GME plug-ins are described below. They are grouped into two sections:

- Ones that determine how analysis is carried out. See Parameters That Affect Analysis.
- Ones that control the rendering of the output. See Parameters That Affect Rendering.

Parameters That Affect Analysis

The following parameters affect the analysis of the four-corner pin.

Analyse - This is a push button which triggers an analysis of the input clips and calculate a corner pin. Interrupting the analysis does not delete the corner pin keys that have already been calculated.

Render During Analysis - If set, this toggle causes the effect to update the time line and render a freshly analyzed frame so you can see the progress of the effect. Doing so slows down the analysis somewhat, so toggle this off to speed up the general analysis.

Clear Analysis - Pressing this push button deletes all keyframes from the corner pin parameters, allowing you to force a re-analysis if you feel the need to.

Analysis Range - This controls the range of frames any analysis will be run over. It can be one of:

- **Specified Range** - which looks at the parameters **Analysis Start** and **Analysis Stop** for the range of frames to analyze,
- **Source Clip Range** - which automatically determines the range of frames to analyze from the length of the input clip.

Analysis Start - The first frame to analyze from if **AnalysisRange** is set to **Specified Range**.

Analysis Stop - The last frame to analyze from if **Analysis Range** is set to **Specified Range**.

Scale - A toggle that indicates whether the calculated corner pin can include a scaling factor.

Rotate - A toggle that indicates whether the calculated corner pin can include rotations.

Translate - A toggle that indicates whether the calculated corner pin can include translations in x and y.

Perspective - A toggle that indicates whether the calculated corner pin can include perspective transforms.

Analysis Region - This is the region analyzed to calculate the four-corner pin. This is especially useful when doing any form of frame locking, in which case, go to the lock frame, look at the reference clip and position the box over the area you want locked.

- **Analysis Region BL** - controls the position of the bottom-left corner of the analysis region.
- **Analysis Region TR** - controls the position of the top-right corner of the analysis region.

Accuracy - This controls the time/accuracy trade off in the GME engine. The higher this is, the slower it goes, but you have a better likelihood of a good result.

Parameters That Affect Rendering

These following parameters control how a GME effect renders the four-corner pin. Some of them are set during the analysis pass.

Filtering - This controls the quality of the rendering.

- **Low** - uses nearest neighbor filtering. This gives low quality but is quick to render.
- **Medium** - uses a bilinear filter. This gives good results and is quicker to render than high filtering.

- **High** - uses a sinc filter to interpolate pixels giving a sharper repair. This gives the best results but takes longer to process.

Invert - If set, then the inverse of the calculated four-corner pin is used during render.

Four Corner Pin - The corner pins calculated during the analysis pass:

- **Bottom Left** - the lower left corner pin.
- **Bottom Right** - the lower right corner pin.
- **Top Left** - the upper left corner pin.
- **Top Right** - the upper right corner pin.

Widgets

All the Analyzing GME effects have two on-screen widgets: one to provide feedback and one to set up the analysis region.

Analysis Region Widget - This is a rectangle widget which you use to set the analysis region over the reference image.

Four Corner Widget - This is a widget that shows the state of the four-corner pin that has been calculated. You can change it by grabbing any of the corners and tweaking the shape of the pin. To give you more feedback as to what frames have been analyzed, it will be drawn solid if there is a key in the corner pin at the frame being displayed; otherwise, it will be drawn dashed.

Using F_Align

This chapter looks at how to use F_Align to register (line up) two shots that are of the same scene, but have slightly different camera motion and foreground objects. This can be useful, for example, for doubling up the crowd size by lining up and comping together two shots of the same scene, or locking your freshly generated clean plate to the original.

Introduction

F_Align takes two sequences that were shot of the same scene and lines them up spatially. It uses Global Motion Estimation (GME) to calculate a four-corner pin so that each frame in one shot (the source input) will be aligned with the corresponding frame in a second shot (the reference input). The result is the source image which has been repositioned to line up with the reference image.



Source image. Notice the position of the background.



Reference Image.



Output from F_Align. The source image has been repositioned so that the background lines up with the reference image.



The output of F_align comped together with the reference image.

To be able to align the sequences, F_Align analyzes them for global motion. This analysis can be triggered for the complete sequence, specified frame range, or a single frame when you press the **Analyse** button in the F_Align controls. Alternatively, it can be done on the fly for a single frame when you move to a new frame on the timeline. The advantage of pressing **Analyse** is that during the analysis, F_Align stores the calculated four-corner pin as key-framed parameters. When you then render the output of the plug-in later, F_Align can use these key frames without having to calculate them again.

If you analyze on the fly, you won't have direct access to the calculated corner pins. Any re-rendering will also be significantly slower, as the 'on the fly' calculations will have been lost and F_Align will have to analyze again.

If at any stage you modify the effect in such a way to invalidate the key-framed analysis (for example by changing the **Accuracy** parameter), a warning will be posted and the effect will analyze on-the-fly during render, ignoring the keyed analysis.

The on-screen widget and the **Analysis Region**parameters are used to control which section of the **Reference**frame is being matched to each **Source** frame. Typically, leaving the region at its default is good enough. However, a heavy mismatch in foreground detail may make it necessary to change the region to a section that is shared between shots.

The transformation in F_Align is concatenated with other NukeX transform nodes. This means that if you add a row of F_Align and NukeX transform nodes to a tree, their functions are combined. Because the image is only re-sampled once, there is no loss of image quality and processing time is decreased. However, as in NukeX, certain nodes, including color correction nodes, can break the concatenation.

For more of an overview of Global Motion Effects and a description of the common way of working many of these effects have, please see Global Motion Estimation.

Quick Start

This section gives a very brief outline of how to use the plug-in. It covers both analyzing using the **Analyse** button and analyzing on the fly.

Analyzing with the Analyse button

To align two shots and store the results as keyframed parameters, do the following:

1. Find two shots that are of the same scene, but have slightly different camera motion and foreground objects. Select **Image** > **Read** to load both these shots.

2. Select **FurnaceCore** > **F_Align**. Connect the shot you want to reposition to the source (**Src**) input of F_Align and the shot you want to match to the reference (**Ref**) input. View the output from F_Align.

   The source will be immediately repositioned so that it aligns to the reference shot without any need for analysis.

   You will see the following banner in the overlay:

   No valid key framed analysis found. Analyzing during render.
3. Depending on the exact difference between the two shots, you may need to enable the **Scale** and/or the **Perspective** toggles to get a decent alignment.
4. You may also need to reposition the **Analysis Region** depending on the differences in foreground detail. However, leaving it at the default position works well for most shots.
5. Click on the **Analyse** button.

   F_Align will now start analyzing each frame in the shot, figuring out the four-corner pin and writing it as key frames to the corner pin parameters, **Bottom Left**, **Bottom Right**, **Top Left**, and **Top Right**. You will find these parameters under **Advanced** > **Four Corner Pin**.

   F_Align will update the timeline at each frame, and you will see the aligned image render in the output. If you don't want this to happen, uncheck **Render During Analysis**.

   If you interrupt the analysis, the pins it has keyed until that point will be retained.
6. Play or scrub through the aligned frames. The rendering will be faster as F_Align will no longer need to analyze on the fly. However, if you scrub to a frame where a corner pin has not been keyed, F_Align will re-analyze that frame on the fly.
7. To see how closely the two clips have aligned, press **M** on the Node Graph to insert a Merge node. Connect the Merge node's **A** input to F_Align and the **B** input to the reference image. View the output from the Merge node. Then, adjust the **mix** slider in the Merge controls to dissolve between F_Align's output and the reference clip you wanted to match.

Analyzing On The Fly

To align two shots and calculate the alignment on the fly, do the following:

1. Find two shots that are of the same scene, but have slightly different camera motion and foreground objects. Select **Image** > **Read** to load both these shots.
2. Select **FurnaceCore** > **F_Align**.
3. Connect one of the shots to the source (**Src**) input of F_Align and the other to the reference (**Ref**) input. View the output from F_Align.

   The **Src** will be immediately repositioned so that it aligns to the **Reference** shot without any need for analysis.

   You will see the following banner in the overlay:

   No valid key framed analysis found. Analyzing during render.

4. Depending on the exact difference between the two shots, you may need to enable the **Scale** and/or the **Perspective** toggles to get a decent alignment.
5. You may also need to reposition the **Analysis Region** depending on the differences in foreground detail. However, leaving it at the default position works well for most shots.
6. To see how closely the two clips have aligned, press **M** on the Node Graph to insert a Merge node. Connect the Merge node's **A** input to F_Align and the **B** input to the reference image. View the output from the Merge node. Then, adjust the **mix** slider in the Merge controls to dissolve between F_Align's output and the reference clip you wanted to match.

Parameters

The parameters for this plug-in are described below.

Analyse - This is a push button which will trigger an analysis of the input clips and calculate a corner pin. Interrupting the analysis will not delete the corner pin keys that have already been calculated.

Render During Analysis - If set, this toggle will cause the effect to update the time line and render a freshly analyzed frame in the Viewer so you can see the progress of the effect. Doing so will slow down the analysis somewhat, so toggle this off to speed up the general analysis.

Clear Analysis - Pressing this push button will delete all keyframes from the corner pin parameters, allowing you to force a re-analysis if you feel the need to.

Analysis Range - This controls the range of frames any analysis will be run over. It can be one of:

- **Specified Range** - which will look at the parameters **Analysis Start** and **Analysis Stop** for the range of frames to analyze,
- **Source Clip Range** - which will automatically determine the range of frames to analyze from the length of the input clip.
- **Current Frame** - the analysis occurs only on the current frame. This is useful for correcting any errors that may have occurred while analyzing the entire clip.

Analysis Start - The first frame to analyze from if **Analysis Range** is set to **Specified Range**.

Analysis Stop - The last frame to analyze from if **Analysis Range** is set to **Specified Range**.

Scale - A toggle that indicates whether the calculated corner pin can include a scaling factor.

Rotate - A toggle that indicates whether the calculated corner pin can include rotations.

Translate - A toggle that indicates whether the calculated corner pin can include translations in x and y.

Perspective - A toggle that indicates whether the calculated corner pin can include perspective transforms.

Analysis Region - This is the region analyzed to calculate the four-corner pin. This is especially useful when doing any form of frame locking, in which case, go to the lock frame, look at the reference clip and position the box over the area you want locked.

Analysis Region BL - controls the position of the bottom left corner of the analysis region.

Analysis Region TR - controls the position of the top right corner of the analysis region.

Advanced - The lesser used refinement controls.

Accuracy - This controls the time/accuracy trade off. The higher this is, the slower the analysis, but you have a better likelihood of a good result.

Filtering - Sets the filtering quality.

- **Low** - low quality but quick to render.
- **Medium** - uses a bilinear filter. This gives good results and is quicker to render than high filtering.
- **High** - uses a sinc filter to interpolate pixels giving a sharper repair. This gives the best results but takes longer to process.

Invert - if set, then the inverse of the calculated four-corner pin is used during render.

Four Corner Pin - The corner pins calculated during the analysis pass:

Bottom Left - the lower left corner pin.

Bottom Right - the lower right corner pin.

Top Left - the upper left corner pin.

Top Right - the upper right corner pin.

Using F_DeFlicker2

When working in film, you sometimes have to deal with shots that have a luminance flicker. This chapter concentrates on removing flicker using F_DeFlicker2.

Introduction

F_Deflicker2 is used to remove flicker. It is particularly suitable for removing flicker that is localized and dependent on the geometry of the scene (that is, flicker that is not present across the whole of the image), such as that caused by an unsynchronized fluorescent light in a shot. It works by calculating the gain between the current frame and each frame in a small analysis range surrounding it. It then tries to adjust the gain so that it varies smoothly over this frame range. This means it is better at reducing fast flicker than flicker which varies slowly over the image sequence, as the latter will already appear smooth over the small frame range and F_DeFlicker2 will leave it largely untouched.

The algorithm used by F_DeFlicker2 can introduce blurring in areas where there is rapid motion. This problem could be alleviated by using local motion estimation before deflickering the frames. However, this process is complicated by the fact that the presence of flicker can adversely affect the results of the motion estimation. F_DeFlicker2 therefore adopts a two-stage approach to this problem. First, the normal deflickering process is performed. Then, the motion vectors for the sequence are calculated on the resulting deflickered frames, and applied to the original frames in order to align them. The deflicker calculation is then performed on the aligned frames to give the final result. To use this approach, turn on **Use Motion** in F_DeFlicker2.



Note: Because F_DeFlicker2 looks at input frames outside the current frame when performing its calculation, it can be a computationally expensive plug-in. As such, using more than two instances of F_DeFlicker2 in a node tree will dramatically increase render times. It is strongly advised therefore, that you render each instance out separately.

Quick Start

To remove flicker from a sequence:

1. Select **Image** > **Read** to load the sequence you want to remove flicker from.
2. Select **FurnaceCore** > **DeFlicker2** to apply DeFlicker2. View its output.
3. If you're not happy with the results, adjust the DeFlicker2 parameters. The available parameters are described below.

Parameters

The parameters for F_DeFlicker2 are described below.

- **DeFlicker Amount** - Use this to reduce flicker without removing it entirely; smaller values mean more will be left behind.
- **Block Size** - To find where a certain pixel is located in the analysis range, the deflicker algorithm looks for a block of pixels centered around that pixel. Block size defines the width and height of these blocks (in pixels). On rare occasions, a large block size can produce data that's lacking in detail. This is because a small feature can fit into a large block, causing the motion estimation to concentrate on the background motion and ignore the small feature. A small value, instead, can produce a noisy motion field, as there aren't enough constraints in a small block to fit the motion accurately. In most cases, however, the default value is small enough so that details aren't lost, and the smoothing step of the algorithm ensures the motion field isn't too noisy. Therefore, this value very rarely needs editing.
- **Use Motion** - Turn this on to do a second deflicker pass using motion-compensated frames. This can improve results in areas where there is fast motion, where the initial deflicker pass can introduce blurring.
- **Vector Detail** - Determines the density of the motion vectors used when **Use Motion** is turned on. The maximum value of 1 will generate one vector per pixel. This will produce the most accurate vectors but will take longer to render. A value of 0.5 will generate a vector at every other pixel.
- **Analysis Range** - The number of frames searched each side of the current frame when calculating the flicker. Higher values may give better results, but can also bring in erroneous information and take longer to process.

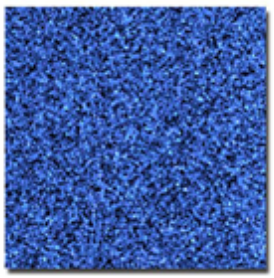
Using F_ReGrain

This chapter looks at adding grain to sequences using F_ReGrain.

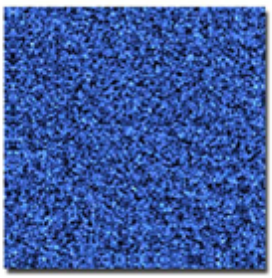
Introduction

F_ReGrain is used to add grain to a sequence. It has been designed to sample an area of grain from one image and then to generate unlimited amounts of this grain with exactly the same statistics as the original. This new grain can then be applied to another image.

The figure on the left shows an enlarged and exaggerated sample of grain from Kodak 320 film stock. F_ReGrain was used to sample the original Kodak 320 stock and synthesize a plate of grain. The result is shown in the figure on the right. Note that the grain characteristics closely match the original.

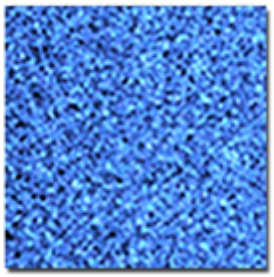


Kodak 320.

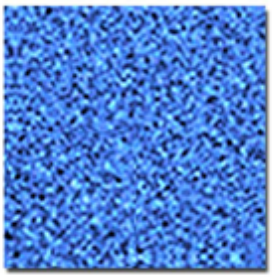


F_ReGrain.

Similarly, below the figure on the left is a sample from Kodak 500 film stock and the figure on the right shows this replicated using F_ReGrain.



Kodak 500.



F_ReGrain.

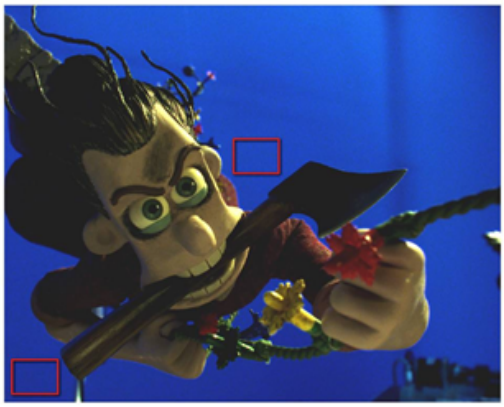
Quick Start

You can sample grain from an image and apply it to another or select from a variety of pre-sampled, standard grain types.

Adding Sampled Grain

To add grain to a sequence, do the following:

1. Select **Image** > **Read** to load the sequence you want to add grain to. Then, load the image you want to sample grain from.
2. Make sure you are working at full resolution and not proxy resolution. F_ReGrain will not work at proxy resolution. (See Proxy Resolutions.)
3. Select **FurnaceCore** > **F_ReGrain**.
4. Connect the sequence that you want to have grain to F_ReGrain's source (**Src**) input. Then, connect the sequence you want to sample grain from to the **Grain** input. View the output from F_ReGrain.
5. Set **Grain Type** to **From Grain Clip**.
6. Position the on-screen sample region over an area of the **Grain** sequence just containing grain and no picture detail. See the figure below.



This shows two possible selection regions that contain no edge detail and little luminance variation.

It is important to get your selection right. You should avoid any image detail or even a plain area that has luminance variations underneath the grain. The better this initial selection, the better the result will be. If you can't find a decent sample area on the current frame, then try other frames from the same film stock. The default size of the sample area should be enough to gather information about the grain

characteristics of your image. However, you may need to change its size and shape to fit over a plain area free of image detail.



Warning: There is a minimum size of this sample area below which the statistical analysis of the grain will be unreliable. If the sample area you select is too small, you will see a warning message which prompts you to select a larger region. (See Proxy Resolutions.)

7. View the output of F_ReGrain to judge the results. The output will now contain the **Src** image with grain from the **Grain** image applied. Both the size and the luminance of the new grain can be manually tweaked using **Grain Size** and **Grain Amount** respectively. It helps to view the **Grain** input while editing the parameters of F_ReGrain.

The grain is sampled on a single frame which is set when you adjust the sample area (or by manual adjustment of the **Analysis Frame** parameter). Although it is sampled on only one frame, the algorithmically created grain will change from frame to frame but mirror the characteristics of the sample grain.

Using Pre-Sampled, Standard Grain Types

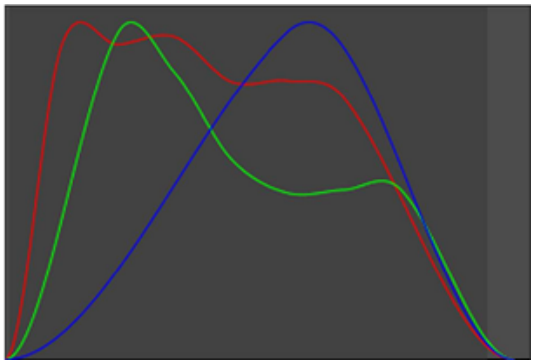
If you don't have an image to sample grain from, you can also select from a variety of pre-sampled, standard grain types. Do the following:

1. Select **Image** > **Read** to load the sequence you want to add grain to.
2. Make sure you are working at full resolution and not proxy resolution. F_ReGrain will not work at proxy resolution. (See Proxy Resolutions.)
3. Select **FurnaceCore** > **F_ReGrain**.
4. Connect the sequence that you want to have grain to F_ReGrain's source (**Src**) input.
5. Set **Grain Type** to **Preset Stock**.
6. Try the different grain types using the **Preset Stock** dropdown menu. 2K, 4K, aperture corrected, and non aperture corrected stocks are included. Individual color channels can be selected and adjusted using the **Advanced** parameters.

Response

In its default setting, F_ReGrain adds the same amount of grain over the whole image. However, the amount of grain on an image is normally a function of luminance. Various parameters in the **Grain Response** group allow you to adjust how the amount of grain added varies with luminance:

- Pressing **Sample Grain Response** will cause the variation of the amount of grain with luminance to be calculated from the **Grain** input, and switching on **Use Sampled Response** will apply these curves to the grain added to the **Src** sequence.
- To view the sampled response curves, switch on **Draw Response**; an example is shown in the figure below.
- The amount of grain added to the lowlights, midtones and highlights of the image can be adjusted using the **Low Gain**, **Mid Gain** and **High Gain** parameters. The effect of adjusting these can also be seen on the response curves.

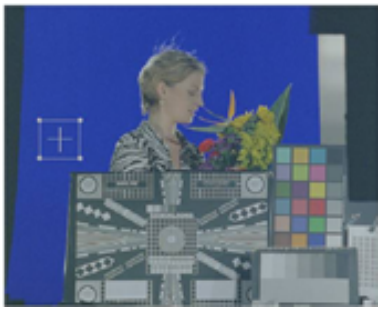


This shows an example of the grain response with luminance. The x axis represents luminance and the y axis the amount of grain.

Checking the Result

To test that the new grain is the same as the old grain, set **Output** to **Grain Plate**.

This generates a sheet of grain with the same luminance level as the mean of the sample region. The sample region with the original grain is also displayed. It should be impossible to differentiate between the two regions. The figure on the left shows a good selection area giving a good test plate of grain in the figure on the right.



Good selection area ...



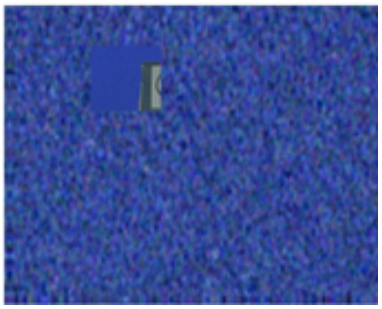
... producing a good test plate of grain, free of

artifacts.

Below, the figure on the left shows a poor selection area since it contains image detail. The figure on the right shows the resulting test plate which clearly highlights the problem.



Bad selection area ...



... producing a poor result.

Proxy Resolutions

Grain manipulation at proxy resolution should be avoided as the results are unreliable. The grain selection area may be too small at proxy resolution to give a good result, and making this area larger may drag in unwanted detail from the image. If you try to use F_ReGrain at proxy resolution, we simply pass the image through untouched and issue the following warning:

Cannot work at proxy scale.

We decided that this was preferable behaviour to doing poor grain replication at proxy resolution. You can, of course, crop the input clip and work with that rather than the proxy. There is a minimum size for the selection box, which is about 37x37 at base resolution. If the box you select is smaller, you will get this warning along the top of the viewer:

Sample box is too small - please select a larger sample of the grain.

Parameters

The parameters for this plug-in are described below:

- **Grain Type** - Selects whether the grain is sampled from the **Grain** image (**From Grain Clip**) or from a set of standard stocks.

    - **Preset Stock** - grain characteristics are sampled from a supplied film stock. 2K, 4K, aperture corrected and non aperture corrected stocks are supplied. Although standard stocks are included, it is recommended where possible that you sample from the film stock you are trying to match.
    - **From Grain Clip** - samples and reconstructs the grain characteristics from the **Grain** input.
- **Preset Stock** - The film stock the grain characteristics are sampled from when **Grain Type** has been set to **Preset Stock**. Common Fuji and Kodak stocks are supplied. The exposure can be under, over, or, if left blank, non aperture corrected. The size is either 2K or 4K pixels. For example, **FUJIF500 2K** refers to the grain characteristics sampled from a 2K plate of Fuji Film 500 film stock non aperture corrected.
- **Grain Amount** - Adjusts the brightness of the grain. Setting this to 0 means no grain is added.
- **Grain Size** - Adjusts the size of the grain granules. The larger the value, the bigger and softer the granules.
- **Output** - Sets whether to render the result or a test image.
    - **Result** - shows the **Source** image with the grain applied.
    - **Grain Plate** - shows a test image with the grain applied. This test image is composed from a section of the input image surrounded by a uniform solid color sampled from the image with the grain applied. If the inner area is indistinguishable from the outer area, then you have a good grain sample.

Analyse - This is a push button which will trigger an analysis of the input clip. Press this button if the input clip from which the grain was analyzed has changed but you do not want to move the analysis region to trigger re-analysis. Whenever the input clip changes, you will see the following warning in the Viewer:

The clip from which the grain was analyzed has changed. Press Analyse or move the analysis region to re-analyze grain.

- **Analysis Region** - A selection box that marks the region of image used to analyze the grain when **Grain Type** is set to **From Grain Clip**. This part of the frame must contain no image detail, only grain.
- **Analysis Region BL** - controls the position of the bottom left corner of the analysis region.
- **Analysis Region TR** - controls the position of the top right corner of the analysis region.
- **Analysis Frame** - sets the frame to sample the grain from.
- **Grain Colour Space** - This tells F_ReGrain what color space the grain sample clip was in when the grain originated. Setting this correctly ensures that the grain is not exaggerated by any color space conversions prior to sampling.
    - **Cineon**
    - **sRGB**
    - **Linear**
- **Advanced** - The parameters under Advanced allow detailed adjustment of the grain.
    - **Process Red** - Switch this on to process the red channel.
    - **Red Amount** - sets the brightness of the grain in the red channel.
    - **Red Size** - adjusts the size of the grain granules in the red channel.
    - **Process Green** - Switch this on to process the green channel.

- **Green Amount** - sets the brightness of the grain in the green channel.
- **Green Size** - adjusts the size of the grain granules in the green channel.
- **Process Blue** - Switch this on to process the blue channel.
- **Blue Amount** - sets the brightness of the grain in the blue channel.
- **Blue Size** - adjusts the size of the grain granules in the blue channel.

- **Grain Response** - The parameters under **Grain Response** allow the amount of grain added to be varied as a function of the image luminance.
- **Apply Grain In** - This controls what color space the grain sample is re-applied to the image.

  Generally, this should be set to **Grain Colour Space** to ensure the most accurate recreation.

  You may want to override this though for some looks or special cases.
  - **Cineon** / **sRGB** / **Linear** - The grain sample will be applied in the specified space.
  - **Grain Colour Space** - The Grain sample will be applied in the color space set in the **Grain Colour Space** dropdown menu, in the Grain Sample section.
- **Low Gain** - adjusts the gain of the grain in the lowlights.
- **Mid Gain** - adjusts the gain of the grain in the midtones.
- **High Gain** - adjusts the gain of the grain in the highlights.
- **Use Sampled Response** - switch this on to scale the brightness of the grain as a function of the luminance of the Grain image.
- **Sampled Response Mix** - this control is usually set to 1. Decreasing it reduces the effect of the response curves until, at 0, they have no effect on the output. This parameter is only available if **Use Sampled Response** is on.
- **Sample Grain Response** - press this to update the response curves from the current frame. Multiple presses accumulate the grain response rather than resetting every time. This parameter is only available if **Use Sampled Response** is on.
- **Reset Grain Response** - press this to reset the grain curves to their default (flat) response. This parameter is only available if **Use Sampled Response** is on.
- **Draw Response** - overlays the response curves on the bottom left corner of the viewer. This parameter is only available if **Use Sampled Response** is on.

Color Space in FurnaceCore Plug-ins

Some of the algorithms in the FurnaceCore tool set are sensitive to the color space of the source footage. If the footage is not in the expected color space, you may get poor results from some of the plug-ins.

Like Nuke, FurnaceCore expects all footage to be in **Linear** space. Nuke converts all footage to Linear upon import, so unless you have changed the colorspace in your **Read** nodes (or in the node tree by using a Colorspace node before a FurnaceCore plug-in), your footage should be Linear by the time it reaches the plug-in anyway.

If you know that the input to a plug-ins isn't Linear, you should use a Colorspace node before and after the plug-in to convert to and from Linear for processing. This should ensure optimal results.

Color Space in F_ReGrain

Because color space transformations can distort the look of grain, and by default, Nuke converts your footage to linear, if you are sampling grain from your own plate, you need to make sure you tell the plug-in what space the plate was in originally, so the sample isn't distorted.

This is accomplished by setting the **Grain Colour Space** dropdown menu in the F_ReGrain controls to the right space. For example, if you were sampling grain from a film scan, you would want to set this to **Cineon**. If you had footage from a digital video camera, this would most likely be **sRGB**. F_ReGrain automatically sets the right color space when using one of the pre-sampled grain clips, which are in **sRGB**.

F_ReGrain also works best when applying grain in the same color space that the sampled grain originally existed. There is a dropdown menu in the **Response** section of the F_ReGrain controls which allows you to match or override this. It defaults to applying the grain in the same space as the sample.

Using F_RigRemoval

This chapter looks at the removal of unwanted objects (rigs) from image sequences without accurate rotoscoping or keying to produce a clean plate.

Introduction

In this context, we define a rig as a foreground element in a sequence that moves over a background element. The plug-in will only work satisfactorily if it is possible to model the background motion by a global 3D transform. For example, if the background contains multiple objects moving in different directions, the results will be poor. Typically, good results will only be achieved in situations where a skilled artist could generate, and track in, a clean plate in order to repair the sequence. However, this plug-in should make the process quicker and easier. The rig removal algorithm works by estimating the background motion between successive frames, ignoring the foreground object, and then using the motion information to look forward and backward in the sequence in order to find the correct piece of background to fill in the missing region. The size of the missing region and the speed of the background dictate how far away from the current frame it is necessary to search to find the correct information.



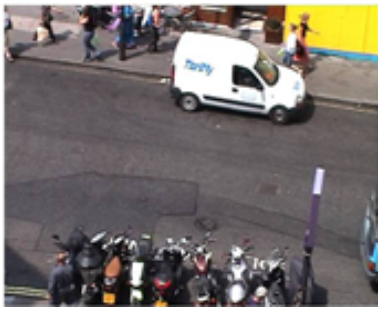
Before with bird.



After applying F_RigRemoval.



Before with taxi.



After applying F_RigRemoval.

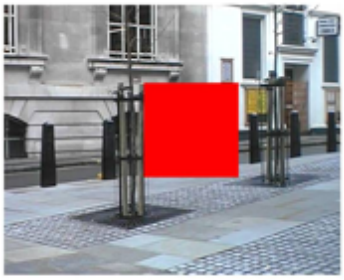
Quick Start

To remove an unwanted object from an image sequence, do the following:

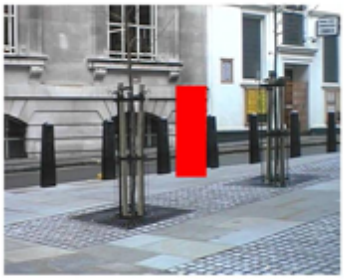
1. Select **Image** > **Read** to load the sequence with an unwanted object.
2. Select **FurnaceCore** > **F_RigRemoval** and connect your image sequence to F_RigRemoval's **Src** input. View the output of F_RigRemoval.
3. On each frame, define the area that will be repaired. You can do this in the following three ways:
   - If the image sequence has an embedded alpha channel, then you can use that alpha to define the area to be repaired. To do so, set **Rig Region** to **Src Alpha**.
   - Using F_RigRemoval's **RigMask** input, you can feed in a matte sequence to define the area to be repaired. To use this matte, set **Rig Region** to **Rig Mask Alpha** (or one of the other **Rig Mask** options).
   - You can use the on-screen rectangle to define the area to be repaired. To do so, set **Rig Region** to **Box**. Position the on-screen rectangle on top of the unwanted object. To key-frame the position to follow the object, select **Set key** from the animation menus next to **Rig Region BL** and **Rig Region TR** in the **Rig Region Box** parameter group. Move to a new frame and reposition the on-screen rectangle. A new key frame is set automatically. Repeat as necessary until the rectangle covers the object on every frame you want to remove the object from.

   Whichever method you choose, the region does not need to be the exact foreground region, but just a rough outline. However, you should avoid making it unnecessarily large as this will increase rendering time.
4. Having defined the region to repair throughout the clip, set **Frame Range** to the number of frames that the plug-in needs to analyze forwards and backwards to find enough data to repair the sequence. On the first frame, this will be quite time consuming as the algorithm needs to estimate the motion between each pair of frames. Subsequent frames will be much quicker.
5. If it has not been possible to replace all the foreground pixels, either because **Frame Range** was set too low or the background information does not exist anywhere within the sequence, the pixels will be displayed in red. Try to adjust **Frame Range** until no red pixels are visible and then render the sequence.

   Below, in the figure on the left, we are using a box to define the pixels to replace, and **Frame Range** is set to zero. Increasing this value, as shown in the figure on the right, gathers pixels from other frames and improves the result. To completely remove the red pixels, you'd need a **Frame Range** value of 5.



Frame Range = 0.



Frame Range = 3.

6. View the results.



Original image.



The output of F_RigRemoval.

Tip: F_RigRemoval is fairly slow to process, which can make the initial keyframing of the rectangular region frustrating.

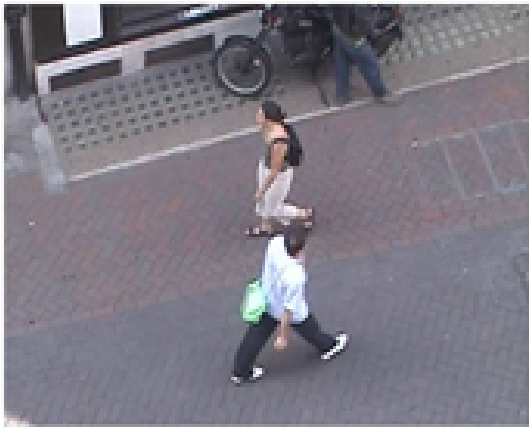
Sometimes, the easiest way to adjust the region is to load up F_RigRemoval, and to view the source so the effect is not processed, but the parameters are visible. Then, animate the rectangular region over the foreground object you're trying to remove throughout the sequence. When you're happy with the region position, click back onto F_RigRemoval's output and wait for it to update. Slowly increase the **Frame Range** parameter until the whole region is repaired, then check the output on other frames.

Note: F_RigRemoval uses frames from a wide range around the current frame. Feeding the output of one F_RigRemoval node into another will greatly increase the memory over head, as the second F_RigRemoval node will require the first F_RigRemoval node to calculate all its frames before passing the result on. Therefore, it is strongly recommended that you process out the first result and then use the processed result with the second F_RigRemoval node.

Occlusions

The algorithm used in F_RigRemoval is unable to differentiate between multiple foreground objects. If there is another foreground object in the sequence that moves through the background region that is being used in the repair, this second foreground object will also be cut up and used, resulting in an incorrect repair. To try and assist in these situations, it is possible to mark regions of the image as not to be used for repair by setting their alpha value to mid gray. This will ensure that spurious bits of other foreground objects do not appear in the repair.

In the figure below, we are trying to remove the woman in the center of the screen as she walks from left to right down the street. At this frame, a man walks in the opposite direction and her feet and his head overlap.

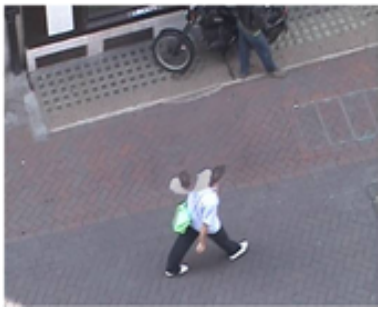


Original shot.

Below, the figure on the left shows the normal matte for the woman, and the figure on the right shows the result of using this in F_RigRemoval. Note that the man's head interferes with the repair and the reconstruction of the pavement is particularly bad, probably due to the man becoming the dominant motion source.



The normal matte.



The output.

To fix this, we can adapt the matte to include a mid gray area over the man. This tells the rig removal algorithm to ignore that area in the repair. This matte is shown below in the figure on the left, and the result is shown in the figure on the right. Note that the repair on the pavement is improved, and the man is simply clipped rather than being used in the repair.



The matte with a mid-gray area.



The output.

Parameters

The parameters for this plug-in are described below.

Rig Region - Defines the area to repair.

- **Box** - repair the area inside a rectangular box, controlled by the box parameters below or the on-screen box.
- **Src Alpha** - repair the region defined by the alpha of the source input.
- **Src Inverted Alpha** - repair the region defined by the inverted alpha of the source input.
- **RigMask Luminance** - repair the region defined by the luminance of the **Rig Mask** input.
- **RigMask Inverted Luminance** - repair the region defined by the inverted luminance of the **Rig Mask** input.
- **RigMask Alpha** - repair the region defined by the alpha of the **Rig Mask** input.
- **RigMask Inverted Alpha** - repair the region defined by the inverted alpha of the **Rig Mask** input.

Frames Searched - Sets whether to search forwards, backwards, or in both directions to find missing data.

- **Forward and Backward** - searches before and after the current frame.
- **Forward** - searches frames after the current frame.
- **Backward** - searches frames before the current frame.

Frame Range - Sets the number of frames the algorithm should look forwards and backwards in the sequence to find the missing data. If you are getting red pixels, then increase this value. See Quick Start .

Frames Used in Range - If **Frame Range** has to be set to a large number to make an effective repair, the rendering time can be prohibitive. **Frames Used in Range** can speed up the repair by not using every frame to fill the foreground region, effectively skipping frames. However, this may reduce the quality of the result.

- **All Frames** - use every frame in the searched frame range to construct the repair.
- **Half of Frames** - use every other frame in the searched frame range to construct the repair.
- **Quarter of Frames** - use every fourth frame in the searched frame range to construct the repair.
- **10% of Frames** - use every tenth frame in the searched frame range to construct the repair.
- **Max 25 Frames** - use no more than 25 frames from the searched frame range to construct the repair. This option can be useful if **Frame Range** has been set to a very large number.

Max Rig Movement - To avoid perspective changes, F_RigRemoval searches for the missing data inside an area immediately around the rig region. **Max Rig Movement** defines the width of this area (in pixels). Fast movement in the **Src** footage requires a higher value than slow movement. However, because the area used for the repair may be from a different part of the image, high values can introduce perspective problems.

Rig Region Box - The rectangular area used to define the repair when **Rig Region** is set to **Box**.

Rig Region BL - controls the position of the bottom left corner of the rig region.

Rig Region TR - controls the position of the top right corner of the rig region.

Advanced - The lesser used refinement controls.

Filtering - Sets the filtering quality.

- **Low** - low quality but quick to render.
- **Medium** - uses a bilinear filter. This gives good results and is quicker to render than high filtering.
- **High** - uses a sinc filter to interpolate pixels giving a sharper repair. This gives the best results but takes longer to process.

Luminance Correct - Switch this on to correct for luminance changes from information taken from other frames. This is particularly important if the lighting changes throughout the sequence.

Perspective Correct - Switch this on to correct for minor perspective changes.

Overlap Correct - The repair is built up using slices of information from other frames in the sequence. These slices can be overlapped and blended to give a more natural looking repair. This parameter controls how much the regions overlap. Increasing this parameter too much will degrade image sharpness.

Repair Fail Marker Opacity - Sets the level of transparency of the red pixels used to show where the repair has failed.

Preserve Alpha - Switch this on to preserve the original alpha channel. By default, this is switched off, and the alpha channel is set to white where the repair has failed and black everywhere else.

Using F_Steadiness

This chapter looks at how to stabilize a shot using F_Steadiness.

Introduction

F_Steadiness uses Global Motion Estimation (GME) to calculate a four- corner pin, so that camera motion within a single shot can be smoothed out over a range of frames or removed by locking to a specific frame.

F_Steadiness needs to analyze the input clip before it can render useful output. This analysis is done when you press **Analyse** in the F_Steadiness controls. During the analysis, F_Steadiness keyframes a four-corner pin which will stabilize the clip in subsequent renders. Without having performed an analysis pass, F_ Steadiness will not do anything useful on render.

F_Steadiness can work in two ways. These are:

1. **Smooth** - A range of frames around each frame is analyzed for motion and an average of that motion used to calculate the corner pin. Use this to keep the overall camera motion, but to smooth out sharp bumps and kicks.
2. **Lock** - A lock frame is specified and steadiness attempts to register each individual frame to that lock frame. Use this to completely remove camera motion from the sequence.

In lock mode, each frame in the clip must share a substantial amount of the scene with the lock frame, so you can't lock each frame in a 360 degree pan to the first one. However, in smooth mode, because F_Steadiness is only working on a small window of frames around the current frame, you can use it in shots which change completely over time.

The analysis region is used to control which section of the reference frame is being matched to each source frame. In lock mode, the reference is the lock frame, so you should position the analysis region when looking at the lock frame. In smooth mode, it looks at the incremental differences between frames, in which case you should place the analysis region in the area you want to appear with smooth motion.

The transformation in F_Steadiness is concatenated with other NukeX transform nodes. This means that if you add a row of F_Steadiness and NukeX transform nodes to a tree, their functions are combined. Because the image is only resampled once, there is no loss of image quality and processing time is decreased. However, as in NukeX, certain nodes, including color correction nodes, can break the concatenation.

For an overview of Global Motion Estimation, and a description of the common way Global Motion Effects work, please see Global Motion Estimation.

Quick Start

This section gives a very brief outline of how to use the plug-in.

Smoothing Out Camera Motion

To keep the overall camera motion but to smooth out sharp bumps and kicks, do the following:

1. Find a shot that has some camera shake in it and select **Image** > **Read** to load it.
2. Select **FurnaceCore** > **F_Steadiness** to apply F_Steadiness, and view its output.

   You will see the following banner in the overlay:

   No valid key framed analysis found, please press Analyse
3. In the **Advanced** F_Steadiness controls, make sure **Mode** has been set to **Smooth**.
4. Click on the **Analyse** button.

   F_Steadiness will now start analyzing each frame in the shot, figuring out the smoothing corner pin and writing it as key frames to the corner pin parameters, **Bottom Left**, **Bottom Right**, **Top Left**, and **Top Right**.

   After a brief pause (while F_Steadiness calculates the transforms in the frame range), F_Steadiness will update the timeline and you will see the steadied image render in the viewer.

   If at any point you interrupt the analysis, the pins it has calculated until that point will be retained.
5. Play or scrub through the stabilized frames.
6. If you want to make the result smoother, increase the **Smoothing** parameter in the **Advanced** parameter group and the corner pin will be recalculated immediately to give a smoother shot. You don't need to re-analyze the sequence if you do this; as F_Steadiness has kept the raw inter-frame transforms cached away, all it needs to do is re-write the keys for the average smoothing pin.

Locking To A Frame

To completely remove camera motion from a sequence, do the following:

1. Find a shot that has camera shake but where all frames share scene information and select **Image** > **Read** to load it.
2. Select **FurnaceCore** > **F_Steadiness** to apply F_Steadiness, and view its output.

You will see the following banner in the overlay:

No valid key framed analysis found, please press Analyse.

3. In the **Advanced** F_Steadiness controls, set **Mode** to **Incremental Lock**.
4. Choose a frame somewhere in the middle of the sequence that you want to lock to, and set the **Lock Frame** parameter to that frame.
5. Scrub back and forth through the shot and look for a region of the shot that is shared by all frames and doesn't change much (for example, avoid a region with people walking in front of it).
6. Whilst looking at the lock frame, position the on-screen widget for the **Analysis Region** over that region.
7. Hit **Analyse**.

   The effect will start to analyze, working first forward from the lock frame, then backwards from it, until all frames to be analyzed are done.

   The timeline will immediately update to give you feedback in the viewer.

Parameters

The parameters for this plug-in are described below.

Analyse - This is a push button which will trigger an analysis of the input clip and calculate a corner pin. Interrupting the analysis will not delete the corner pin keys that have already been calculated.

Render During Analysis - If set, this toggle will cause the effect to update the time line and render a freshly analyzed frame so you can see the progress of the effect. Doing so will slow down the analysis somewhat, so toggle this off to speed up the general analysis.

Clear Analysis - Pressing this push button will delete all key frames from the corner pin parameters, allowing you to force a re-analysis if you feel the need to.

Analysis Range - This controls the range of frames any analysis will be run over. It can be one of:

- **Specified Range** - which will look at the parameters **Analysis Start** and **Analysis Stop** for the range of frames to analyze,
- **Source Clip Range** - which will automatically determine the range of frames to analyze from the length of the input clip.

Analysis Start - The first frame to analyze from if **Analysis Range** is set to **Specified Range**.

Analysis Stop - The last frame to analyze from if **Analysis Range** is set to **Specified Range**.

Scale - A toggle that indicates whether the calculated corner pin can include a scaling factor.

Rotate - A toggle that indicates whether the calculated corner pin can include rotations.

Translate - A toggle that indicates whether the calculated corner pin can include translations in x and y.

Perspective - A toggle that indicates whether the calculated corner pin can include perspective transforms.

Analysis Region - This is the region analyzed to calculate the four-corner pin. This is especially useful when doing any form of frame locking, in which case, go to the lock frame, look at the reference clip and position the box over the area you want locked.

Analysis Region BL - controls the position of the bottom left corner of the analysis region.

Analysis Region TR - controls the position of the top right corner of the analysis region.

Advanced - the lesser used refinement controls.

Mode - This parameter controls whether F_Steadiness is smoothing the shot while keeping the overall camera motion, or locking the shot to a single frame to completely remove camera motion. It can be set to:

- **Incremental Lock** - in which case, it will calculate the pin that takes each frame to the lock frame. This calculates the pin by working from the lock frame out to each frame, calculating the GME between each frame incrementally and accumulating it to create the corner pin.
- **Absolute Lock** - this also calculates a pin that takes each frame to the lock frame. However, it does so by doing GME directly from the frame in question directly to the lock frame.
- **Smooth** - in which case, we are smoothing the shot for a range of frames described by the **Smoothing** parameter. This is the default value. It can be used to keep the overall camera motion, but to smooth out sharp bumps and kicks.

Incremental locks work better in some situations, whilst absolute locks work better in others. However, absolute locks, when they work, are often more accurate. If, for example, you want to get a lock between frame 0 and frame 100, try using absolute lock first. If absolute lock doesn't work, it can be because frame 0 and frame 100 are too different. In this case, incremental lock can help if there is a gradual spatial progression from frame 0 to frame 100.

Smoothing - This controls the range of frames to average motion over when **Mode** is set to **Smooth**. The default value is 10.

Lock Frame - This controls the frame which will be locked to when **Mode** is set to either of the lock modes. By default, this is set to 0. Note that what is frame 0 for F_Steadiness is frame 1 in NukeX. Therefore, if you are looking at frame 3 in the Viewer and want to use that frame as the lock frame, you need to enter 2 as the **Lock Frame** value.

Accuracy - This controls the time/accuracy trade off. The higher this is, the slower the analysis, but you have a better likelihood of a good result.

Filtering - Sets the filtering quality.

- **Low** - low quality but quick to render.
- **Medium** - uses a bilinear filter. This gives good results and is quicker to render than high filtering.
- **High** - uses a sinc filter to interpolate pixels giving a sharper repair. This gives the best results but takes longer to process.

Invert - If set, then the inverse of the calculated four-corner pin is used during render. This works best with the lock modes, and can be used to track static locked-off plates back into a shot.

Auto Scale - To smooth out or remove camera motion, F_Steadiness translates and rotates the frames in the source clip. This leaves black pixels around the image. The Auto Scale parameter lets you fill in the black gaps at the edges by scaling the output image up. A value of 1 (the default) scales the image up until no black is visible, whereas a value of 0 disables scaling and leaves the black edges untouched. Auto Scale uses the minimum scale necessary to remove the black gaps at the edges to preserve as much detail as possible.

Four Corner Pin - The corner pins calculated during the analysis pass:

Bottom Left - the lower left corner pin.

Bottom Right - the lower right corner pin.

Top Left - the upper left corner pin.

Top Right - the upper right corner pin.

Using F_WireRemoval

This chapter looks at the removal of wires from images using F_WireRemoval.

Introduction

Many effects movies feature complicated stunts that require an actor to be suspended by wires for their safety. These wires need to be digitally removed.

There are many ways of doing this, including painting the wires out frame by frame and replacing large parts of the background to cover the wires. The method used depends on the type of image under the wire. F_WireRemoval is particularly good at replacing the painting method but it also includes tools to assist in clean plating when new backgrounds have to be tracked into place.



Three wires.



Two of the wires removed using F_WireRemoval.

Background

Clean Plates

The use of clean plates in wire removal is very common and gives good results in certain situations.

Consider a scene shot with an actor suspended from wires and then the same scene shot again without the actor. This second sequence is called the clean plate. The wires from the first shot can be painted out using pixels from the clean plate leaving the actor suspended in thin air.

Shooting a clean plate if the camera is locked off is easy. If the camera moves, then motion control rigs can be used to exactly reproduce the first pass. But it doesn't always work, particularly if the scene is shot against backgrounds that don't look the same on the second pass, such as clouds, sky, or smoke. Motion control rigs are also expensive and that makes them a rarity. Often, a clean plate is not shot during the filming and the compositor is left to create one by merging together unobstructed pixels from many frames. This single frame can then be tracked into place to cover the wires.

FurnaceCore

FurnaceCore's wire removal plug-in should make the process of wire removal much easier. It is particularly good at removing wires over heavily motion blurred backgrounds or wires over smoke, dust, or clouds. It can be used to remove each wire in a sequence or to quickly create a clean plate which can then be tracked into place.

The reconstruction of the background behind the wire can be done spatially, in other words, using only information from the current frame. Alternatively, motion estimation techniques can be used to warp frames from before and after onto the current frame so that, where available, background pixels from these frames can be used to improve the repair. Our reconstruction methods are unique in that they remove the wire without removing and reapplying the grain. They are also tuned to remove long thin objects, leaving other objects untouched. For example, if transient foreground objects cover the wire, they will be left alone.

Reconstruction Methods

Our four reconstruction methods are:

- Spatial
- Temporal With Static Scene
- Temporal With Moving Scene
- Clean Plate

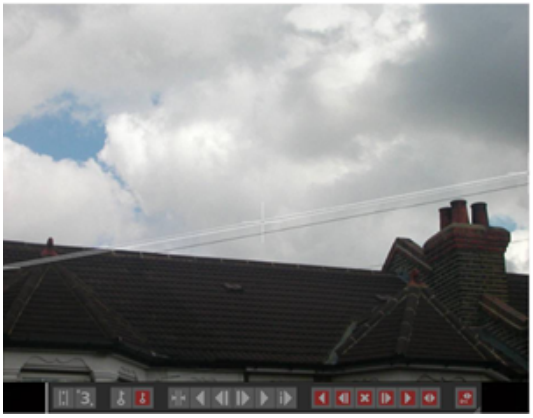
The spatial method takes the background information from adjacent pixels in the current frame, and the clean plate method takes the information from a separate clean plate input. The other methods are temporal and attempt to get background information from frames either side of the current frame. Where this information is not available, for example, because the wire covers part of the same region in one or more of the other frames, the spatial method will be used for the repair.

The spatial method is fastest. It uses a slope-dependent filter that interpolates across the wire at the most likely angle, given the image behind the wire. It works well when the wire is over backgrounds such as sky, clouds, or smoke, and can also cope with some backgrounds where there is a noticeable gradient, such as the edge of a roof, building, or car. If this fails and the wire is moving relative to the background, you should try one of the temporal methods. These look at frames before and after the current one to find likely pixels with which to repair the region and use the spatial method in areas where there are none available.

Where traditional clean plates are possible or give better results than using F_WireRemoval to repair the wire on each frame, you can supply a clean plate as the fourth input to the plug-in. It will then automatically match-move it to repair each frame. If the overall luminance of the background is different to that of the clean plate or varies during the sequence, turn on **Luminance Correct**. The luminance of the background pixels used for the reconstruction will then be adjusted before the repair is made. Of course, various FurnaceCore tools can be used to create a clean plate, including an F_WireRemoval pass on a single frame where the repair has been successful.

Tracker

F_WireRemoval incorporates a tracker which can automatically track a moving wire through a clip. This tracker has its own control panel, which will float inside the NukeX viewer if you have checked **Show On Screen Controls** in the WireRemoval controls. Below is a screenshot that illustrates this.



Screenshot with tracker panel.

You can resize the panel by dragging its edges. For details of the rest of the controls, see the section on Tracker Controls on Tracker Controls. All the controls are also available in the WireRemoval properties.

Quick Start

To remove a wire from an image sequence, do the following:

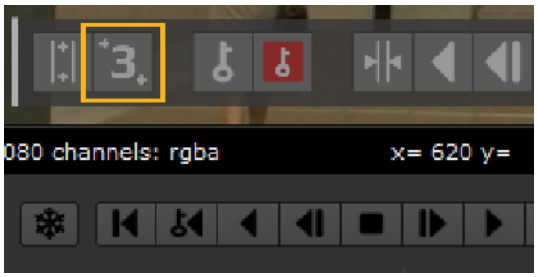
1. Select **Image** > **Read** to load the clip that needs a wire removed.
2. Select **FurnaceCore** > **F_WireRemoval**. Connect your image sequence to F_WireRemoval's **Source** input. View the output from F_WireRemoval.
3. Make sure **Output** is set to **Source**, and use the on-screen tools to draw a region that defines the position and shape of the wire to be removed. For more information, see Positioning the On-Screen Wire Tool.
4. If the wire you want to remove is moving, start the wire tracker by clicking on the Track Forwards button, so that it follows the wire during the clip. For more information, see Tracker Controls.
5. From the **Repair** dropdown menu, choose a repair method that removes the wire. If you choose to use the **Clean Plate** method, connect a clean plate image to F_WireRemoval's **CleanPlate** input. For more information on the available methods, see Reconstruction Methods.
6. To see the output, set **Output** to **Repair**.
7. If necessary, increase **Overall Width** to expand the width of the repair region until the wire disappears.

Positioning the On-Screen Wire Tool

To position the on-screen wire tool, do the following:

1. In the F_WireRemoval controls, set **Output** to **Source**. This way, you will be able to see the wire you're trying to remove but won't have to wait for F_WireRemoval to repair the wire every time you change wire tool's position.
2. Choose the number of points that are needed to describe the wire: two for straight lines, three for simple curves, or five for more complex shapes. This can be done using the **Type** dropdown in the properties panel, or by toggling the Number of Points button in the on-screen tracker panel.

Tip: To show the on-screen tracker panel, check **Show On Screen Controls** in the WireRemoval properties.



3. Position the on-screen wire tool so that it roughly fits the wire you want to remove. To move the entire tool, drag its central line to a new location. To move an individual wire point on the tool, drag its center. To move an individual point in either a horizontal or vertical direction only, drag the horizontal or vertical line on the cross that marks the point.
4. Press the Snap To button  in the WireRemoval properties or the on-screen tracker panel. This will find the edges of the wire and adjust the removal region to fit it more closely. It does this by locating the areas of highest gradient, which should correspond to the edges of the wire.
5. In practice, the edges of the wire will be slightly soft, so the resulting region might not cover the whole width of the wire. To correct for this, you can adjust the **Overall Width** parameter. This expands the region outwards, over all key frames, to ensure the entire width of the wire is covered.

Tracking

F_WireRemoval incorporates a tracker that tracks the region to be repaired through the image sequence.

To use the tracker, do the following:

1. Drag the on-screen wire tool to position the repair region on top of the wire on one frame. This sets a user keyframe for the current frame, enabling tracking.
2. To start the tracker, press the Track Forwards button .

   If you haven't set a user keyframe (step 1), the tracker cannot be started and F_WireRemoval displays the following message:

   Please select a user key frame before tracking.
3. When the track is ready, check that the wire has been correctly tracked across the other frames and adjust the track if necessary. The tracking controls in the tracker panel are outlined under Tracker Controls. Any points initially positioned off the image will remain off the image. This way, the tracker will not lose the wire at the edges of the image.
4. If after tracking the wire you adjust the position of the on-screen wire tool manually on one frame, you may want to create a smooth transition between the user keyframe you created and the track keyframes

around it. The easiest way to do this is to go on the previous and next frame on the timeline and press the Delete track keyframe button . This deletes the track keyframes around the frame you adjusted and forces NukeX to interpolate smoothly around the adjustment.

Similarly, if there is too much jitter in the track between some frames, you can delete the track keyframes between these frames. This way, NukeX will interpolate smoothly from one frame to another.

If it is not possible to track the wire automatically, the wire can be manually keyframed. Adjusting the on-screen wire tool on each frame will automatically set user keyframes. Alternatively, they can be set by pressing the Create User Keyframe button  in the tracker panel.

User Keyframes and Track Keyframes

User keyframes are keyframes you set manually. When a user keyframe has been set on the current frame, this is indicated by a red key that appears on the on-screen wire tool. Track keyframes are keyframes set by F_WireRemoval's tracker.

User keyframes override track keyframes on the same frame. This has the following consequences:

- Once you set a user keyframe, the track keyframe on that frame will be lost.
- When you track a wire, existing user keyframes will not be replaced by track keyframes.

Indicators on the On-Screen Wire Tool

You may see the following indicators appear on the on-screen wire tool:

Control	Description
	The current frame is inside the range of frames to track the wire over.
	The current frame is not inside the range of frames to track the wire over.
	The current frame is the first frame in the range of frames to track the wire over.
	The current frame is the last frame in the range of frames to track the wire over.
	A user key frame has been set on the current frame.

Tracker Controls

The following controls appear in the on-screen tracker panel that you can display in the Viewer by checking **Show On Screen Controls** in the WireRemoval properties.

Control	Description
	**Toggle display mode** - Toggle the display mode for the on-screen wire tool: show the points and lines, show just the points, or hide both points and lines.
	**Number of points** - Changes the number of points used to describe the wire.
	**Create user key frame** - Creates a user keyframe.

Control	Description
	**Delete user key frame** - Deletes a user keyframe.
	**Snap to wire** - Finds the edges of the wire and snaps the edges of the region onto them.
	**Track backwards** - Plays backwards through the sequence tracking from frame to frame.
	**Step backward** - Tracks backwards one frame.
	**Step forward** - Tracks forward one frame.
	**Track forwards** - Plays forwards through the sequence tracking from frame to frame.
	**Smart track** - Tracks from beginning to end of frame range in an intelligent order.
	**Delete track key frames backwards** - Deletes track keyframes backwards through the sequence until either a user keyframe or the beginning of the sequence is reached.
	**Delete track key frame and step backward** - Deletes a track keyframe and steps backwards one frame.
	**Delete track key frame** - Delete the current track keyframe.
	**Delete track key frame and step forwards** - Deletes a track keyframe and steps forwards one frame.
	**Delete track key frames forwards** - Deletes track keyframes forwards through the sequence until either a user keyframe or the end of the sequence is reached.
	**Delete all track key frames** - Deletes all track keyframes from the sequence.

Control	Description
	**Delete all track and user key frames** - Deletes both track keyframes and user keyframes.

The corresponding controls are also available in the WireRemoval properties.

Parameters

The parameters for this plug-in are described below.

Tracker buttons - most of the on-screen tracker panel buttons also appear here. For more information, see Tracker Controls.

Type - this parameter controls the number of points on the on-screen wire tool. Choose the number of points needed to describe the wire you wish to remove.

- **Two Points** - choose this if your wire is straight.
- **Three Points** - choose this if your wire is a simple curve.
- **Five Points** - choose this if your wire has an s-shaped curve.

On-Screen Wire - the display mode for the on-screen wire tool.

- **Show** - show both points and lines.
- **Hide** - hide both points and lines.
- **Points only** - only show the points.

Show On Screen Controls - use this to show or hide the tracker panel in the Viewer.

Output - sets the output mode for the plug-in.

- **Source** - output the untouched source image. Use this output mode to position the on-screen wire tool over the wire you wish to remove.
- **Repair** - output the repaired source image, with the wire removed from under the on-screen tool.
- **Wire Matte** - renders a matte for the wire. This may be useful if the wire has been tracked but cannot be repaired using F_WireRemoval and other techniques have to be used.
- **Repair Matted** - output the repaired source image and a matte in the alpha channel. If you want, you can adjust your image further manually using the matte.

Track Range - the range of frames to track the wire over.

- **Specified Range** - use the **Track Start** and **Track End** parameters to specify the range over which to track the wire.
- **Source Clip Range** - track the wire over the entire range of the **Source** clip.

Track Start - use this to specify the start of the tracking range when **Track Range** has been set to **Specified Range**.

Track End - use this to specify the end of the tracking range when **Track Range** has been set to **Specified Range**.

Repair - sets the algorithm used to remove the wire from under the grain.

- **Spatial** - this method uses a slope dependent filter that interpolates across the wire at the most likely angle, given the image behind the wire. It uses information from the current frame only.
- **Temporal With Static Scene** - this method uses local motion estimation to align frames from before and after onto the current frame. If the wire is not in the same place with respect to the background in these two frames, it can then use background information from them to fill in the background behind the wire in the current frame. Where no such information is available, for example, if the wire covers part of the same region in all three frames, the spatial repair method above will be used. This is useful for sequences where the wire is moving and where the motion in the rest of the scene is non-uniform, for example, if the camera has been locked to place but there are objects moving in the area surrounding the wire.
- **Temporal With Moving Scene** - also aligns frames from before and after onto the current frame, but uses global motion estimation. Again, it gets background information from these two frames where it can and uses the spatial repair method to fill in the rest. This is useful for sequences where the wire is moving and the motion in the rest of the scene is fairly uniform, for example, if the entire scene is moving in the same direction as a result of a camera pan.
- **Clean Plate** - choose this method if you have a clean plate you wish to use for the repair, or if F_WireRemoval does not do a good job of removing the wire from each frame. Connect the clean plate or single frame with the wire removed to the **CleanPlate** input. F_WireRemoval will then align this frame to the frame to be repaired in order to reconstruct the background behind the wire.

Filter Size - this is a trade off between the amount of grain removed and the blurring of image along the direction of the wire. If the wire you are trying to remove has detail within it (for example, a steel wire in which the twisted threads are reflecting light), then the algorithm may leave these alone, thinking that they are grain. In this situation, you should decrease the filter size. A value of zero will definitely remove the artifacts but also the grain, which would then have to be put back using FurnaceCore's F_ReGrain.

Temporal Offset - the time offset of the additional frames to use for the **Temporal With Static Scene** or **Temporal With Moving Scene** methods. This determines which two frames before and after the current frame are used to fill in the background behind the wire. For example, if this is set to 2 and the current frame is 30, frames 28 and 32 are used.

Luminance Correct - turn this on for methods other than Spatial repair where there are global luminance shifts between one frame of the sequence and the next, or between a frame of the sequence and a clean plate you are using for the repair. With **Luminance Correct** turned on, F_WireRemoval will adjust the luminance of the background pixels it uses to the correct level before doing the repair.

Lum Block Size - change this value if luminance correction is not working as well as it should. The luminance correction uses overlapping blocks and matches the luminance within those blocks; changing the size of the blocks will change the regions covered by each block and could result in a better match.

Points - Parameters to set the position of the points used to define the wire.

Point 1 - the position of the start point on the wire.

Point 2 - the position of the point on the wire between the start point and the mid point (this is only active if **Type** is **Five Points**).

Point 3 - the position of the mid point on the wire.

Point 4 - the position of the point on the wire between the mid point and the end point (this is only active if **Type** is **Five Points**).

Point 5 - the position of the end point on the wire.

Start Width - the width of the wire at point 1 of the on-screen wire tool.

End Width - the width of the wire at point 5 of the on-screen wire tool. This allows you to make your repair region wider at one end than the other, which is useful for cases where there is motion blur on the wire.

Overall Width - increase this parameter to expand the width of the repair region along its entire length, and for all key frames.

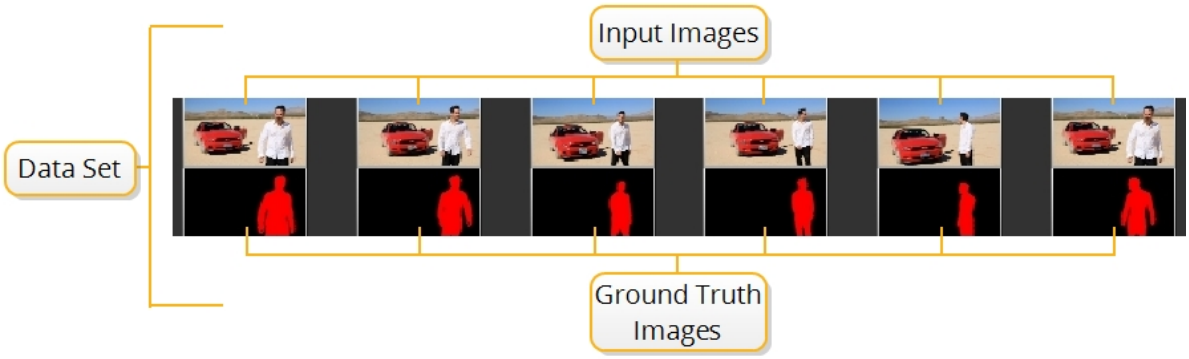
Train Neural Networks to Replicate Effects Using Machine Learning

The CopyCat node (NukeX and Nuke Studio only) copies sequence-specific effects, such as garbage matting, beauty repairs, or deblurring, from a small number of frames in a sequence and then trains a network using machine learning to replicate this effect on a full sequence. CopyCat outputs a trained network in a **.cat** file ready for the Inference node to apply your effect.

Note: NVIDIA CUDA is not supported on macOS so training defaults to the CPU, which may result in longer training times.

CopyCat ingests any number of image pairs called **Input** and **Ground Truth** images, which are referred to as the data set. The data set is used to train a neural network using machine learning to manipulate the **Input** images to replicate the **Ground Truth**. For example, a data set could be six frames from a sequence that you want to garbage matte and six manually roto'd masks, as shown in the image.

Note: CopyCat can ingest sequential formats, such as **.jpg** and **.exr**, and container formats, such as **.mov** and **.mxf**, but the set up is slightly different in each case. See Create a Data Set for Training for more information.



As the network learns, it creates **.cat** files in the **Data Directory**, which can then be used by an Inference node to reproduce the effect learned from a few frames across the whole sequence.

Creating a Data Set for Training

Learn how to create a data set to train a network to perform a specific task.

Training and Monitoring the Network

Train your network using the data set to replicate the desired effect.

Applying and Improving the Results

Apply a trained network to a sequence using the Inference node.

Training on Multiple GPUs Simultaneously

Machines with multiple GPUs can leverage the extra processing power in two ways:

- Train a single network using all GPUs up to 30% faster than with a single GPU. See Train the Network to Perform a Task for more details.
- Run different training sessions on each GPU, allowing you to train multiple models at the same time. See Train Networks from the Command Line for more information.

BigCat

Most CopyCat workflows involve propagating edits from a small number of frames across the rest of a shot or sequence. While CopyCat is capable of generalizing to a wider variety of shots - this requires significantly larger training datasets.

To support workflows at that scale, we've begun extending the CopyCat feature set into a new node called BigCat. This separation is intentional: features designed for large-scale generalization can be counterproductive in typical CopyCat scenarios, where the goal is to deliberately overfit to a specific shot or sequence. This also avoids adding additional complexity to CopyCat for typical usage.

Using BigCat

Learn when and how to use BigCat for larger datasets.

Create a Data Set for Training

A CopyCat data set is a number of image pairs used to train a network to perform a specific task. Image pairs consist of one **Input** image and one **Ground Truth** image, before and after the effect is applied. The network is attempting to learn how to create the **Ground Truth** from the **Input** image.



Note: The **Input** and **Ground Truth** image pairs must be the same format size.

CopyCat can ingest sequential formats, such as **.jpg** and **.exr**, and container formats, such as **.mov** and **.mxf**, but the set up is slightly different in each case.

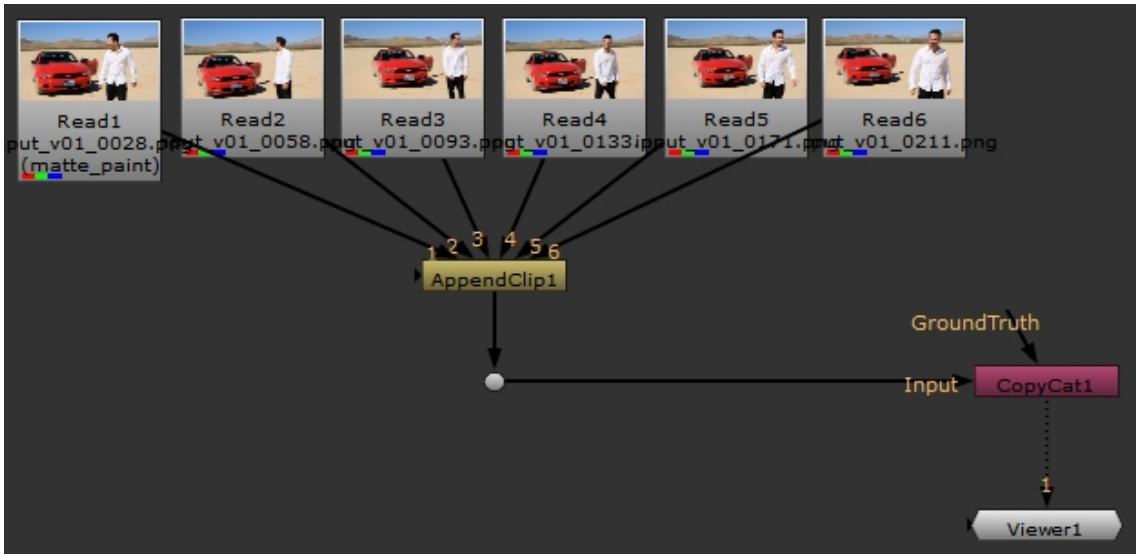
Sequential Format Data Sets

Sequences of **.jpg** and **.exr** files are the simplest data sets to create because they're already broken down into the individual frames required for training. This example shows a data set you could use to teach a network to apply a matte, but the principle is the same for clean up, deblurring, or any other manual work.

1. Play through your sequence and select a number of frames to use for the data set.

> **Tip:** A diverse selection of images tends to produce the best results and the more frames you use, the better the results are likely to be. For example, if you're training a network to mask an object, try to pick frames that represent a wide variety of mask shapes.

2. Create a Read node for each frame selected and set the **File** control to point to the required frame number. For example:
   `../assets/input/input_v01_0211.png`
3. Create an AppendClip node and connect all the Read nodes.
4. Connect the AppendClip node to the CopyCat node's **Input** pipe.



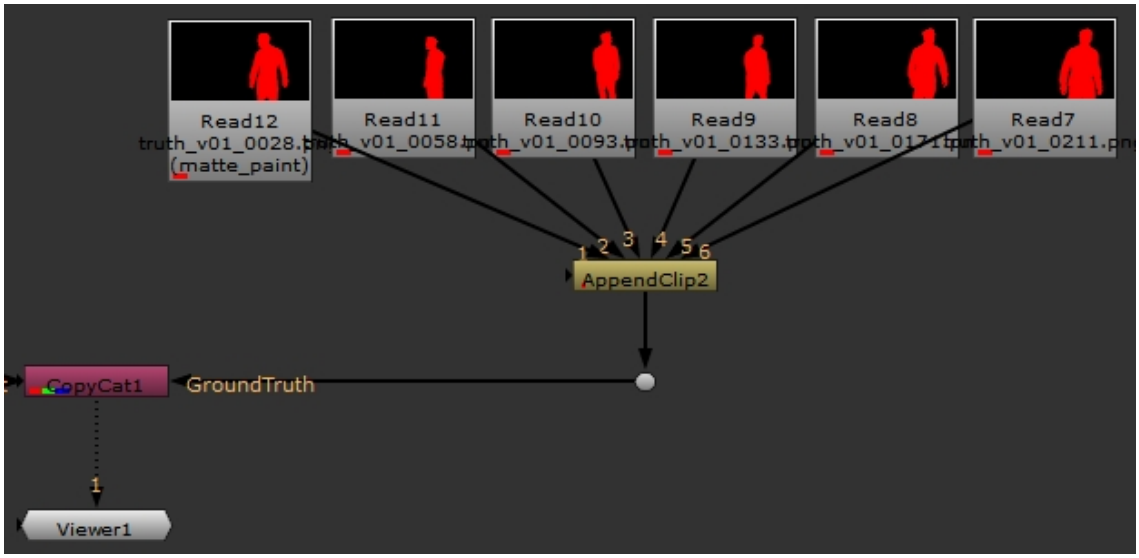
5. Using the same frames, roto out the masks for the **Ground Truth**.
   See Using RotoPaint for more information on Roto and RotoPaint.

> **Note:** In this example, the alpha channel was written to disk separately, but CopyCat can read the alpha from the Roto shapes in the Node Graph.

6. Create a Read node for each of the masks and set the **File** control to point to the required frame number. For example:
   `../assets/groundtruth/truth_v01_0211.png`
7. Create an AppendClip node and connect all the Read nodes.

Warning: Ensure that your **Ground Truth** images are connected to the AppendClip node in the same order as your **Input** images. Image pairs that don't match affect the quality of the resulting model.

8. Connect the AppendClip node to the CopyCat node's **Ground Truth** pipe.



9. Proceed to Train and Monitor the Network for more details on CopyCat settings and training.

Container Format Data Sets

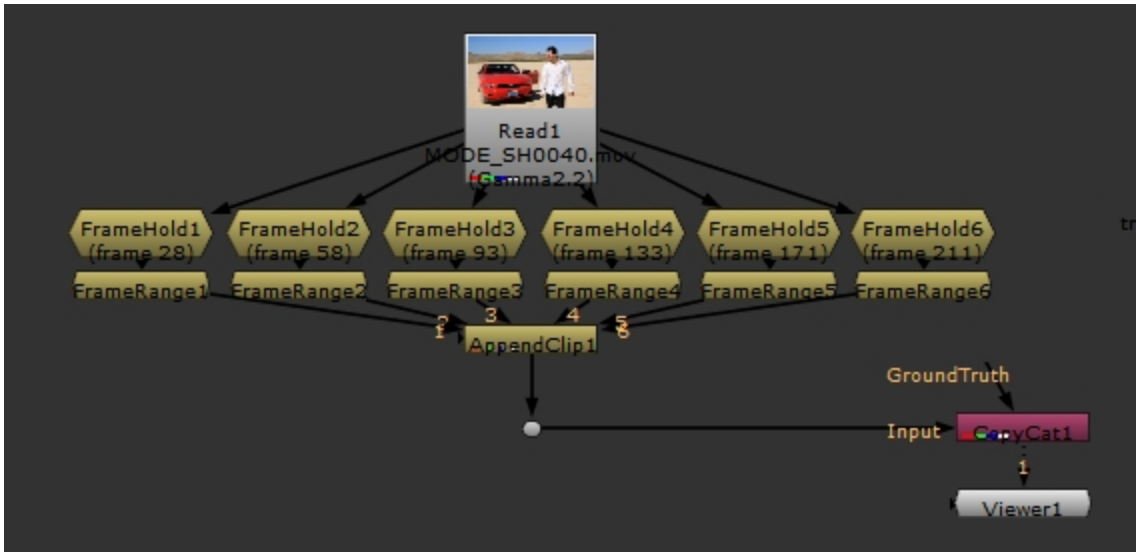
Container formats like **.mov** and **.mxf** files require a bit more work to separate out the individual frames required for training. This example shows a data set you could use to teach a network to apply a matte, but the principle is the same for clean up, deblurring, or any other manual work.

1. Play through your sequence and select a number of frames to use for the data set.

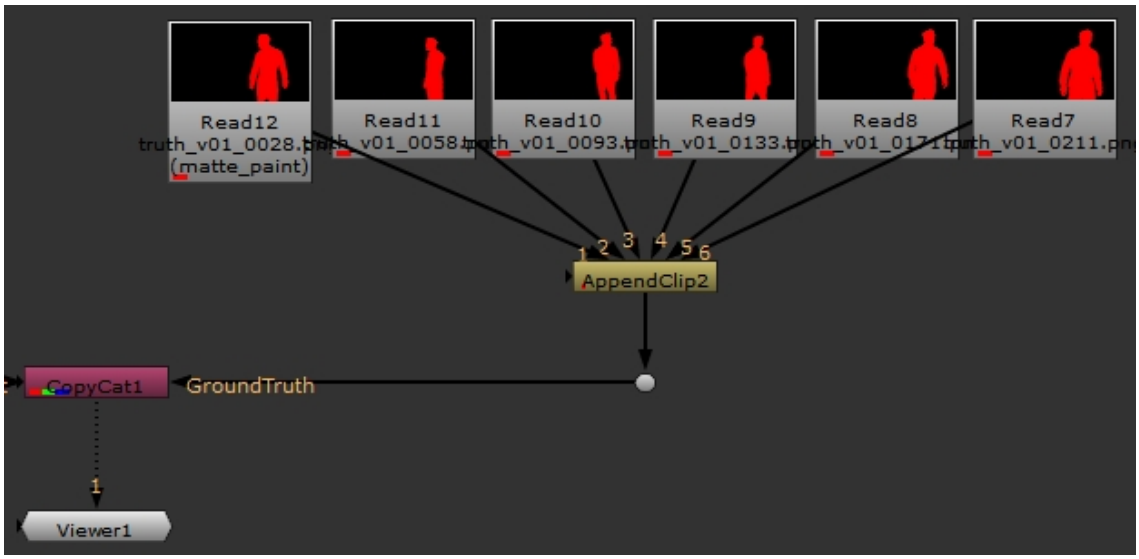
   A diverse selection of images tends to produce the best results and the more frames you use, the better the results are likely to be. For example, if you're training a network to mask an object, try to pick frames that represent a wide variety of mask shapes.
2. Create a Read node and set the **File** control to point to the full sequence. For example: `../assets/input/input_v01.mov`
3. Create a FrameHold and FrameRange node for each frame selected in the data set.
4. Set each FrameHold node's **first frame** control to the required frame and the FrameRange node's **frame range** control to 1-1.

Note: The FrameRange nodes ensure that only 1 frame is ingested by CopyCat per Read node, the frame specified by the FrameHold node.

5. Create an AppendClip node and connect all the FrameRange nodes.
6. Connect the AppendClip node to the CopyCat node's **Input** pipe.



7. Using the same frames, roto out the masks for the **Ground Truth** and write the alpha channel to disk. See Using RotoPaint for more information on Roto and RotoPaint.
8. Create a Read node for each of the masks and set the **File** control to point to the required frame number. For example:
   `../assets/groundtruth/truth_v01_0211.png`
9. Create an AppendClip node and connect all the Read nodes.
10. **Warning:** Ensure that your **Ground Truth** images are connected to the AppendClip node in the same order as your **Input** images. Image pairs that don't match affect the quality of the resulting model.
11. Connect the AppendClip node to the CopyCat node's **Ground Truth** pipe.



Training and Monitoring the Network

Train your network using the data set to replicate the desired effect.

Applying and Improving the Results

Apply a trained network to a sequence using the Inference node.

Train and Monitor the Network

For more details on Machine Learning with CopyCat in NukeX, see https://learn.foundry.com/nuke#machine-learning-compositing-in-nuke.

Train the Network to Perform a Task

After setting up your data set of **Input** and **Ground Truth** image pairs, you're ready to train the network using machine learning to replicate the desired effect. There are a number of settings available to fine-tune your model, but they are generally trade-offs between training speed and the quality of the result from the network.

1. Enter the path to the **Data Directory**.

   The output files created by the training process are stored here in the following formats:

   • **.png** - contact sheets and previews showing the progress of the training.



Note: Preview images are only available if the optional **Preview** input is connected. See Previewing the Result for more information.

- **.cat** - files produced by the training and used by the Inference node to apply the learned effect to the sequence. The **.cat** files are saved at the **Checkpoint Interval** and contain all the training data up to that point.
- **.ost** - progress files that allow you to resume training should the process fail or be stopped manually.
- **.summary** - records the loss value at each step so that the results of multiple Runs can be displayed in CopyCat's **Graphs** tab.

See Monitor Training to Get the Best Result for more information about the contents of the **Data Directory**.

2. Set the **Epochs** or number of times CopyCat processes the entire data set during training. **Epochs** are also used to determine the number of **Total Steps** plotted on the **Graphs** tab using the following equation:

```
Total Steps = Epochs * (Data Set Size) / (Batch Size)
```

Higher values generally produce better trained networks, but at the cost of longer processing times.

3. In the **Advanced** controls, you can add **Initial Weights** for certain tasks and set the **Model Size**, **Batch Size**, and **Crop Size**. These controls are trade-offs between training speed and the quality of the result from the network. Larger values may also cause your GPU to run out of memory.
   - **Initial Weights** - Sets whether training begins from scratch or from weighting defined by a previous model:
     - **None** - the training begins from scratch with no weighting applied.
     - **Checkpoint** - the training uses weighting from a previously trained **.cat** file, which is saved to the **Data Directory** every 1000 steps by default.
     - **Deblur** - the training starts with a model weighted towards deblur effects, which can improve the training results for similar operations.
     - **Upscale** - the training starts with a model weighted towards upscaling, which can improve the training results for similar operations.
     - **Human Matting** - the training starts with a model weighted towards rotoscoping humanoid shapes, which can improve the training results for similar operations.

       In this workflow example, selecting **Human Matting** may decrease the number of **Epochs** required to achieve the same result.
   - **Model Size** - trade off speed and memory use against better results. **Small** models are faster to train and use the least GPU memory, but **Large** models may produce better results for more complex tasks.

- **Batch Size** - calculated automatically using the available GPU memory by default. You can set **Batch Size** to **Manual** and set the size to suit your data set and GPU memory, but the size must be less than or equal to the number of image pairs in your data set.

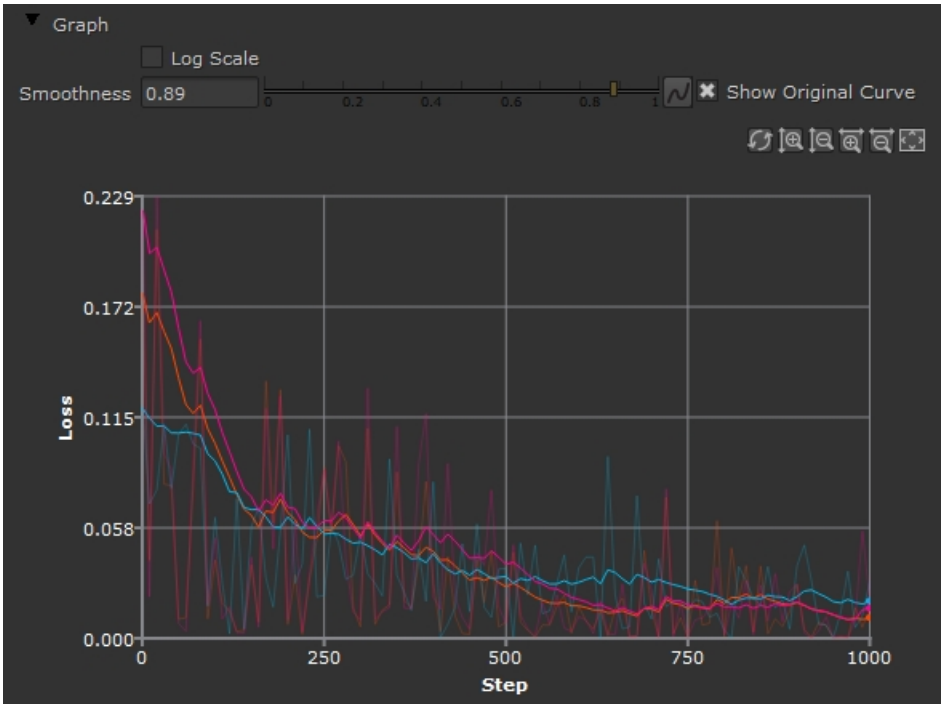
4. 

   Note: Multi-GPU processing requires a **Batch Size** greater than or equal to the number of GPUs you want to use. For example, if you have two GPUs, CopyCat's **Batch Size** must be set to two or higher to use both GPUs. If you have four GPUs and you set the **Batch Size** to two, CopyCat only uses two GPUs.
   If you want to use GPUs individually to train multiple networks at the same time, see Train Networks from the Command Line.

   Tip: The automatic **Batch Size** calculation is designed to be quite conservative, so feel free to adjust it manually if you think your machine can handle larger batches.

   - **Crop Size** - the size of the random crops taken from the data set image pairs. Larger values generally produce more accurate results, but at the expense of processing time and memory.
     In our tests, the default **256** was suitable for most scripts, but data sets of larger images sizes may require a larger **Crop Size** to train effectively.
     The **Crop Size** also affects the size of the grids in the contact sheets written to the **Data Directory**. See Monitor Training to Get the Best Result for more information about the contents of the **Data Directory**.
     If you find that training takes a long time or uses too much memory, try reducing the **Crop Size** control.
5. You can use the **Checkpoint Interval** and **Contact Sheet Interval** to determine how often **.cat** and **.png** files are written to the **Data Directory**. For example, a **Total Steps** value of 5000 and **Checkpoint Interval** of 1000 would produce 5 **.cat** files over the course of the training.

   See Monitor Training to Get the Best Result for more information about the contents of the **Data Directory**.
6. Click **Start Training** to begin teaching the network how to replicate the desired effect.
7. Click the **Progress** tab to view the training progress represented as a graph of Step/Loss.



Monitor Training to Get the Best Result

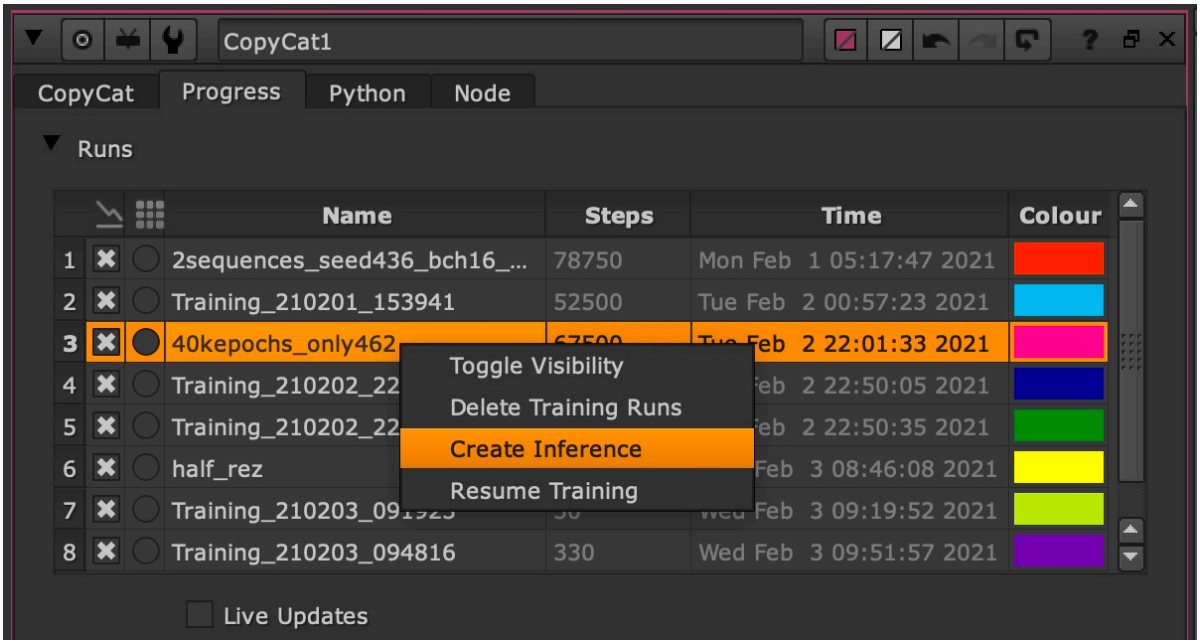
As the training progresses, it's important to monitor the progress of the machine learning to get the best result. You can monitor progress using CopyCat's **Properties** panel **Progress** tab, by examining the Viewer crop overlay and corresponding contact sheets, and by connecting the optional **Preview** input on the CopyCat node to the source sequence.

Runs Table

The Runs table allows you to manage the different training runs contained within your chosen data directory. Each new training run will appear in the Runs table.

- Name - Rename runs to keep track of different experiments. The default naming convention is based on the time the training session was started `[Training_<year><month><day>_<hour><minute><second>]`.
- Graph Visibility - toggle the visibility of each run in the Graph.
- Contact Sheet - choose which run to display in the contact sheet.

You can right-click on one or more training runs in order to toggle the **visibility** within the graph; **delete** the files associated with that training run; create an **inference** node referencing the cat file produced by that training run; or simply **resume training** that run.

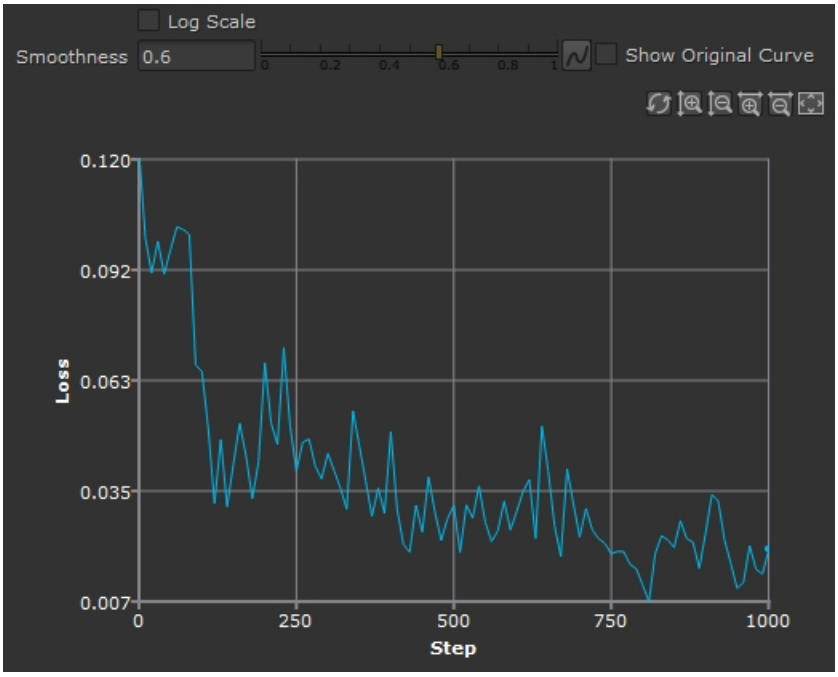


If you're using another session of Nuke, either locally or via distributed training, enable the **Live Updates** option and Nuke will actively monitor the files in your data directory and update the graph and contact sheet.

Monitoring the Graph

As the network completes passes through the data set, you'll see the graph for the current training steadily decline toward zero on the **Loss** axis. You'll also notice that the Viewer displays a crop overlay showing a grid of images.

1. Open CopyCat's **Properties** panel and switch to the **Progress** tab.



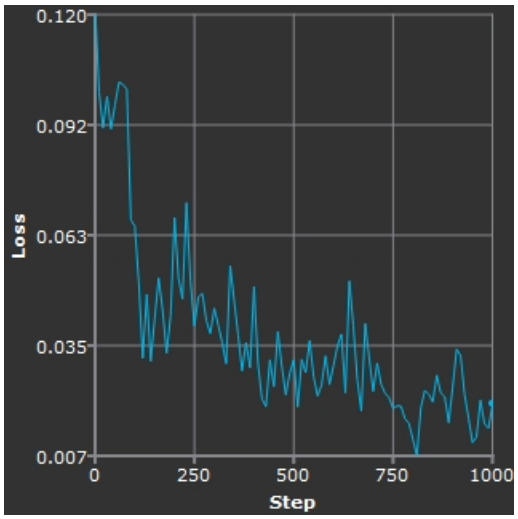
The graph plots the number of steps against loss. Steps begins at 0 and ends at the total number of steps calculated using the following equation:

```
Total Steps = Epochs * (Data Set Size) / (Batch Size)
```

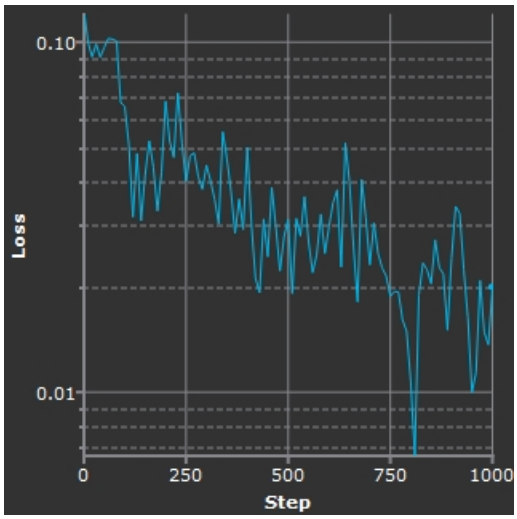
The **Data Set** is the number of **Input** and **Ground Truth** image pairs connected to CopyCat.

For example, 10000 **Epochs** over a data set of 6 image pairs with a **Batch Size** of 4 is equal to 15000 steps.

2. Use the zoom controls above the graph to control the scale:
   - Refresh - click to force a graph refresh between automatic refreshes of the data.
   - Zoom Y in/out - increase or decrease the zoom level on the Y axis.
   - Zoom X in/out - increase or decrease the zoom level on the X axis.
   - Reset Zoom - return the graph zoom state to the default values.
3. You can enable **Log Scale** to allow you to see more detail as the graph approaches 0 on the Loss axis.

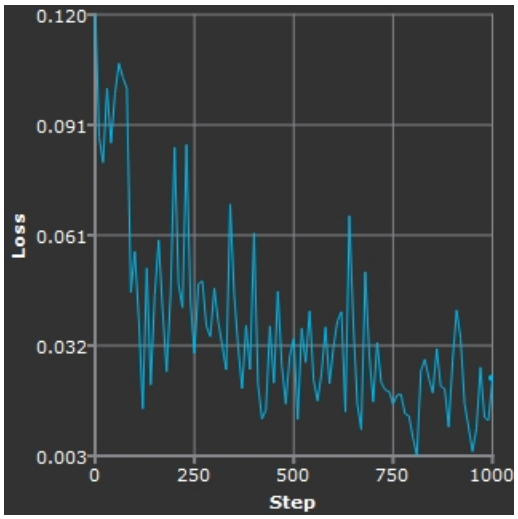


Linear scale.

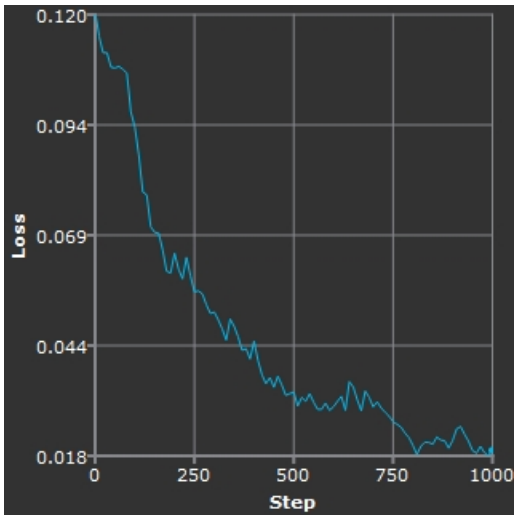


Log Scale showing low values.

4. You can also use the **Smoothness** control to get a better idea of the general trend of the graph as training progresses. Higher values smooth the curve to show a more accurate trend over time and lower values show the actual values, which can fluctuate dramatically.



Low **Smoothness**.



High **Smoothness**.

Monitoring Contact Sheets

As the machine learning completes passes through the data set, as well as the graph in the **Properties** panel, you'll see a crop overlay grid in the **Contact Sheet** panel. As training progresses, you'll notice the **output** images change steadily from random garbage to replicate the **groundtruth** images.

Contact sheets allow you to see if your data set is aligned correctly and display a visual representation of how the network behaves on certain frames.

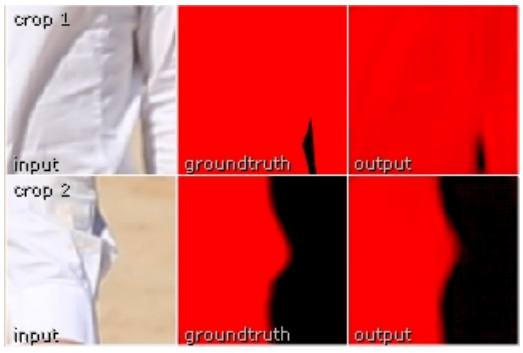


The overlay is split into three random crop rows, each with three columns. The size of the crop square is determined by the **Crop Size** control and generally, a larger area produces a better overall result. Some older GPUs may struggle with large **Crop Size** though, so you may have to reduce the area.

Each random crop row contains three images:

- **input** - a random sample from the **Input** side of the data set image pair. This is a frame from the source sequence that we'll apply the trained network to later on.
- **groundtruth** - the same sample area, but taken from the **GroundTruth** side of the data set image pair. This is what the network is learning to replicate.
- **output** - the same sample area, but with the effect that the network has learned so far applied to the source image.

In the example, **crop 1** and **crop 2** have a reasonable approximation of the **groundtruth** shown in the **output** column.



However, **crop 3** is blank. This is because the **groundtruth** contains no alpha mask in that particular random crop.



Training a network requires both kinds of data to replicate the effect reliably, but you don't have to watch the Viewer overlay to evaluate progress. The overlay is saved as a contact sheet to the **Data Directory** every 100 steps by default. You can change how often a contact sheet is saved using the **Contact Sheet Interval** control in the **Properties** panel.

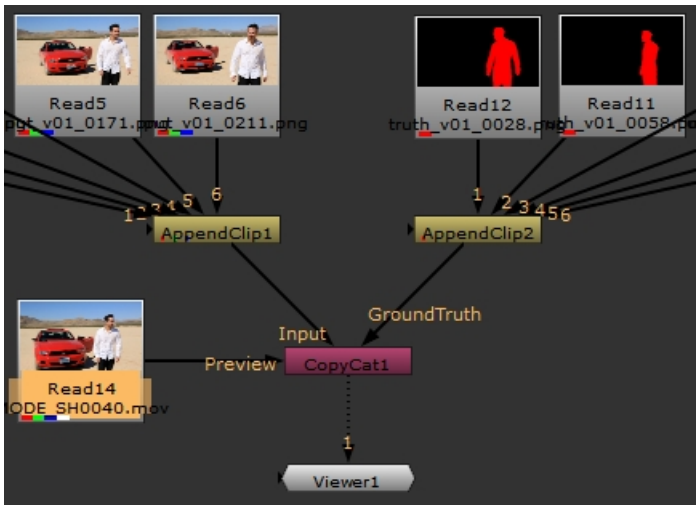
As training progresses, you'll see **.png** files populate the **Data Directory** at the specified **Contact Sheet Interval**. You can see how well the network is learning by examining the images over time. As training progresses, you'll notice the **output** images change steadily from random garbage in the first and second panels below to replicate the **groundtruth** images in later panels.

Tip: If the images in the **output** column appear to replicate the images in the **groundtruth** column before the training run finishes, you can stop training early. You don't have to complete the run if you see the result you want halfway through.

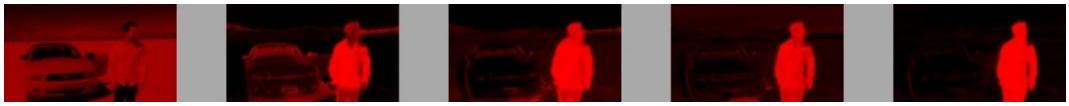


Previewing the Result

CopyCat's **Preview** input allows you to connect the source sequence so that you can view the machine learning progression in the Viewer. The **Preview** allows you to see how the training is progressing on frames outside the data set because the whole source sequence is connected to CopyCat.



You don't have to watch the Viewer overlay to evaluate progress, a preview **.png** file is saved to the **Data Directory** every 1000 steps by default. You can change how often a preview is saved using the **Checkpoint Interval** control in the **Properties** panel.



Reading the Results

You can examine the graph and contact sheets and previews in the **Data Directory**, but what indicates that you might need to tweak some settings or restart the training from a checkpoint or specific weighting?

Contact sheets are designed to be used alongside graph data, so you may not need to restart training if the contact sheet results are not perfect as long as the Step/Loss graph is trending toward zero on the Y axis.

- Check that you have connected the **Input** and **Ground Truth** images in the same order.
- Try adjusting the **Model Size** - more complex operations, such as beauty work, may require a larger **Model Size**.
- Try adjusting the **Crop Size** - higher resolution data sets may produce better models with a larger **Crop Size**.
- If the Step/Loss graph is trending toward zero, but the **output** column is still not replicating the **groundtruth**, try increasing the number of **Epochs** in the run.
- Add more image pairs to the data set - a diverse selection of images tends to produce the best results and the more frames you use, the better the results are likely to be. For example, if you're training a network to mask an object, try to pick frames that represent a wide variety of mask shapes.

If the **Preview** does not look correct on frames outside the data set, even if the graph is trending downwards and the contact sheet output column looks good, there may be image differences in the source sequence

that are not represented by the **Input** and **Ground Truth** images you're using to train the network. These differences include defocus, different framing, shadows covering part of the effect, and so on.

- Add more image pairs to the data set - a diverse selection of images tends to produce the best results and the more frames you use, the better the results are likely to be. For example, if you're training a network to mask an object, try to pick frames that represent a wide variety of mask shapes.
- Try faking the effect on image pairs already in the data set. This can be particularly effective in certain situations such as shadows in the affected area.

Apply and Improve the Results

Apply a trained network to a sequence using the Inference node.

Train Networks from the Command Line

Training networks from the command line, rather than from the Nuke interface, allows you to start training on remote machines that may be more powerful than your local machine. If you have multiple GPUs, you can start multiple training runs simultaneously.



Note: NVIDIA CUDA is not supported on macOS so training defaults to the CPU, which may result in longer training times.

Running Nuke from the command line means typing commands into an operating system tool, such as Terminal on macOS or PowerShell on Windows, rather than launching Nuke in interactive mode from a desktop icon. Command line operation is sometimes called "headless" because you never see the Nuke interface, Nuke processes the script you specify under the hood. See What Is a Terminal and How Do I Use One? for more information and Command Line Operations for other examples of command line flags.

The basic syntax to start training from an existing CopyCat script is as follows: [Nuke executable] [Nuke mode] [arguments] [script location]



Note: You must have a NukeX license to train a network from the command line and include the **--nukex** argument in the command.

For example, from the install location:

```
./Nuke14.0 --nukex -i -F 1 -X CopyCat1 --gpu /path/to/myFirstTraining.nk
```

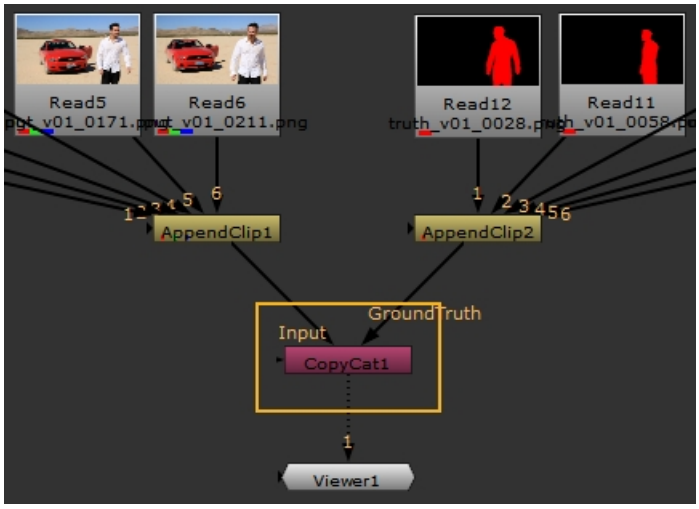
Note: Make sure to include the **--gpu** argument otherwise training defaults to using the CPU, which takes significantly longer to process.

Tip: The **-F 1** argument ensures that the training loops over a single frame. If you omit **-F**, a warning is displayed but the training is not affected.

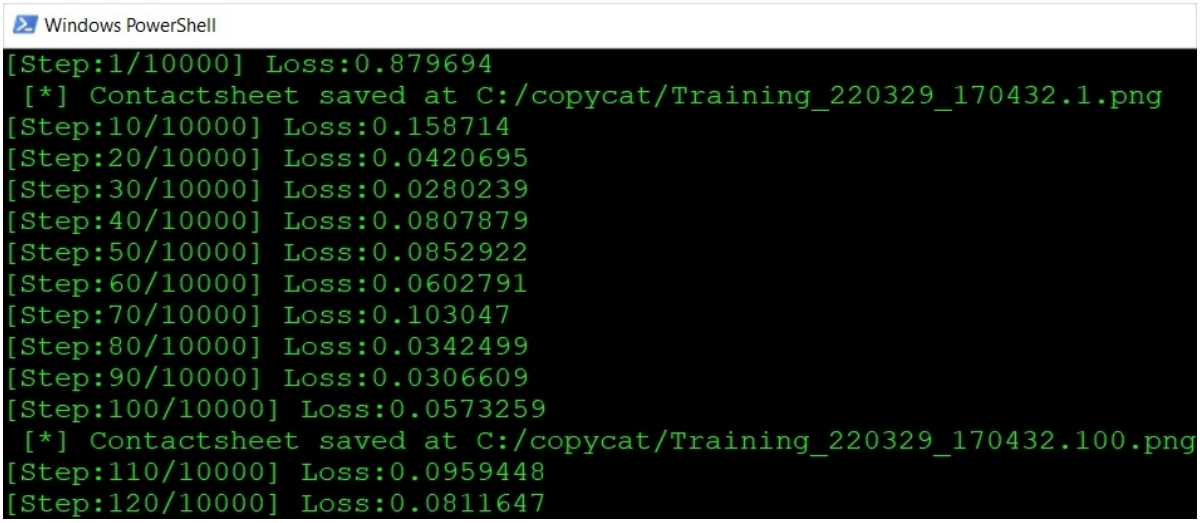
The command starts training with a NukeX interactive license, from the node CopyCat1, on the GPU, and using the script **myTraining.nk** containing the dataset. The name of the node supplied is the CopyCat node that is handling the training with all the **Input** and **GroundTruth** images already set up. In the example shown, CopyCat1 is highlighted, but you can use any CopyCat node in the specified script.

See Create a Data Set for Training for more details on how to create a dataset.

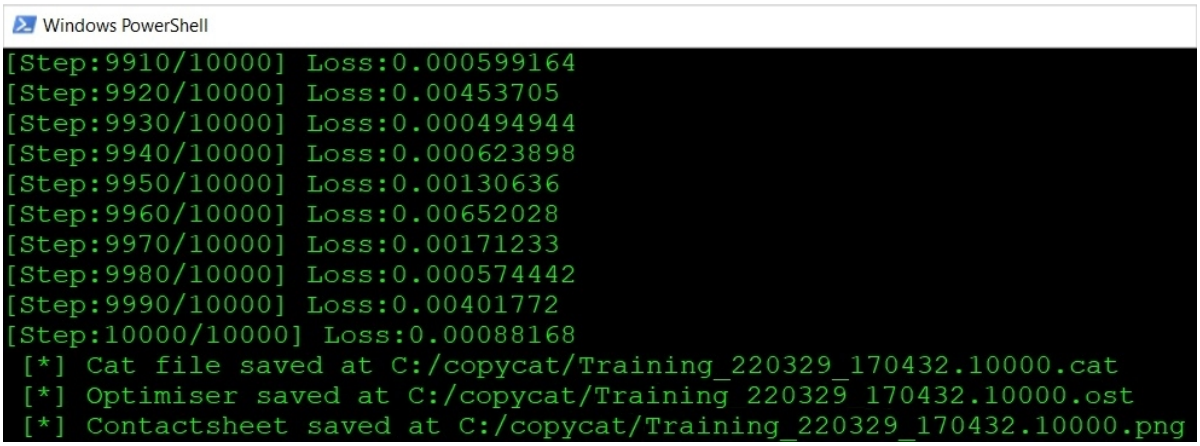
Note: Multi-GPU processing requires a **Batch Size** greater than or equal to the number of GPUs you want to use. For example, if you have two GPUs, CopyCat's **Batch Size** must be set to two or higher to use both GPUs. If you have four GPUs and you set the **Batch Size** to two, CopyCat only uses two GPUs.



Running a script from the command line trains the network in exactly the same way as with an interactive GUI session, generating the same **.png**, **.cat**, **.ost**, and **.summary** files in the chosen **Data Directory**. Contact sheets and Step/Loss graphs are not displayed in command line mode, but the **Step** count and current **Loss** value are printed on-screen.



Training is complete when the specified number of steps are processed or you stop training manually by closing the command line window or pressing the escape hotkey **Ctrl**/**Cmd**+**C**.



You can then apply the trained model to your footage using the Inference node. See Apply and Improve the Results from Your Network for more information.

Train Multiple Networks Simultaneously

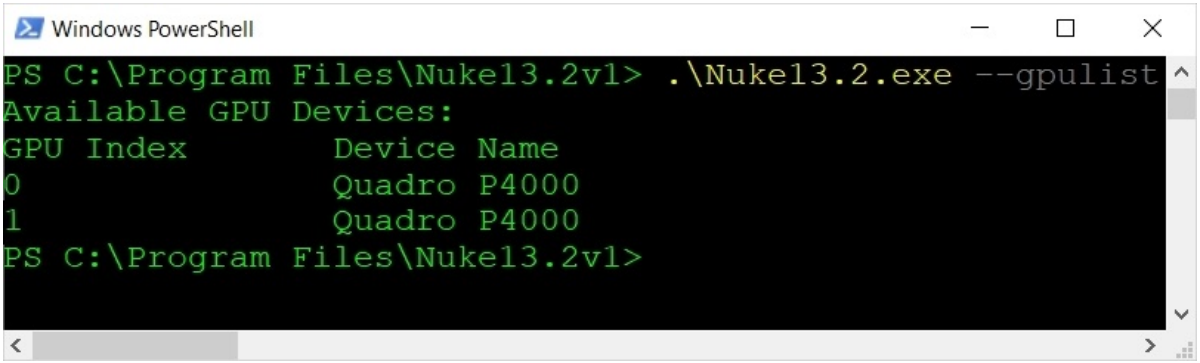
The previous example uses all available GPUs to train a single network. If you want to start simultaneous training on multiple GPUs independently, you need to add the **--gpu [index]** argument to the command line as follows:

```
./Nuke14.0 --nukex -i -F 1 -X CopyCat1 --gpu 0 /path/to/myFirstTraining.nk
./Nuke14.0 --nukex -i -F 1 -X CopyCat1 --gpu 1 /path/to/mySecondTraining.nk
```

You can find out a GPUs index using the **--gpulist** argument. For example, using Windows PowerShell:

```
./Nuke13.2 --gpulist
```

Returns:



Share the Load with Distributed CopyCat Training

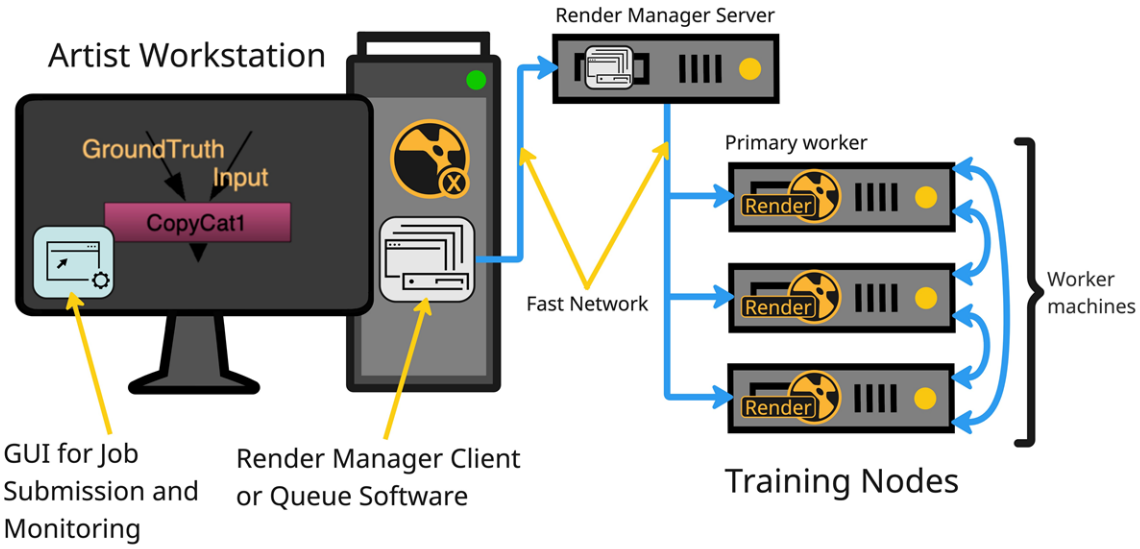
CopyCat can share the training load across multiple machines, with a familiar interface, so that distributing training feels just like distributing rendering using standard render farm applications such as OpenCue.

Offloading the training onto different machines also means that you can continue working when CopyCat is training models in the background.

Note: Distributed CopyCat training is qualified on Nuke 14.1 and 15.0.
Distributed CopyCat training via OpenCue has been validated internally using 14.1. We expect it would work using 15.0 but this has not been tested.

As an artist, you work as normal on your workstation and the render manager handles the distribution of training tasks to the available workers. The image shows a typical distributed training set up using a render manager to distribute the training load.

Note: You must also install Nuke on the worker machines for distributed training to operate as expected.



Note: CopyCat can train using Nuke Render Licences on distributed machines

Distributed Training in CopyCat is controlled through the following environment variables:

- COPYCAT_MAIN_ADDR: the main address. Process 0 runs on this IP address and only this process saves the contact sheets, graphs, and training model checkpoints.
- COPYCAT_MAIN_PORT: the main port for process 0.
- COPYCAT_RANK: the current process rank relative to other processes. COPYCAT_RANK must be < COPYCAT_WORLD_SIZE.
- COPYCAT_WORLD_SIZE: the total number of processes between which training is distributed. For example, if COPYCAT_WORLD_SIZE = 4, four processes are launched on the same Nuke script and CopyCat node with the same COPYCAT_MAIN_ADDR and COPYCAT_MAIN_PORT, and with COPYCAT_RANK being 0, 1, 2, and 3.
- COPYCAT_SYNC_INTERVAL (optional): the interval at which gradients are shared between processes. By default, synchronization happens every 1 step. You can increase this value for better network latency.
- COPYCAT_LOCAL_ADDR (optional/required for speed): the IP address of the local machine on which you are running distributed training. COPYCAT_LOCAL_ADDR can be used to make sure you are running distributed training on fast network lines.

Note: If you omit one of the four required environment variables, CopyCat runs in non-distributed mode and Nuke is locked until training completes or is stopped.

Manual Distributed Training Example

Set up distributed training from the command line.

Automated Distributed Training Example

Set up distributed training using a render manager.

Manage and Monitor Training in CopyCat

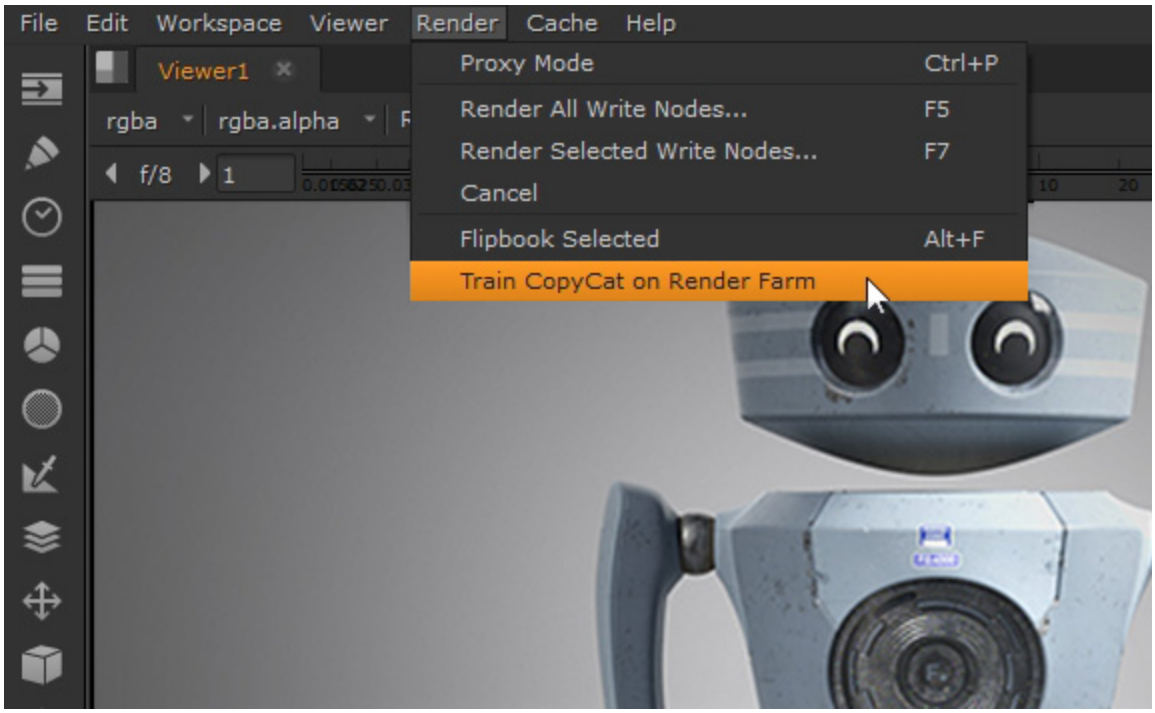
When distributed training is set up and working correctly, you can start training in Nuke's menu bar, rather than in CopyCat's **Properties** panel. Custom menu items can be added using a Python script, an example of which you can find on the Automated Distributed Training Example page. In this example, we're using OpenCue's CueGUI application to manage training runs from within Nuke and we've added a **Train CopyCat on Render Farm** function to Nuke's **Render** menu.

Tip: If you train CopyCat models without distributed training, Nuke is locked until training completes or is stopped.

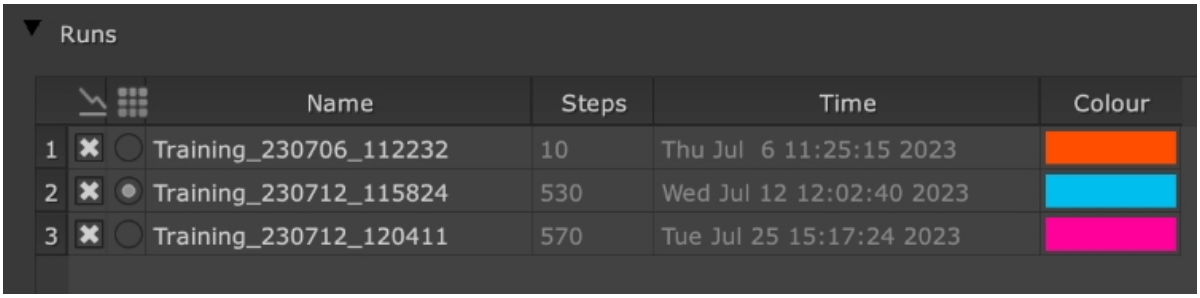
1. Go to **Render** > **Train CopyCat on Render Farm** to start a training run.



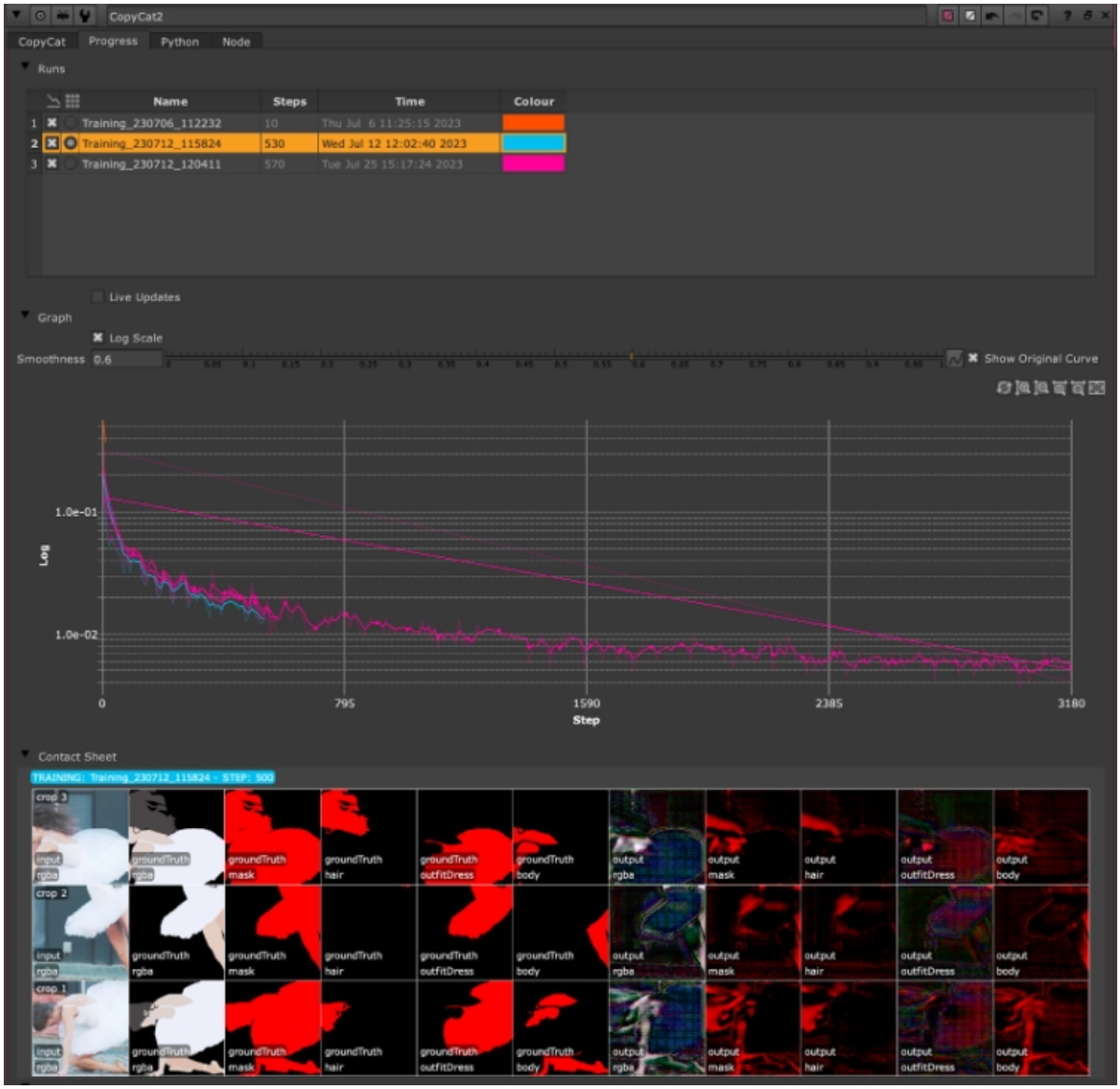
Note: You won't see this option if you haven't added a custom menu item using the **copycatExample.py** script.



A new run is added to the CopyCat **Progress** tab under **Runs**.



2. Each run is shown in the **Graph** dropdown and **Contact Sheet** dropdown by default, but you can toggle off and on the visibility using the  show graph and  show contact sheet radio buttons for each run.



3. If you want to see live updates, turn on **Live Updates** under the **Runs** table. The **Graph** information is not updated by default to save unnecessary network polling.

Optimizing Distributed Training: How Many Machines Should You Use?

Determining the optimal number of machines for distributed training can be complex. However, here's a guide to help you make an informed decision.

Tip: Distributed training scales effectively when each GPU is heavily utilized. In simpler terms, the more work each GPU has, the better the performance when using multiple GPUs.

Factors Influencing Scalability

Batch Size

- In distributed training, the batch is divided evenly among the GPUs. For instance, with a batch size of 8 on 4 GPUs, each GPU processes 2 crops per step.
- A larger batch size means more work for each GPU, leading to better scalability.

Crop Size & Model Size

- Larger crop or model sizes mean more computations per GPU per step, enhancing the scalability when using multiple GPUs.

GPU Specifications

- High-end GPUs can process the same batch faster and with less utilization compared to lower-end GPUs. Thus, with top-tier GPUs, you might not see significant benefits from scaling unless you're also increasing batch or crop sizes.

Examples:

Scenario A

- Batch Size: 12
- Crop Size: 256
- Model Size: Medium
- GPU Spec: Low
- Recommendation: Using four GPUs might double your training speed.

Scenario B

- Batch Size: 8

• Crop Size: 128
• Model Size: Small
• GPU Spec: High

Recommendation: There might be minimal benefit in using more than one GPU.

Scenario C

• Batch Size: 128
• Crop Size: 256
• Model Size: Large
• GPU Spec: High

Recommendation: Deploying up to 8 GPUs could potentially octuple your training speed.

In conclusion, while there's no one-size-fits-all answer, considering the above factors can guide you in optimizing your distributed training setup.

Apply and Improve the Results from Your Network

For more details on Machine Learning with CopyCat in NukeX, see https://learn.foundry.com/nuke#machine-learning.

Apply the Model to the Sequence

Trained networks are applied to sequences using the Inference node. The node reads in a **.cat** file and applies the effect learned from the data set to the whole sequence.

1. Add an Inference node to the Node Graph or click the **Create Inference** button in CopyCat's **Properties** panel.
2. Connect the Inference node to the source sequence Read node.
3. In the Inference node's **Properties** panel, enter the file path of the **.cat** file in the **Model File** control or browse to its location on disk using the  browse button.
4. Connect the Viewer to the Inference node to apply the effect to the current frame.

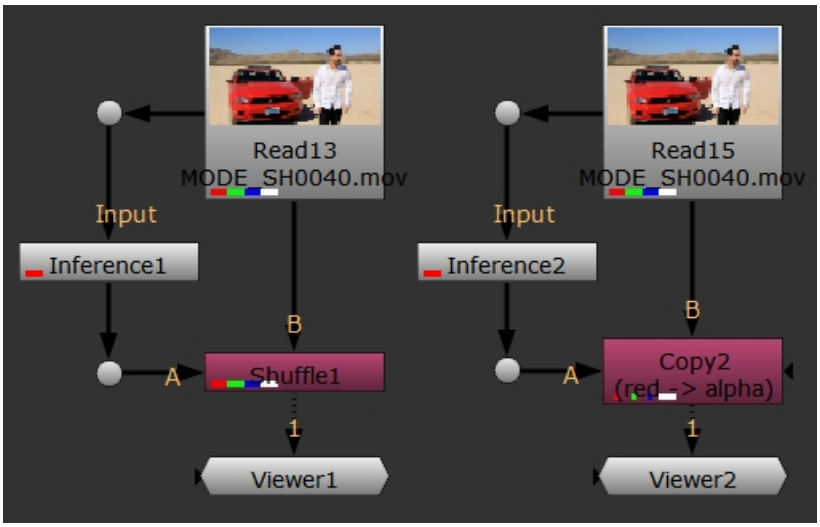
Inference times depend on your GPU's capabilities, but the results are displayed in the Viewer when complete.

Tip: You can enable **Optimize for Speed and Memory** in the Inference node's **Properties** panel to produce a faster result or process larger images, but this can result in artifacts with some trained networks.



5. Skip through the sequence to examine the results on different frames. Inference applies the model on each frame, so you may not be able to playback in real-time, depending on your machine's GPU and RAM specifications.

   You can see from the example frame that a few artifacts remain in the areas outside the mask. Have a look at Improve the Results Your Model Produces for more information about retraining and cleaning up your results.
6. If you're satisfied with the results, all you need to do is Shuffle or Copy the mask channel back into the source sequence and render the results.



Improve the Results Your Model Produces

One of the simplest errors to correct is when you have connected the **Input** and **Ground Truth** images in a different order. You can look at the AppendClip nodes in the Node Graph to check the numbering, but this can be difficult if you have a large data set. An easy way to check is by connecting a Merge node in **difference** mode to compare the **Input** and **Ground Truth** images.

1. Add a Merge node downstream of the **Input** and **Ground Truth** AppendClip nodes.
2. In the **Properties** panel, set **operation** to **difference**.
3. Check each frame in the data set for mismatches by skipping through the image pairs.



Image pair matching.



Image pair mismatch.

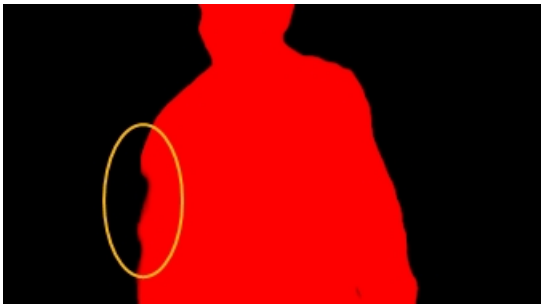
If the result still isn't what you expected, it could be that the image pairs in the data set don't contain enough diverse examples or that elements such as shadows and defocus are not represented in the data set.

Augment Image Pairs in the Data Set to Improve Results

You can fake the elements not represented in an existing data set, which requires less effort than creating brand new image pairs. This can work particularly well for shadow elements or areas where the matte passes over different backgrounds in the source sequence.

Playing through the example source sequence doesn't provide any examples of the where the actor passes directly over the car in the background, with the red color on either side of where the matte passes. This could cause problems for the neural network as it may only expect the actor against predominantly sand, mountains, and sky colors.

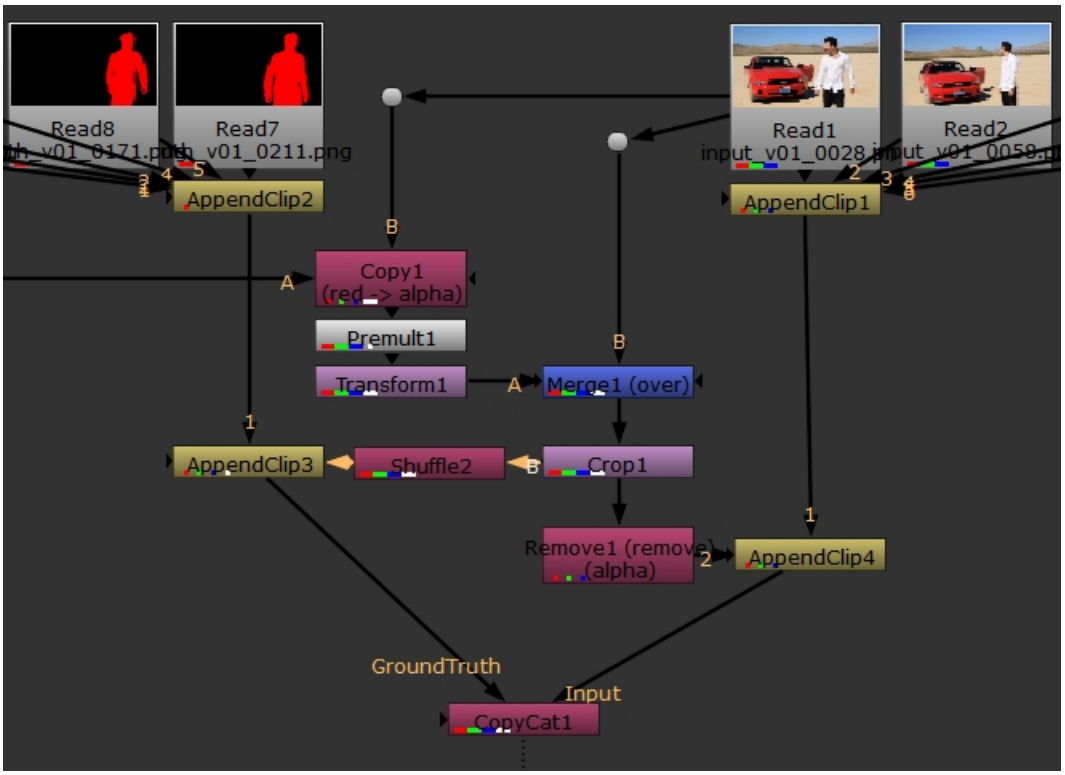





You can fake new foreground and background combinations by copying a **Ground Truth** matte into an **Input** image and then transforming the matte of the actor over the car body. You can also crop the new image pair to focus on just the areas where the actor and car overlap.

Tip: It's a good idea to add your additional image pairs using new AppendClip nodes. This way, you can easily isolate your original data set from faked image pairs if the new pairs aren't working to improve the output.

This simple example shows one possible workflow to produce any number of fake image pairs to add to the data set.



1. Copy the alpha from your chosen **Ground Truth** image into the corresponding **Input** image using a Copy node.
2. Add a Premultiply node to remove the background.
3. Translate the matte over the car using a Transform node.
4. Merge the result with the **Input** image using the **over** operation.
5. Crop the new image to the combination of foreground and background elements you want.
6. Shuffle the alpha back into the **rgba.red** channel for the **Ground Truth** image.
7. Remove the alpha from the **Input** image using a Remove or Shuffle node.
8. The two new images are then added to the data set using new AppendClip nodes.

You can produce a lot of extra image pairs quickly using this method. Feel free to experiment with flipped and flopped mattes as well, any image with the correct combination of foreground and background elements is useful to train a network.







9. When you're happy with the augmented data set, restart training from a checkpoint by setting **Initial Weights** to **Checkpoint** and selecting a **.cat** file from the **Data Directory**.

   Choosing a checkpoint after fewer steps into the training can help to avoid sending the training down the wrong path, but you can also restart from the latest checkpoint by setting **Initial Weights** to **None**. Bear in mind that later checkpoints might contain bias towards an inferior result.
10. Click **Resume Training** to pick up from the selected checkpoint.

Add Image Pairs to the Data Set for Variety

Adding new image pairs to the data set is another solution, but can be very time consuming. For example, the sequence shown in this topic contains a few frames where the actor crosses into the shadow of the car.

1. Play through the source sequence and look for examples of where shadow, defocus, and so on are affecting the matte. In these example, the actor's arm strays into the shadow of the car, which is not represented in the initial data set.



2. Roto the selected frames manually to create the **Ground Truth** mattes.
3. Add the new image pairs to the data set. See Create a Data Set for Training for more information.
4. Click **Resume Training** to continue from the latest **.cat** file in the **Data Directory**.

   You could also start a fresh run from a checkpoint early in the run by selecting **Initial Weights** > **Checkpoint** , entering the path to the **.cat** file in the **Checkpoint File** control, and then clicking **Start Training**. Using a checkpoint from earlier in the run speeds up the training process and can remove the risk of incorrect bias from later checkpoints.

Retrain Your Model Using Checkpoints and Weights

Training a network may not always produce the results you're looking for on the first run, so CopyCat allows you to resume training from the latest checkpoint in the **Data Directory** or restart from a previously trained model. You can select the required option from the **Advanced** > **Initial Weights** dropdown:

- **Checkpoint** - the training uses weighting from an existing **.cat** file. Enter the path to the **.cat** file in the **Checkpoint File** control or browse to its location.

- **Deblur** - the training starts, or resumes from the most recent **.cat** file, with a model weighted towards deblur effects, which can improve the training results for similar operations. For example, you could use the Deblur node on a sequence, and then try to improve the result on frames where it didn't work particularly well using Copycat.
- **Upscale** - the training starts, or resumes from the most recent **.cat** file, with a model weighted towards upscaling, which can improve the training results for similar operations. For example, you could use the Upscale node on a sequence, and then try to improve the result on frames where it didn't work particularly well using Copycat.

Thresholding and Filling the Matte

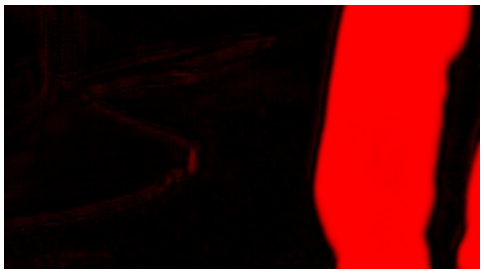
If there are still some artifacts in the background or holes in the foreground matte, you can use the Grade node or RotoPaint nodes to complete the matte. Thresholding refers to adjusting the black point and white point in the Grade node's controls to erode artifacts while leaving the edges of your matte intact. Filling describes manually filling holes in your matte using a RotoPaint node or similar.

In this exaggerated example, the background contains a lot of artifacts of the car and the mountains.



1. Add a Grade node downstream of the Inference node applying the effect.
2. In the Grade node's **Properties**, steadily increase the **blackpoint** to remove the background artifacts.

   As the **blackpoint** increases, you may notice foreground artifacts on the matte or the edges of the matte may start to erode.



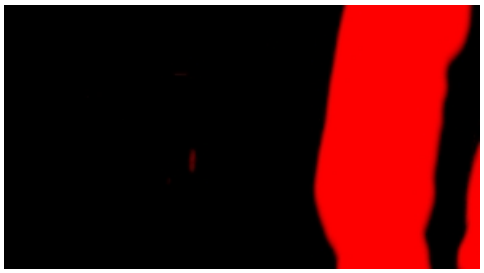
Low **blackpoint** values.



High **blackpoint** values.

3. Decrease the **whitepoint** to remove the foreground artifacts and replace eroded edges.

   As the **whitepoint** decreases, you may notice that the background artifacts return or the edges of the matte may start to dilate.



High **whitepoint** values.



Low **whitepoint** values.

4. Try to find the balance between the **blackpoint** and **whitepoint** that work best.
5. You can cover holes or artifacts in the matte or replace eroded edges using the RotoPaint node. See Using the Brush tool for more information.

Using BigCat

We've begun extending the CopyCat feature set into a new node called BigCat.

When to Use BigCat?

BigCat builds on the same model network as CopyCat but introduces features designed for generalizing across larger datasets.

Use CopyCat when working with smaller datasets and focused tasks - like training a model for a single shot or a set of very similar shots. BigCat, on the other hand, is geared toward broader generalisation across hundreds or even thousands of images.

This approach is particularly effective when you can programmatically assemble a dataset from previous work - for example, reusing mattes from an earlier film when working on a sequel. It's also well suited for projects that incorporate procedurally generated synthetic training data, as discussed in a previous Foundry Live session.

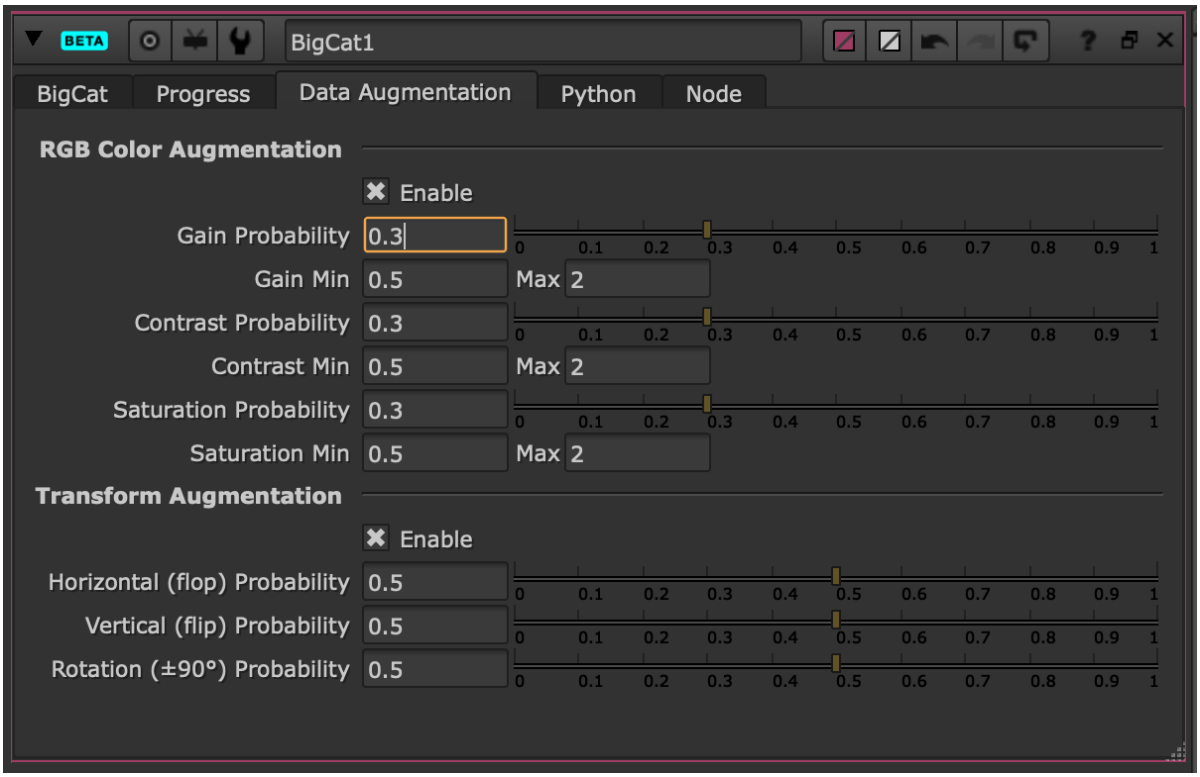
CopyCat generated **.cat** files are fully compatible with BigCat. If you've already trained a model for a specific shot using CopyCat and want to generalise it for other shots, you can use that model as a starting point. However, because such models are often overfitted to their original shots, you may get better results by training a new model from scratch with your larger dataset.

BigCat **.cat** files can be used with the CopyCat node. However if you continue training, using CopyCat the additional BigCat specific metadata such as the custom loss graph may be lost.

BigCat Features

Automatic Data Augmentation

This tab introduces probabilistic grade and transform changes to help create extra variation in the dataset. The extra variation helps BigCat be more robust to different lighting conditions and movements within unseen frames/shots. However if you're training BigCat for just a single shot, this additional variation in the dataset can hinder the training.



Right now the default probability is set to zero however in our use of this augmentation feature during the training of the CopyCat Initial Weights, we used a probability value of 0.5.

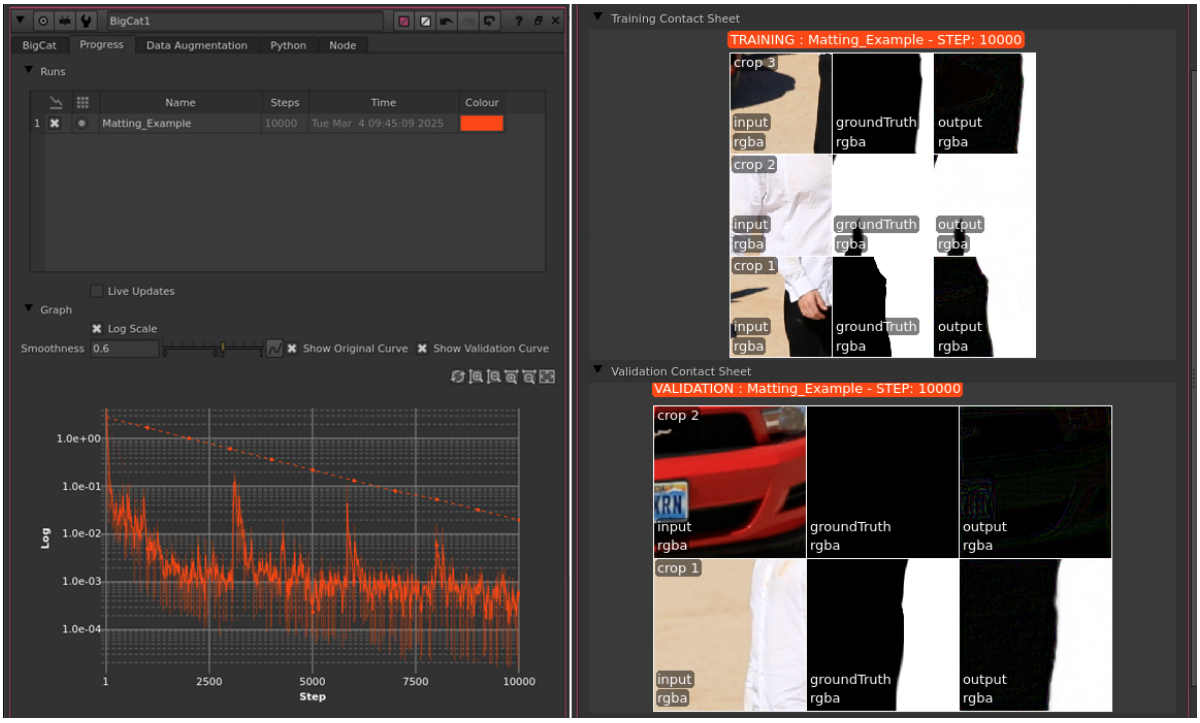
Note: These are the types of augmentation which we've used in training the CopyCat initial weights, there is scope to add further methods of augmentation in the future.

Data Validation

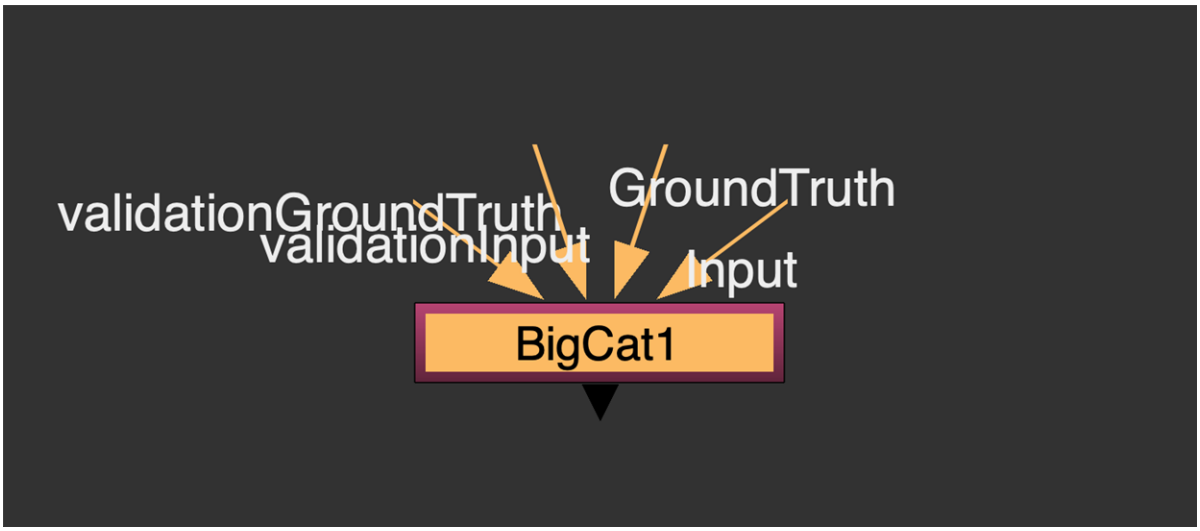
These optional new inputs to the node to allow you to feed in a validation dataset. These should be input/gt frames not included in the main dataset, which can be used to measure how well the model is performing on unseen data. The validation loss will be plotted alongside the training loss in the training graph. This helps when you want to prevent the model overfitting on the training dataset, so the model performs well across a wider range of shots. Validation loss is plotted as a dotted line and can be disabled using the **Show Validation Curve** knob.

Validation loss is calculated at the same interval as the checkpoints are written to disk, which can be adjusted in the advanced settings.

An additional contact sheet has been added to the progress tab, so that inference on the validation dataset can be monitored also.



The BigCat node has additional inputs for this validation dataset, which like the training inputs, also required an input and ground truth.



Custom Loss Functions

BigCat introduces control over loss functions used to guide the training process. In addition to the existing pixel based loss (MSE) we've added a perceptual loss (e.g., LPIPS), which should push BigCat to match structures and semantics rather than raw pixels. This should make the network less brittle to shot-to-shot variation and small misalignments, so a single model trained on diverse examples generalizes better than one trained with pixel losses alone.

In summary, introducing a perceptual loss function to the training process:

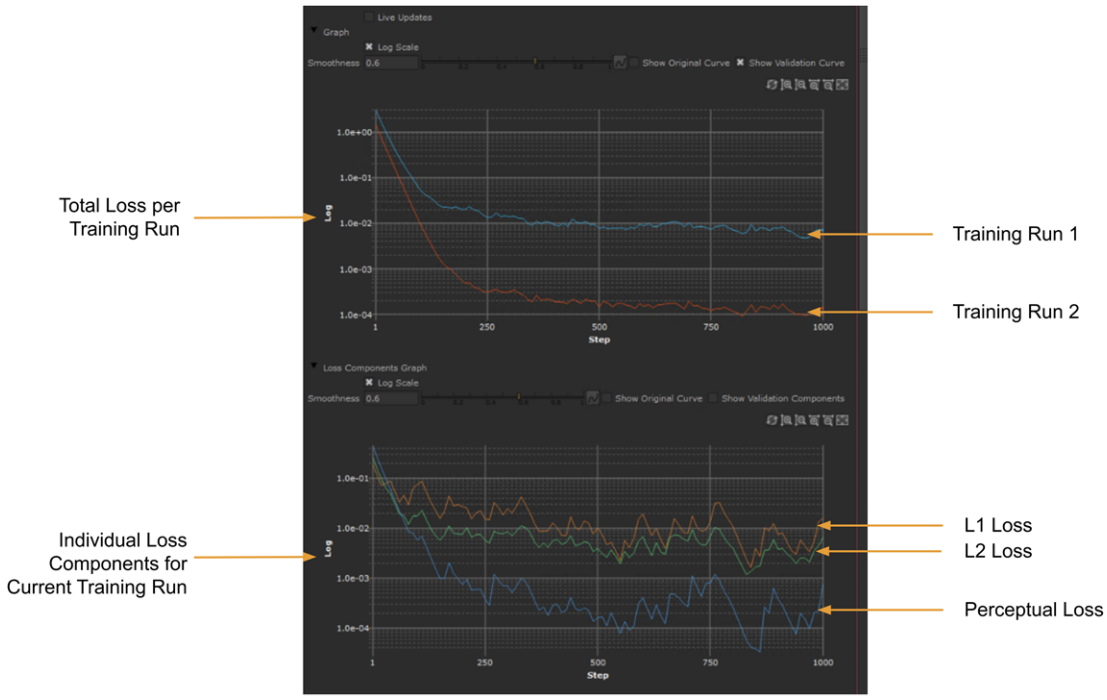
- **Learns what viewers care about.** Perceptual losses compare high-level features instead of per-pixel differences.
- **More tolerant to minor misregistration** Pixel losses punish 1–2px shifts harshly.

Because a perceptual loss can "hallucinate" detail, it's important to maintain a contribution from the pixel loss function. So on the 'Custom Loss' tab an L1 option as well as the current L2 (Mean square error). (Rule of thumb: L2 (MSE) punishes big errors more; L1 (MAE) is more robust to outliers).

Tracking & Balancing the Loss Functions

Each loss function will have a different range and depending on the scenario, you will likely want to adjust the contribution of each loss function. Additionally the loss output will change over time as the model gets better, the values for each loss function will change. Balancing the weighting of each could be tricky.

The loss values for each of the loss functions will be plotted on a second training graph on the progress tab called Loss Components Graph.



Loss Functions and Weighting

The final training loss is a weighted sum of all active loss functions. By adjusting these weights, you guide the model to prioritize different aspects of image reconstruction.

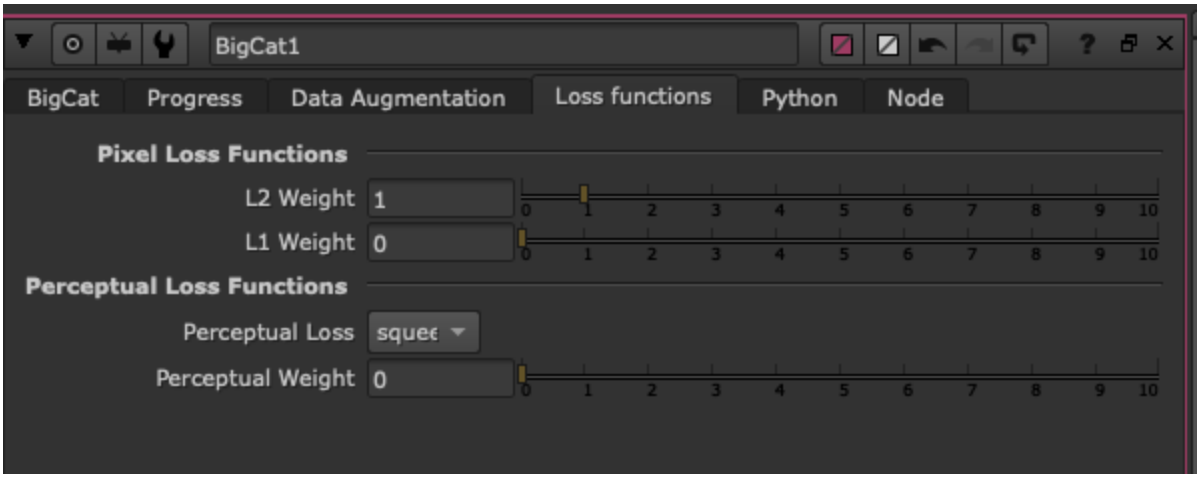
Pixel-Based Losses

Pixel-level losses compare the generated output directly against the ground truth on a pixel-by-pixel basis.

- **L1 (Mean Absolute Error)** Minimizes the absolute difference between pixels. It is generally more robust to outliers and tends to produce sharper results than L2, as it does not disproportionately penalize large pixel differences.
- **L2 (Mean Squared Error)** The default loss used in CopyCat. L2 squares the differences, placing a high penalty on large errors. This often results in a "smoother" or slightly blurrier output, as the model seeks a statistical average to minimize the risk of large deviations.

Perceptual Losses

LPIPS (Learned Perceptual Image Patch Similarity): Unlike pixel-based metrics, LPIPS uses a pre-trained deep network trained on a dataset of human judgments, where the model learns to mimic how humans rank the similarity of two images. This allows it to prioritize textures and structures over exact mathematical pixel values.



Note: Perceptual loss functions utilize neural networks which are typically not trained on commercially permissive datasets. Consequently, these networks will not be bundled with Nuke by default; they must be downloaded from the Cattery. Perceptual loss will only become visible in the properties panel provided the required cat files are present in the designated location within your Cattery directory.

Import Pre-Trained PyTorch Models

The CatFileCreator node (NukeX and Nuke Studio only) creates custom **.cat** files from PyTorch TorchScript **.pt** files within Nuke, opening up a wide range of pre-existing, trained models. You can import TorchScript files from GitHub, Model Zoo, or your own internal R&D department and slot them into your existing pipeline through NukeX's CatFileCreator.

For more details on Machine Learning with CopyCat in NukeX, see https://learn.foundry.com/nuke#machine-learning.

CatFileCreator allows you to add user knobs to your **.cat** files that are automatically passed to NukeX's Inference node so that you can have complete control over your pre-trained machine learning models. To create **.cat** files with exposed user knobs, you need a valid TorchScript **.pt** file containing correctly defined variables that link to the user knobs you define in Nuke.



Warning: It is important to note that not all PyTorch models work without modification. In order to correctly construct a TorchScript model from a PyTorch model please follow the guidelines laid out in the PyTorch user guide at https://pytorch.org/docs/stable/jit.html and the information provided by Foundry in the Nuke **.cat** File Creation Guide here: https://learn.foundry.com/nuke/developers/17.0/catfilecreationreferenceguide/index.html

Converting .pt Files into .cat Files

PyTorch models must be converted to TorchScript **.pt** file so that CatFileCreator can convert them to Inference-readable **.cat** files. In order to create a custom **.cat** file with exposed user knobs, you will first need a valid TorchScript **.pt** file that contains correctly defined variables. These variables will be linked to the user knobs you define in Nuke. The combination of a custom TorchScript and user knobs enables you to create and use a wide variety of different PyTorch architectures in the Inference node, as well as the ability to control their function within Nuke using the user knob definitions.

As noted previously, not all PyTorch models can be converted without modification, so have a look at the Introduction to TorchScript guide provided by PyTorch.

To convert a valid **.pt** file into a **.cat** file:

1. Navigate to **AIR Tools** > **CatFileCreator** in the left-hand node toolbar,

   OR

   Press **Tab** in the Node Graph and start typing **CatFileCreator** to display a list of selectable nodes.

   A CatFileCreator node is added to the Node Graph and its **Properties** panel is displayed.

2. Enter the location of the valid **.pt** file on disk in the **Torchscript File** field or browse to the location.
3. Enter the location to save the converted **.cat** file in the **Cat File** field or browse to the location.
4. Enter a comma separated list that defines the names of the **Channels In** that your model computes. For example: `rgba.red,rgba.green,rgba.blue`

5. Enter a comma separated list that defines the names of the **Channels Out** that your model outputs. For example, if your model automates object selection you might want to output a matte into the `rgba.red` channel.

   Note: Refer to the Nuke **.cat** File Creation Guide https://learn.foundry.com/nuke/developers/17.0/catfilecreationreferenceguide/index.html for more information on supported channels.

6. Enter a user-defined **Model ID** that you can use to identify the model. The **Model ID** is encoded in the **.cat** file.

   Note: Leaving this field blank stores an empty string in the **.cat** file.

7. The **Output Scale** defines how much the width and height of the input images are scaled by the model. For example, setting the **Output Scale** to 1.5 would mean that you are expecting the model to scale up the input by 50% so that a 512x512 input image would create a 768x768 output.

   Note: This value is model-dependent.

8. If your model doesn't require any artist interaction, such as upscale or deblur, click **Create .cat file and Inference** to convert the specified **.pt** file into an Inference-readable **.cat** file,

   OR

   If your model contains variables in TorchScript that need to be controlled using knobs, such as object labels that the model has been trained to recognize, proceed to Adding User Knobs to find out how to add knobs to control the output of your model.

Adding User Knobs

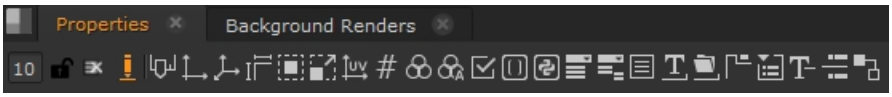
CatFileCreator allows you to add user knobs to your **.cat** files that are automatically passed to NukeX's Inference node so that you can have complete control over your pre-trained machine learning models.

Note: Any user knobs you add must be associated with variables in the TorchScript file and have the same name.

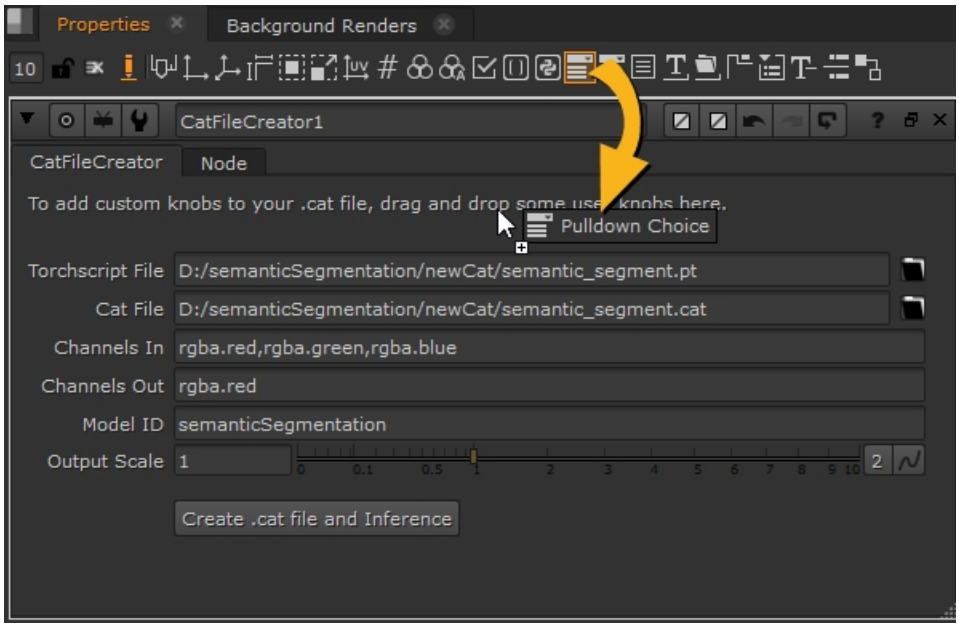
In this example, we'll be adding a dropdown to allow artists to choose from a number of objects in a semantic segmentation model, but there are several other types of user knob available. See Adding Knobs Using Drag-and-Drop for more information.

To add user knobs to CatFileCreator:

1. Click the pencil icon above the node **Properties** panel to display the available user knobs.

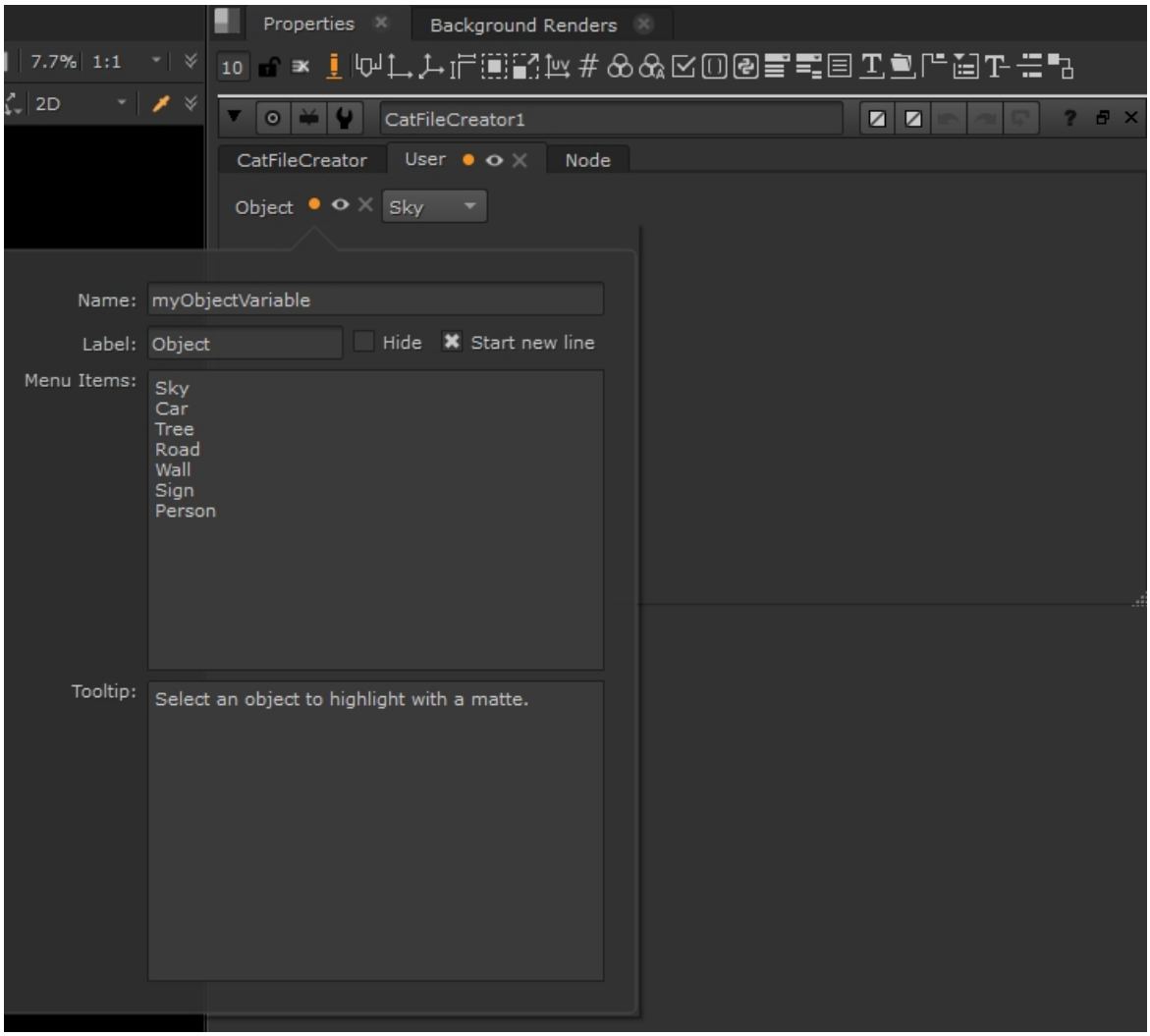


2. Drag a **Pulldown Choice** knob from the toolbar into the **Properties** panel to add it to the node.



   A new **User** tab is added and the user knob is displayed.

3. Click the orange edit button to display the knob's properties.



4. Enter the required properties:
   - **Name** - the name of the variable used in the **forward()** function in PyTorch to control the output of the model. See https://pytorch.org/docs/stable/jit.html and the Nuke **.cat** File Creation Guide https://learn.foundry.com/nuke/developers/17.0/catfilecreationreferenceguide/index.html for more details.
   - **Label** - the label displayed in the node **Properties** panel for this user knob.
   - **Menu Items** - the selectable items that are displayed in the dropdown menu.
   - **Tooltip** - an informational message to help artists by providing a little extra information when they hover over the knob.
5. Click the edit button again to close the knob properties.
6. Proceed to next section to apply the model to your source footage.

Applying the Model to Your Footage

To apply the result of your model conversion, either:

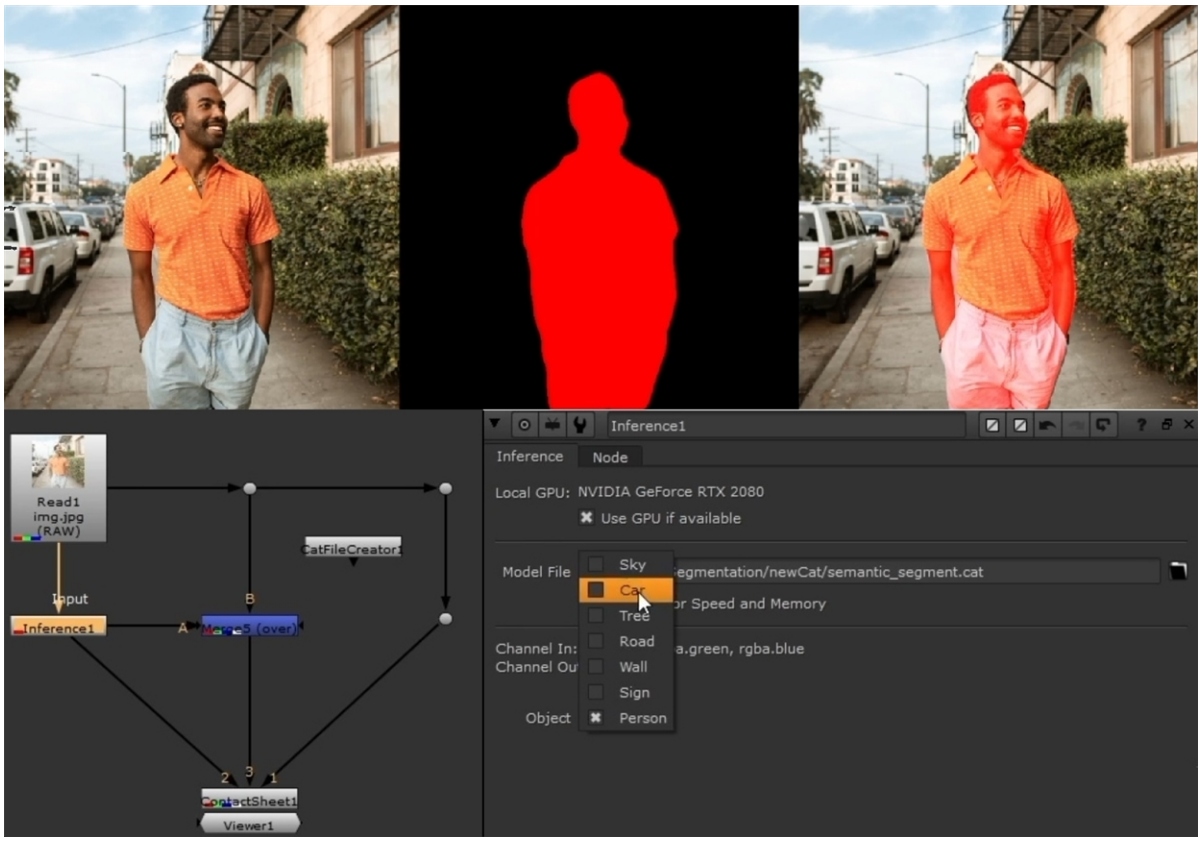
1. Click the **Create .cat file and Inference** button in CatFileCreator,

   OR

   If you have already converted a file, add an Inference node to the Node Graph and enter the location of the **.cat** file in the **Model File** field or browse to its location on disk.

2. Connect the Inference node to your input image and then select the required operation, if your model contains more than one variable.

   For example, if your model performs semantic segmentation your user knob could be named **Object**, causing your model to output a matte to `rgba.red`. You could then merge the Inference output over the original plate to highlight the selected objects in the Viewer, or create a ContactSheet node to output several variations.



Visualize Unreal Scenes in NukeX and Nuke Indie

The UnrealReader node connects NukeX or Nuke Indie to Unreal Editor using the Nuke Server plug-in. UnrealReader makes it quick and easy for you to generate live renders from Unreal Editor and control the results in Nuke, by breaking objects into layers, separating passes, building environment maps, and tweaking shot framing.

Note: UnrealReader is available with Nuke Studio, NukeX, Nuke Indie and Nuke Non-Commercial licenses. See About Nuke Indie for more details. You can also use UnrealReader in the non-commercial versions of Nuke Studio and NukeX. See About Nuke Non-commercial for more details.

All images courtesy of Epic Games Subway Sequencer project, available in Unreal Editor from the Epic Games Launcher.

UnrealReader synchronizes Nuke with a sequence in your Unreal Map and streams render passes directly into Nuke. UnrealReader connects over a TCP/IP connection, allowing Unreal Editor to be running on the same machine as Nuke or on another machine, even across different operating systems. Under the hood, UnrealReader incorporates Unreal's Movie Render Queue system to provide an interactive compositing-oriented workflow directly inside the Nuke toolset.

Install Nuke Server to Connect Unreal Editor and Nuke

To install the Nuke Server and connect Unreal Editor to Nuke, follow along with this short installation video or scroll down for step by step guidance.

1. Download the Nuke Server plug-in for your operating system from: https://www.foundry.com/products/nuke/download/unreal-nuke-server

Note: You must be using Unreal Editor version 4.27.2, 5.0.3, 5.1.0, 5.2.0, 5.3.1, 5.4.3, 5.5.3, 5.6.1, or 5.7.4. Click the link above for more information.

2. Extract the contents of the archive and copy the **Foundry** subfolder to your Unreal plug-ins location. The default Unreal Editor installation locations are:

   Windows
   `C:\Program Files\Epic Games\<version>\Engine\Plugins`

   macOS
   `Macintosh HD/Users/Shared/Epic Games/<version>/Engine/Plugins`

   OR
   `/Applications/Epic Games/<version>/Engine/Plugins`

   Linux
   `<install_directory>/<version>/Engine/Plugins`

3. Open your Unreal Editor project, then use the search functionality to find and then enable the Nuke Server plug-in under **Edit** > **Plugins**.
4. Enable the plug-in and restart Unreal Editor if required.
5. Open the Nuke Server panel in Unreal Editor from **Window** > **Nuke Server**.
6. Click **Start Server** in the **Nuke Server** panel to start the server connection to Nuke.

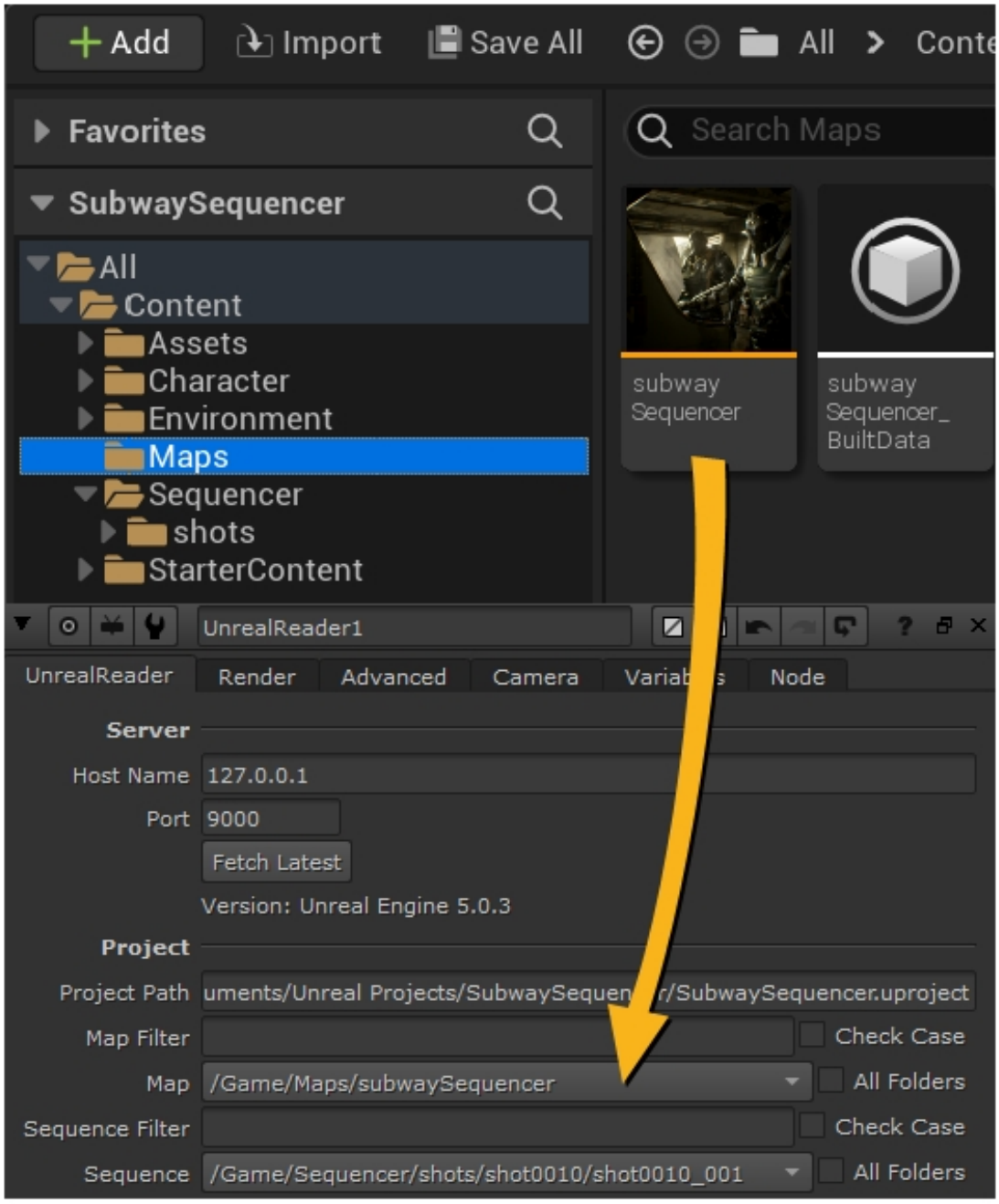
Note: You can change the port number here if required, but make sure that you use the same port number in Nuke's UnrealReader node.

7. In Nuke, create an UnrealReader node and connect it to a Viewer.
8. In the UnrealReader node **Properties** panel, click **Connect to Server** to establish the connection to Unreal Editor.

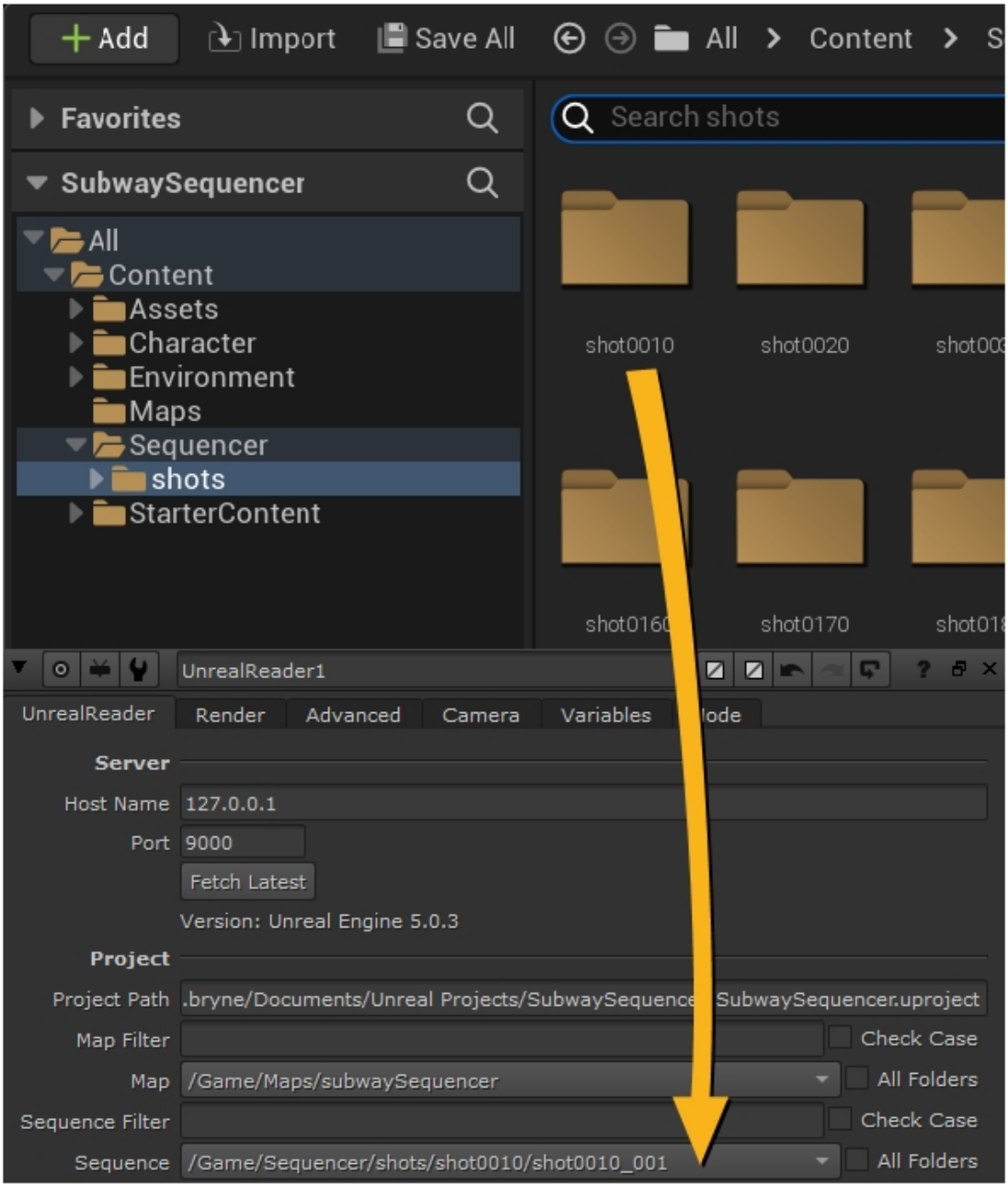
   The Unreal Editor project is now connected to Nuke.

Choose Unreal Maps and Sequences in Nuke

Nuke Server gives you the choice of which **Map** and **Sequence** to view from the live Unreal Editor session. You can select individual shot sequences or a source sequence to bring into Nuke's Viewer.



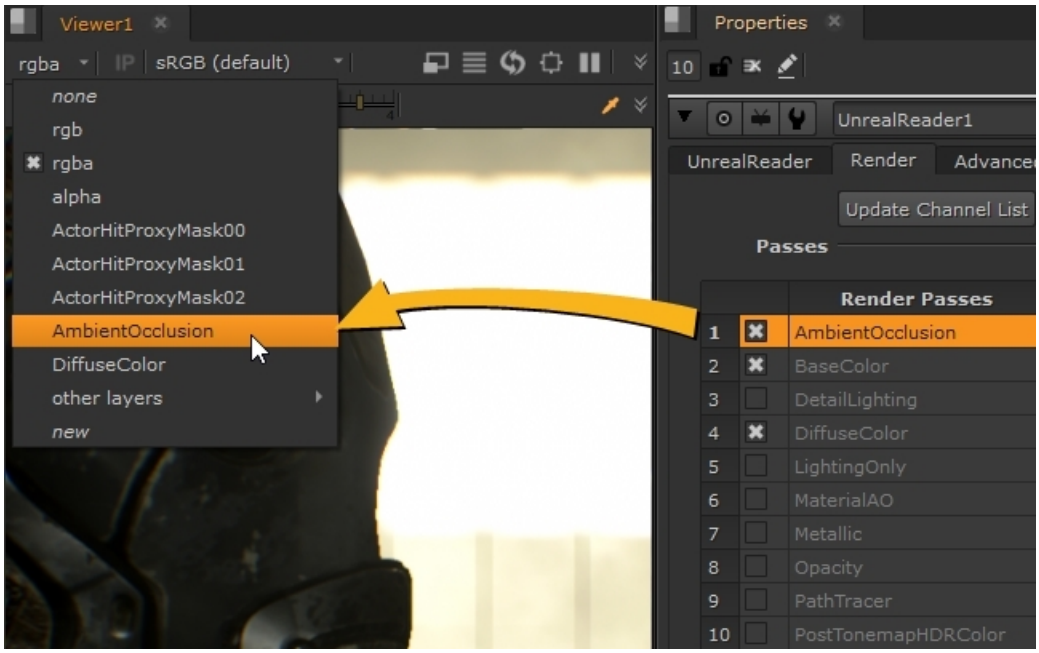
Selecting Unreal Maps in Nuke.



Selecting Unreal Sequences in Nuke.

Load Only the Render Passes You Need

In the node **Properties** panel on the **Render Passes** tab, select which passes to bring in from Unreal and then click **Update Channel List** to bring in the channels you need for your work. If you need to add or remove channels, check and uncheck the required Unreal passes and click **Update Channel List** to update the channels in the Nuke script.

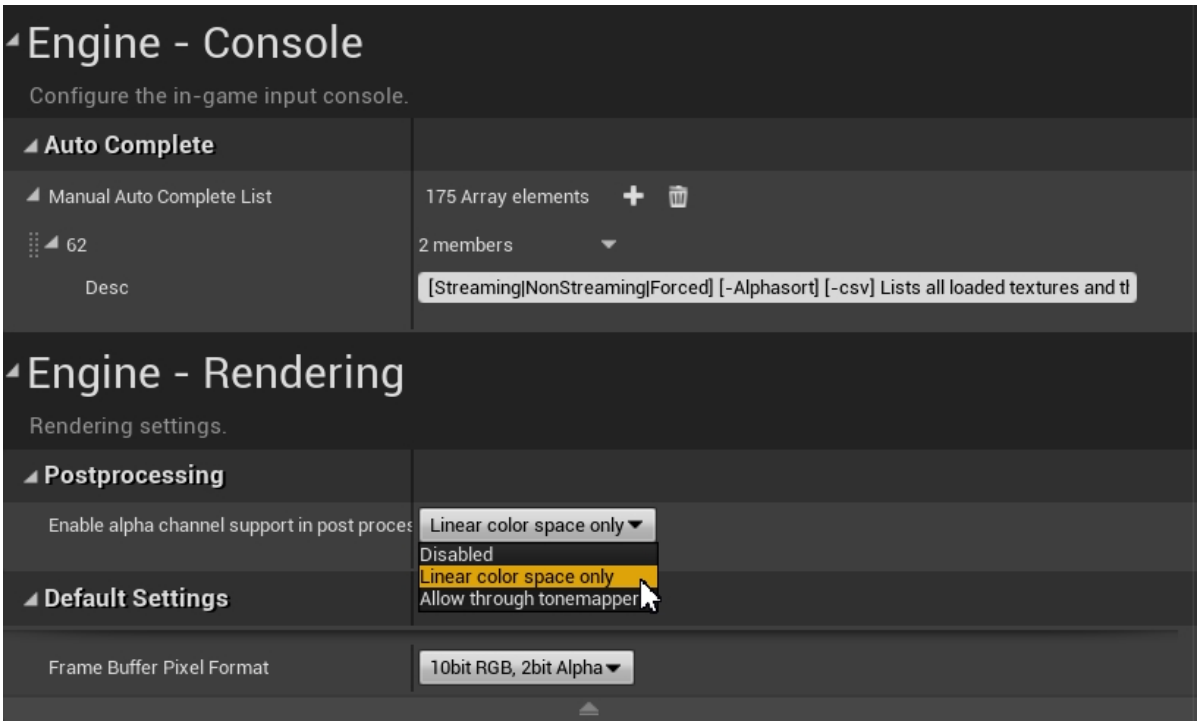


Load Stencil Layers from Unreal in Nuke

You can render individual Unreal actors in their own layers and remove occluding foreground elements and dynamic shadows, and select which render passes are generated with stencil layers. This gives you the flexibility to generate mattes for specific objects, or to layer up objects piece-by-piece, as you would in a traditional composite.

To get alpha channels that match your stencil passes, you must enable the following setting in Unreal: **Edit** > **Project Settings** > **Engine - Rendering** > **Postprocessing** > **Enable alpha channel support in post processing (experimental)** > **Linear color space only**.

Note: You must restart Unreal Editor for this option to take effect.



To view stencil layers in the Nuke script:

1. In the node **Properties** panel on the **UnrealReader** tab, set the **Render Mode** to **stencil layers**.
2. On the **Render** tab, enable **Preview Layers** to display the stencil layers in the Viewer.

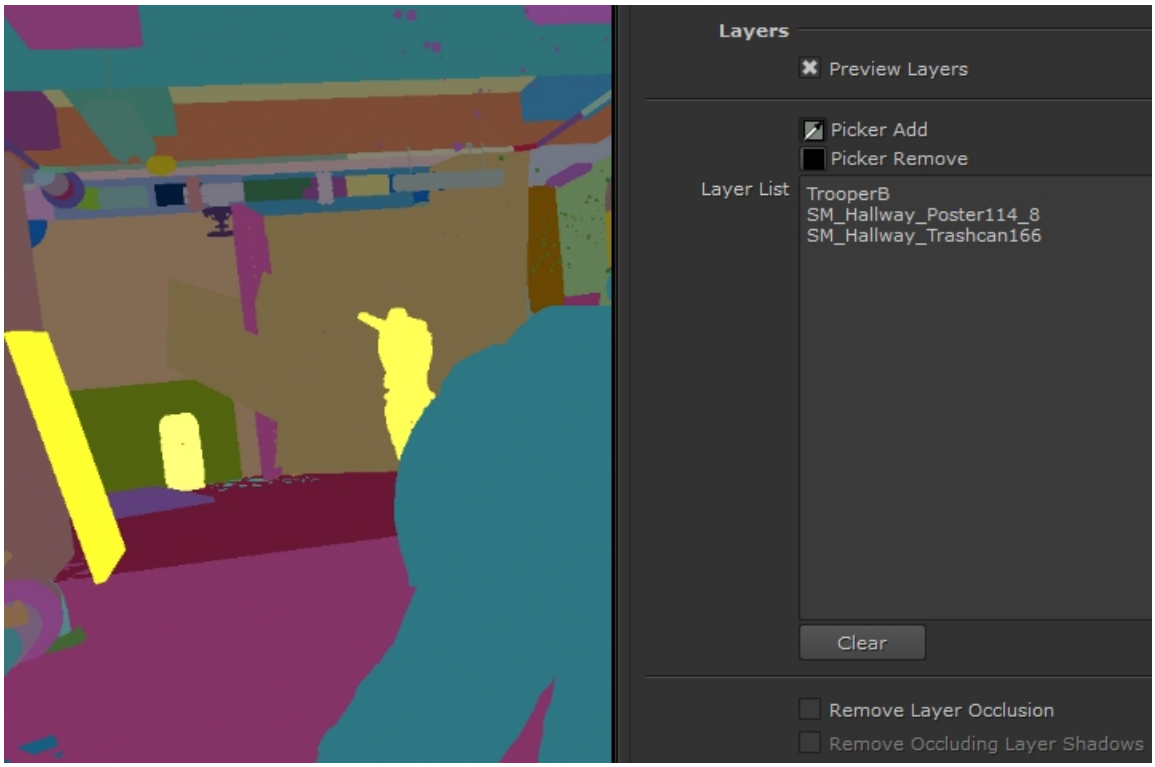


The full image



Stencil layers only

3. Use the pickers and **Layer List** to select which layers to bring in from Unreal.
4. Click the color swatch  next to **Picker Add** in the **Properties** to enable the eyedropper color picker.
5. **Ctrl/Cmd**+click a layer to add it to the selection.
   Selected layers are displayed in yellow in the **Viewer** and the layer names are added to the **Layer List**.



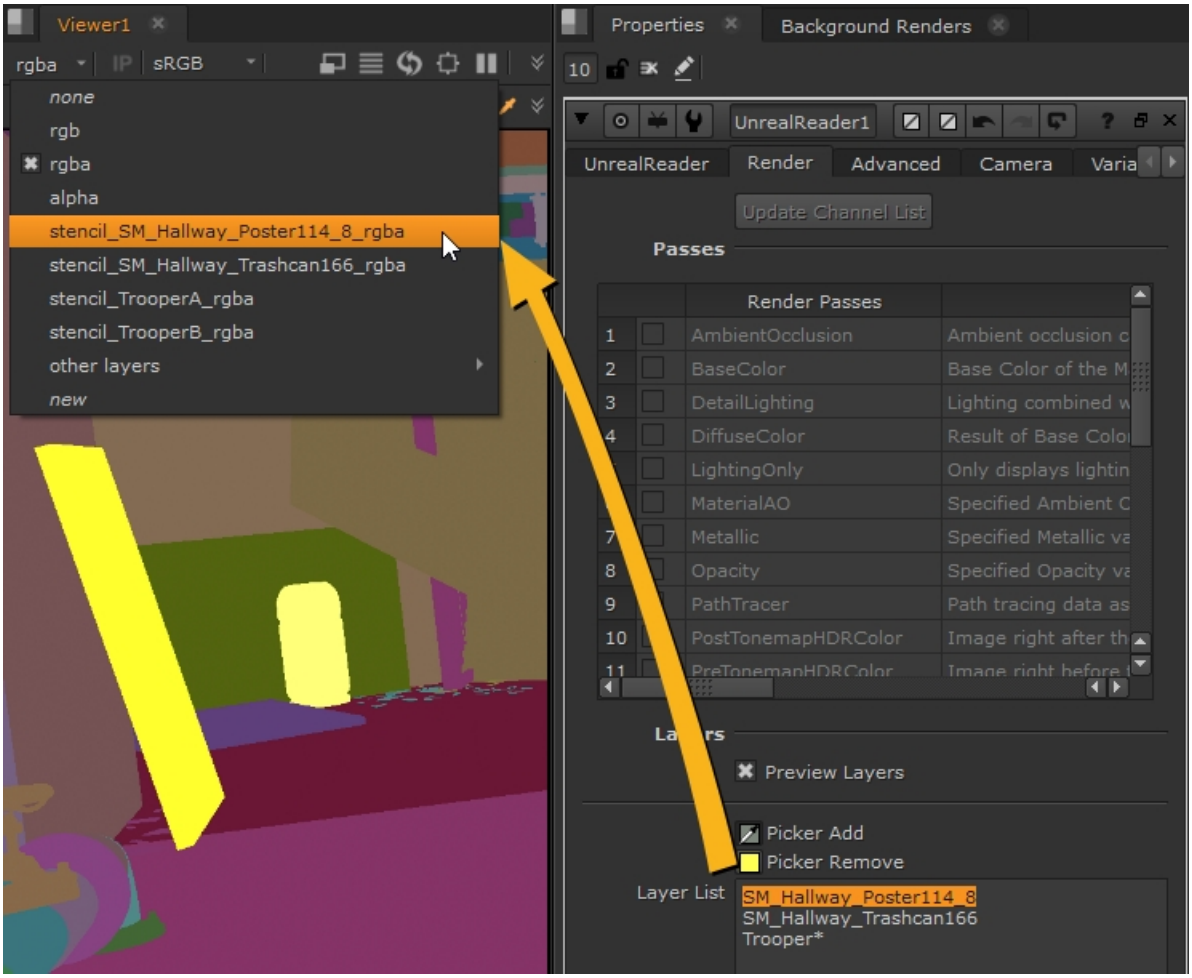
6. **Ctrl/Cmd** + **Click** to add more layers to the selection.

> **Tip:** You can marquee select multiple layers by holding **Ctrl/Cmd**+**Shift** and dragging in the Viewer.

7. When you are finished selecting layers, click the color swatch  next to **Picker Add** in the **Properties** to disable the eyedropper color picker. You can also **Ctrl/Cmd** + right click in the Viewer to disable the eyedropper color picker.

   The **Layer List** works in the same way as Cryptomatte's layer picking system. See Using the Matte List for more information.

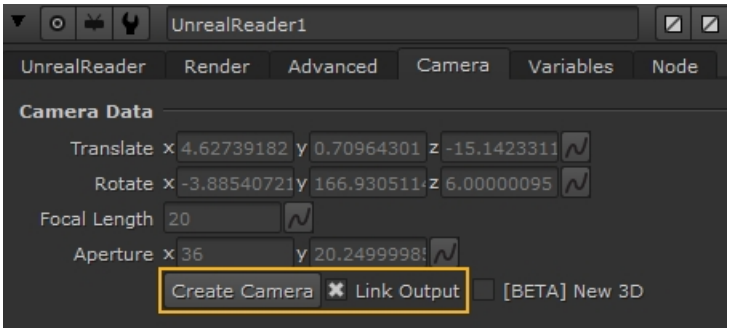
8. Click **Update Channel List** to bring in the channels you need for your work.



We have added new Cryptomatte render pass **ID Type** grouping for segmenting your Unreal scene. Among the options now available for Cryptomatte division are per **actor**, per **folder**, and per **material**.

Match and Override Unreal Cameras

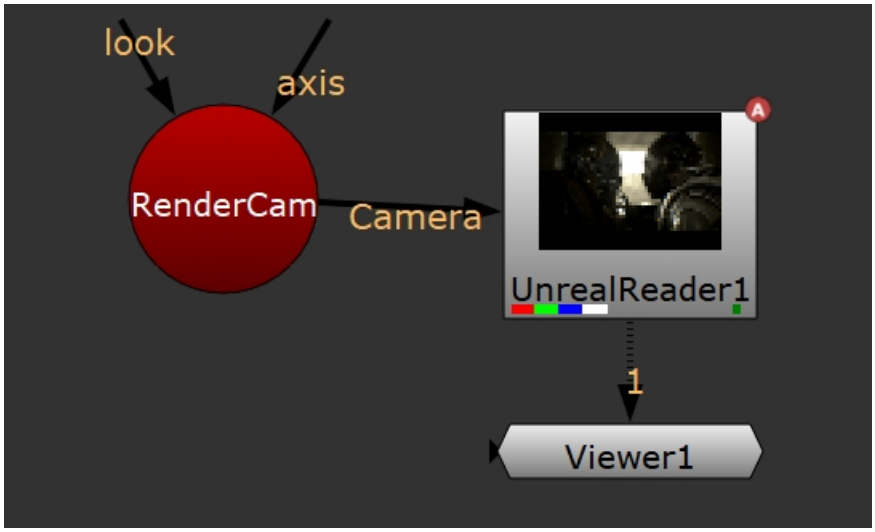
You can create a linked or unlinked camera that matches the Unreal sequence camera to integrate new elements into the scene, visualize the scene through point clouds, and enable comp of elements in 3D. In the node **Properties** panel on the **Camera** tab, click **Create Camera** to add the camera to the scene.



Linked cameras allow you to update the camera in Nuke with any changes applied in the Unreal Editor session when you click **UnrealReader** > **Fetch Latest** in the node **Properties**. Unlinked cameras do not update, they retain the camera settings as they were when the Camera node was created.

Note: UnrealReader cameras are compatible with the classic Nuke 3D system by default. If you're working in the new USD-based 3D system, enable the **New 3D** checkbox to create a camera compatible with the new system.

If you want to render from a certain angle in Nuke without updating the Unreal Editor project, you can override the Unreal camera's **translation** and **rotation** at render time, directly from Nuke, by connecting a Nuke Camera node to the UnrealReader's Camera input. This is a non-destructive override that does not persist in the Unreal Editor project file.



Render Using the Unreal Movie Render Queue from Nuke

Nuke uses the **Write** controls on the **UnrealReader** tab to access the Unreal Editor Movie Render Queue to render your shots. UnrealReader creates a Read node that references the render automatically by default, but you can disable Create Read Node to change this behavior. Files are written to a location on the machine running Unreal Editor, specified by the **File** control.

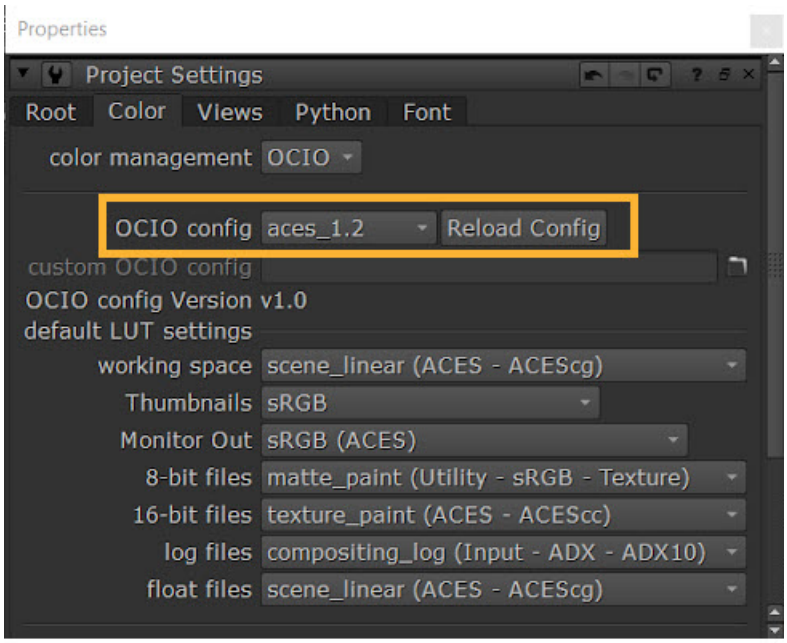
We've added environment variable support in the UnrealReader **Write** > **File** control to assist with cross-operating system support when writing to shared network drives. You can now use the familiar Nuke variable syntax as follows:

```
[getenv <your_environment_variable>]
```

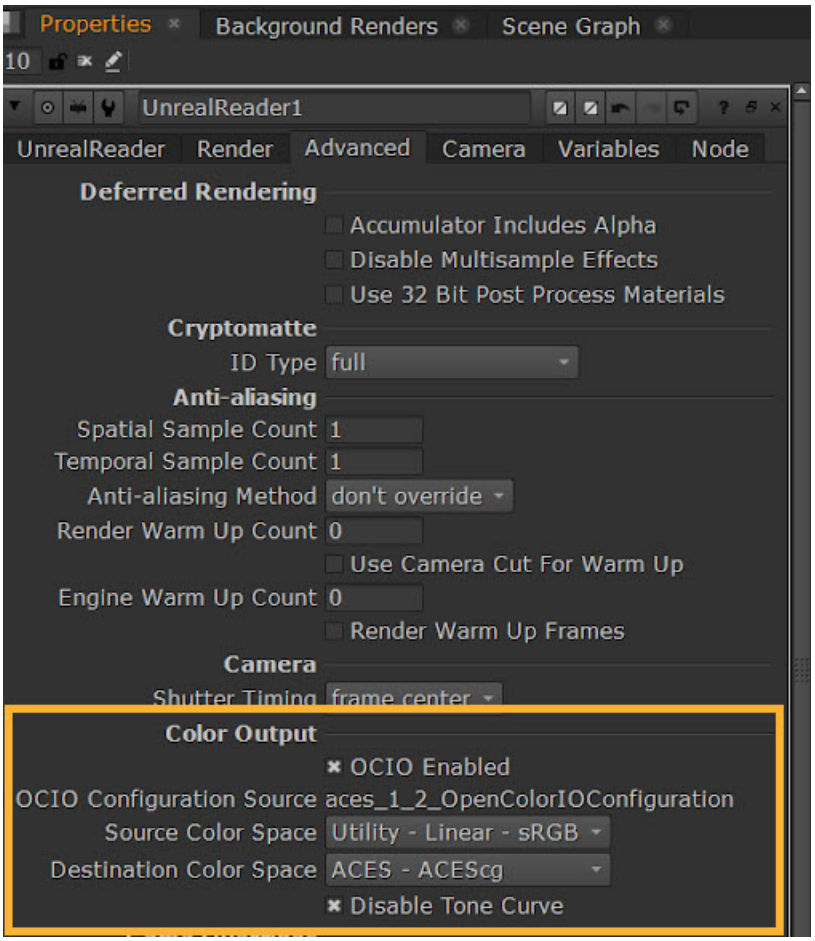
OpenColorIOConfiguration

In Unreal Editor, there is an *Open Color Configuration* section that has been added to the Nuke Server plugin panel. In Unreal Editor's *OCIOConfig* field, you can select any OpenColorIOConfiguration asset that exists within your Unreal Engine project.

To set this up in Nuke, select the same file under **Project Settings** > **Color** > **OCIO config** to give access to those same desired color spaces as selected in Unreal Editor.



Then, the file and any color space option chosen will now be available to the UnrealReader node in Nuke.



Note: We currently only support a single OpenColorIOConfiguration file being chosen to pass through to Nuke at this time.

In Nuke's **UnrealReader** node under **Advanced** tab > **Color Output** section there are options, outlined below, which match those in the standard Movie Render Queue config section for OCIO.

- **OCIO Configuration Source** - represents the *OCIOConfig* that has been chosen on the Unreal Editor side in the Nuke Server panel.
- **Source Color Space** - a drop-down menu that offers all the color spaces chosen in the *Destination Color Space* section of the OpenColorIOConfiguration file that was selected as the *OCIOConfig* in the Nuke Server panel in Unreal Editor.
- **Destination Color Space** - offers the same as the Source Color Space drop-down menu above.

In Unreal Editor, you can change the view mode. Select the *View Mode* menu on the top Viewer menu bar, and go down to the *Colour Management* > *OCIO Display* section.

- Select a OpenColorIOConfiguration asset.
- In top <*Invalid*> Available Color Spaces dropdown choose *Utility - Linear - sRGB*.

- In bottom <*Invalid*> Available Color Spaces dropdown choose *Output - sRGB*.
- Click on *Enable Display* (so a tick mark shows).



Note: If you change the *OCIOConfig* selected in the Unreal Editor's Nuke Server panel, you must click the **Fetch Latest** button in Nuke's **UnrealReader** node to update it. After fetching, you'll need to reselect your Source and Destination color space options. This change does not happen automatically on frame updates, unlike some other UnrealReader settings.

Important Information on Running Nuke Server

- UnrealReader is available with Nuke Studio, NukeX, and Nuke Indie licenses with valid maintenance. See About Nuke Indie for more details. You can also use UnrealReader in the non-commercial versions of Nuke Studio and NukeX. See About Nuke Non-commercial for more details.
- The Nuke Server plug-in supports Windows, macOS, and Linux. See the Nuke Server download page for more information.



Article: If you are running a customized build of Unreal Editor and have issues loading the Nuke Server plug-in, please see this article for more information:
https://support.foundry.com/hc/en-us/articles/4410634179858

- You can connect multiple Unreal sessions to Nuke using several UnrealReader nodes. Connecting Unreal to Nuke between two separate machines and connecting multiple Unreal sessions simultaneously has not been tested for all combinations of operating systems.
- The UnrealReader does not work when Unreal Editor is in **Play Mode**.
- To get alpha channels that match your stencil passes, you must enable the following setting in Unreal: **Project Settings** > **Engine - Rendering** > **Postprocessing** > **Enable alpha channel support in post processing (experimental)** > **Linear color space only**

Using the BlinkScript Node

The BlinkScript node runs Foundry's Blink framework enabling you to write your code once and run it on any supported device. This is achieved through code translation, in which the Blink code is turned into specific code for each target device. Code is generated and compiled on-the-fly, allowing you to switch between devices at will.

About BlinkScript

BlinkScript runs a Blink "kernel" over every pixel in the output, where a Blink kernel is similar to a C++ class, but with some special parameter types and functions. Through translation, the code in the BlinkScript node can be turned into normal C++ or SIMD code for the CPU, or OpenCL or CUDA for the GPU.

The Blink framework streamlines plug-in development workflow significantly, as you no longer have to exit Nuke to compile your code.



Note: For more information on GPU support, see the Release Notes for your version of Nuke.



Note: BlinkScript is disabled in the Non-commercial version of Nuke.

You can publish kernels in Group nodes which can then be saved as gizmos, if required. Published kernels can be encoded to protect your IP using BlinkScript's built-in kernel protection. Protected kernels are not readable when the published node is saved to a script.



Warning: BlinkScript is very flexible, as there are no restrictions on the code you can write within a kernel. As a result, code compiled from the Kernel Source can cause Nuke to crash, so please use caution!

For more information, see Nuke's **Help** menu under **Documentation > Guide to Writing Blink Kernels** or navigate to: https://learn.foundry.com/nuke/developers/17.0/BlinkUserGuide/BlinkKernelAPIReference.html

Quick Start

To get started with BlinkScript:

1. Connect the BlinkScript node to your sequence or image. See Connecting the BlinkScript Node for more information.
2. Edit the default SwirlomaticKernel, load an example kernel, or write your own from scratch. See Using and Writing Kernels for more information.
3. Set the standard Kernel Parameters, those that are present in all kernels, such as **Use GPU if available** and **format**. See Setting Kernel Parameters for more information.
4. Publish your kernel and create a gizmo, if required. See Baking and Protecting Your Kernels for more information.

Connecting the BlinkScript Node

1. Select **Other** > **BlinkScript** to create a BlinkScript node in the Node Graph.
2. Connect the **src** input to your sequence or image.



Note: BlinkScript supports as many inputs as you create in your kernel, but the default SaturationKernel only has a **src** input.

3. Connect a Viewer to the output of the BlinkScript node.

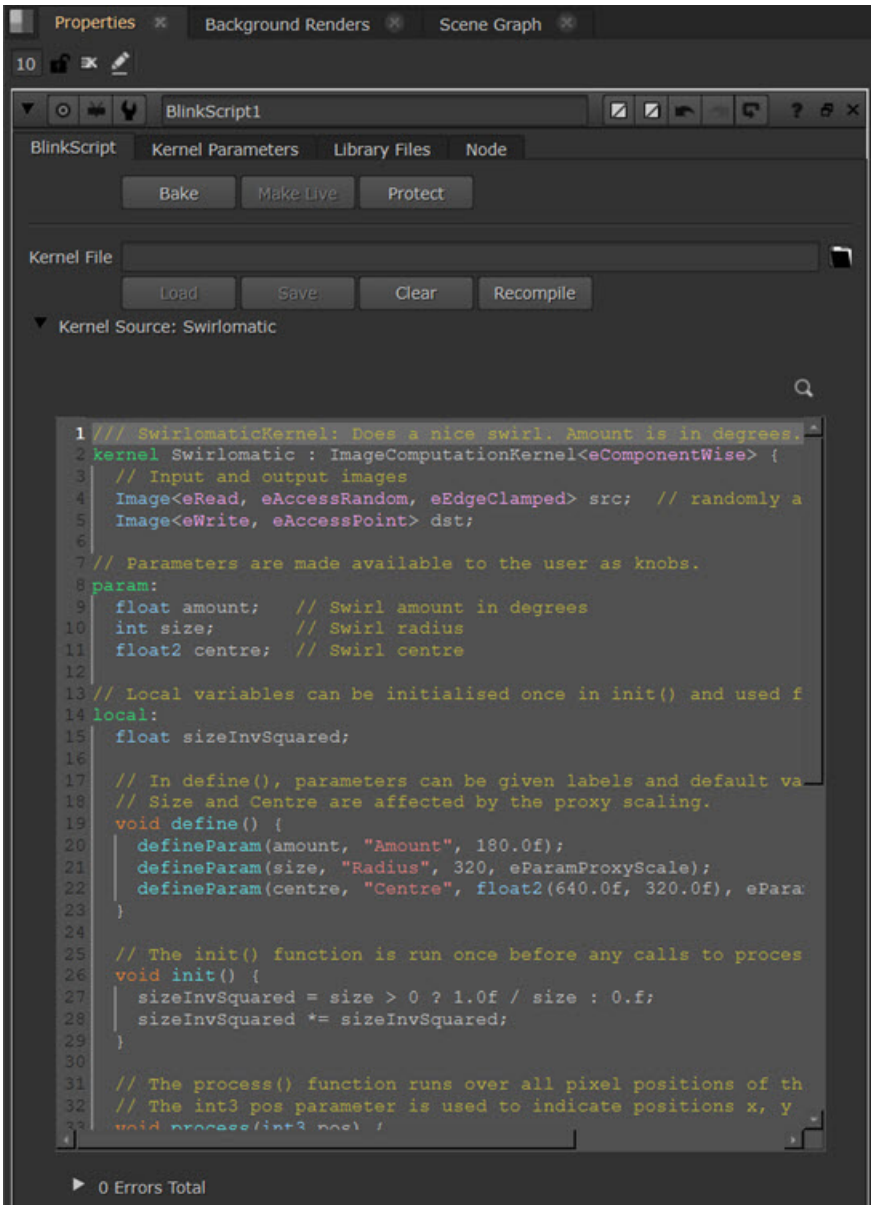
   The Viewer displays your image.

Using and Writing Kernels

Kernel management is taken care of on the first tab of the BlinkScript properties panel. Double-click the BlinkScript node in the Node Graph to display its properties.



Note: Loading, editing, and saving kernels is only available if you have a NukeX license.



Loading Kernels

1. Enter the file-path in the **Kernel File** field or click the folder icon to browse to the kernel's location.

Tip: BlinkScript kernels use the **.rpp** file extension.

2. Click **Load**.

   The selected kernel is compiled and read into the **Kernel Source** field.

Editing Kernels

You can edit existing kernels, or write your own from scratch by clicking **Clear** and entering code in the **Kernel Source** field. You won't see any results in the Viewer until you click **Recompile**.

Tip: You can change how text in the **Kernel Source** appears using the controls in the **Preferences** > **Panels** > **Script Editor** tab. After saving your preferences, close and then re-open the **Kernel Source** in the **Properties** panel to apply the changes.

The first line in a kernel is always a declaration, similar to a C++ class and derived from ImageComputationKernel, which describes a kernel used to produce an output image.



Note: BlinkScript in Nuke only works with ImageComputation kernels. It does not work with reduction or rolling kernels. Reduction and rolling kernels can, however, be used in the Blink API as part of an NDK plug-in written in C++.

In the case of the default SwirlomaticKernel:

```
kernel SwirlomaticKernel : ImageComputationKernel<eComponentWise>
```

Parameters for the kernel are declared in the `param` section, in the same way as you would declare member variables in a C++ class. If your kernel doesn't require any parameters, you can omit this section.

For example, the SwirlomaticKernel has three parameters **amount**, **size** and **centre**.

```
param:

  float amount; //Swirl amount in degrees.

  int size; //Swirl radius.

  float2 centre; //Swirl centre.
```

When a kernel is compiled inside the BlinkScript node, controls are generated for each of the kernel's parameters and added to the **Kernel Parameters** tab. In the case of SwirlomaticKernel, its three custom parameters are accessible in the **Kernel Parameters** tab.

Kernel parameters can be C++ built-in types such as **float** or **int**. Vector and matrix parameters are also supported. The following table contains the control types that the Kernel Source can expose on the **Kernel Parameters** tab in the properties panel.

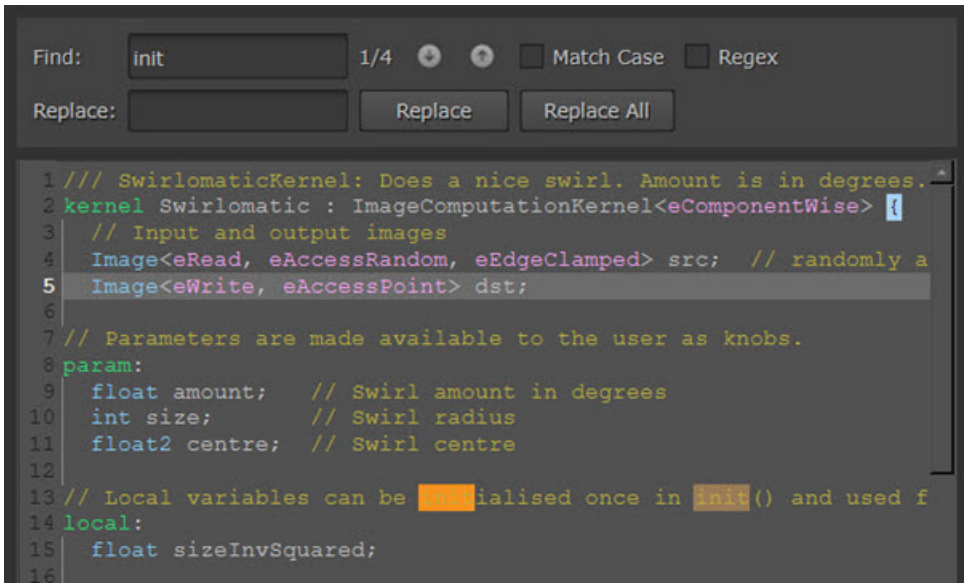
Control Type	Description
Bool_knob	Parameters of type **bool** are represented by a Bool_knob, or single checkbox.
Int_knob	Parameters of type **int** generate an Int_knob with a single numeric input box. Right-clicking in the input box displays a menu allowing you to animate the value.
MultiInt_knob	Parameters of type **int2**, **int3**, **int4**, or **int[ ]** generate a MultiInt_knob, with multiple numeric input boxes. As with the Int_knob, right-clicking in the input boxes displays a menu allowing you to animate the values.
Float_knob	Parameters of type **float** generate a Float_knob with a numeric input box, linear slider, and animation menu button. The slider ranges from 0 to twice the default value of the parameter, or 0 to 1 if no default value is set.
XY_knob	Parameters with two floating-point values, types float2 or float[2], are interpreted as positions in 2D space and generate an XY_knob with an adjustable handle in the Viewer.
XYZ_knob	Parameters with three floating-point values, types float3 or float[3], are interpreted as positions in 3D space and generate an XYZ_knob with an adjustable handle in the 3D Viewer.
AColor_knob	Parameters with four floating-point values, type float4 or float[4], are interpreted as colors with alpha, and generate an AColor_knob. This initially displays a single entry box and slider, a button to split to four entry boxes, a button to display the Nuke color wheel, and a swatch showing the current color with an eyedropper for sampling in the Viewer.
Array_knob	Parameters with nine floating-point values, including float3x3, are displayed as an Array_knob with a 3x3 grid. Parameters with sixteen floating-point values, including float4x4, are displayed as an Array_knob with a 4x4 grid.
MultiFloat_knob	Parameters with numbers of floating-point values not already listed above, such as float[5], generate a MultiFloat_knob with a numerical entry box for each value, a single linear slider, and an animation menu button. The slider ranges from 0 to 1.

The BlinkScript Editor in the Kernel Source field has a variety of features to help with writing and editing code.

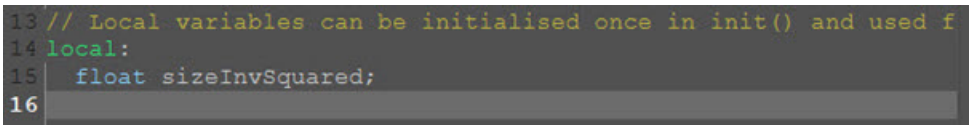
It may also be helpful to consult the Blink Developer Guide. There are also several example kernels to get you started.

Editing Features

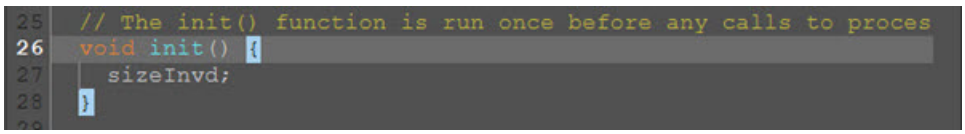
- **Find and Replace** - by clicking the magnifying glass icon or **CTRL+F**, you can find and replace a particular string or regular expression in your code.



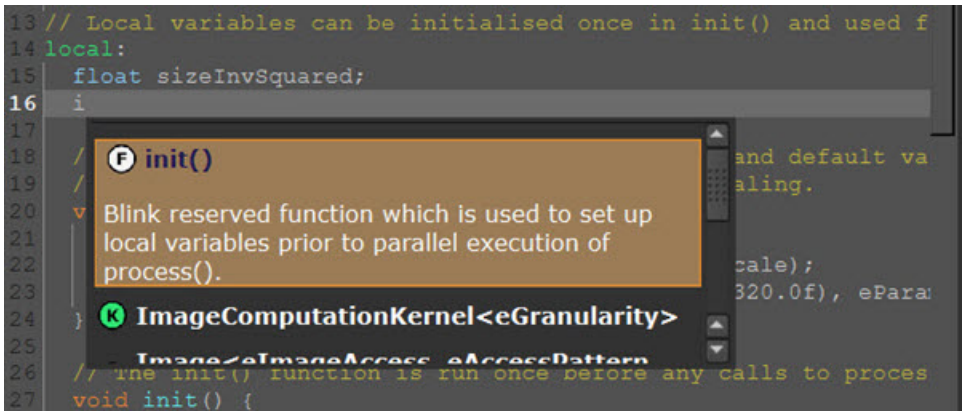
- **Indentation** - the editor will auto indent new lines and show indentation guide line overlays.



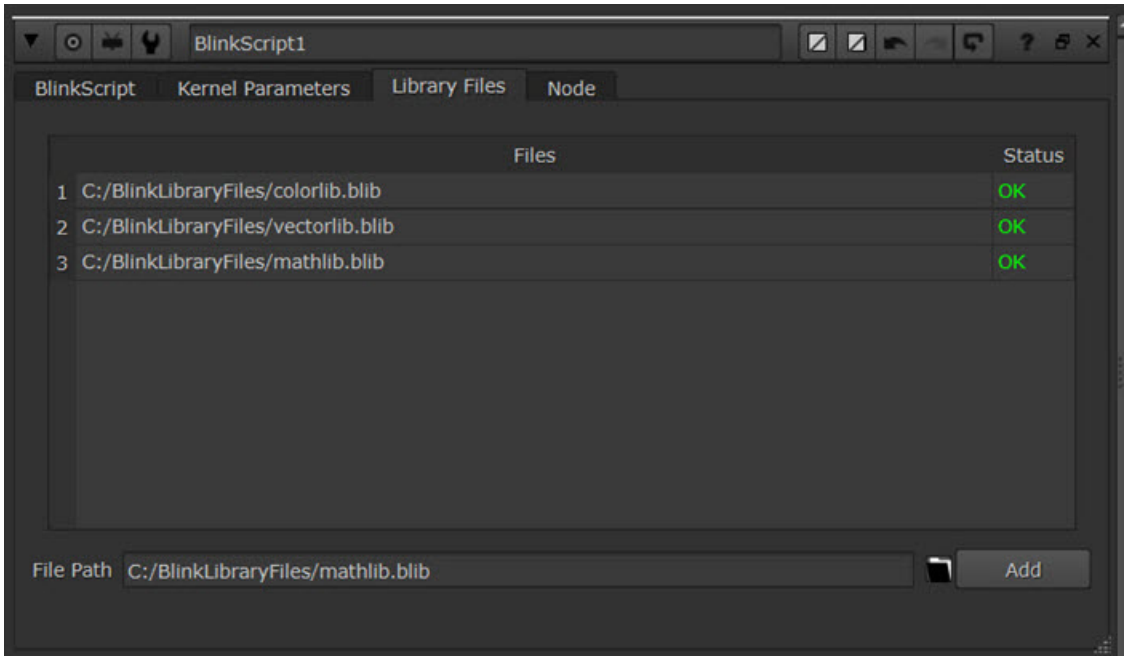
- **Closing brackets** - the editor will auto close brackets when an open bracket is typed.
- **Dynamic highlighting** - the editor will auto highlight the brackets of the current function.



- **Tab Menu** - when you begin to type a string or function, you can hit **Tab** to see a list of possible options or functions. When clicked into, the item will give a more detailed description of the current keyword or argument.



- **Library Files** - add common functions and code snippets from an external file through the Library Files tab. You can then reuse these functions without having to rewrite them - just call the function name in the kernel. Read the Loading Library Files page for more information about this feature.



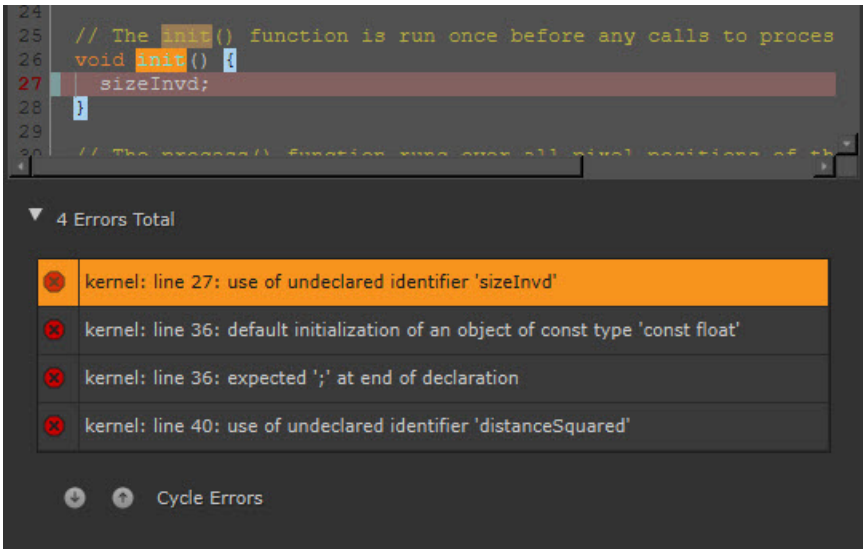
Note: Note: Documentation for Library Files functions can be added to the tab menu of the BlinkScript editor. Type with a '///' comment before the function to add documentation.

- **Keyboard Shortcuts**-

Keyboard Shortcut	Function
cmd/ctrl+F	Find and replace.

Keyboard Shortcut	Function
tab	Trigger tab menu.
cmd/ctrl+Enter	Recompile BlinkScript.
cmd/ctrl+/	Create multiple comments.
cmd/ctrl+tab **OR** cmd+shift+tab	Tabbing multiple lines.

- **Error Accordion** - all error messages are displayed directly in the editor when the kernel is recompiled, and can be cycled through. Any Library File errors will also appear here.



Saving Kernels

1. Enter the file path in the **Kernel File** field or click the folder icon to browse to the intended location.

Tip: BlinkScript kernels use the **.rpp** file extension.

2. Click **Save**.

   The selected kernel is saved to the specified location. There is no compile step during saving as the kernel is compiled when loaded.

Setting Kernel Parameters

Some BlinkScript controls are used by all kernels. Rather than write these individually in each instance, the BlinkScript properties panel includes the **Kernel Parameters** tab, providing easy access to the standard kernel controls.

The **Kernel Parameters** tab also contains any controls exposed by specific kernel commands. For example, the default **SwirlomaticKernel** exposes the **Swirlomatic_Amount** control. See Editing Kernels for more information on exposing controls.

The **Bake** button also resides on the **Kernel Parameters** tab, but we'll get into that in more detail in Baking and Protecting Your Kernels.

GPU or CPU?

The read-only Local GPU field displays the graphics card installed locally, if there is one available. You can toggle the GPU on and off using the **Use GPU if available** control. If you have multiple GPUs available, select the required device using the **Preferences** > **Performance** > **Hardware** > **default blink device** dropdown.



Note: Selecting a different GPU requires you to restart Nuke before the change takes effect.

Tip: Even if there is no GPU available locally, you can still enable **Use GPU if available**. BlinkScript then uses the GPU as and when one becomes available.
You should also enable this if you wish to render from the command line with the **--gpu** option.

Nuke's GPU support includes an **Enable multi-GPU support** option. When enabled in the preferences, GPU processing is shared between the available GPUs for extra processing speed.



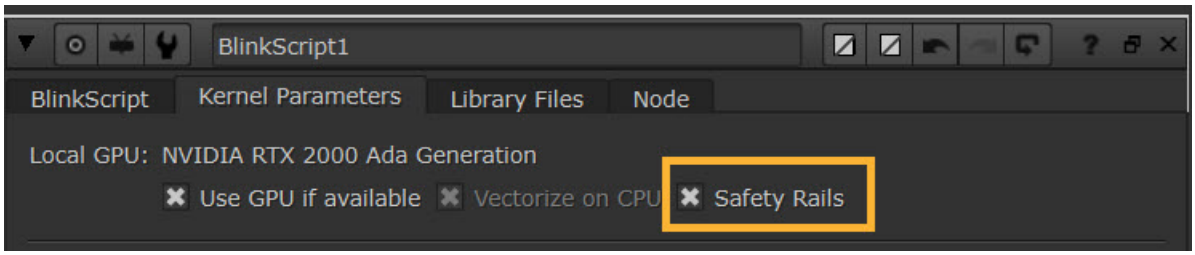
Note: For more information on GPU support, see the Release Notes for your version of Nuke.

Additionally, enable **Vectorize on CPU** to use SIMD acceleration on the CPU where possible. See the Blink Developer Guide for more information on kernel execution.

Safety Rails

Turn on **Safety Rails** in the **Kernel Parameters** tab to catch common errors when prototyping BlinkScript kernels. When this option is enabled, additional code is generated to check for, catch and safely handle out-of-bounds memory accesses, which are a common cause of BlinkScript crashes.

When you recompile the BlinkScript Kernel, invalid pixel accesses are printed to the Nuke terminal and any errors caught are displayed in the error accordion in the BlinkScript editor.



Warning: When disabled, the code is generated to maximize performance without the Safety Rails preventing any crashes.

It is recommended to enable Safety Rails during development and to disable once complete.

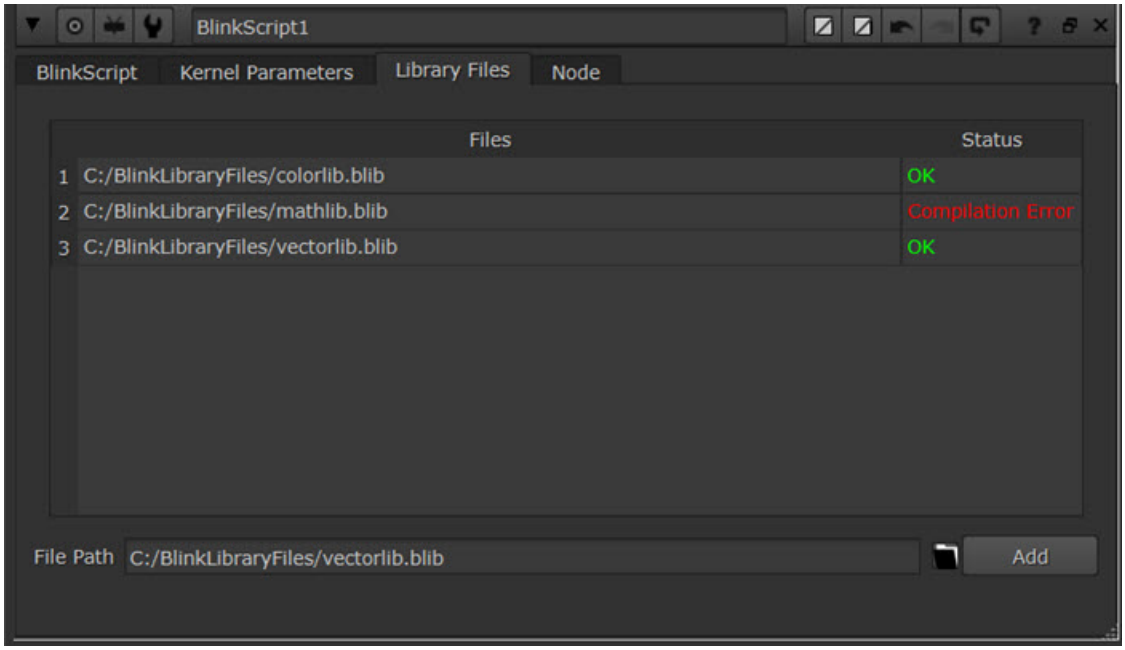
Specifying the Output Format

BlinkScript's output defaults to the union of all its inputs, but you can specify a different output format by enabling **Specify output format** and selecting the required ratio from the **format** dropdown.

Tip: Enabling **Specify output format** also adds the format control to the kernel once it has been published. See Baking and Protecting Your Kernels for more information.

Loading Library Files

In BlinkScript, Library Files enable you to share common functions and code snippets. This means you can reuse common functions across multiple kernels and projects without having to rewrite them.



To access Library Files in your BlinkScript Node:

1. In BlinkScript Node Properties, click on the **Library Files** tab.
2. Click the folder icon to choose the **.blib** file to import.
3. Click **Add**.

   The .blib file (and location) will now appear in the kernel's loaded Library Files. This list also contains information on the file status:

   - **OK** - the file is found and loaded.
   - **File Not Found** - no file at the specified path.
   - **Compilation Error** - the file contains compilation errors.
4. Call these functions in your main kernel by writing the function in your kernel source (similar to calling any inbuilt function).

Note: Documentation for Library Files functions can be added to the tab menu of the BlinkScript editor. Type with a '///' comment before the function to add documentation.

Note: Errors in Library Files are caught by the BlinkScript Editor's error handling.

Baking and Protecting Your Kernels

When you're happy with your kernel, you can bake it for use elsewhere in the script or in an entirely separate Nuke script. Baked kernels are locked, with the Library Files baked into the kernel.

Blink kernels can also be encoded using the built-in kernel protection. The Protect feature creates a new version of the node, with only the Kernel Parameters tab visible. Protected kernels are not readable when the published node is saved to a script.

Note: Kernel protection offers a fairly low level of security and might not be sufficient to protect valuable source code.

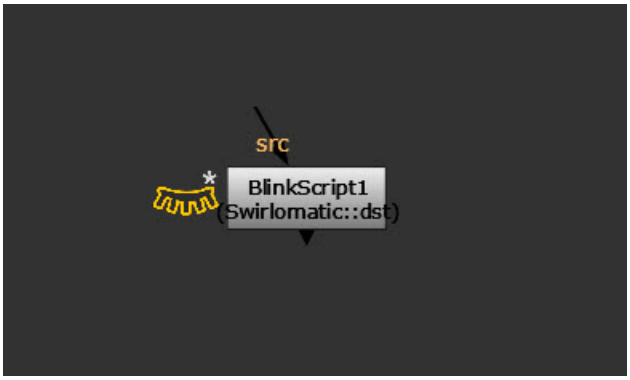
To **bake** a kernel:

1. Double-click the BlinkScript node to open the properties panel.
2. In the BlinkScript tab, click **Bake**.

The BlinkScript kernel is now locked and has the 'baked' icon next to it. Any library files included in the kernel are saved within the node.



To access and edit the kernel of a baked Blinkscript node by clicking **Make Live** in the **BlinkScript** tab. The contents of the kernel can be edited and the live icon will appear next to the node.



When a kernel is made live, you can either save the node's baked Library Files to disk, or discard them and default back to the path listed in the kernel.

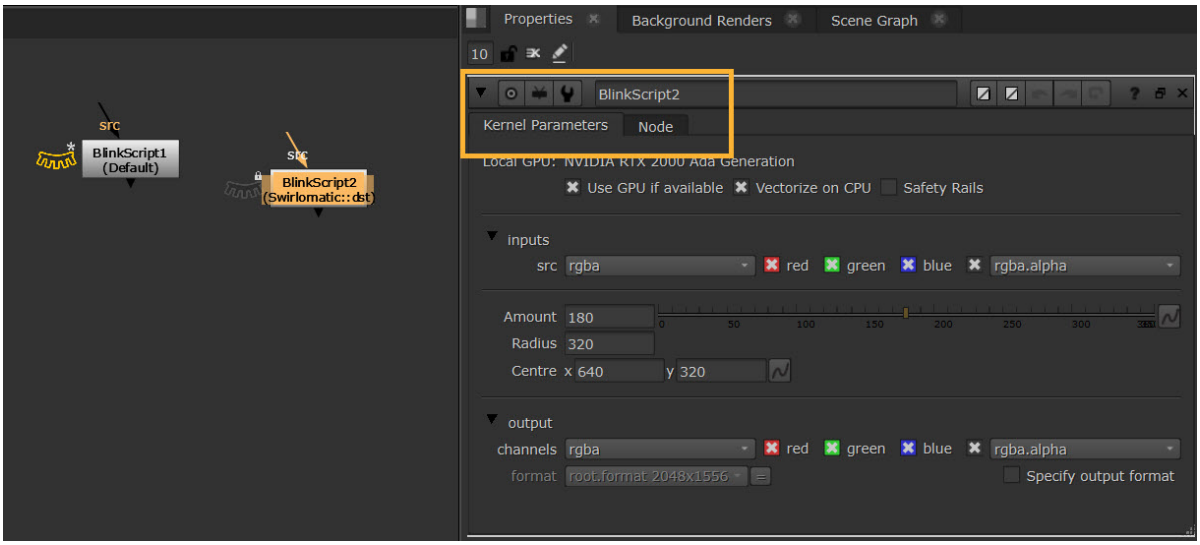
Warning: Warning: this discard operation is destructive.

The **Protect** feature can be applied to either a Live or Baked node.

To **protect** a kernel:

1. Double-click the BlinkScript node to open the Properties panel.
2. In the BlinkScript tab, click **Protect**.

The protected state creates a new version of the node with only the kernel parameters tab. For example, using the Swirlomatic Kernel supplied in the documentation produces the following copied node.



Note: Note: If a live BlinkScript node is protected, the Library Files will not be baked into the node.

In a Nuke Script, protected kernels appear as shown in the example below.



Certain BlinkScript functions are only available with a NukeX license. The following table describes the various levels of access for the Nuke family.

Product	Feature		
	Create node	Load, edit, and save	Adjust controls
Nuke Assist	●		
Nuke	●		●
NukeX	●	●	●
Nuke Studio	●	●	●

Limitations and Known Issues

GPU

Complicated kernels can cause driver timeouts on the GPU if they take too long to execute. The lower the specification of your GPU, the more likely this is to happen.

Note: For more information on GPU support, see the Release Notes for your version of Nuke.

Vectorization

Vectorization will attempt to optimize Blink kernels on the CPU with SIMD instructions. This may result in noticeably longer compile times for very large kernels, and there is no guarantee that your code will be vectorizable.

Crashing and Infinite Loops

Since the BlinkScript node allows arbitrary code to be written and run, it is possible to either crash or lock up Nuke. The BlinkScript node does not check your code to make sure it is sensible before running it, so we advise you to use caution!

To prevent crashes caused by out-of-bound memory accesses, turn on the Safety Rails option in the Kernel Parameters. When this option is enabled, additional code is generated to check for, catch and safely handle these out-of-bounds accesses, which are a common cause of BlinkScript crashes.

When you recompile the BlinkScript Kernel, full error messaging is logged in the Nuke terminal and the BlinkScript Editor Error Accordion to assist with debugging. See Editing Kernels for more information.

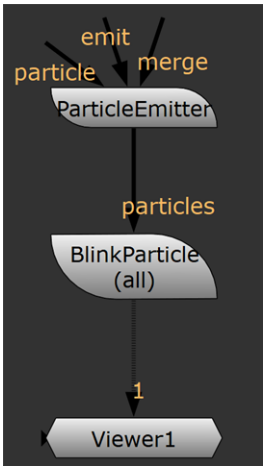
Tip: Tip: It is recommended to enable Safety Rails during development and to disable once complete.

Using the ParticleBlinkScript Node

The ParticleBlinkScript node is similar to the BlinkScript node, but instead of working on images, it works on a particle system. It is a particle node which lets you write a Blink kernel to modify the particles, allowing custom behavior that can't be achieved with Nuke's built in particle nodes.

To use the ParticleBlinkScript node, connect it to a particle system in the same way as any other particle node.

Note: For a full breakdown of the ParticleBlinkScript node Properties, see ParticleBlinkScript in the Reference Guide.



A simple example of a ParticleBlinkScript node graph

In the ParticleBlinkScript **Properties** tab, in the **script** section, write a Blink kernel to modify the particle attributes. Declare each attribute you want to read or write as an Image. The node binds each Image you declare in your kernel to the particle attribute of the same name.

For example, if you want to modify particle positions, declare an Image called p_position. All particle image names must start with the prefix p_.

Note: The prefix is removed before matching with an attribute name. The reason for the prefix is that there are other types of special image which can be declared (see below).

Blink Kernel Example

Nuke has a ParticleDrag op which multiplies the velocity of each particle by a drag coefficient. The following is an example Blink kernel which does the same.

```
kernel ParticleDragKernel :
ImageComputationKernel<ePixelWise>
{

 // Declare images for the particle attributes we're
 interested in
```

```
  Image<eRead> p_startTime;

  Image<eReadWrite> p_velocity;

  param:

  // Declare parameters
    float _drag; // This will be a knob for the drag
    factor
    float _dt; // The time step for the kernel
    float _endTime; // The particle system time at the
    end of the step

  void define() {
    defineParam(_drag, "pa_drag", 1.0f);
    defineParam(_dt, "_dt", 1.0f);
    defineParam(_endTime, "_endTime", 1.0f);

  }

  void process() {
    // Work out the time step for this particle
    float dt = max(0.0f, min(_dt, float(_endTime - p_
    startTime())));
    // As this is a drag force, it's more accurate to
    give it exponential decay over dt. For other
    forces, we may just want to multiply by dt.
    float t = pow(1.0f-_drag, dt);
    p_velocity() *= t;

  }
};
```

An Image is declared for the particle velocity attribute, p_velocity. In this case, the Image is going to be read and written so it must be declared as eReadWrite. In this kernel, the Image has a width of 1, and a height of the number of particles in the particle system.



Note: Often it is only necessary for an attribute to be read, in which case it can be declared as eRead.

There is a single parameter, pa_drag, and the velocity is multiplied by the drag in the process() method. Nuke calls the process method once for every particle.

Note: The prefix pa_ added to the drag parameter name is not essential, but is good practice as it prevents name clashes when ParticleBlinkScript creates knobs for the kernel parameters.

For more information, see Nuke's **Help** menu under **Documentation** > **Guide to Writing Blink Kernels** or navigate to: https://learn.foundry.com/nuke/developers/17.0/BlinkUserGuide/BlinkKernelAPIReference.html

Warning: ParticleBlinkScript is very flexible, as there are no restrictions on the code you can write within a kernel. As a result, code compiled from the Kernel Source can cause Nuke to crash, so please use caution!

ParticleBlinkScript Attributes and Parameters

The following particle attributes are available within a ParticleBlinkScript kernel:

float3 p_position	The current position of the particles.
float3 p_initialPosition	The starting position of the particles.
float3 p_lastPosition	The previous position of the particles.
float3 p_velocity	The velocity of the particles.
float4 p_color	The color of the particles.
float3 p_size	The size of the particles.
float4 p_orientation	The orientation of the particles.
float3 p_rotationAxis	The rotation axis of the particles.
float p_rotationAngle	The rotation angle of the particles.
float p_rotationVelocity	The rotation velocity of the particles.
float p_mass	The mass of the particles.

float p_life	The lifespan of the particles.
float p_startTime	The start time of the particles.

All the particle attribute arrays are the size of the number of particles in the system, so you can pass as many as you want to a kernel and it will iterate over all of them simultaneously. This allows you to write kernels which can, for example, change particle color depending on velocity and position, or write new forces.

Creating Knobs in ParticleBlinkScript

Parameters can also be declared in a kernel. When using the BlinkScript node, parameter knobs are automatically created. The ParticleBlinkScript node works differently. The disadvantage of automatically creating the knobs is that you can't choose what type of knob to create for a given parameter and you can't set ranges for sliders or tooltips. A **float3** attribute could require a **Color_knob**, or a **XYZ_knob**. Automatically generating the knobs also means that you have no control over the layout in the control panel.

ParticleBlinkScript instead uses **User Knobs** and binds them by matching their names to the parameters. If your kernel has a **float4** parameter called **pa_color**, then an RGBA color User Knob called **pa_color** must also be created which will bind to the **float4** parameter.

The ParticleBlinkScript method may take a little more time but is far more controllable.



Note: If you don't create a User Knob for a parameter, it is set to the default value. The **Create Knobs** button creates knobs for any parameters which don't already have one. These can be removed, replaced and edited as much as needed.

Magic Attributes

In the kernel, it may be useful to have some extra information, such as the current frame. ParticleBlinkScript provides this if the attributes are defined with certain magic names. These are:

float _frame	The current frame Nuke is rendering.
float _systemTime	The current time of the particle system.
float _dt	The time delta for this frame.

float _startTime	The beginning of the time-step.
float _endTime	The end of the time-step.

All of these start with an underscore _ to distinguish them from **p_** kernel attributes. The most common ones to use are **_startTime**, **_endTime** and **_dt**.

These attributes can be used together to get various information, for example:

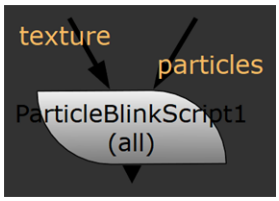
_systemTime-p_startTime()	The age of a particle.
(_systemTime-p_startTime())/life()	The age of a particle as a proportion of its life.
float _dt = max(0.0f, min(_dt, float(_endTime - p_startTime())));	The correct time delta for applying forces.

Note: For more information on writing kernels for the ParticleBlinkScript node, refer to the Nuke Developer guide.

Image Inputs

As well as particle inputs, image inputs can be provided to a ParticleBlinkScript node which can be connected to nodes producing images. These must be declared with a name starting with **image_**, and with **eAccessRandom**, otherwise they are iterated in the iteration space of the particles.

For example, declaring an image called **image_texture** creates a new input called **texture**.



Example ParticleBlinkScript node with extra image input

The extra image inputs can be used to pass more data to the kernel, such as flow maps or height maps, and can be used for various purposes such as coloring particles by their position in the scene.

Example ParticleBlinkScript Kernels

These kernel examples demonstrate various ways in which the ParticleBlinkScript node can be used.

Coloring Particles

This example kernel is for changing particle color over the course of their lifetime.

Constraining Particles

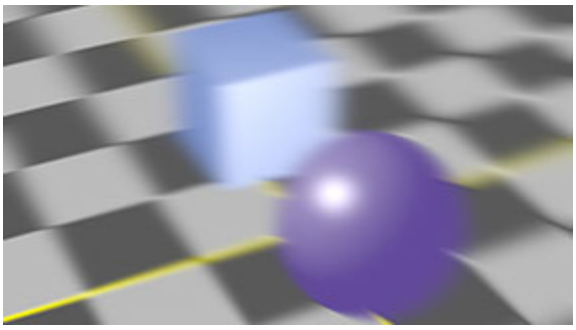
An example kernel demonstrating how the ParticleBlinkScript node can be used to constrain particles to the surface of a sphere.

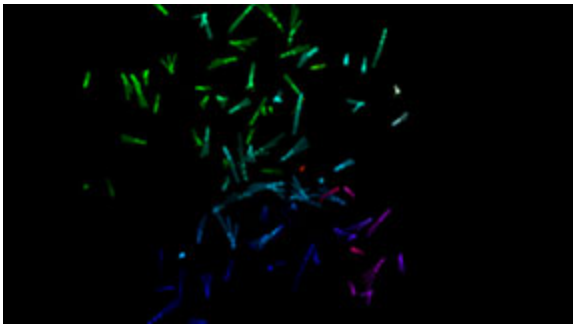
Flowing Particles

An example kernel for making particles flow around, or avoid, an object by creating a vector field.

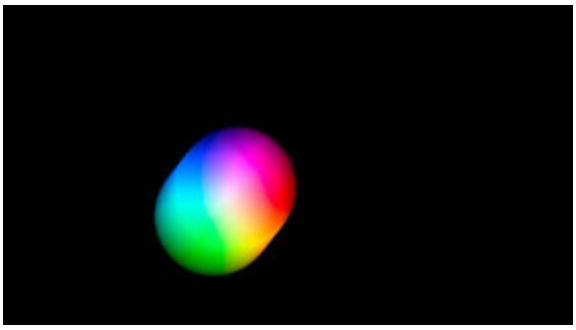
Image Projection

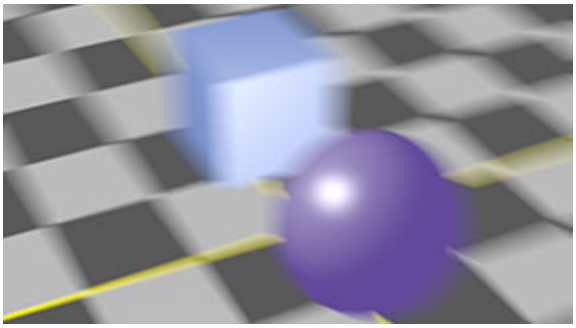
This kernel example projects an image onto the particle system and colors the particles according to their position.

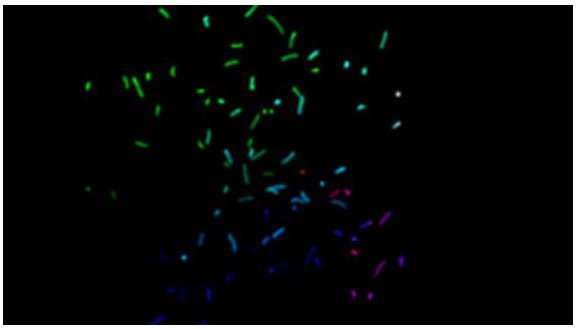




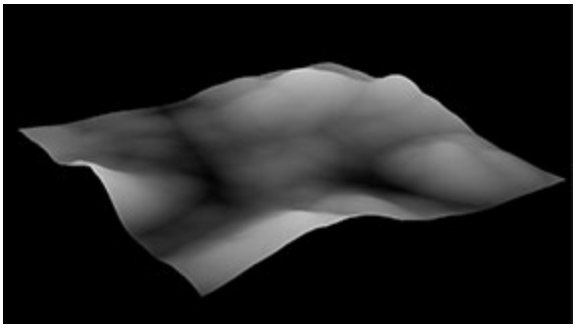
Nuke 9.0 or Earlier







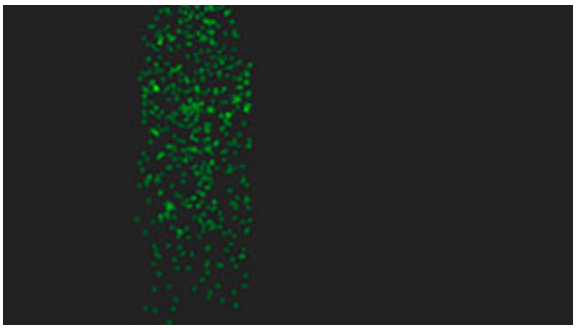
3D Compositing



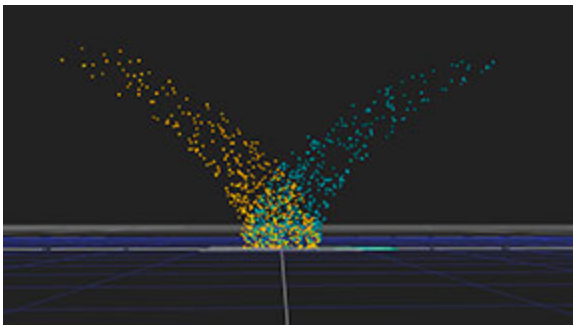
Generating 3D Particles

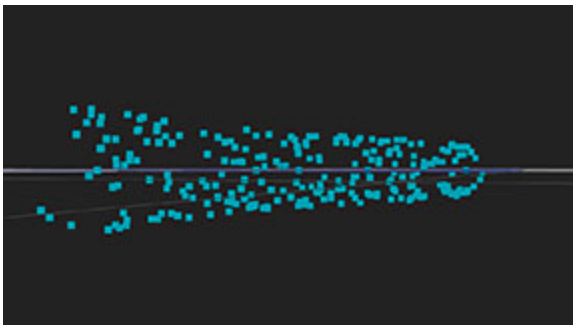
Using particles to create raindrops (Click to Download)

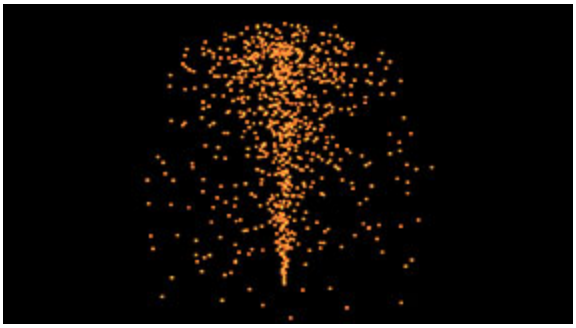


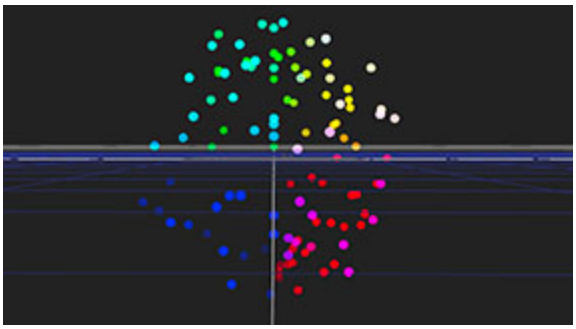


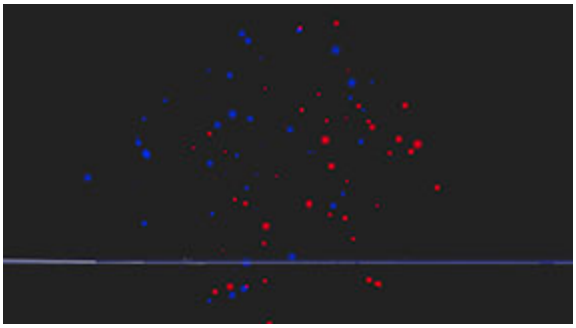


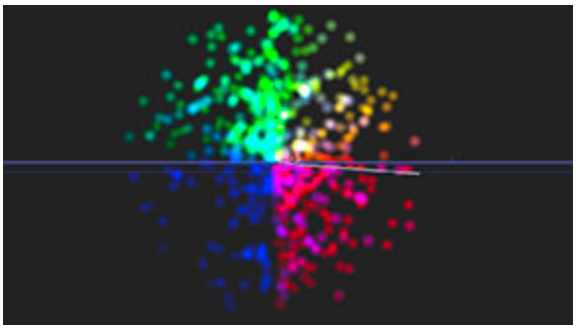


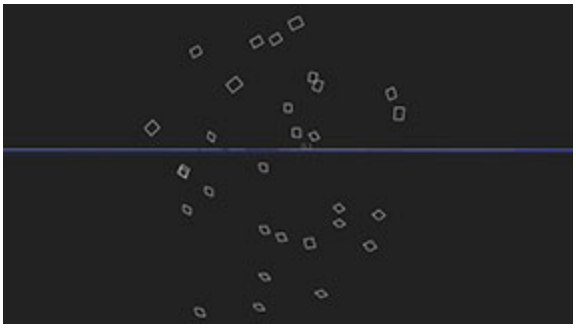


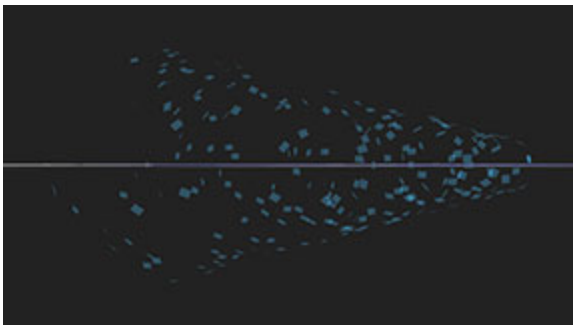


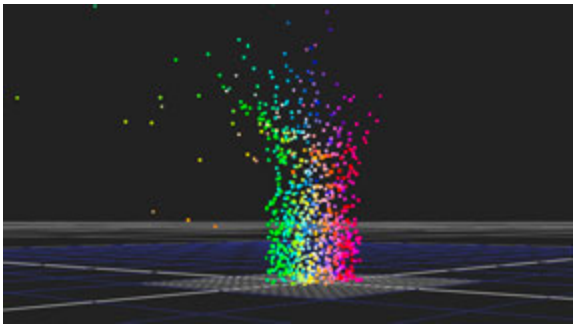




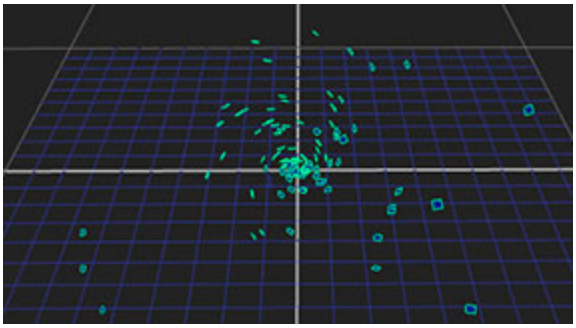


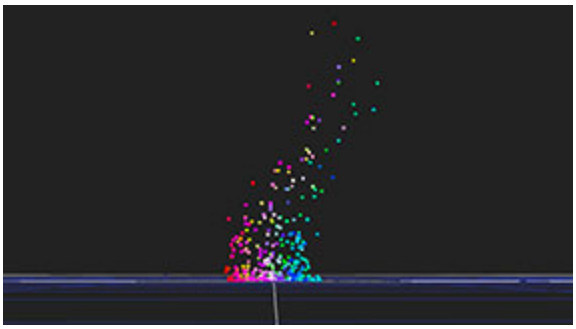


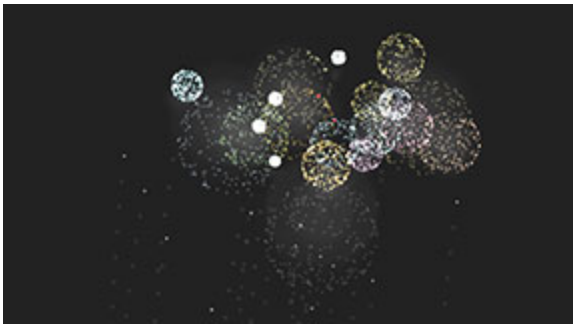


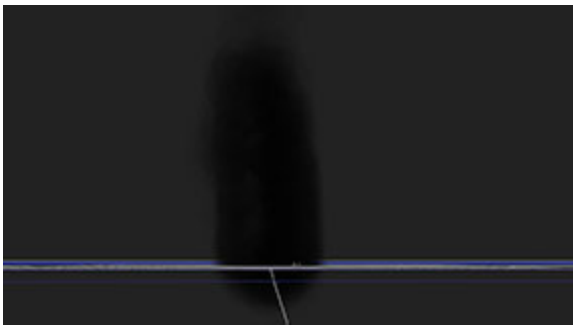


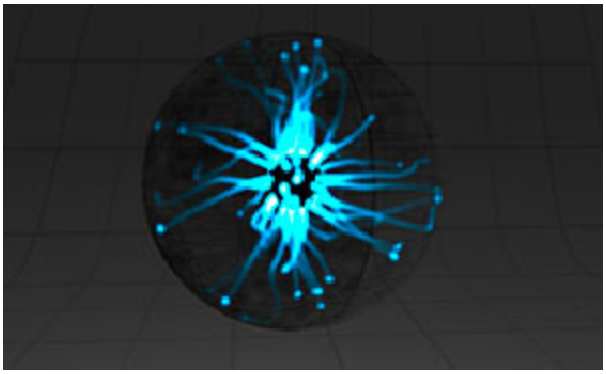






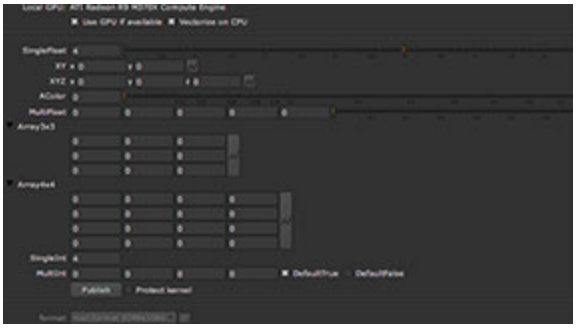


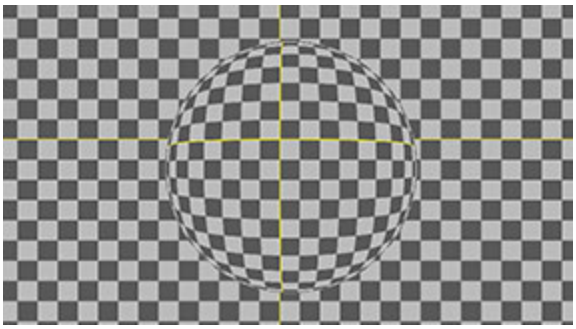






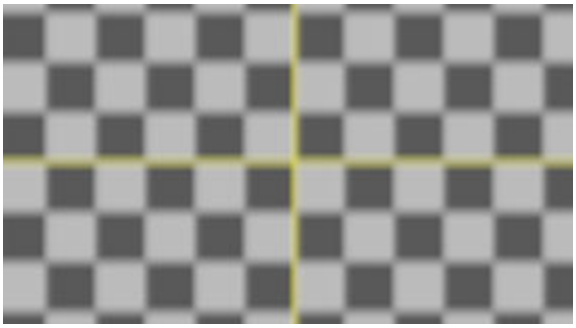
BlinkScript

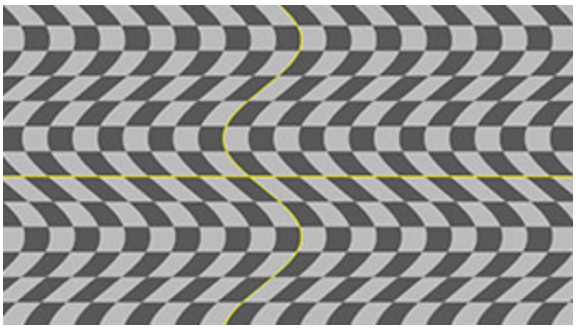


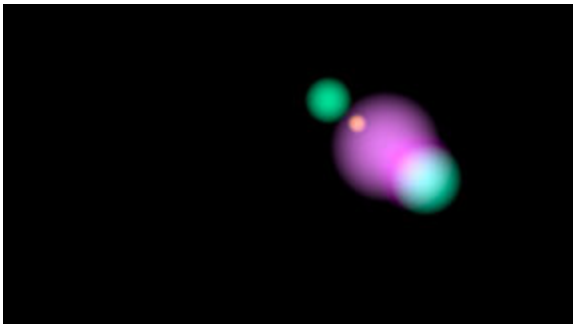


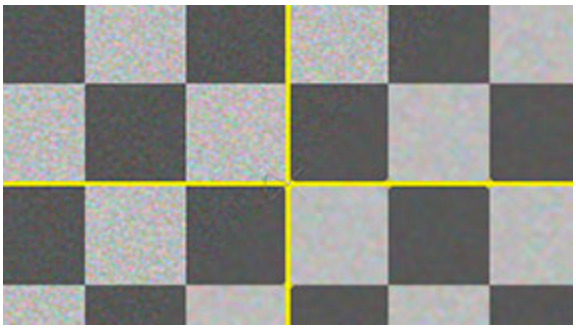


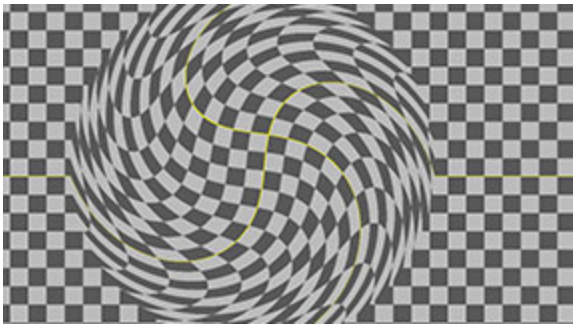




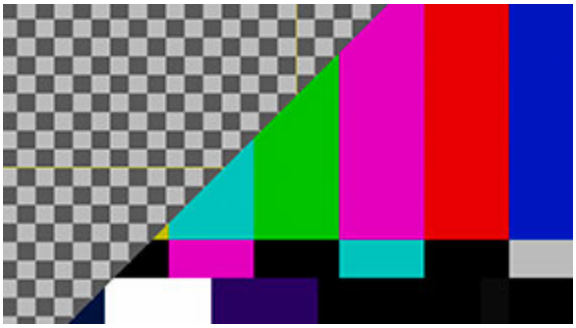




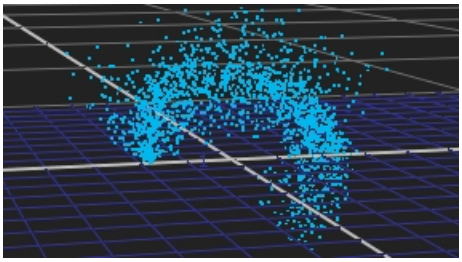


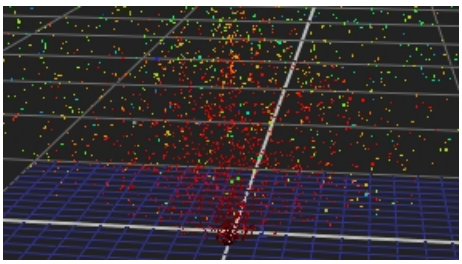


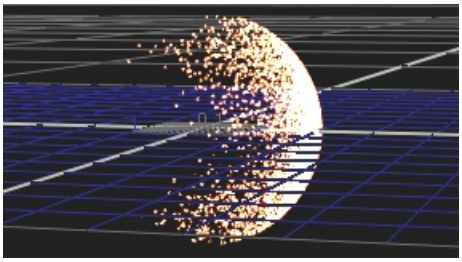


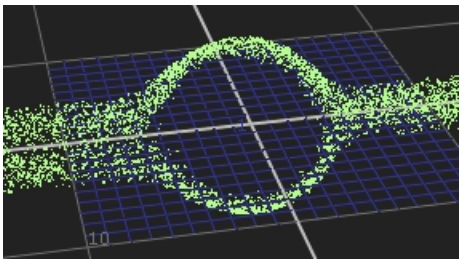


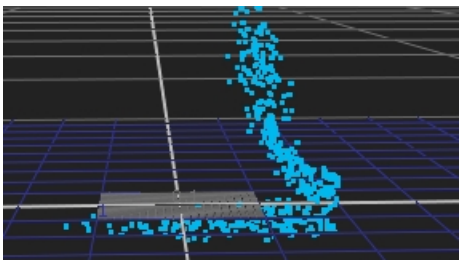
ParticleBlinkScript

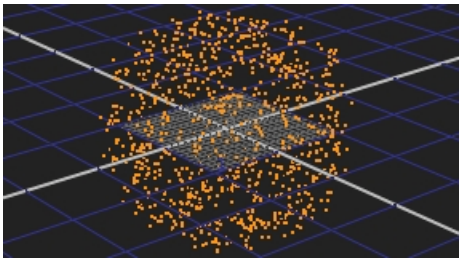


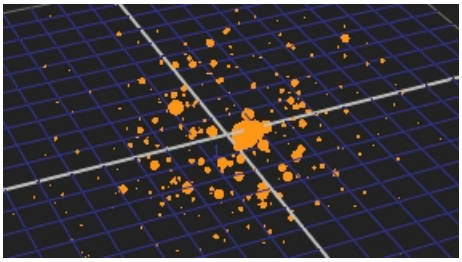


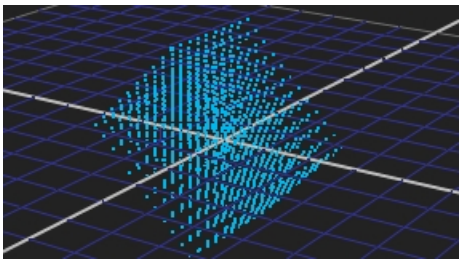


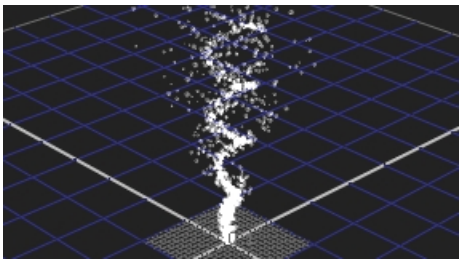


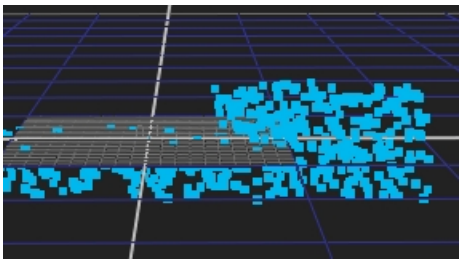


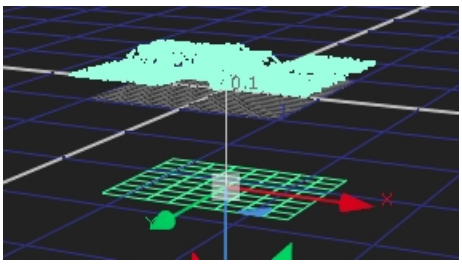


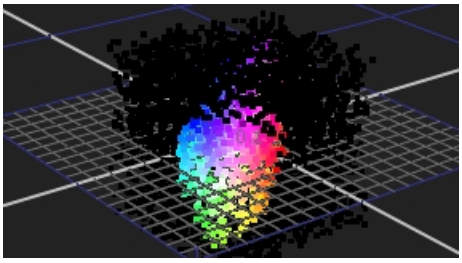


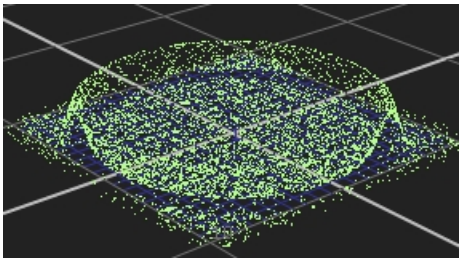






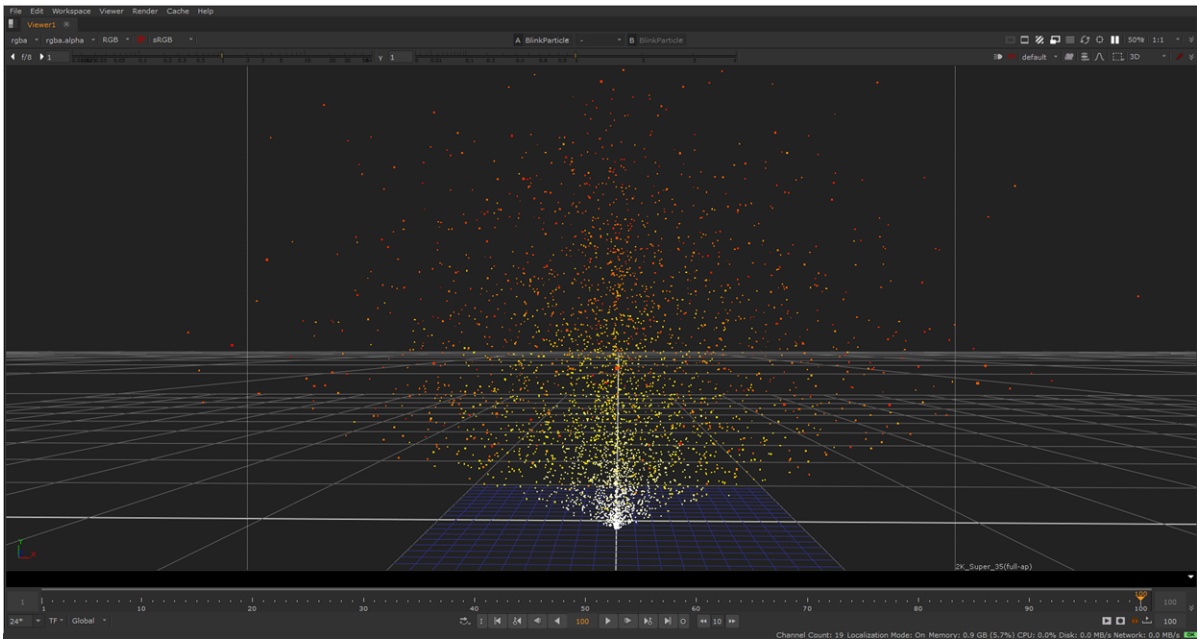




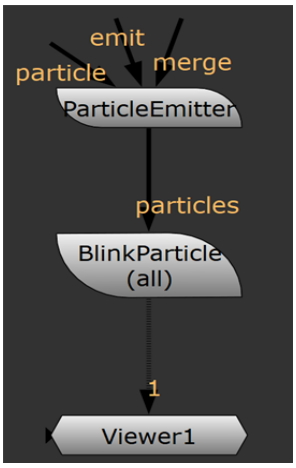


Coloring Particles

The ParticleBlinkScript node can be used to change the color of particles according to things like their age or velocity. The following example kernel changes the color of particles over their lifetime. Individual knobs for four colors are created and a Catmull-Rom spline is used to interpolate smoothly between the colors as the particles age.



The resulting effect of the following particle coloring kernel



Example node graph for the following kernel

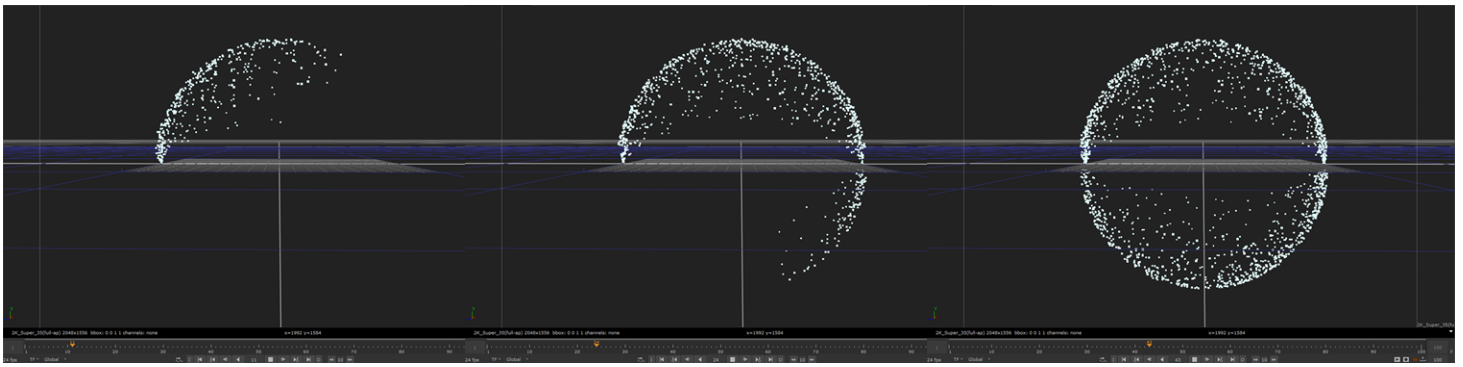
```
kernel ParticleColorKernel : ImageComputationKernel<ePixelWise>
{
 Image<eReadWrite> p_color;
 Image<eReadWrite> p_startTime;
 Image<eReadWrite> p_life;
```

```
param:
  float4 _color1;
  float4 _color2;
  float4 _color3;
  float4 _color4;
  float _systemTime;


local:
  float4 _colors[6];


void define() {
  defineParam(_color1, "pa_color1", float4(1.0f, 0.0f, 0.0f,
  1.0f));
  defineParam(_color2, "pa_color2", float4(1.0f, 0.0f, 0.0f,
  1.0f));
  defineParam(_color3, "pa_color3", float4(1.0f, 0.0f, 0.0f,
  1.0f));
  defineParam(_color4, "pa_color4", float4(1.0f, 0.0f, 0.0f,
  1.0f));
  defineParam(_systemTime, "_systemTime", 0.0f);
}


void init() {
  // Repeat the first and last color for the spline tangents
  _colors[0] = _color1;
  _colors[1] = _color1;
  _colors[2] = _color2;
  _colors[3] = _color3;
  _colors[4] = _color4;
  _colors[5] = _color4;
}


// Catmull-Rom spline
```

```
float4 spline(float x, int numKnots)
{
  int numSpans = numKnots - 1;

  float t = max(0.0f, min(1.0f, x)) * (numSpans-2);
  int span = int(floor(t));
  t -= span;

  float4 k0 = _colors[span];
  float4 k1 = _colors[span+1];
  float4 k2 = _colors[int(min(numSpans, span+2))];
  float4 k3 = _colors[int(min(numSpans, span+3))];

  return 0.5f*(2.0f*k1
   + (-k0 + k2)*t
   + (2.0f*k0 - 5.0f*k1 + 4.0f*k2 - k3)*t*t
   + (-k0 + 3.0f*k1 - 3.0f*k2 + k3)*t*t*t);
}

void process() {
  float t = (_systemTime-p_startTime(0))/p_life(0);
  p_color() = spline(t, 6);
}
};
```

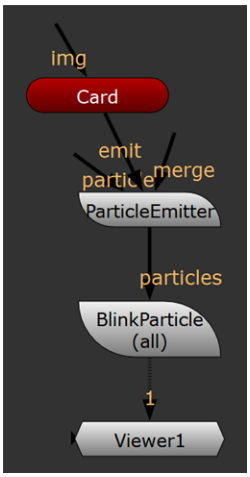
Constraining Particles

The ParticleBlinkScript node can be used to constrain particle positions. As an example, this kernel constrains particles to the surface of a sphere. Each particle is moved outwards or inwards to the radius of the sphere and its velocity rotated to point tangent to the sphere's surface.



The resulting effect of the following particle constraining kernel

Note: A more realistic version of this kernel would involve the gradual moving of the particles, rather than doing it instantaneously.



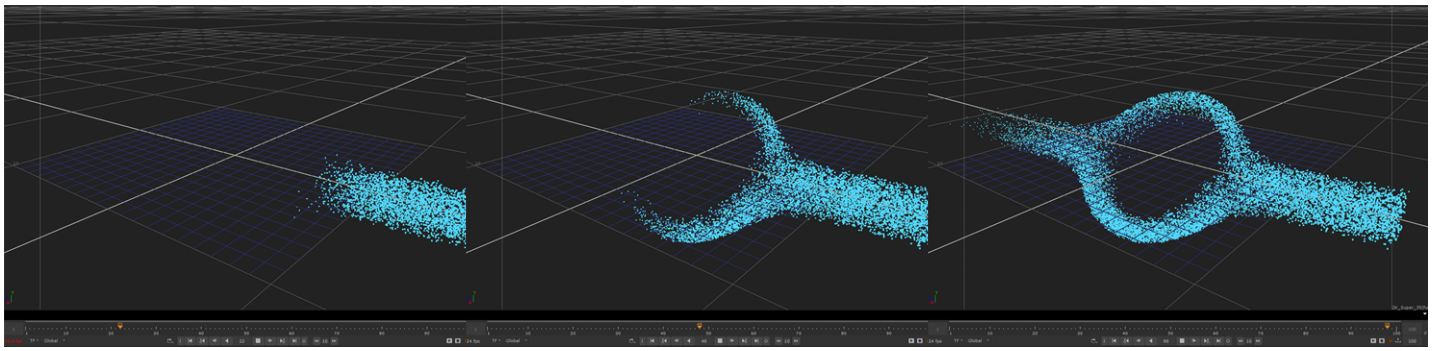
Example node graph for the following kernel

```
kernel ParticleSphereKernel :
ImageComputationKernel<ePixelWise>
{

 Image<eReadWrite> p_position;

 Image<eReadWrite> p_velocity;


 param:
   float3 _centre;
   float _radius;
```

```
 void define() {
   defineParam(_centre, "pa_position", float3(0.0f, 0.0f,
   0.0f));
   defineParam(_radius, "pa_radius", 1.0f);

 }

 void process() {
   float3  d = p_position()-_centre;
   float r = length(d);
   if ( r != 0.0f ) {
    d /= r;
    // Move the particle to the sphere surface
    p_position() = _centre + d*_radius;

    // Change the velocity to follow the surface
    float3 v = p_velocity();
    float lv = length(v);
    if ( lv != 0.0f ) {
         v /= lv;
         p_velocity() = cross(cross(d, v), d)*lv;
    }
   }

 }
};
```

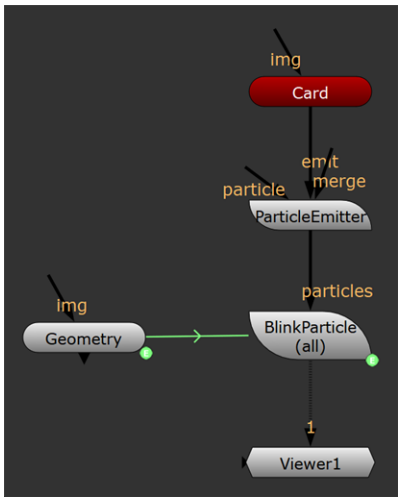
Flowing Particles

This is a slightly more involved example of what can be achieved with the ParticleBlinkScript node, the result is making particles flow around an object.



The resulting effect of the following example flowing particles kernel

Geometry cannot be accessed from the ParticleBlinkScript node, so expressions can be used within the **translate**, **rotate** and **scale** knobs of the ParticleBlinkScript node to access the **translate**, **rotate** and **scale** values from a **Cylinder** geometry node. The **radius** can be linked using an expression as well.



Example node graph for the following kernel

The basic principle of this kernel is to generate a vector field which gives the direction of flow of the particles at every point in space. The velocity vector of each particle is changed to point in the direction of flow.

Tip: Avoiding objects is just one of the ways this example can be used, many different effects can be achieved by using different flow field functions.

Here is the relevant part of the kernel which implements this. The **fieldAt** function returns the direction of flow at a point in space and the **process** function redirects the velocity vector of each particle to point along the flow vector at its position.

```
float3 fieldAt( float3 pos )
```

```
{
 return pos; // A field which just points away
 from the origin
}
void process()
{
 float3 p = p_position();
 float3 v = p_velocity();
 float3 field = fieldAt( p );
 field.normalize();
 p_velocity() = field*v.length();
}
```

The following example kernel implements a function causing particles to flow around an imaginary cylinder.

```
kernel ParticleCylinderFlowKernel :
ImageComputationKernel<ePixelWise>
{
 Image<eReadWrite> p_position;
 Image<eReadWrite> p_velocity;
 Image<eReadWrite> p_orientation;
 param:
   float3 _origin; // Origin of the cylinder
   float3 _axis; // Axis of the cylinder
   float _radius; // Radius of the cylinder
   float3 _flow; // Direction of flow
   float _strength; // The strength of the interaction with the
   flow
   float _falloff; // The speed at which the force falls off
   with distance from the surface
 local:
   float3 _normalizedAxis;
 void define() {
```

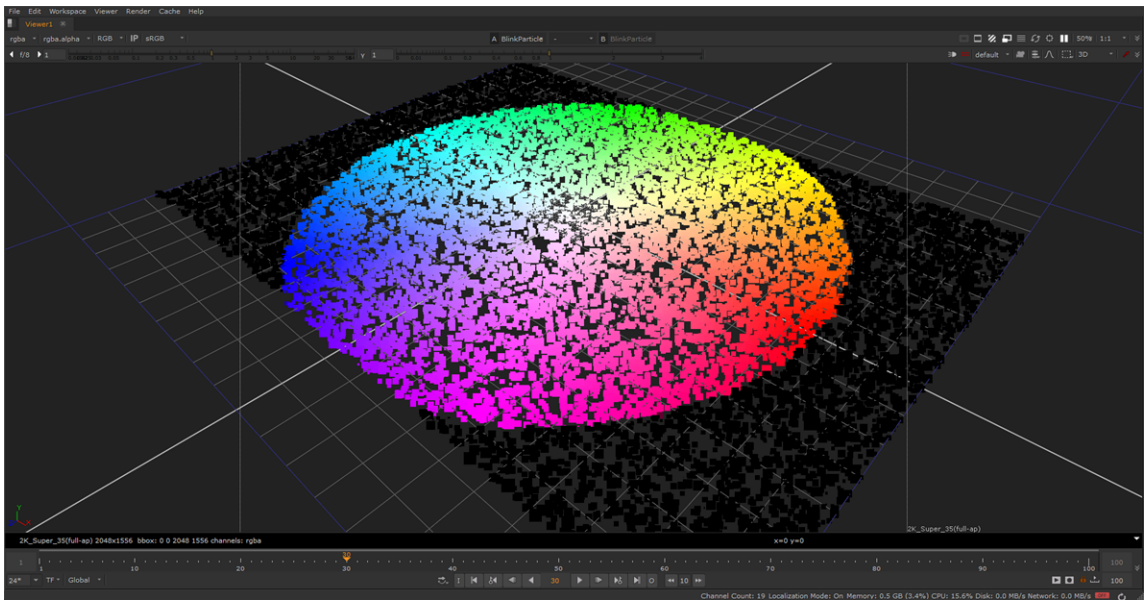
```
  defineParam(_origin, "pa_origin", float3(0.0f, 0.0f, 0.0f));
  defineParam(_axis, "pa_axis", float3(0.0f, 1.0f, 0.0f));
  defineParam(_flow, "pa_flow", float3(0.0f, 0.0f, -1.0f));
  defineParam(_radius, "pa_radius", 1.0f);
  defineParam(_strength, "pa_strength", 1.0f);
  defineParam(_falloff, "pa_falloff", 1.0f);

}

void init() {
  _normalizedAxis = normalize(_axis);

}

float3 mix(float3 a, float3 b, float t ) {
  return a+t*(b-a);

}

void process() {
  float3 p = p_position()-_origin;
  float l = length(p);
  if ( l != 0.0f ) {
   float3 axis = _axis/l;
   p -= _origin+_normalizedAxis*dot(p, _normalizedAxis);
   float r = length(p);
   float3 normal = p;
   float3 d = cross(normal, _flow);
   float3 tangent = cross(d, normal);
   float fall;
   if ( r >= _radius )
        fall = exp(-_falloff*(r-_radius));
   else
        fall = 1.0f;
   float3 force = tangent*_strength;
   force = mix( _flow, force, fall );
   l = length(force);
   if ( l != 0.0f ) {
        force = force/l;
        float speed = length(p_velocity());
```

```
            p_velocity() = speed*force;
            p_orientation() = float4(0.0, force.x, force.y,
      force.z);
      }
    }

  }
};
```

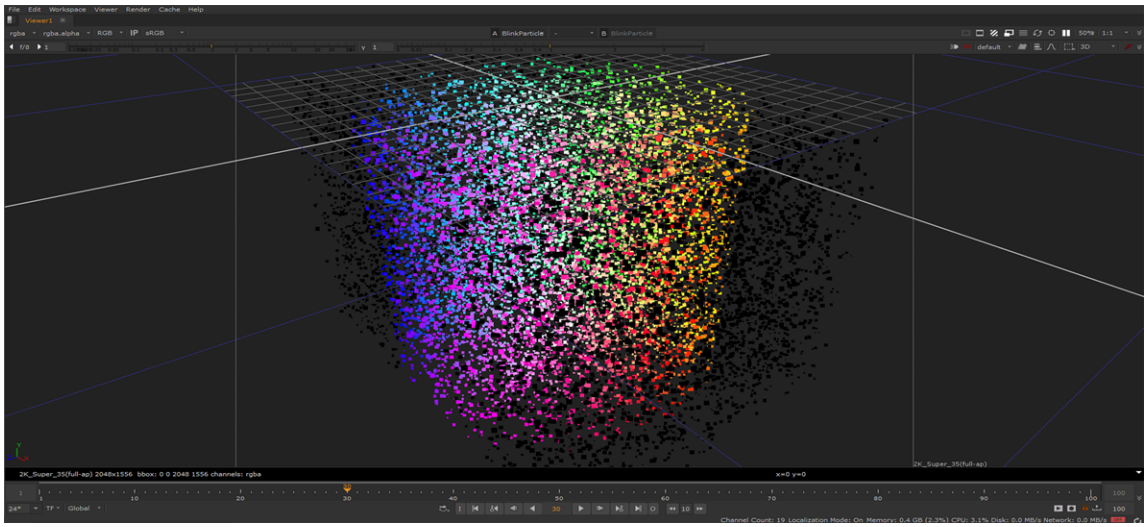

Tip: There are many ways to experiment with this kernel, try using a fluid simulation to generate the flow field for simulating smoke.

Image Input Projection

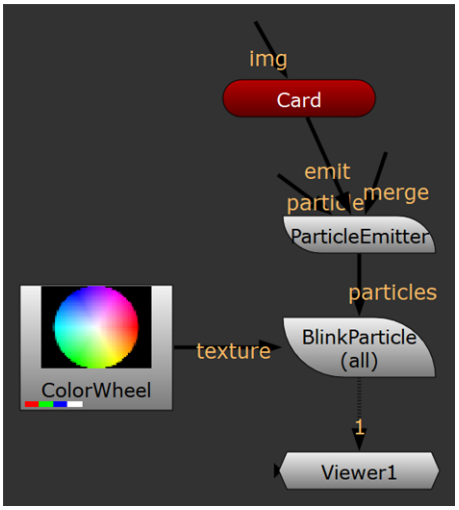
This is an example of a kernel which uses an image input to project onto the particle system, coloring particles according to their position. The image is assumed to lie on the XZ plane and two knobs are provided to set the top-left and bottom-right corners of the image in that plane.



The resulting effect of a **ColorWheel** image projection onto particles with a **velocity** of **0**



The resulting effect of a **ColorWheel** image projection onto particles with a **velocity** of **0.05**



Example node graph for the image projection kernel

```
kernel ParticleImageProjectKernel :
ImageComputationKernel<ePixelWise>
{

 Image<eRead,eAccessRandom,eEdgeClamped> image_texture;

 Image<eReadWrite> p_position;

 Image<eReadWrite> p_color;

 param:
   float _amount;
```

```
  float _dt;
  float2 _imageTL;
  float2 _imageBR;

local:
  float2 _imageSize;

void define() {
  defineParam(_imageTL, "pa_imageTL", float2(0.0f));
  defineParam(_imageBR, "pa_imageBR", float2(1.0f));
  defineParam(_dt, "_dt", 1.0f);

}

void init() {
  _imageSize = float2(image_texture.bounds.width(), image_
  texture.bounds.height());

}

void process() {
  float3 p = p_position();
  float2 xz = float2(p.x, p.z);
  float2 uv = xz = (xz-_imageTL)/(_imageBR-_imageTL);
  float2 st = uv*_imageSize;
  float4 c = bilinear(image_texture, st.x, st.y);
  p_color() = float4(c.x, c.y, c.z, 1.0f);

 }
};
```

Stitching Rigs with CaraVR

CaraVR is now included with NukeX and Nuke Studio, allowing you to solve and stitch mono and stereo VR rigs and perform a number of mono 360 compositing tasks. CaraVR operations can be separated into three broad categories: stitching, workflow, and review.



Note: CaraVR's stereo compositing nodes are not included in NukeX and Nuke Studio's plug-ins. Visit https://www.foundry.com/products/cara-vr to see the full power of CaraVR's stereo toolset.

Stitching

The Stitching nodes take care of the process of aligning your camera rig output and joining the separate images into a single, cohesive frame. You can also perform some optional global color corrections at this stage.

- **C_CameraSolver** - calculates the geometry for the multi-camera rig. This defines the location and orientation for each camera so that the images can be stitched. See Preparing Camera Rigs for more information.
- **C_CameraIngest** - takes manually solved or pre-tracked footage from third-party applications and adds CaraVR metadata to the stream to build a model of the relationship between the cameras. See C_CameraIngest for more information.
- **C_Stitcher** - automatically generates a spherical lat-long by warping and blending between the different input images. See Stitching Images Together for more information.
- **C_GlobalWarp** - an optional step to produce a preview stitch using metadata, passed downstream from a C_CameraSolver, to help minimize ghosting in overlap regions. See Matching and Solving Warps for more information.
- **C_ColourMatcher** - an optional step to perform global gain-based color correction for all the views in a rig to balance out differences in exposure and white balance. See Matching Colors Across All Cameras for more information.



Tip: The CaraVR plug-ins include example scripts in Nuke's toolbar under **CaraVR** > **Toolsets**.

Workflow

The Workflow nodes deal with the bread and butter functions required in compositing, but within a VR environment.



Note: The functionality of C_Bilateral and C_SphericalTransform nodes are now part of standard Nuke's Bilateral and SphericalTransform nodes. See Bilateral Filtering and Transforming and Projecting with SphericalTransform for more information.

- **C_AlphaGenerator** - a convenience tool that can be used to create a rectangular or elliptical mask in the alpha channel. See Generating Alpha Masks for more information.

- **C_Blender** - is used as a Merge node to combine all images together after manually correcting a stitch. See Blending Multiple Views Together for more information.
- **C_Blur** - similar to Nuke's standard Blur, but allows you to apply blur to a latlong image and produce a sensible result across the entire frame, as if the blur were applied to a rectilinear image all around. See Applying LatLong Blur for more information.
- **C_STMap** - a GPU accelerated version of the Nuke STMap node, C_STMap allows you to move pixels around in an image using a stitch_map or ppass_map to figure out where each pixel in the resulting image should come from in the input channels. You can re-use the map to warp another image, such as when applying lens distortion. See Warping Using STMaps for more information.
- **C_Tracker** - similar to Nuke's standard Tracker, but with the addition of CameraTracker-style auto-tracking and calibrated for pattern tracking in lat-long space. See Tracking and Stabilizing for more information.
- **RayRender** - a ray tracing renderer alternative to Nuke's ScanlineRender. See Rendering Using RayRender for more information. It has a number of advantages over scanline-based renderers in a VR context, including:
  - the ability to render polar regions in a spherical mapping correctly, without the artifacts inherent in scanline-based rendering in these areas, and
  - support for lens shaders, allowing slit scan rendering of multi-view spherical maps. This provides a more natural viewing environment for VR material.
  - allows you to quickly preview output from Facebook Surround, Google Jump, and Nokia OZO data to construct point clouds for use with depth-dependent workflows. The preview can be useful for positioning 3D elements accurately and then rendering into 2D through RayRender. See Compositing Using Facebook Surround Data, Compositing Using Google Jump Data, and Compositing Using Nokia OZO Data for more information.
- **C_GenerateMap** - outputs a stitch_map or ppass_map for UV or XYZ coordinates, which can be warped and then piped into C_STMap. See Generating Stitch and PPass Maps for more information.
- **Split and Join Selectively** - similar to Nuke's vanilla Split and Join node, but gives you more control over which views are affected. See Split and Join Selectively for more information.

Review

CaraVR provides a monitor out plug-in for the Oculus CV1, Oculus Quest 2, and HTC Vive, which work in a similar way to Nuke's existing SDI plug-ins. They're handled by the official Oculus SDK on Windows, by OpenHMD on Linux, and by SteamVR for the HTC Vive. See Reviewing on VR Headsets (HMD) for more information.

Stitching Camera Rigs

The Stitching nodes, C_CameraSolver and C_Stitcher, take care of aligning your camera rig imagery and joining them into a single cohesive 360 frame.

If you have manually solved or pre-tracked footage from third-party applications, C_CameraIngest can be used to add CaraVR metadata to the stream to build a model of the relationship between the cameras to pass to the stitcher or as the basis for further refinement with the C_CameraSolver.

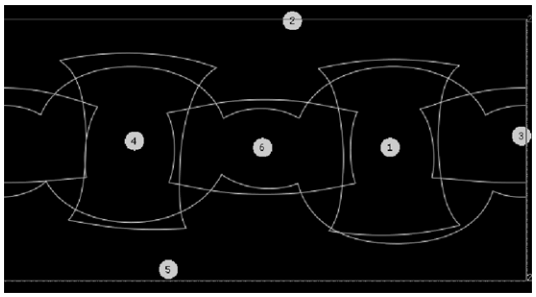
You can use C_GlobalWarp to reduce ghosting and constrain particular areas to create a fast preview before stitching. During stitching, you can also fix a number of common problems seen when shooting 360 material, using the C_ColourMatcher, C_Tracker, and C_SphericalTransform nodes.

Tip: The CaraVR plug-ins include example scripts in Nuke's toolbar under **CaraVR** > **Toolsets**.

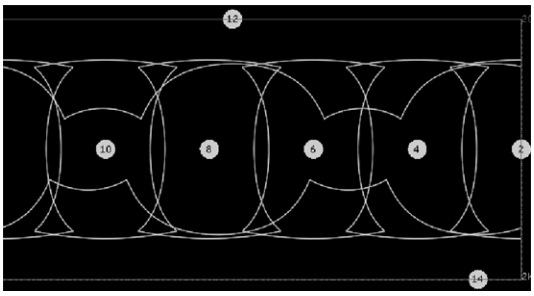
C_CameraSolver

C_CameraSolver allows you to calibrate your rig, building a model of the relationship between the cameras and producing a preview of the stitch. C_CameraSolver includes a list of rig presets that you can use as a starting point for a solve, such as the 6 camera Freedom360 and 14 camera H3PRO14. CaraVR also supports custom rigs that you calibrate manually and an import function for rigs created in PTGui and Autopano using the **.pts** and **.pano** formats.

Note: CaraVR only supports PTGui version 10 **.pts** files, or earlier. If you're using a later version of PTGui, make sure you use the **PTGui 10 Project** export option.



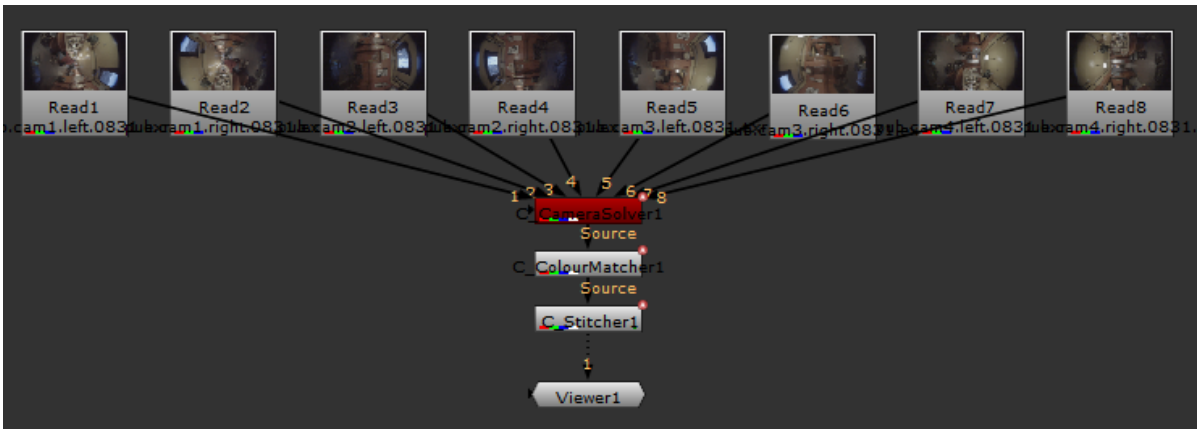
Freedom360 Rig



H3PRO14 Rig

See Preparing Camera Rigs for more information.

C_CameraSolver feeds metadata about the rig and the original source imagery downstream for later nodes to access, most importantly C_Stitcher and C_ColourMatcher. Both these nodes require this information to perform their tasks, making C_CameraSolver an engine tool, found at the top of every stitch script.



C_CameraSolver can build both nodal and spherical representations for the rig. Spherical rigs are often more physically correct. The solver can estimate rig layout and lens parameters from scratch if necessary, and offers a number of tools for fine-tuning and manually configuring the layout.

See Matching and Solving Cameras for more information.

Preparing Camera Rigs

The first step to creating a rig is to read in your camera footage using standard Nuke Read nodes and connect them to a C_CameraSolver node. The footage can be stills or sequences with single or multiple views per file.

Warning: The Read nodes representing the cameras in the rig must be pre-synchronized, because CaraVR and Nuke do not support auto-synchronization.

1. Navigate to the **Project Settings** and set the **full size format** control to the required latlong format, such as **2K_LatLong 2048x1024**.

Note: You can set the format independently using the C_CameraSolver, C_ColourMatcher, and C_Stitcher **Properties** panels.

2. Read in all your footage using Read nodes. If you're using multi-view footage, Nuke asks you to create the relevant views before continuing.
3. Select all your Read nodes and then add a C_CameraSolver node to the Node Graph and connect the Viewer.
4. If your footage contains multiple views per file, enable **Stereo** on the **Cameras** tab and select the views you want to solve using the **Input Views** dropdown.
5. In the node properties **C_CameraSolver** tab, click the **Preset** dropdown and select the required preset rig information to load:
   - **Custom** - allows you to specify any number of cameras for the rig, with custom settings for each available in the **Cameras** tab.
   - **PTGui** - allows you to specify an existing rig created in PTGui. See Importing Rigs for more information.

Note: CaraVR only supports PTGui version 10 **.pts** files, or earlier. If you're using a later version of PTGui, make sure you use the **PTGui 10 Project** export option.

   - **Preset Rigs** - select the required preset rig from the list to create the relevant rig with the correct number of cameras. These are rarely accurate enough to 'just work', but they provide a good basis for a solve in some cases.

     If you're using Nokia OZO, Google Jump, or Autopano rigs, you'll also need to import a **.txt**, **.json**, or **.pano** file. Select the required rig from the **Preset** dropdown and then click **Setup Rig** to browse to the required file. See Importing Rigs for more information.

Note: If you're using an imported or preset rig, ensure that the input cameras are connected in the same order in which the rig was configured. If the preview stitch produced by C_CameraSolver shows immediate mismatches between views, check that the camera order is correct.

Tip: If you want to change to an alternate rig, select the rig from the **Preset** dropdown and then click **Setup Rig**.

6. If you use a preset or imported rig, CaraVR makes an informed guess about the cameras in the rig. If you are using a **Custom** rig, enable **Set global camera parameters** and enter the lens parameters. The more accurate information you enter here, the better your chances of getting a good result from the solver.

Note: When you're setting up a **Custom** rig, all camera settings in the rig are assumed to be identical and are replicated in the **Cameras** tab. If your rig contains different cameras, click the **Cameras** tab, and then select a camera from the table to change the camera's lens parameters.

7. Proceed to Matching and Solving Cameras.

Importing Rigs

CaraVR allows you to import rigs from PTGui, Nokia OZO, Google Jump, and Autopano using the **.pts**, **.txt**, **.json**, and **.pano** file formats.

Note: If you're using an imported rig, ensure that the input cameras are connected in the same order in which the rig was configured. If the preview stitch produced by C_CameraSolver shows immediate mismatches between views, check that the camera order is correct.

To import a rig:

1. Read in all your footage using Read nodes. If you're using multi-view footage, Nuke asks you to create the relevant views before continuing.
2. Select all your Read nodes and then add a C_CameraSolver node to the Node Graph.
3. In the node properties, click the **Preset** dropdown and select **PTGui**, **Nokia OZO**, **Google Jump**, or **Autopano** and then click **Setup Rig**.

Note: CaraVR only supports PTGui version 10 **.pts** files, or earlier. If you're using a later version of PTGui, make sure you use the **PTGui 10 Project** export option.

   The file browser displays.
4. Navigate to the required template file and click **Open** to import the rig.

   The camera and lens parameters are filled in automatically from the metadata in the file.
5. Proceed to Matching and Solving Cameras.

Notes on Autopano Imports

The current limitations on **.pano** imports are as follows:

• Feature matches in Autopano are not converted to matches in CaraVR.

• Only rigs employing fisheye lenses can be imported.

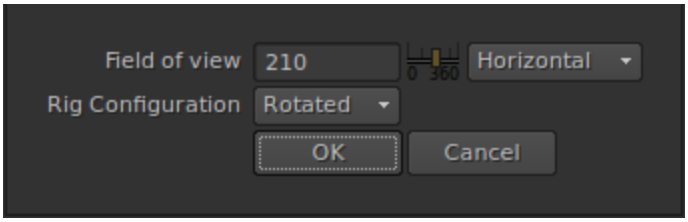
- The fisheye model in Autopano is different to that implemented by CaraVR. As with import procedures in other applications, the values are converted and so distortion does not match exactly.

Importing Back to Back Rigs

The **Back to Back** preset is designed to solve two-camera fisheye rigs with a field of view greater than 180°. To use the preset:

1. Select **Back to Back** from the **Preset** dropdown.
2. Click **Setup Rig**.

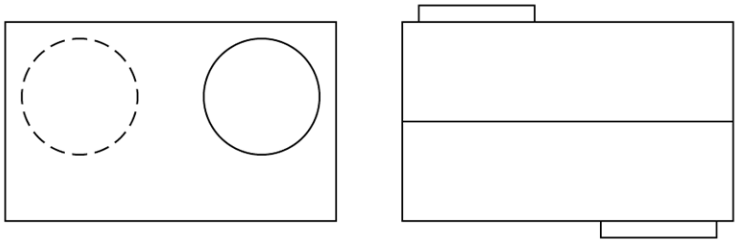
   The rig setup dialog displays.



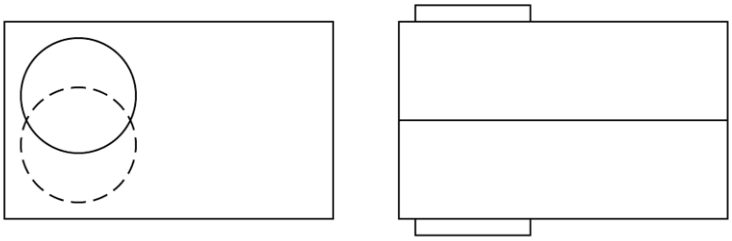
3. Enter the **Field of view** and select **Horizontal**, **Vertical**, or **Diagonal** depending on your rig.

Tip: If you are unsure of the **Field of view**, estimate and then adjust the C_CameraSolver's global **focal length** accordingly after clicking **OK**.

4. Set the **Rig Configuration** to account for different lens positions:
   - **Rotated** - the lenses are aligned vertically, but not horizontally.



   - **Flipped** - the lenses are aligned horizontally, but not vertically.



5. Click **OK**.
6. Proceed to Matching and Solving Cameras.



Note: Two camera, back to back rigs can cause problems for C_Stitcher because of the lack of overlap between the cameras. See Improving Overlap for Back to Back Rigs for more information.

Matching and Solving Cameras

The next step after setting up the rig is to set the camera properties for the rig, locate matches between overlapping cameras, and solve the camera positions relative to each other.

1. The **Analysis** section enables you to change the camera properties for the rig. By default, CaraVR assumes that all cameras have the same focal distance, no lens distortion, and a spherical layout.

   Set the **Focal Length** to account for:

   - **Known Focal Length** - fixes the existing focal length in the solve.
   - **Optimize per Camera** - calculates the focal length for each camera in the rig.
   - **Optimize Single** - calculates a single optimized focal length for all cameras in the rig.
2. Set the **Lens Distortion**, if required. CaraVR assumes **No Lens Distortion** by default, but like **Focal Length**, you can account for:
   - **Known Distortion** - fixes the lens distortion in the solve.
   - **Optimize per Camera** - calculates the lens distortion for each camera in the rig.
   - **Optimize Single** - calculates a single optimized lens distortion for all cameras in the rig.
3. Set the **Camera Layout** you intend to use:
   - **Nodal Layout** - all cameras are located at the scene origin. For example, a single DSLR shooting 360 stills would use this nodal layout setting.
   - **Spherical Layout** - the cameras in the rig are placed on a sphere whose diameter, defined by the **Rig Size** control on the **Cameras** tab, defaults to 30 cm.
   - **Horizontal** / **Vertical** / **Planar Layout** - the cameras in the rig are placed on a single line or plane depending on selection.
   - **Free Layout** - no constraints are enforced on camera positions.

Note: The **Horizontal**, **Vertical**, **Planar**, and **Free** layouts are experimental. Please contact support.foundry.com if you have any feedback you'd like to share with Foundry.

Tip: Setting the **Rig Size** control to the actual size of the rig in use allows you to enter real world measurements for other **Cameras** controls, such as **Focal Length**.

Rigs that utilize multiple cameras, filming concurrently, usually have a spherical configuration allowing you to alter the **Convergence** between cameras to check for image alignment at different depths in the scene. See Adjusting Convergence in Spherical Rigs for more information.

4. CaraVR automatically adds a single keyframe at the beginning of the sequence to define a set of feature matches that correspond to fixed points in the scene, but you can add keyframes manually by:
   - scrubbing in the Viewer and then clicking **Add Key** ( ) to add a keyframe at the current frame, or
   - clicking **Key All** to add keyframes throughout the sequence at the **Step** interval.
   - clicking **Import** and selecting the node from which you want to import keyframes. You can import keyframes from any instance of C_CameraSolver, C_Stitcher, or C_ColourMatcher in the script.

   The more keyframes you add, the longer the process takes, though you may get better results.

Note: C_Stitcher only computes optical flow camera warps at designated keyframes, it ignores upstream animation that does not fall on the same keyframes. Use the **Import** button to copy over all keyframes from C_CameraSolver. This way, any additional per-camera, manual keyframes added in the **Cameras** tab are also copied.

Tip: If you're solving a rig with a nodal layout, you can click the **Settings** tab and enable **Solve for animated cameras** to calculate animated values for the camera parameters at each of the analysis key frames.

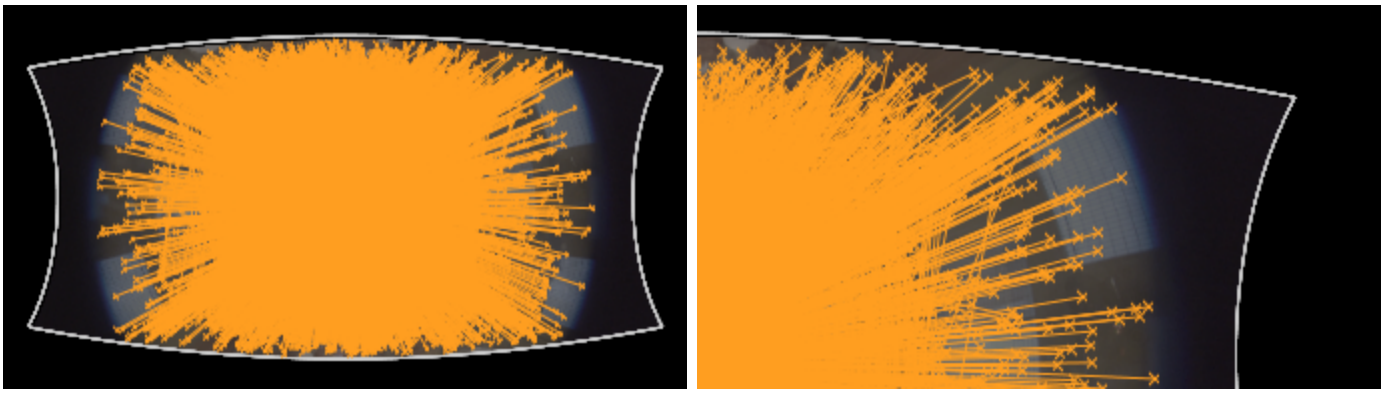
This setting can help to eliminate ghosting in mono 360 nodal stitches by automatically adjusting the cameras to compensate for differing scene depths.

Bear in mind, however, that this setting may produce odd results if the camera position is fixed. Any changes in camera rotation at the analysis keys may cause the background scene to shift.

5. Click **Match** to compare keyframes on overlapping cameras for shared features. CaraVR only looks for matches in cameras that overlap by default.

Note: **Match** calculates the feature matches for all cameras in the rig at the same time, so there is no need to calculate them separately for left and right images in a stereo setup. See Stitching Stereographic Rigs for more information.

Initially, all the matched features for the cameras overlay each other in the Viewer. Individual matches won't be visible until you solve the cameras on the rig, or zoom into the Viewer as shown in the second image.



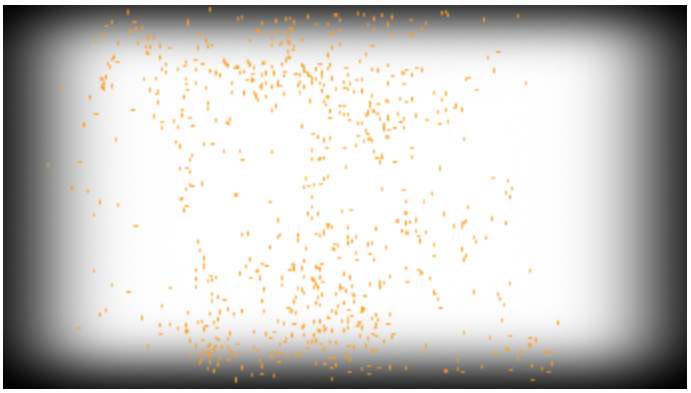
Feature matches for all cameras at different Viewer zoom levels.

Note: Matches are only displayed when the playhead is on a keyframe.

You can step through the cameras on the rig to examine the matches using the views buttons above the Viewer. If you switch the Viewer to display only the alpha channel, you can also examine the default masks applied to each view.

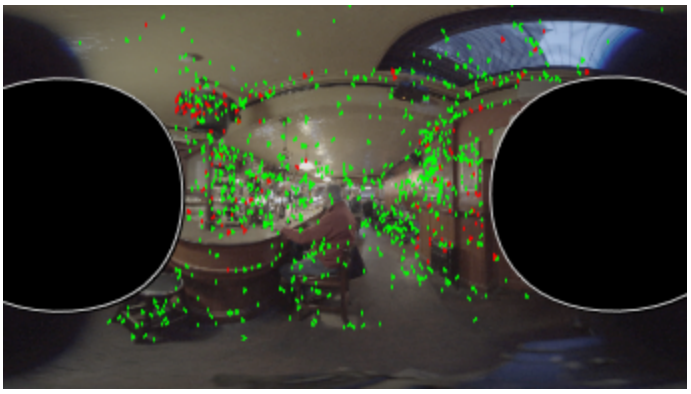


Feature matches in a single camera.



The default alpha channel mask.

When the rig is solved, you can also view matches between overlapping cameras in latlong space by switching the **Cam Projection** dropdown to **Latlong**.



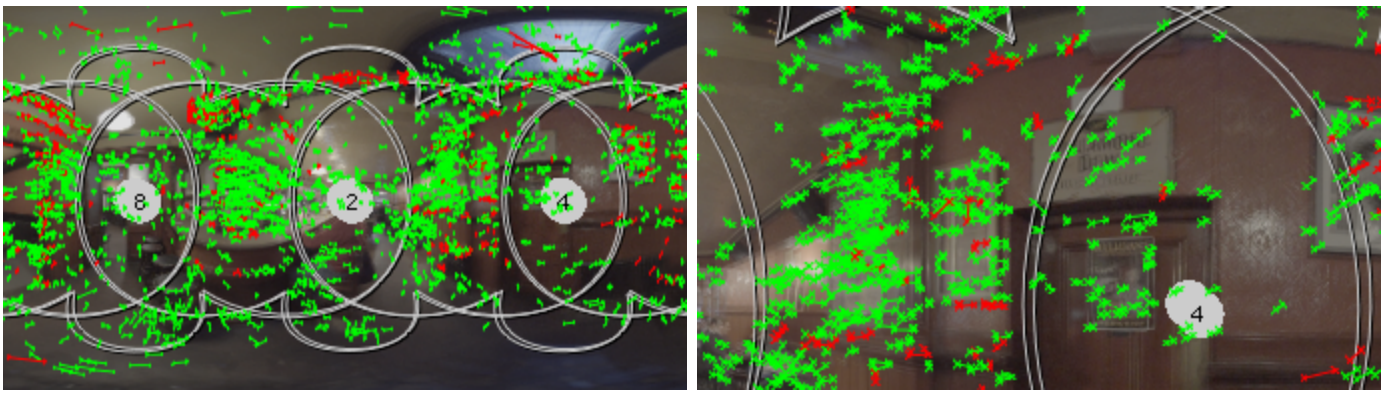
Feature matches in a single camera
in latlong space.

See Troubleshooting Matches and Solves for more information on improving match results.

6. Click **Solve** to calculate the geometry for the selected rig to define the location and orientation of each camera, relative to the other cameras in the rig, and arrange the matched images in the correct order.

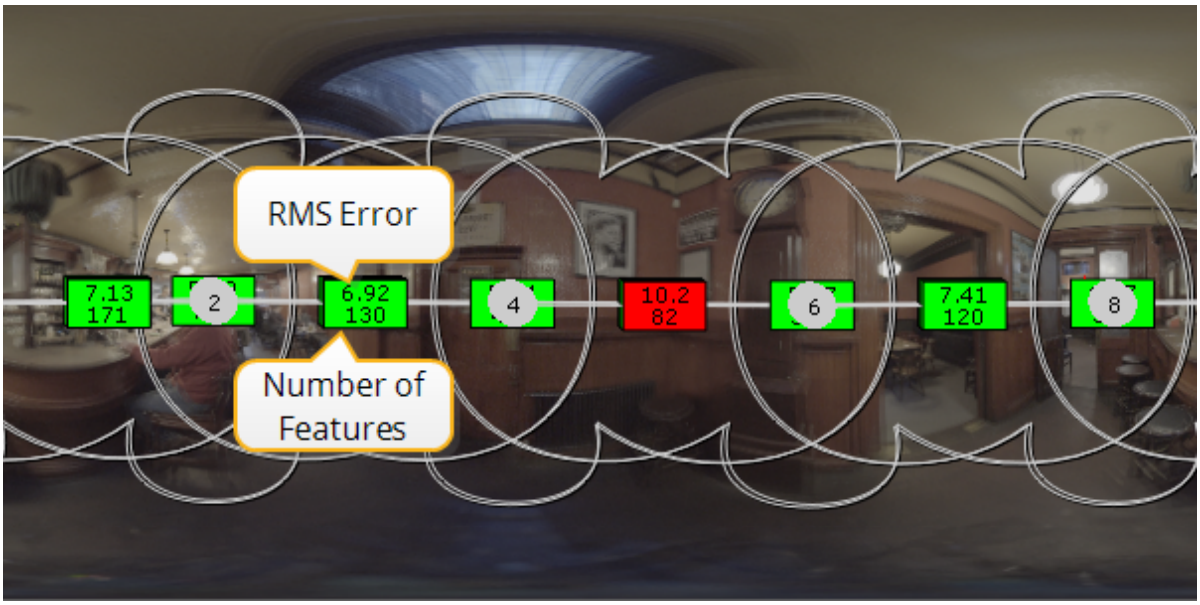
Note: The solve calculates the geometry for all cameras in the rig at the same time, so there is no need to calculate them separately for left and right images in a stereo setup. See Stitching Stereographic Rigs for more information.

The solved preview image displays in the Viewer, including all matched points for the cameras in the rig.



The solved cameras and feature matches at different Viewer zoom levels.

7. Switch to **Camera** mode ( ) in the Viewer tools to display an overlay containing the number of features shared by overlapping cameras and the RMS (root-mean-square) errors calculated by the solve for them.



Green overlays represent RMS errors below the **Error Threshold** and red overlays those over the threshold.

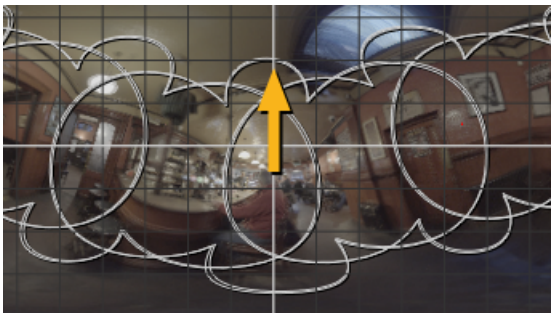
Note: The overlay is only displayed when the playhead is on a keyframe.

See Troubleshooting Matches and Solves for information on how to refine your matched features and solve.

8. Turn off the overlay by pressing **Q** over the Viewer to see the result.
9. The solved cameras may need a little work to get the horizon line correct before you proceed to stitching the views together. CaraVR may not automatically place your images in the correct orientation.

   CaraVR offers two methods for aligning the horizon, the **Transform** controls on the **Settings** tab and the Viewer toolbar.

   - **Transform Controls** - Open up the C_CameraSolver **Properties** panel and click the **Settings** tab. Adjust the **rotate x**,**y**, and **z** controls to align the cameras as flat line halfway down the frame, or the equator.
   - **Viewer toolbar** - enable the **Horizon** tool and then hold **Ctrl**/**Cmd**+**Alt** and drag in the Viewer to adjust the horizon line.





Note: You can also adjust individual cameras this way, by selecting the **Camera** tool instead.

10. Set the required output resolution using the **Format** dropdown, if required.

Tip: CaraVR adds a tab to the **Properties** panel of Nuke's Write node allowing you quickly select groups of views such as **stereo** and **cams** when rendering previews or stitches. You can still select views manually using the **views** dropdown, but **View Presets** can make the process faster.

11. For more information about inaccurate feature matches or solved cameras, have a look at the Troubleshooting Matches and Solves section before stitching the images.

    Once you have your solved rig, you can export to a variety of preset nodes (see Exporting to Preset Nodes for more information), skip to Stitching Images Together to stitch the preview together, or proceed to C_ColourMatcher using C_ColourMatcher prior to stitching.

Troubleshooting Matches and Solves

Some sequences are inevitably going to cause problems while matching and solving. There are a number of refinement workflows to assist CaraVR when solving rigs.

Tip: CaraVR adds a tab to the **Properties** panel of Nuke's Write node allowing you quickly select groups of views such as **stereo** and **cams** when rendering previews or stitches. You can still select views manually using the **views** dropdown, but **View Presets** can make the process faster.

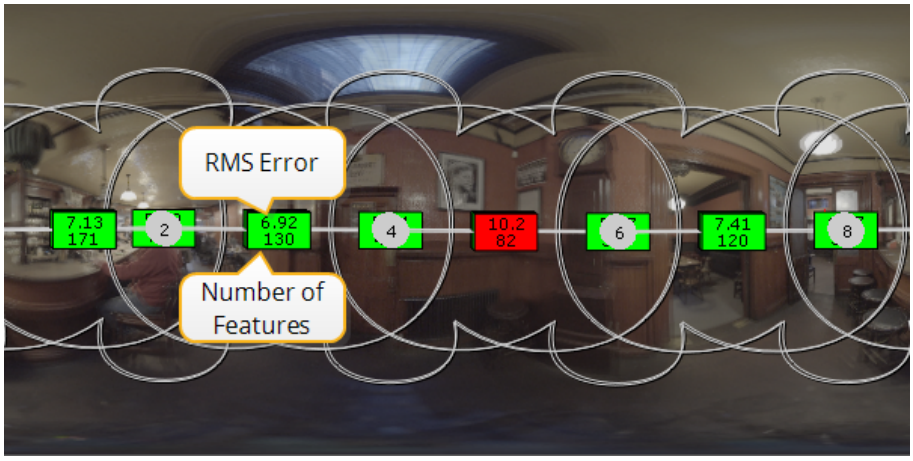
Rejecting and Refining Matches

After matching and solving the cameras, you'll see some features that are outside the specified **Error Threshold** tolerance, marked in red.

Generally speaking, matched pairs should trend in the same direction for a given area of the sequence, so that's a good place to start refining the matches. Another good sign that a match is poor is the distance between the matched features, when compared to other pairs in the same region.

1. Switch to **Camera** mode ( ) in the Viewer tools to display an overlay containing the number of features shared by overlapping cameras and the RMS (root-mean-square) errors calculated by the solve for them.

Note: This overlay is only displayed on frames where a keyframe is present.

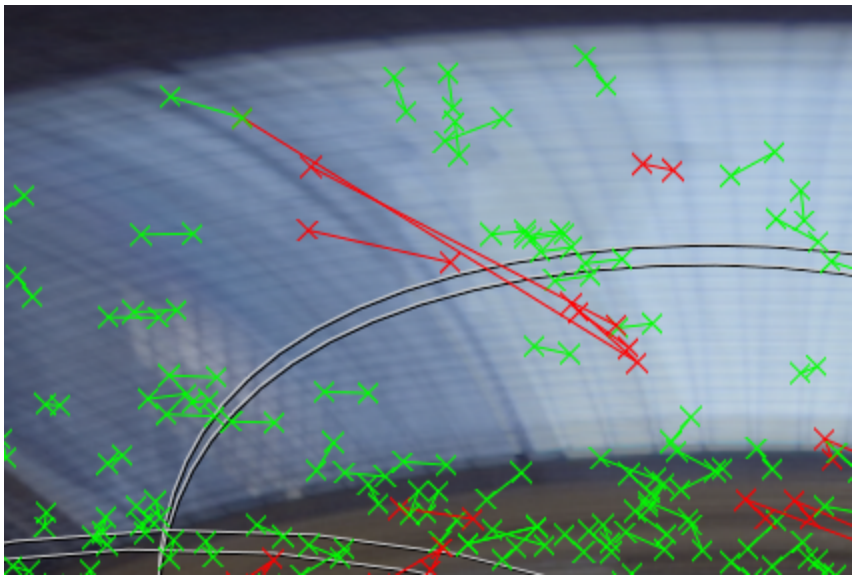


In the example, the matches within the overlap for cameras 4 and 6 register an RMS error of 10.2, which is quite high.

2. The first thing to try is dialing in some lens distortion using the **Analysis** section of the node properties.
3. Click **Lens Distortion** and select **Optimise Single** to use one lens model for all the cameras in the rig.
4. Click **Refine**. You may find that the RMS errors fall to more acceptable levels.

Note: You can refine the solve further by setting **Lens Distortion** to **Optimse per Camera** and then clicking **Refine** again for certain rigs.

5. Switch back to **Matches** mode ( ) to view the individual matches and zoom into problem regions.



The example shows matches that CaraVR has rejected automatically. They're quite far apart, when compared to the other matches in the area, but you'd see the other matches turn red as well if you lowered the **Error Threshold** control.

6. To remove the rejected matches, you can either:
   - click the **Reject** button in the C_CameraSolver properties, or
   - select individual matches and press **Backspace** or **Delete** on your keyboard.
7. After cleaning up the rejected matches, click **Refine** in the C_CameraSolver properties to recalculate the solve based on the current solve.

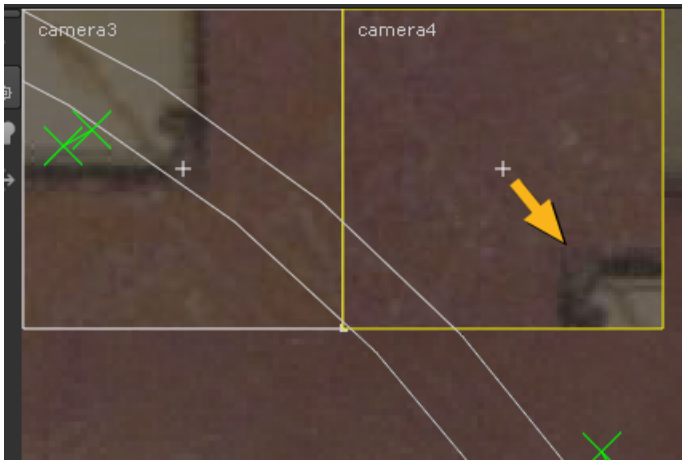
   You should see the RMS error decrease, along with the number of features.

Adding User Matches

User matches are placed manually, rather then being automatically detected by C_CameraSolver, and can be used to improve automatic matching. You can add as many user matches as required.

1. Switch to **Add Match** mode ( ) in the Viewer tools.
2. Hold **Ctrl**/**Cmd**+**Alt** and click a feature in the Viewer that is present in more than one camera to add a match marker ( ).
3. Click the user match to display a zoom window for each camera in which the feature resides.

Tip: You may find that panning the Viewer to move the automatic features makes it easier to see the zoom windows.



4. Refine the match position by clicking and dragging in the camera zoom window for each instance of the match.

   A match marker is added for each camera.



5. Add as many user matches as required and then click **Refine** in the C_CameraSolver properties to recalculate the solve based on the current solve. User matches carry more weight than automatic matches when solving the rig, so the more accurate matches you add, the better the chances of getting a good solve.

Tip: You can adjust the **Settings** > **User Matches Weight** control to instruct CaraVR to give user matches more or less 'weight' when refining a solve. If you haven't added many user matches, you might consider reducing the amount of weight so that the solve depends more on the automatically generated matches.

6. Switch back to **Camera** mode ( ) to check that your user matches have improved the solve. You should see the RMS error decrease.

Adjusting Convergence in Spherical Rigs

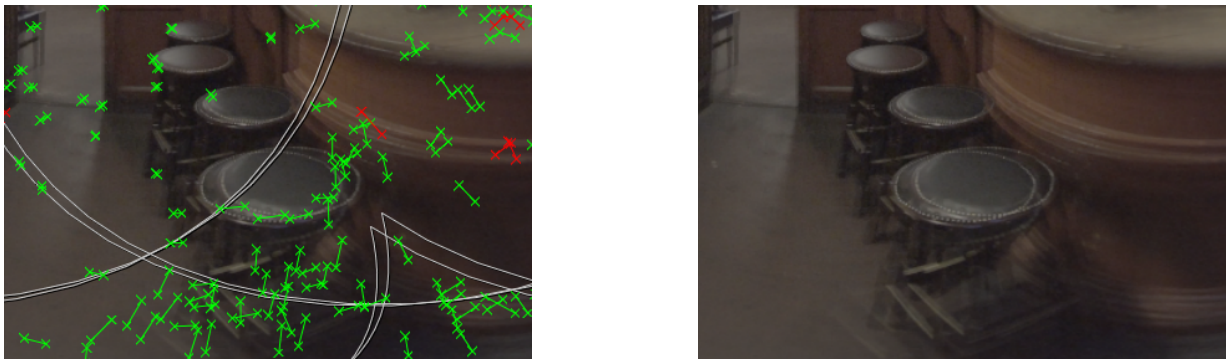
If you're using a spherical camera layout, you can adjust the **Converge** control to alter the depth at which the cameras overlap, allowing you to bring particular areas in the scene into focus. You can use convergence to determine the distance of objects from the camera by adjusting the slider until the ghosting disappears at the required depth.

Convergence in C_CameraSolver is used to test the quality of the solve, the final convergence is set using the controls in C_Stitcher. See Setting the Convergence Point in Spherical Rigs for more information.

Note: Convergence doesn't have any affect on nodal rigs because all the cameras share the same origin.

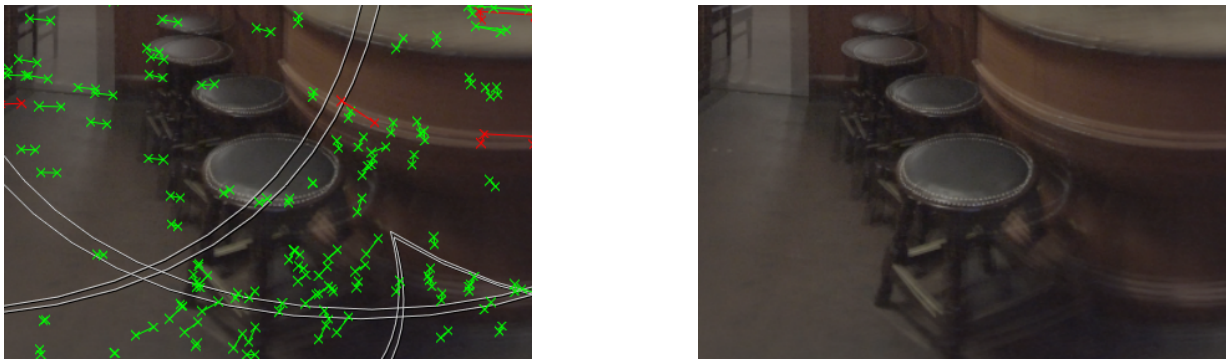
The first step is to set the approximate diameter of the sphere on which the cameras sit using the **Rig Size** control on the **Cameras** tab. This normalizes the 3D representation of the rig into close to real world units, depending on the quality of the solve. The **Converge** control then defines the depth at which the preview stitch is converged, bringing different areas of the shot into focus.

The example shows converge depths using the default **Rig Size** setting of **0.3** (or 30 cm). The **Converge** control defaults to **10**, which is a little high when considering the stool in the foreground.



Default rig settings showing an offset between overlapping cameras.

Decreasing the **Converge** value to around **1.6** metres brings the stool into sharper focus. You can also see that the matched feature pairs overlay each other.



Decreasing the converge depth aligns the offset nicely.

You can remove any feature matches that are shown in red by clicking **Reject** in the node properties, and then **Refine** the solve.

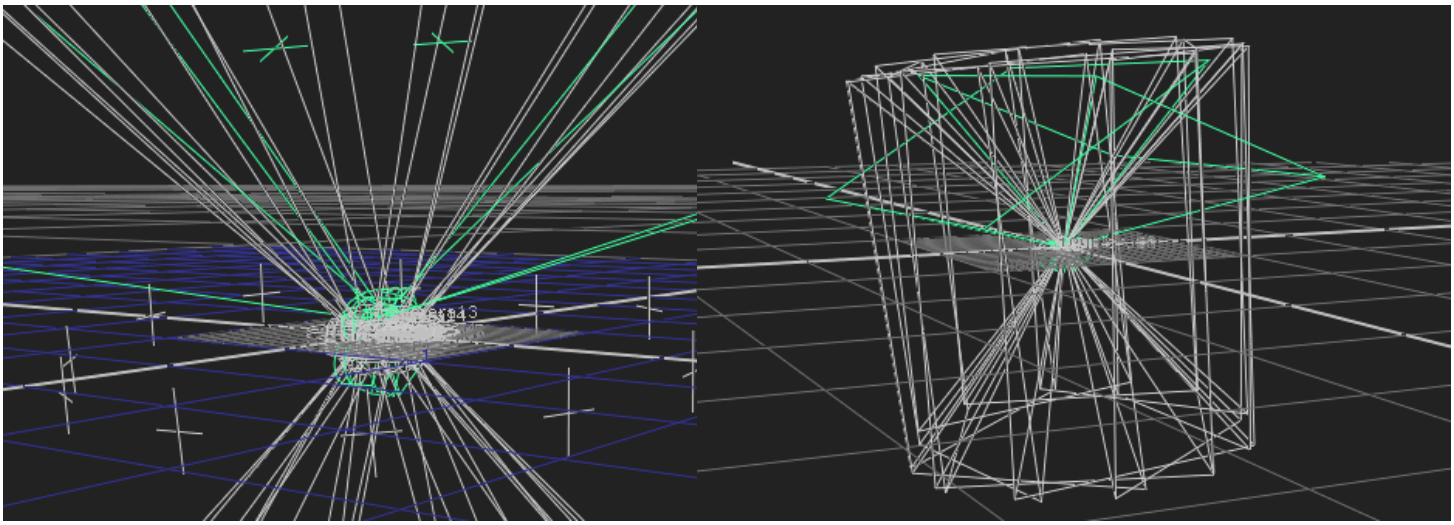
Exporting to Preset Nodes

C_CameraSolver includes an **Export** dropdown, which automatically adds preset script components allowing you to quickly create node trees for common VR tasks.

Select the required operation from the dropdown and then click **Create** to add the script to the Node Graph:

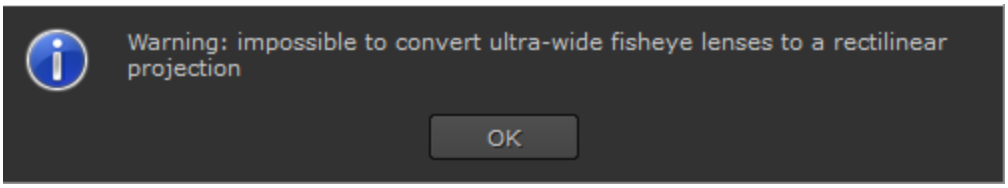
• **Cameras** - creates Camera nodes, attached to an Axis and Scene, for each camera in the rig.

Note: The Axis node contains any global rotations you applied to the rig.



This option allows you to examine the orientation and field of view for each camera in the rig in Nuke's 3D Viewer.

If you're solving rigs with extreme fisheye focal lengths, Nuke displays a warning message.



This occurs when Nuke can't automatically convert the focal length to rectilinear. In this case, you can export using the **Camera Ingest** option described below, to pass through the focal length unchanged, allowing you to ingest extreme fisheye camera data without issues.

- **Transforms (split)** - creates a single C_SphericalTransform containing rotation data for each camera in the rig. Click through the views above the Viewer to examine the rotation of each camera.

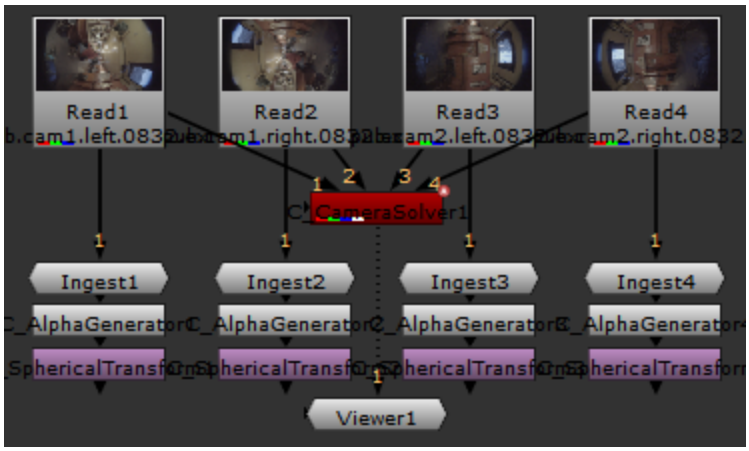


In the **Properties** panel, open the **Rotation Angles** control to see the rotation information for each view.

Note: C_Stitcher can stitch cameras together from the exported transforms without the need for a C_CameraSolver in the script. See Stitching Pre-Projected Images Together for more information.

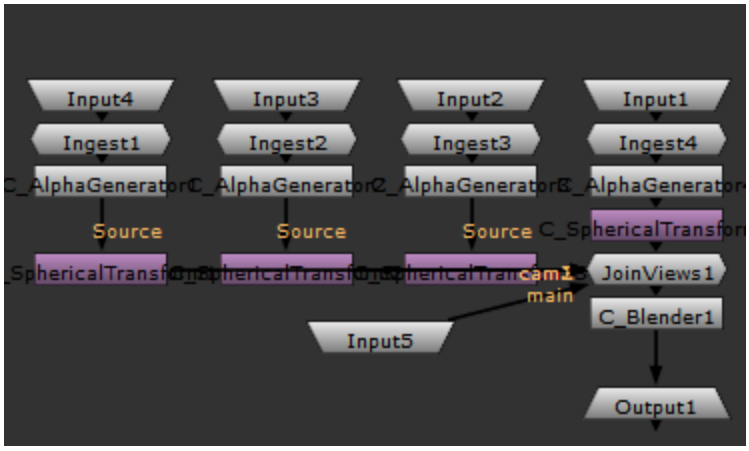
- **Transforms (separate)** - creates separate rotation data for each view using a OneView and C_SphericalTransform node for each camera in the rig.

  This export is similar to **Transforms (split)**, but the views are separated out into individual streams.



Note: C_Stitcher can stitch cameras together from the exported transforms without the need for a C_CameraSolver in the script. See Stitching Pre-Projected Images Together for more information.

- **Manual 2D Stitch** - creates a RigWorkflow2D group containing a C_SphericalTransform for each view, passed into a JoinViews and C_Blender node to re-assemble the views into a coherent whole.



The AddMaskGenerator nodes for each view allow you to adjust the masks used to stitch views together, giving you more control over coverage and feathering between matched camera pairs.

- **Manual 3D Stitch** - creates a RigWorkflow3D group containing a C_SphericalTransform and Camera for each view, projected onto a Sphere and rendered through RayRender.



Note: Projections in Nuke's 3D system expect rectilinear footage, which is inherently limited to a 180° field of view. As a result, C_CameraSolver's manual 3D workflow is not applicable to rigs using fisheye lenses.

- **Ingest (joined)** - creates ShuffleView nodes for all inputs, allowing you to assign inputs to views manually and then joins the views together. Double-click the ShuffleView node for an input to specify which view to output using the **get** and **from** controls.
- **Ingest (separate)** - creates Ingest groups containing ShuffleView and OneView nodes for all inputs, allowing you to assign inputs to views manually. Double-click the ShuffleView node for an input to specify which view to output using the **get** and **from** controls.

  This export is similar to **Ingest (joined)**, but the views are separated out into individual streams.
- **Contact Sheet** - creates a ContactSheet containing all solved views for review purposes.
- **Camera Ingest** - automatically creates Camera nodes for each view, including an Axis node containing any global rotation, and links the images and cameras to a C_CameraIngest node.

  This export option sets the **Cameras** > **Focal Length** control to **Default (pass through)**, allowing you to pass through focal lengths unchanged to ingest extreme fisheye cameras without issues.

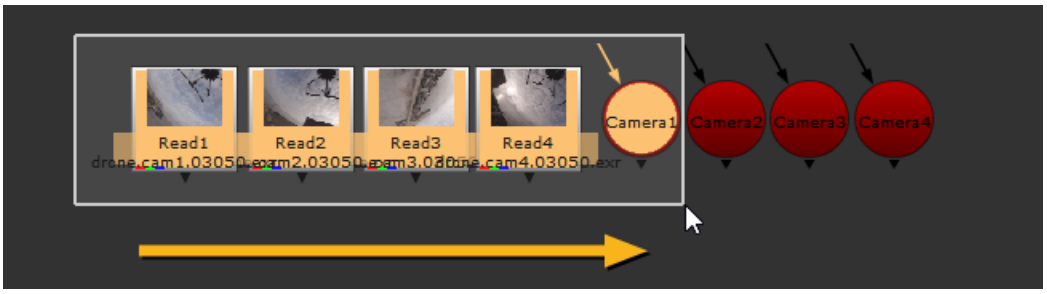
  See C_CameraIngest for more information.

C_CameraIngest

C_CameraIngest takes manually solved or pre-tracked footage from third-party applications and adds CaraVR metadata to the stream to build a model of the relationship between the cameras to pass to the stitcher or as the basis for further refinement with the C_CameraSolver.

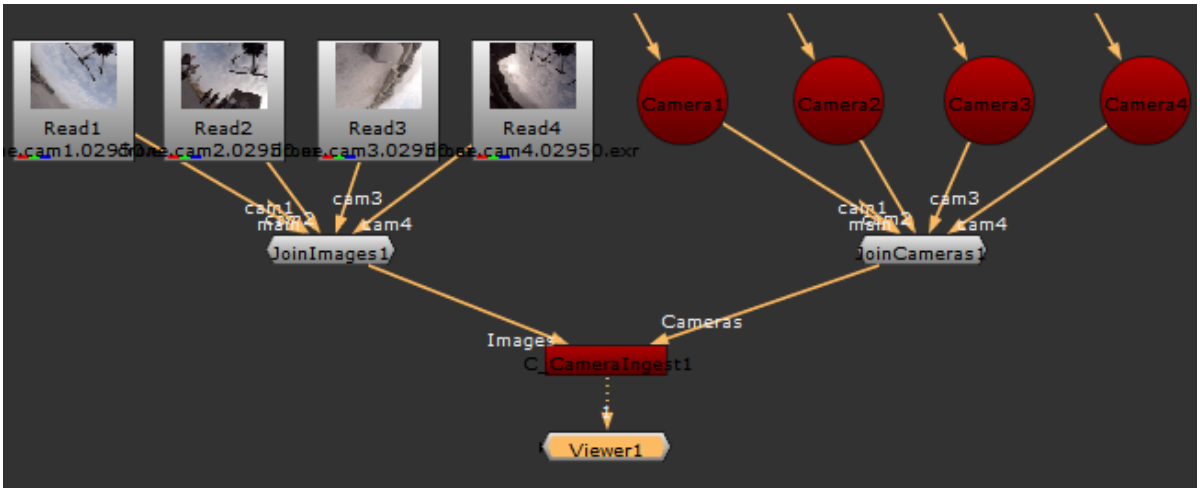
You can also use C_CameraIngest with cameras exported from C_CameraSolver, see Exporting to Preset Nodes for more information. The Axis node associated with the Camera nodes contains any global rotation applied to the rig.

1. Read in your image sequences and camera data using Nuke's standard Read and Camera nodes.
2. Select all the Read nodes and Camera nodes in order to ensure that the inputs are connected up correctly.



3. Add a C_CameraIngest to the Node Graph.

   CaraVR connects all your images and cameras to the C_CameraIngest node using JoinViews nodes with modified names, JoinImages and JoinCameras.



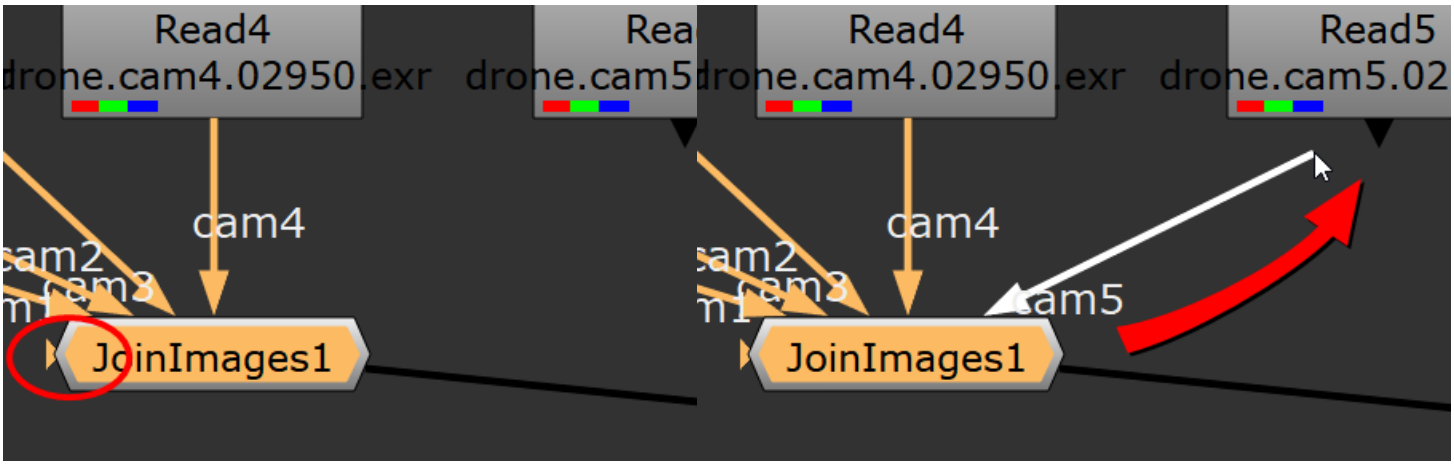
Note: If you selected the nodes in the wrong order, you may need to manually reconnect the node inputs.

4. Connect a Viewer to the C_CameraIngest node to display the main view.
5. Click **Camera Views** > **Create** to add more views, if required.

   The **Create Camera Views** dialog is displayed.
6. Enter the number of views you want to create. For example, if the script already contains four views, entering seven in the dialog creates three new views.

Note: Entering a number less than the current number of views deletes the extra views from the script.

The additional image and camera inputs are added to the left-hand side of the JoinImages and JoinCameras nodes, drag-and-drop them to connect the inputs as normal.



Note: Camera nodes in Nuke don't include lens distortion controls, so C_CameraIngest includes a cameras table in the **Properties** panel on the **Cameras** tab, similar to that found in the C_CameraSolver controls.

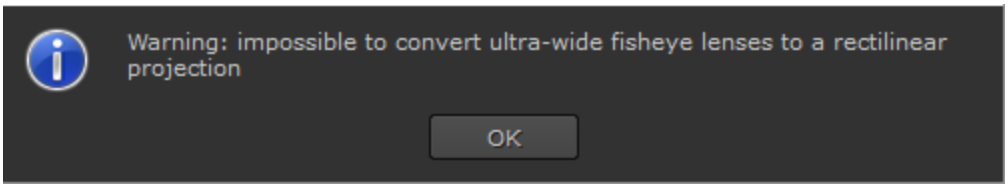
7. Set the **Converge** control to the depth at which the cameras overlap, allowing you to bring a single point of interest in the scene into focus.

Note: The **Converge** control has no effect on **Nodal Layouts**.

8. You can then pass the result to a C_Stitcher or C_CameraSolver for further refinement. See C_Stitcher and C_CameraSolver for more information.

Extreme Fisheye Focal Length

Nuke can't automatically convert extreme fisheye focal lengths to rectilinear. If you attempt to export some fisheye cameras from C_CameraSolver, a warning message is displayed.

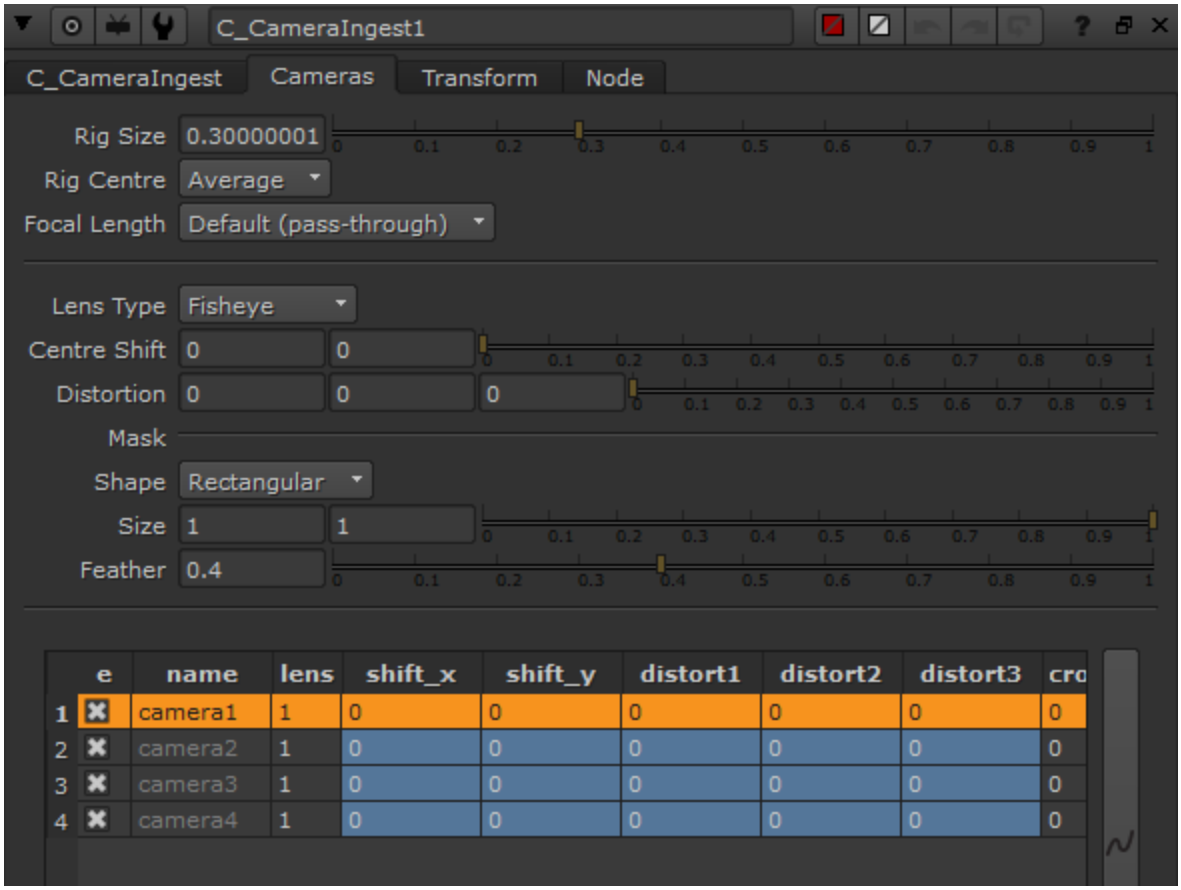


In this case, you can use C_CameraIngest to pass through the focal length unchanged, allowing you to ingest extreme fisheye camera data without issues.

Note: The **Camera Ingest** option described in Exporting to Preset Nodes create Camera nodes with unconverted fisheye focal lengths from C_CameraSolver.

The **Focal Length** control in the **Properties** panel, **Cameras** tab allows you to pass through the focal length unchanged, instead of automatically converting it to rectilinear.

Default (pass-through) is the default setting for the **Focal Length** control. You can then select a camera in the matrix on the **Cameras** tab and enter camera information to improve the result.



C_Stitcher

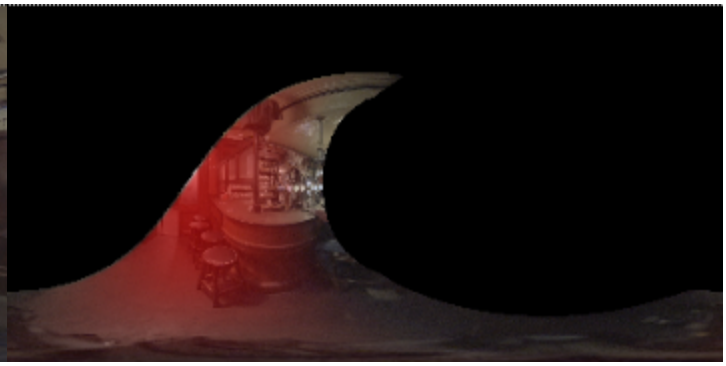
C_Stitcher uses Ocula-style disparity vectors to line up matched features in the overlap areas between overlapping cameras and then blends the results to generate a spherical latlong.

You can adjust the vector detail using the **Vectors** tab to enhance the result and export C_STMaps for use elsewhere in the script. C_Stitcher also include support for stereo output from rigs with ring or ball configurations, such as the Nokia OZO and Jaunt ONE, respectively.

The output of C_Stitcher is the **main** view containing the warped and blended latlong of the input cameras, and the individual source cameras warped into the same latlong space, set up as numbered **cam** views above the Viewer.



The **main** warped and blended view from all cameras.



Individually warped cameras and masks.

Stitching Images Together

CaraVR uses the C_Stitcher to generate a spherical lat-long by warping and blending between the different input images from the C_CameraSolver. See Stitching Stereographic Rigs if you're stitching a stereo rig.

To stitch sequences:

1. Connect a C_Stitcher node to the output of your C_CameraSolver node.
2. CaraVR automatically adds a single keyframe at the current frame, but you can add keyframes manually by either:
   - scrubbing in the Viewer and then clicking **Add Key** ( ) to add a keyframe at the current frame, or
   - clicking **Key All** to add keyframes throughout the sequence at the **Step** interval.

     If you have a lot of movement in your sequence, set the **Step** interval to a lower value so that the greater difference between frames is accounted for in the stitch. The opposite is true for more static sequences, where a higher value may produce better results.
   - clicking **Import** and selecting the node from which you want to import keyframes. You can import keyframes from any instance of C_CameraSolver, C_Stitcher, or C_ColourMatcher in the script.

Note: C_Stitcher only computes optical flow camera warps at designated keyframes, it ignores upstream animation that does not fall on the same keyframes. Use the **Import** button to copy over all keyframes from C_CameraSolver. This way, any additional per-camera, manual keyframes added in the **Cameras** tab are also copied.

Adding keyframes manually can help in blurred areas between existing keyframes where the interpolation is inaccurate.

The more keyframes you add, the longer the process takes, though you may get better results. The stitcher warps and blends the input images to create the final warped frames.

3. To see the before and after images, toggle **Enable Warps** on and off.



Pre-warp sequence



The sequence with warp enabled.

4. Enable **Override Cameras** to use the **Cameras** tab in the **Properties** panel to select which cameras are warped.

   You can view any combination of input cameras in the stitch output by enabling and disabling cameras in the **Selected** tab. The **Preset** dropdown can quickly enable or disable all cameras in the stitch by selecting **All** or **None**.

   For example, if you wanted to examine the stitch between two cameras in isolation, select **None** from the **Preset** dropdown and enable the cameras you're interested in.

5. Use the **Vectors** tab in the properties to control the quality of the vectors used to produce the warp on the input images.

   Higher values tend to produce better warps, but at the expense of compute times.

Note: Vectors are only calculated at keyframes, with faster interpolation taking care of the frames in between. So, the more keyframes you have, the longer the stitch takes.

See Troubleshooting Stitches for more information.

6. Use the **Blend** dropdown to switch between the default **Alpha** mode, **Multi-Band**, and **Spherical Multi-Band** modes. Multi-Band blending can improve stitch results by matching low frequency color changes

over the course of the blend region, but can be slower to process than the default **Alpha** mode. Spherical Multi-Band blending can improve stitch results at the poles, looking up or down.

You can use the **Suppression** control to adjust the amount of blending applied between adjacent views when using **Multi-Band** blending. Lower values can help balance color and exposure between blended images, higher values are closer to the result from the default **Alpha** blending mode.



Note: Multi-band blending has no effect on per view C_STMaps exported from C_Stitcher. See Exporting to Preset Nodes for more information on exporting C_STMaps.

If you intend to export C_STMaps, you'll need to enable **Multi-Band** blending in C_Blender when you merge the separate streams back together. However, the results will not match those output by C_Stitcher because C_Blender uses resampled input images. See Blending Multiple Views Together for more information.

Tip: CaraVR adds a tab to the **Properties** panel of Nuke's Write node allowing you quickly select groups of views such as **stereo** and **cams** when rendering previews or stitches. You can still select views manually using the **views** dropdown, but **View Presets** can make the process faster.

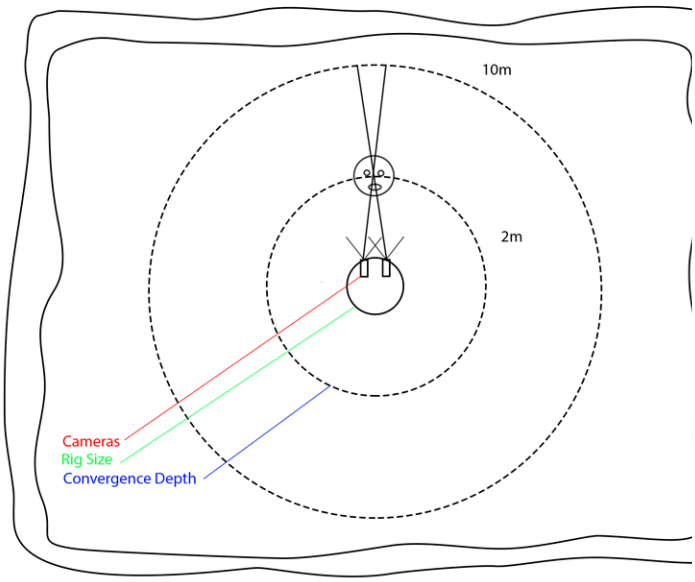
Setting the Convergence Point in Spherical Rigs

If you're using a spherical camera layout, you can set the **Converge** control to the depth at which the cameras overlap, allowing you to bring a single point of interest in the scene into focus.

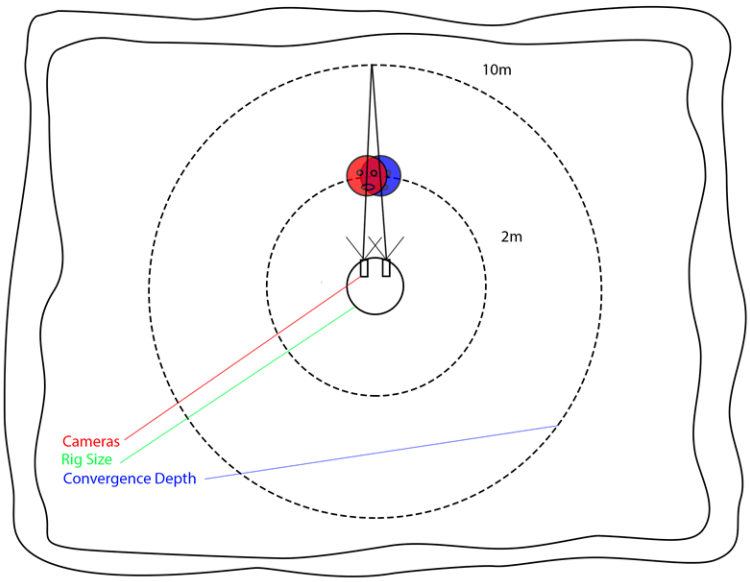


Note: Convergence doesn't have any affect on nodal rigs because all the cameras share the same origin.

Assuming that the cameras sit on a sphere, changing the size of the sphere affects the angle at which the cameras converge. C_Stitcher uses the **Rig Size** set in C_CameraSolver's **Cameras** tab to control the sphere. This normalizes the 3D representation of the rig into close to real world units, depending on the quality of the solve. The **Converge** control then defines the depth at which the stitch is converged, bringing the required point of interest into focus.



Convergence at 2m



Convergence at 10m

Setting the convergence point in the foreground may produce ghosting in the background, as shown on the left. The opposite is also true, as shown on the right, where ghosting may appear in the foreground if the convergence point is in the background.

In some rigs, you'll have to compromise between producing a correct stitch in the background with high convergence values and a correct stitch in the foreground with low convergence values.

Tip: If you set the **Converge** control in the C_CameraSolver, you can automatically import the convergence depth by enabling **Auto** to the right of the **Converge** control in the C_Stitcher **Properties** panel.

The example shows converge depths using the default **Rig Size** setting of **0.3** (or 30 cm). The **Converge** control defaults to **10**, which is a little high when considering the stools in the foreground.



Decreasing the **Converge** value to around **1.6** metres brings the stool into focus nicely.



Stitching Pre-Projected Images Together

If you have pre-projected images from a third-party application, or from an export from C_CameraSolver, C_Stitcher does not require C_CameraSolver metadata to stitch the images together. As long as the transform data and alpha masks available, you can stitch pre-projected images by performing the following steps:

1. Read in the camera footage and projection transform data.

> **Note:** The transform data must include sensible alpha masks, either in the alpha channel of the images or injected using C_AlphaGenerator nodes.

2. Add a JoinViews node to collect the views together, and then connect a C_Stitcher node to the JoinViews node.



3. In the C_Stitcher **Properties** panel, set the **Projection** dropdown to **LatLong**.
   The views are stitched together as if the C_CameraSolver metadata were present.



Troubleshooting Stitches

Some stitches are inevitably going to cause problems. There are a number of refinement workflows to assist CaraVR when stitching images together.

Tip: CaraVR adds a new tab to the **Properties** panel of Nuke's Write node allowing you quickly select groups of views such as **stereo** and **cams** when rendering previews or stitches. You can still select views manually using the **views** dropdown, but **View Presets** can make the process faster.

Improving Vector Quality

The vectors produced by C_Stitcher default to a reasonable compromise between performance and quality, but there a number of controls than you can adjust to improve the vector detail, and subsequent stitch.

Note: Increasing the values in **Vectors** tab controls increases processing time, but can improve your stitch results.

1. Increase the number of **Iterations** and **Warps** to see if the vector results improve:
   - **Iterations** - controls how may times the vectors are generated based on the previous iteration.
   - **Warps** - controls the range of pixel comparison performed during generation.

   For example, increasing the **Iterations** and **Warps** values in the following stitch improves the vector quality around the man's head.



Low **Iteration** and **Warp** values.



High **Iteration** and **Warp** values.

2. Next, increase the **Strength** and **Consistency** values:

- **Strength** - the tolerance applied when matching pixels between matched cameras. Higher values allow you to accurately match similar pixels in one image to another, concentrating on detail matching even if the resulting vectors are jagged. Lower values may miss local detail, but are less likely to provide you with the odd spurious vector, producing smoother results.
- **Consistency** - how accurately the same points between matched cameras are mapped to each other. Increase the value to encourage the vectors to match.

For example, increasing the **Strength** value in the following stitch improves the image quality around the man's head.



Low **Strength** values smooth the vector, but can miss local detail.



High **Strength** values add detail, but can cause jagged edges.

3. Increase the **Vector Detail** value to control the density of the calculated vectors. Higher values pick up finer disparity changes, but take longer to calculate.

   For example, increasing the **Vector Detail** value in the following stitch improves the image quality around the man's head.





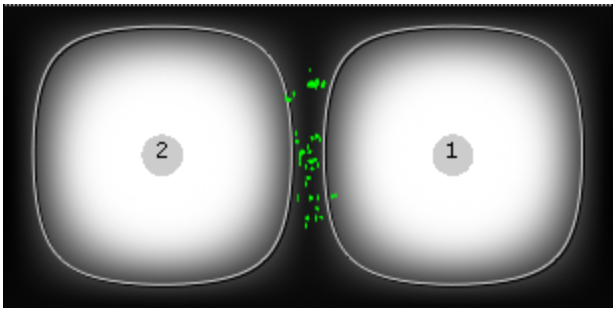
4. You can also increase the Temporal Window to increase the number of frames to either side of keyframes to perform temporal averaging over.

   Increasing this control produces smoother vectors, but significantly increases processing time.

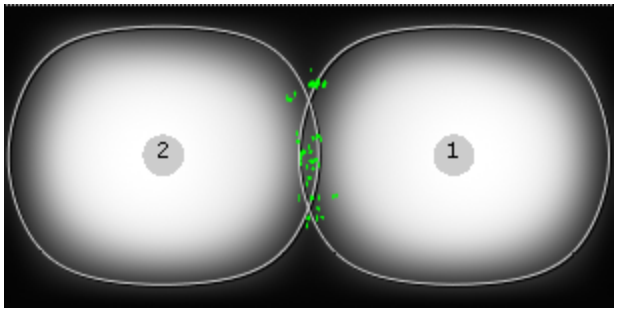
Improving Overlap for Back to Back Rigs

Two camera, back to back rigs can cause problems for C_Stitcher because of the lack of overlap between the cameras. If the stitcher doesn't seem to affect the output, you can try increasing the crop size in the **Cameras** properties in the C_CameraSolver to increase the overlap area.

1. Double-click the C_CameraSolver in the Node Graph to open its **Properties** panel.
2. Click the **Cameras** dropdown to expose the individual camera properties.
3. Scroll to the **crop_x** and **crop_y** properties and double-click to edit the values.



Lower **crop_x** values produce less overlap.



Higher **crop_x** values increase the overlap area.

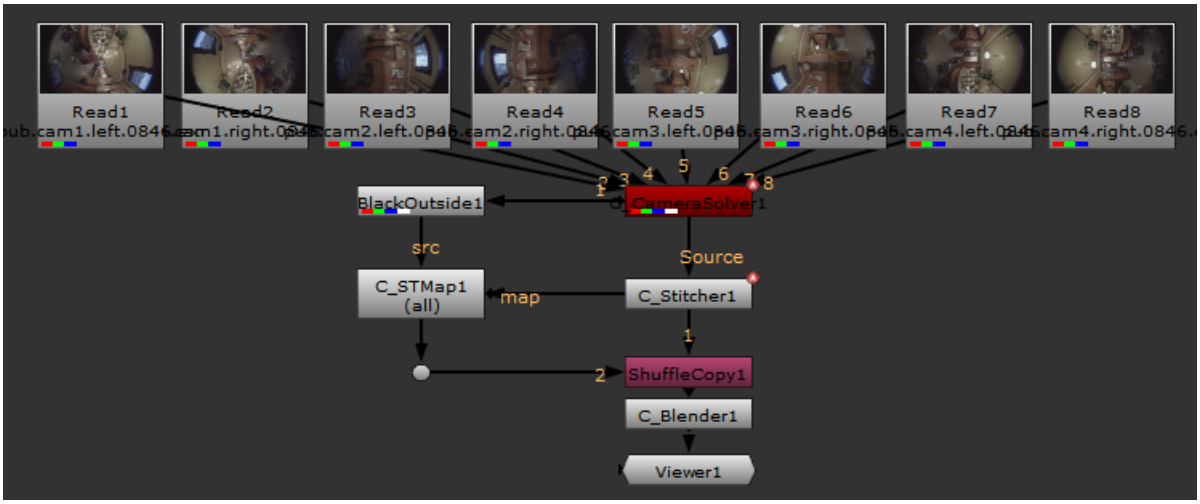
Exporting to Preset Nodes

C_Stitcher includes an **Export** dropdown, which automatically adds preset script components allowing you to quickly create node trees for common VR tasks.

Select the required operation from the dropdown and then click **Create** to add the script to the Node Graph:

- **STMap (split)** - creates a single C_STMap containing a **stitch_map** channel for all views in the stitch. You can examine individual views from the C_Stitcher output by selecting the view above the Viewer and then selecting the **stitch_map** channel from the **channels** dropdown.
- **STMaps (separate)** - creates a separate C_STMap node, containing a **stitch_map** channel, for each view in the stitch.

  This export is similar to **STMap (split)**, but the views are separated out into individual streams using OneView nodes.
- **Manual STMap Stitch (split)** - creates a workflow to distort the original inputs from C_CameraSolver through a single C_STMap containing a **stitch_map** channel for all views in the stitch. The views are then passed to C_Blender to create the stitch.



- **Manual STMap Stitch (separate)** - creates a workflow to distort the original inputs from C_CameraSolver through separate C_STMaps, containing a **stitch_map** channel, for all views in the stitch. The views are then recombined using a JoinViews and the final output passed to C_Blender to create the stitch.

  This export is similar to **Manual STMap Stitch (split)**, but the views are separated out into individual streams using two OneView nodes per view: one for the C_CameraSolver view and one for the C_STMap transform information.
- **PPass (split)** - creates a single C_STMap containing a **ppass_map** channel for all views in the stitch. You can examine individual views from the C_Stitcher output by selecting the view above the Viewer and then selecting the **ppass_map** channel from the **channels** dropdown.
- **PPass (separate)** - creates a separate C_STMap node, containing a **ppass_map** channel, for each view in the stitch.

  This export is similar to **PPass (split)**, but the views are separated out into individual streams using OneView nodes.
- **Manual PPass Stitch (split)** - creates a workflow to distort the original inputs from C_CameraSolver through a single C_STMap containing a **ppass_map** channel for all views in the stitch. The views are then passed to C_Blender to create the stitch.
- **Manual PPass Stitch (separate)** - creates a workflow to distort the original inputs from C_CameraSolver through separate C_STMaps, containing a **ppass_map** channel, for all views in the stitch. The views are then recombined using a JoinViews and the final output passed to C_Blender to create the stitch.

  This export is similar to **Manual PPass Stitch (split)**, but the views are separated out into individual streams using two OneView nodes per view: one for the C_CameraSolver view and one for the C_STMap transform information.

Stitching Stereographic Rigs

C_CameraSolver processes all views by default, so C_Stitcher takes care of splitting the views into stereo views for stitching.

1. Connect a C_Stitcher node to the output of your C_CameraSolver node.
2. In the C_Stitcher **Properties** panel, check **Enable Stereo Stitch**.

   Nuke adds **left** and **right** views and assigns them the default color automatically.
3. CaraVR automatically adds a single keyframe at the current frame, but you can add keyframes manually by either:

   • scrubbing in the Viewer and then clicking **Add Key** ( ) to add a keyframe at the current frame, or

   • clicking **Key All** to add keyframes throughout the sequence at the **Step** interval.

   If you have a lot of movement in your sequence, set the **Step** interval to a lower value so that the greater difference between frames is accounted for in the stitch. The opposite is true for more static sequences, where a higher value may produce better results.

   Adding keyframes manually can help in blurred areas between existing keyframes where the interpolation is inaccurate.

   The more keyframes you add, the longer the process takes, though you may get better results. The stitcher warps and blends the input images to create the final warped frames.
4. In the **Properties** panel **Cameras** tab, enable **Override Cameras** and click the **Linked** tab.
5. Disable the links between non-adjacent cameras using the C_CameraSolver overlay as a guide.

   Stereo stitches generally produce the best results when only adjacent cameras are linked, which also reduces the processing overhead.
6. When the stitch is complete, you can switch between the **left** and **right** views using the buttons over Nuke's Viewer.
7. On C_Stitcher's **Stereo** tab, use the **Left View** and **Right View** dropdowns to control which Nuke view is output as the left and right views.

Note: If you're using a Nokia OZO rig, you may find that setting the **Eye Separation** control to **0.086** more closely matches the true inter-axial distance of each of the lenses on the OZO rig.

Adjusting Convergence and Filtering in Stereo Rigs

You can adjust the **Converge** control to alter the depth at which the cameras overlap, allowing you to stitch at a particular depth in the scene reducing the amount of de-ghosting work you have to do at that depth.

Note: Convergence doesn't have any affect on nodal rigs because all the cameras share the same origin.

The **Converge** control defaults to **10**, which is considered safe in most cases. In some rigs, you'll have to compromise between producing a correct stitch in the background with high convergence values and a correct stitch in the foreground with low convergence values.

You can change the filtering algorithm used during stitching using the Filter dropdown. As a general guideline, the filters increase in quality at the cost of processing time as you move down the list. **Bilinear** filtering is the fastest and **Lanczos** the sharpest.

The difference in filtering can be hard to see with the naked eye, so in the example, the camera views have been swapped out with Checkerboard nodes for illustration purposes.



Stitching with a **Bilinear** filter.



Stitching with a **Lanczos** filter.

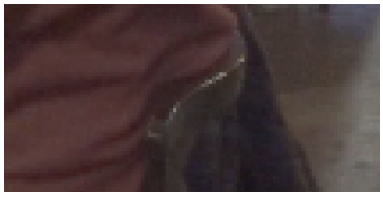
Note: The **Lanczos** filter can produce overshoot and undershoot, or bright and dark highlights at edges in the image, so you may have to clamp or adjust the color in the image if this is the case.

Adjusting Eye Separation

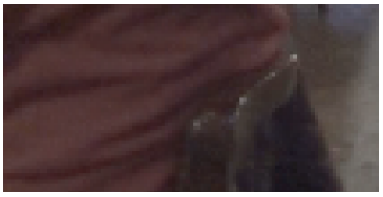
C_Stitcher also allows you to control the **Eye Separation**, or interpupillary distance (IPD), and compensate for latitudinal, or up and down, changes for a more comfortable result.

Adjusting the **Eye Separation** determines how far apart the two views are from a viewer's perspective. If you set the separation too low, objects in the scene appear crushed horizontally, but raising it too high can leave holes in the stitch.

The IPD is measured in the same units as the **Rig Size** control in the upstream C_CameraSolver properties, so adjust it accordingly.



Correct IPD.



Low IPD, relative to Rig Size.

VR headsets have trouble with stereo images at the poles of the 360 sphere, when you look up or down. This effect is minimized by pole merging, which gradually makes the left and right eye the same towards the poles.

C_Stitcher's **Falloff** controls determine how this pole merging is handled:

- **None** - no IPD adjustment occurs towards the poles.
- **Linear** - the views are merged gradually from the **Start Angle** specified toward the pole. Increasing the angle moves the start point toward the poles.

Note: Setting the **Start Angle** to 90 disables pole merging.

- **Cosine** - the views are merged smoothly toward the poles. Reducing the falloff shifts the transition in depth towards the poles.

Note: Setting the **Separation Falloff** to 0 disables pole merging.

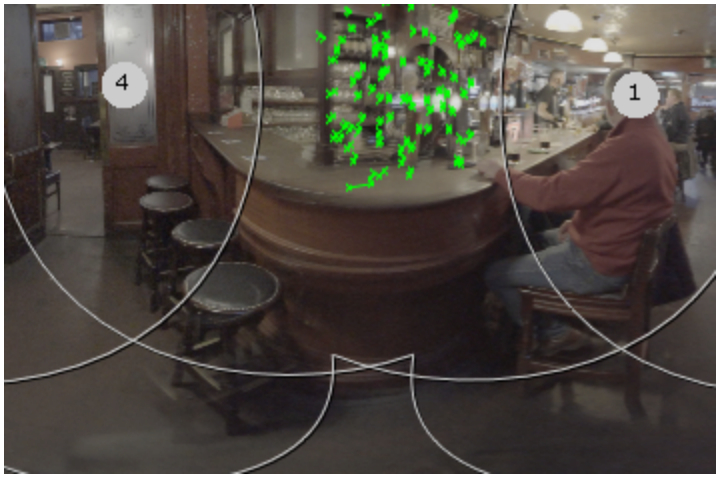
Rotoscoping to Correct Errors in the Source

Viewing the **main** preview from C_CameraSolver allows you to pick out areas of the solve that can be quickly cleaned up by editing the alpha masks in the source, before stitching the results. You'll still need to paint out objects that you don't want in the shot, such as the rig, but this workflow can cut down the amount of work required later on.

Examine the **main** preview to find areas that are bleeding from one camera view to another. The example shows part of a stool bleeding into overlap regions.



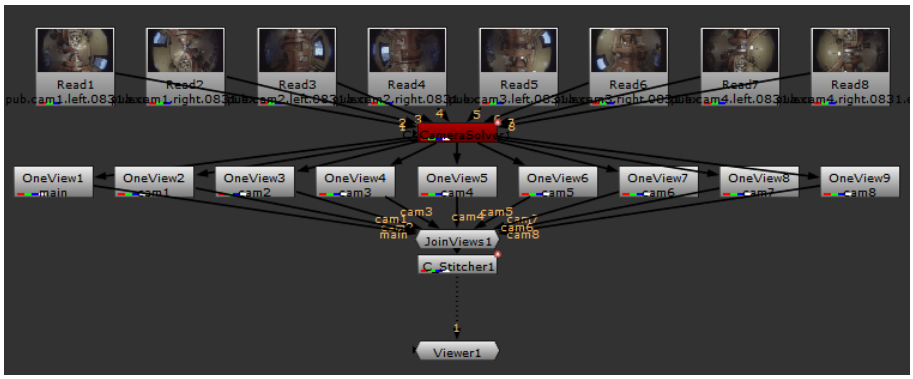
1. The first step is to look through the views to determine where the source is located. Turn on the C_CameraSolver overlay to display the cameras and overlap regions and then click the views above the view to pick the cameras.



In this example, cameras 1 and 4 contain the source.

2. Next, split out the views so that you can work on the individual shots. Select the C_CameraSolver in the Node Graph, press **Tab** to display the node selector, and then add a Split and Join.

Tip: If you only need certain views split out, you can use the Split and Join Selectively node instead. See Split and Join Selectively for more information.

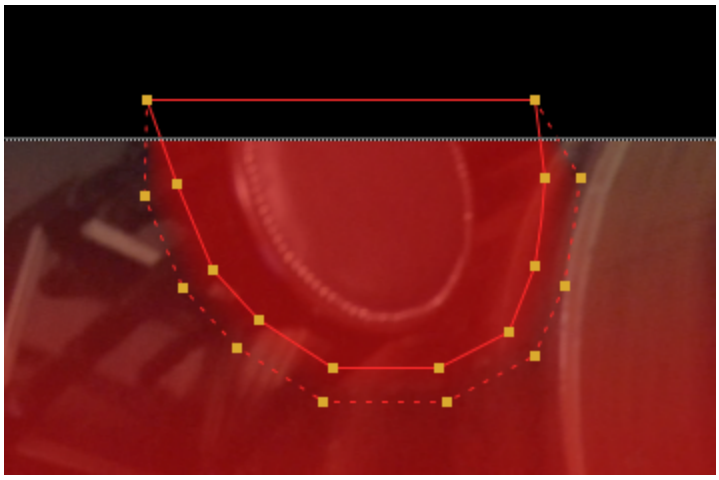


This is a vanilla Nuke operation, which adds a OneView node for each view present upstream, and then connects them together again using a JoinViews node.

Note: You can paint in latlong space as well by adding the Split and Join after C_Stitcher, but remember to add a C_Blender after the join so that the views are recombined into a single stream. See Blending Multiple Views Together for more information.

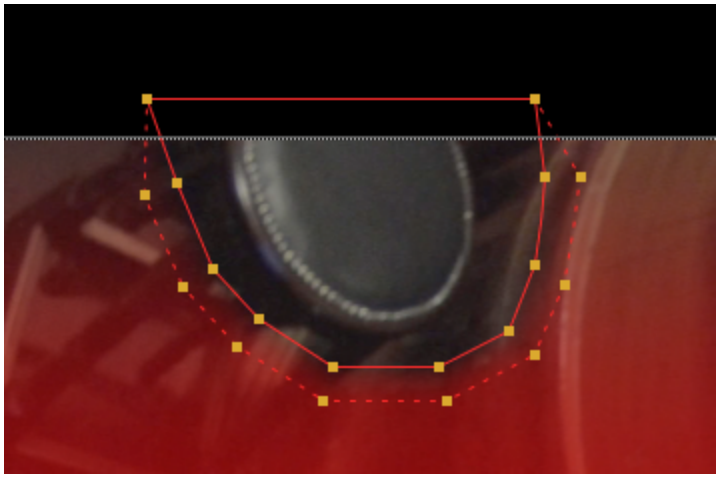
3. Connect the Viewer to the OneView node containing the output from the camera you're interested in and display the alpha as a Mat by pressing **M** over the Viewer. This overlays the alpha mask created by the solve.
4. Add a Roto node to the script and draw a bezier around the area of the shot you don't want.

Tip: It's a good idea to feather the roto shape to produce a better result. You can do this by selecting a point on the shape and pressing **E** a few times. See Using RotoPaint for more information.



5. In the Roto node's **Properties** panel, click the color wheel icon in the shapes list, and set the bezier color to black, or zero alpha.

   The bezier shape now trims the alpha mask created by C_CameraSolver so that it doesn't include the area containing the object.



6. Connect the Viewer to the C_Stitcher node and disable and enable the Roto node to view the results.





C_GlobalWarp

C_GlobalWarp produces a preview stitch using metadata, passed downstream from a C_CameraSolver, to help minimize ghosting in overlap regions. C_GlobalWarp includes controls to adjust a stitch to line up key features and allows you to add constraints to reduce warping on known straight lines in the rectilinear input images.



Results from C_CameraSolver.



Reduced ghosting after C_GlobalWarp.

Global warping relies on solve data and should be performed before stitching the solved camera output together, so ideally, place C_GlobalWarp between C_CameraSolver and C_Stitcher.

Matching and Solving Warps

C_GlobalWarp uses feature detection, similar to that employed by C_CameraSolver, to locate features in the overlap regions of adjacent camera pairs. The warp then uses these features to find the best-fit gridwarp deformation of each input view to reduce ghosting.

To warp a sequence:

1. Connect a C_GlobalWarp node to the output of your C_CameraSolver node.
2. CaraVR automatically adds a single keyframe at the current frame, but you can add keyframes manually by either:
   - scrubbing in the Viewer and then clicking **Add Key** ( ) to add a keyframe at the current frame, or
   - clicking **Key All** to add keyframes throughout the sequence at the **Step** interval.

     If you have a lot of movement in your sequence, set the **Step** interval to a lower value so that the greater difference between frames is accounted for in the stitch. The opposite is true for more static sequences, where a higher value may produce better results.

- clicking **Import** and selecting the node from which you want to import keyframes. You can import keyframes from any instance of C_CameraSolver, C_Stitcher, C_GlobalWarp, or C_ColourMatcher in the script.

  Any matches in C_CameraSolver are also imported by default. If you don't need matches from the solver, disable **Import available matches**.

Note: C_Stitcher only computes optical flow camera warps at designated keyframes, it ignores upstream animation that does not fall on the same keyframes. Use the **Import** button to copy over all keyframes from C_CameraSolver. This way, any additional per-camera, manual keyframes added in the **Cameras** tab are also copied.

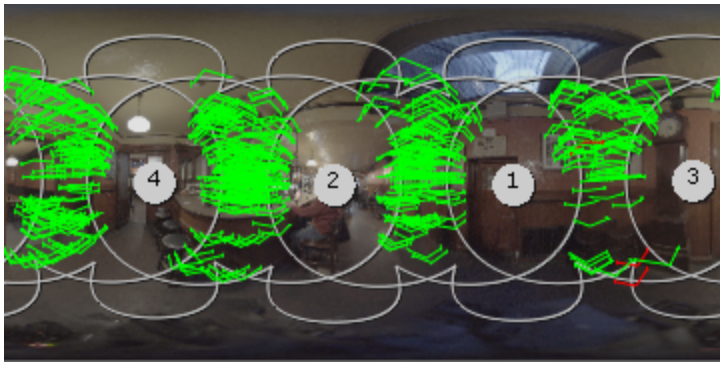
Adding keyframes manually can help in blurred areas between existing keyframes where the interpolation is inaccurate.

The more keyframes you add, the longer the process takes, though you may get better results. The global warper blends the input images to create the final warped frames.

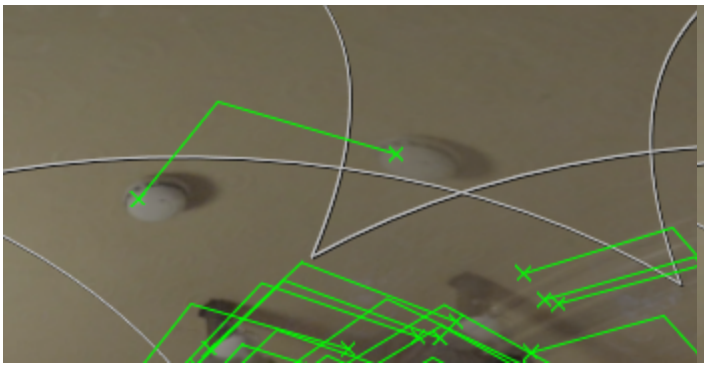
3. Click **Match** to compare keyframes on overlapping cameras for shared features. CaraVR only looks for matches in cameras that overlap by default.

Note: **Match** calculates the feature matches for all cameras in the rig at the same time, so there is no need to calculate them separately for left and right images in a stereo setup. See Stitching Stereographic Rigs for more information.

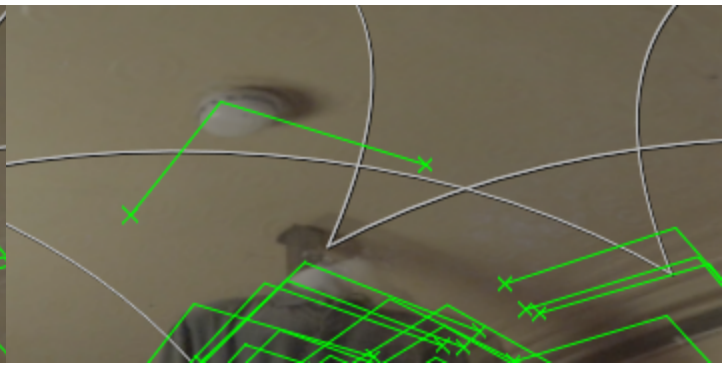
4. Click the **Camera Matches** button to display the matches in overlapping regions, matches aren't displayed by default.



Camera matches represent a single feature in both views, and the estimated warp destination of the feature. For example, the image shows a smoke detector in views 2 and 4 and the warp destination point in between.



Pre-warp matches.



Warp destination.

5. Click **Warp** to calculate the warp.

Note: The warp is calculated for all cameras in the rig at the same time, so there is no need to calculate them separately for left and right images in a stereo setup. See Stitching Stereographic Rigs for more information.

The warped preview image displays in the Viewer. See Troubleshooting Matches and Warps for information on improving warp results.

When you're satisfied with the initial global warp, you can add constraints to reduce warping on known straight lines in the rectilinear input images. See Constraining Warps for more information.

Troubleshooting Matches and Warps

Some sequences are inevitably going to cause problems while matching and warping. There are a number of refinement controls on the **Properties** panel, **Settings** tab to assist CaraVR when warping sequences.

These controls are often a trade-off between processing speed and quality or local and global accuracy. No two sequences are alike, so there's no silver bullet, but adjusting the controls on the **Settings** tab can help you achieve a better result.

Note: You won't see the effect of your fine-tuning until you click **Warp** to apply the new settings.

Grid Size

Sets the size of the grid used by C_GlobalWarp to apply the warp operation in each camera view.

Using larger grids can produce better warping results, but at the expense of longer computation time. Very large grid size values should only be used when a large number of camera matches are available..

Tip: Click the **Camera Matches** button to display the matches in overlapping regions, matches aren't displayed by default.



Low **Grid Size** value.

High **Grid Size** value.

Match Strength

Sets the strength bias for the camera matches. Increasing the strength of a match forces the warped output to converge at the camera matches.



Low **Match Strength** value.

High **Match Strength** value.

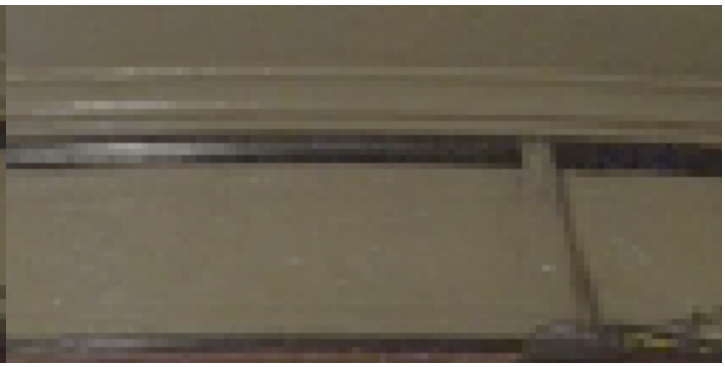
The **User Match Strength** control performs a similar function, but only affects matches that you added manually during camera solving. User matches are given more weight by default and can improve results significantly. See Adding User Matches for more information.

Smoothness

Controls how image edges are used as clues for sharp transitions in the scene. The higher the value, the more regular the deformation within each view. Lowering the **Smoothness** too far can introduce jaggedness between overlap regions.



Very low **Smoothness** value.



High **Smoothness** value.

Consistency and Temporal Window

Sets how consistently keyframes map to each other over time and how many frames each side of the keyframes should be use to calculate the result. Increasing the **Consistency** value forces the warp to match between views at keyframes. Increasing the Temporal Window increases processing time, but can improve results.



Low **Consistency** value.



High **Consistency** value.

Weight Kernel

Allows you to give different weights to each camera match during warping. The weights are a function of the camera RMS error:

- **None** - all matches have the same influence on the image.
- **Linear** - matches have diminishing influence as the RMS error increases.
- **Gaussian** - similar to Linear weighting, but more forgiving with low RMS error values.



Note: If the **Error Threshold** on the **C_GlobalWarp** tab is set to **0**, this control has no effect.

Vector Detail

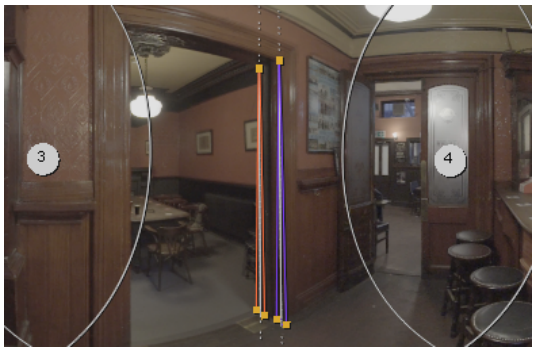
Controls the density of the calculated vectors. Higher values pick up finer disparity changes, but take longer to calculate.

When you're satisfied with the fin-tuned global warp, you can add constraints to reduce warping on known straight lines in the rectilinear input images. See Constraining Warps for more information.

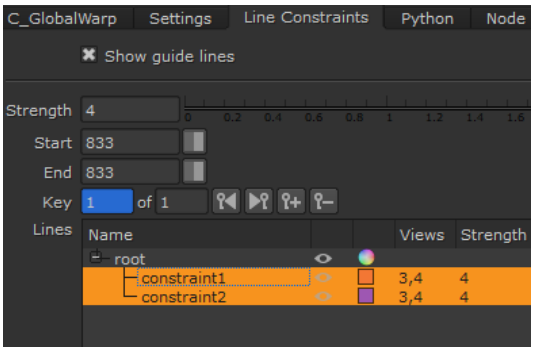
Constraining Warps

Warping a sequence can deform straight lines as part of the global warp applied to the camera views. Adding constraints to known straight lines in the rectilinear input images can reduce this deformation by pinning areas in the global warp.

Typically, constraints are applied to a feature that appears in two adjacent views and then C_GlobalWarp ensures that those marked areas are warped to the same destination.



Constraints marked in Cam3 and Cam4.



Constraints in the **Properties** panel.

Viewing Warped Images

You can view the full output or individual views using the tools at the top-left of the Viewer. Each view is available as the warped result or as the source view in latlong and rectilinear.



1	Output Mode	Switches the Viewer between the warped output and the source latlong from C_CameraSolver.
2	Output View	Sets the view to output. You can select individual views by camera, or examine the **full output**.
3	Previous/Next View	Swap quickly between output views.
4	Projection	Switches between latlong and rectilinear display in the Viewer.
5	Center View	Click to center the view when in rectilinear or single camera view. This button has no effect in **full output** mode.

6	Reset Rotation	Click to reset any rotation applied to the cameras in C_GlobalWarp.
7	Show Cameras	Click to toggle the camera overlay on and off.

Adding Constraints

Constraints pin areas of camera views to ensure that the warp of adjacent cameras have the same destination and all the points overlapping the line overlay are mapped to a straight line. You cannot add constraints in the **full output** mode, so make sure you have either:

- an individual camera view selected using the **view** buttons above the Viewer, or
- if you're in the **main** view, a camera selected from the **Output View** dropdown.

To add constraints:

1. Locate a known linear feature that occurs in two adjacent cameras. Examples include door frames, table tops, tiles, and so on.
2. Select the first view containing the feature and click the **Add Point** button.

Tip: You can also add points by holding **Ctrl**/**Cmd**+**Shift** and clicking in the Viewer.

3. Click in the Viewer where the constraint starts and ends.



A constraint in a rectilinear view.



The same constraint projected into latlong space.

4. Select the second view containing the feature and click in the Viewer where the constraint starts and ends.

   If you switch back to the **main** view or **full output**, the constraint points connect up between views.



A two-view constraint in the main view. A close up of the constraint.

5. Select the constraint in **Properties** > **Line Constraints** > **Lines** table and set the **Strength** and frame range controls.
   - **Strength** - sets the strength bias for the constraint. Increasing the strength of a constraint forces the warped output to converge at the constraint. Constraints have a similar weighting to user matches in C_CameraSolver.
   - **Start** and **End** - sets which frames the constraint influences. You might use these controls if the constraint feature moves out of shot or becomes occluded.
6. Set keyframes for the constraint if the associated feature moves or changes shape. See Animating Parameters for more information on animated key frames. You can also track constraints using C_GlobalWarp's **Tracking** controls. See Tracking Constraints for more information.
7. Add as many constraints as required in the areas you want to preserve.

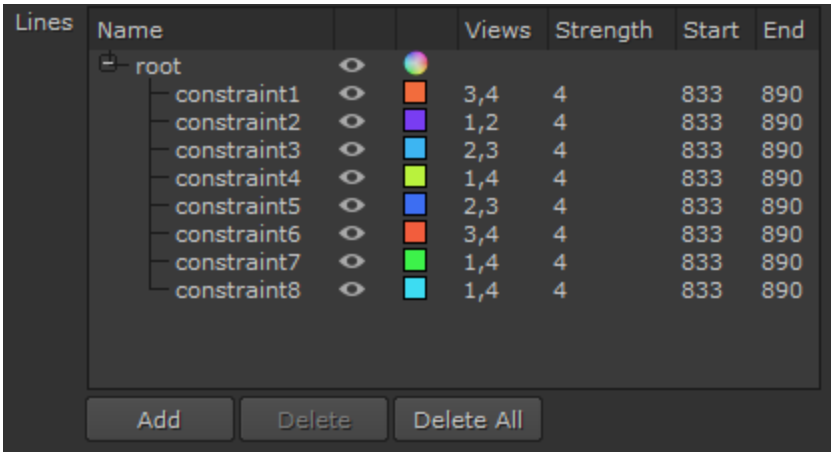
   Make sure you have finished drawing one constraint before starting another, otherwise the next point is added into the previous constraint. The easiest way to complete a constraint is to click an empty space in the **Lines** table.

   If you're adding points to an existing constraint by holding **Ctrl**/**Cmd**+**Shift** and clicking in the Viewer, you can start a new constraint by pressing **N** (the New Constraint keyboard shortcut), and then all subsequent points are added to the new constraint.

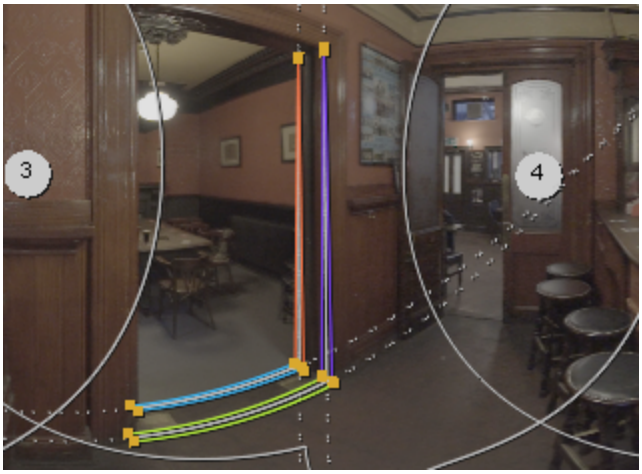
Tip: If you constrain too many areas, the global warp might have difficulty producing sensible warp because 'everything' is pinned.

The **Lines** table displays all the constraints you've added, including which **Views** they're in and the **Strength** bias assigned to them.

Note: You can show and hide individual constraints using the button, but remember that hidden constraints do not affect the warp.



8. When you're finished constraining, click **Warp** to recalculate the global warp taking the constraints into account. You can add more constraints at any point and click **Warp** again to integrate the new data.



Constraints pre-warp.

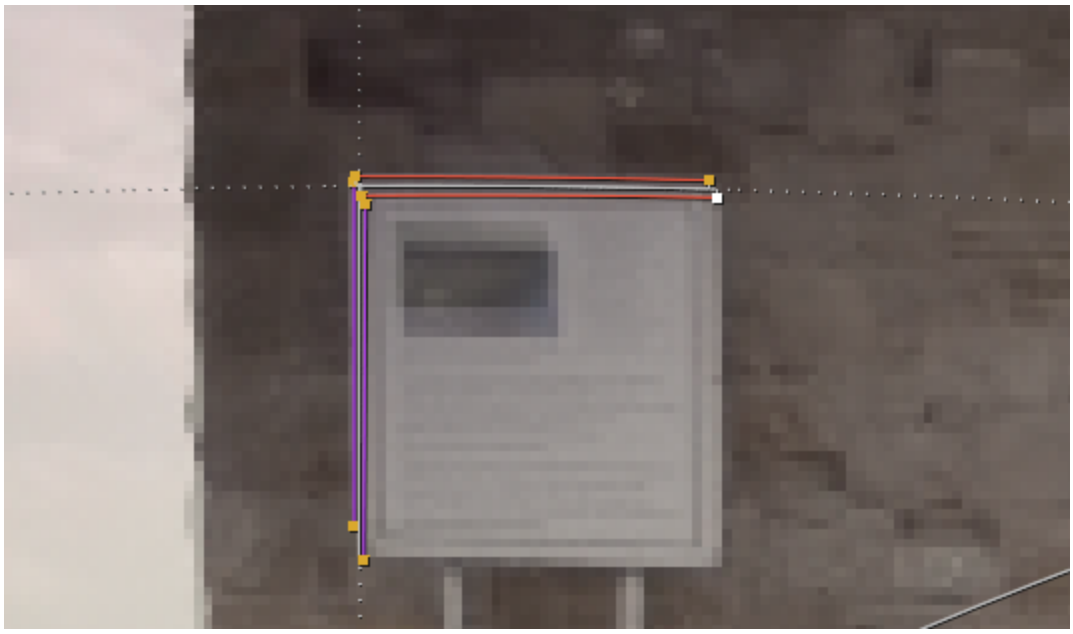


The resulting warp.

Tracking Constraints

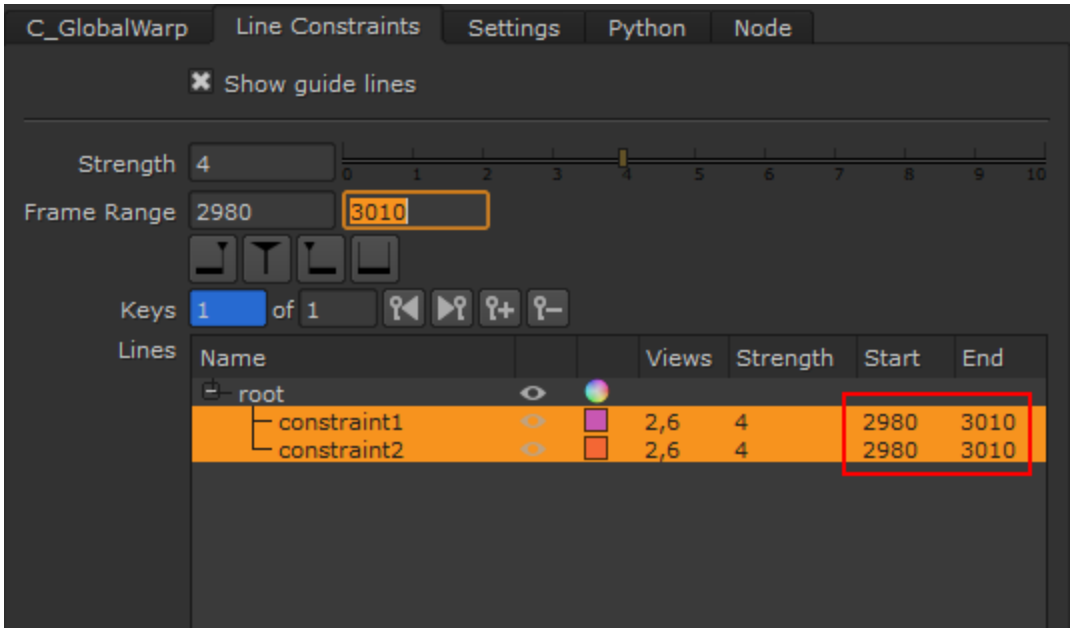
Constrained areas are likely to move through a shot, meaning extra keyframing work to maintain accuracy. C_GlobalWarp's tracking feature can speed up this process by automating some of the work for you. You can also export tracks to individual C_Tracker nodes and then re-import the refined tracks for difficult sequences.

1. Add your constraints as normal. For example:



See Adding Constraints for more information.

2. Select the constraints you want to track in the **Properties** panel **Line** table.
3. Set the **Frame Range** you want to track. The frame range for the individual constraints is updated.



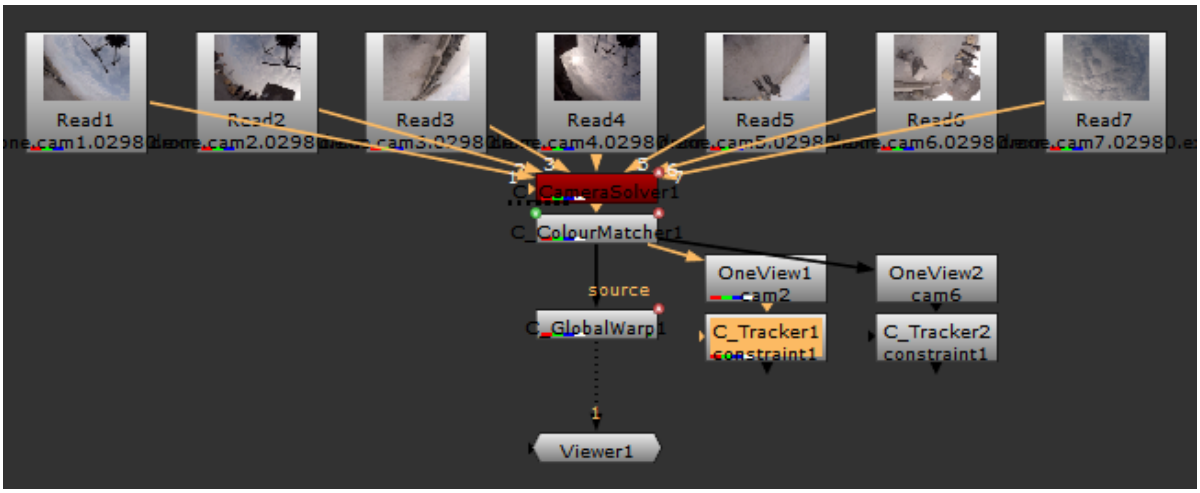
4. Click **Track** to begin tracking the selected constraints.

Note: If you click **Track** without setting a **Frame Range**, an **Enter tracking range** dialog is displayed allowing you to specify the range.

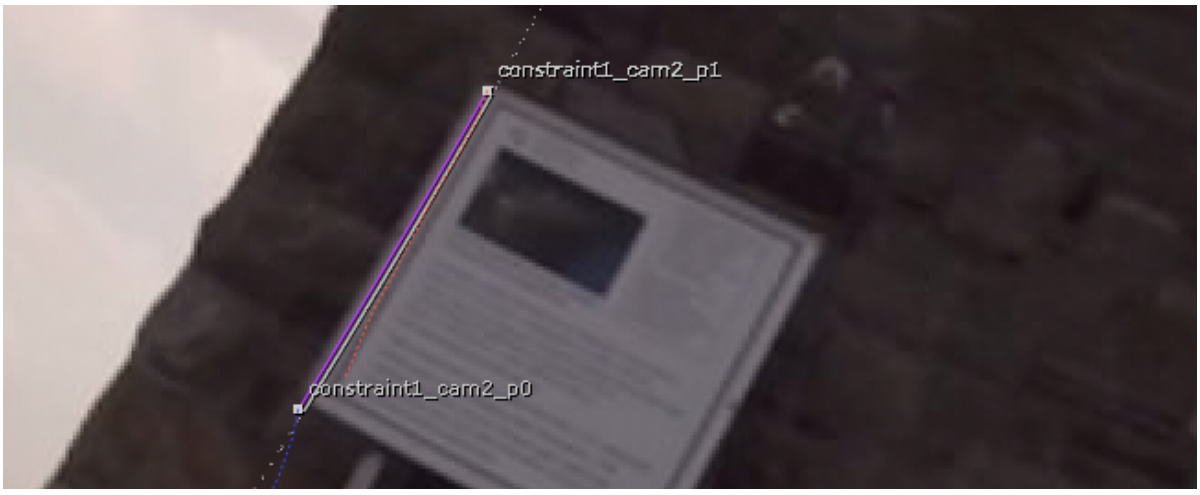
C_GlobalWarp tracks the selected points over the frame range.

5. You can adjust the tracking parameters in the **Properties** panel under **Settings** > **Tracking**:
   - **Step** - sets the number of frames between tracking keyframes. Increasing the steps between keyframes speeds up processing, but can impact accuracy.
   - **Patch Size** - sets the size of the image patch around constraint points. Increasing the patch size can improve track quality, but larger patches increase processing time.
   - **Error Threshold** - sets the image change threshold a constraint track can tolerate before terminating. Reducing this threshold makes tracking more tolerant to image changes, producing longer tracks.
6. You can export the tracks to C_Tracker nodes to improve the results further by selecting the required constraint and clicking **Export**.

   Each constraint is exported per view using OneView nodes. For example, a constraint that occurs in **cam2** and **cam6** produces a Node Graph similar to that shown in the image.



7. Double-click the C_Tracker node to display tracking keyframes for the required view.

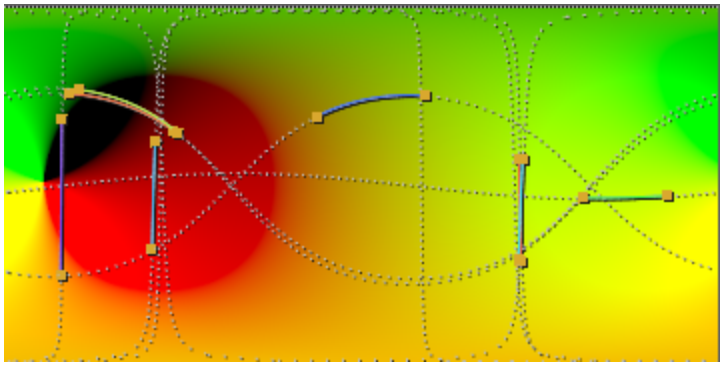


   See Tracking and Stabilizing for more information on how to use C_Tracker in CaraVR.
8. When you're happy with the result from C_Tracker, open the C_GlobalWarp **Properties** panel, select the exported constraint(s), and then click **Import**.

The improved track is imported from C_Tracker and you can then warp the image as normal. See Adding Constraints for more information.

C_GlobalWarp Output

C_GlobalWarp's **Cam Warp** control is set to **Unwarped** by default, so the warp does not affect the camera views and latlong output passed down the node tree, but rather writes the warp to a **stitch_map** layer that C_Stitcher can use to influence the final stitch.



If you want to pass camera warps downstream, set the **Cam Warp** control to **Warped**. The individual camera views are then rendered with the warp applied.

Additionally, C_GlobalWarp overwrites the rig rotation metadata from C_CameraSolver. The Viewer toolbar includes a **resetRotation** button that resets the rig rotation to the original values read from the metadata. C_Stitcher then processes the metadata in the same way as C_CameraSolver metadata. See Stitching Images Together for more information.

C_ColourMatcher

C_ColourMatcher aims to produce a global gain-based color correction across all the views in a rig to balance out differences in exposure and white balance. It solves across all cameras to find the minimum required gain changes to ensure color similarity.

Note: C_ColourMatcher does not compensate for color shifts not resulting from exposure- or gain-type differences. It also does not deal with gamma- or offset-type differences.

Color matching relies on solve data and should be performed before stitching the solved camera output together, so ideally, place C_ColourMatcher between C_CameraSolver and C_Stitcher.

Tip: The CaraVR plug-ins include example scripts in Nuke's toolbar under **CaraVR** > **Toolsets**.

Matching Colors Across All Cameras

To color correct across all cameras in a rig:

1. Add a C_ColourMatcher node to the Node Graph and connect it to the output of the C_CameraSolver node.
2. CaraVR automatically adds a single keyframe at the current frame, but you can add keyframes manually by either:
   - scrubbing in the Viewer and then clicking **Add Key** () to add a keyframe at the current frame, or
   - clicking **Key All** to add keyframes throughout the sequence at the **Step** interval.
   - clicking **Import** and selecting the node from which you want to import keyframes. You can import keyframes from any instance of C_CameraSolver, C_Stitcher, or C_ColourMatcher in the script.

Note: C_Stitcher only computes optical flow camera warps at designated keyframes, it ignores upstream animation that does not fall on the same keyframes. Use the **Import** button to copy over all keyframes from C_CameraSolver. This way, any additional per-camera, manual keyframes added in the **Cameras** tab are also copied.

You can add keys on multiple frames to attempt to match a range of colors as part of the solving process.

Note: C_ColourMatcher can not calculate time variant corrections, such as color shifts over time due to camera movement. You can, however, use multiple passes of the node for the different time ranges.

3. In the **Analysis** section, select **Exposure** or **Exposure and Colour** from the **Match** dropdown, depending on your requirements.
4. Click **Analyse** to match exposure and color across the cameras in the rig.

   CaraVR calculates the required gain per camera and displays the results in the **Properties** panel.



The original image. The same image after exposure and color correction.

5. Adjust the global **Exposure** control to adjust all the cameras in the rig simultaneously.

   If you notice banding in the areas between adjacent cameras, try switching the C_Stitcher's **Blend** mode to **Multi-Band** or **Spherical Multi-Band** blending.



6. If you're using a spherical camera layout, you can adjust the **Converge** control to alter the depth at which the cameras overlap, allowing you to bring particular areas in the scene into focus. If you've adjusted this value upstream, make sure you set the **Converge** control accordingly.

   See Troubleshooting Matches and Solves for more information.

7. Once the correction is complete, you can either:
   - export the correction to Grade and Exposure nodes so that you can make minor adjustments to the result, or
   - export the correction to OCIO CDL Transform nodes for use elsewhere in the script.

See Exporting to Preset Nodes for more information.

Exporting to Preset Nodes

C_ColourMatcher includes an **Export** dropdown, which automatically adds preset script components allowing you to quickly create node trees for common VR tasks.

Select the required operation from the dropdown and then click **Create** to add the script to the Node Graph:

- **Grades (split)** - creates a single Grade and Exposure node containing the calculated exposure and color differences.
- **Grades (separate)** - creates separate Grade and Exposure nodes for each view containing the calculated exposure and color differences.
- **OCIO CDL Transforms (split)** - creates a single OCIOCDLTransform and Exposure node containing the calculated exposure and color differences.
- **OCIO CDL Transforms (separate)** - creates separate OCIOCDLTransform and Exposure nodes for each view containing the calculated exposure and color differences.



Note: OCIOCDLTransform enables you to import and export the color information using **.cc** XML files. See the Nuke Online Help for more information.

Compositing Workflows

The **Compositing Workflow** nodes deal with the bread and butter functions required in compositing, but within a VR environment.

- **C_Blender** - is used as a Merge node to combine all images together after manually correcting a stitch. See Blending Multiple Views Together for more information.
- **C_Blur** - similar to Nuke's standard Blur, but allows you to apply blur to a latlong image and produce a sensible result across the entire frame, as if the blur were applied to a rectilinear image all around. See Applying LatLong Blur for more information.
- **C_Tracker** - similar to Nuke's standard Tracker, but with the addition of CameraTracker-style auto-tracking and calibrated for pattern tracking in lat-long space. See Tracking and Stabilizing for more information.
- **RayRender** - CaraVR adds to Nuke’s RayRender a number of advantages over scanline-based renderers in a VR context, including:

- the ability to render polar regions in a spherical mapping correctly, without the artifacts inherent in scanline-based rendering in these areas, and
- support for lens shaders, allowing slit scan rendering of multi-view spherical maps. This provides a more natural viewing environment for VR material.

See Rendering Using RayRender for more information.

- **C_STMap** - a GPU accelerated version of the Nuke STMap node, C_STPMap allows you to move pixels around in an image using a **stitch_map** or **ppass_map** to figure out where each pixel in the resulting image should come from in the input channels. You can re-use the map to warp another image, such as when applying lens distortion. See Warping Using STMaps for more information.
- **C_AlphaGenerator** - a convenience tool that can be used to create a rectangular or elliptical mask in the alpha channel. See Generating Alpha Masks for more information.
- **C_GenerateMap** - outputs a **stitch_map** or **ppass_map** for UV or XYZ coordinates, which can be warped and then piped into C_STMap. See Generating Stitch and PPass Maps for more information.
- **Split and Join Selectively** - similar to Nuke's Split and Join node, but gives you more control over which views are affected. See Split and Join Selectively for more information.

Transforming and Projecting with SphericalTransform

SphericalTransform converts images between different projections, including 360 work, and takes advantage of Blink GPU acceleration. These view projections can be divided into two broad categories:

- full frame, such as **Latlong**, encompassing the entire 360 world around a single point, and
- partial frame, such as the **Rectilinear** view that Nuke was designed to work in.

The SphericalTransform node can be used for common Nuke operations on 360 material, such a rotoing, comping, and tracking. SphericalTransform allows you to configure the input and output projections for the desired conversion. For partial frame projections, additional projection space parameters are enabled on the **Input** and **Output** tab for the specific camera parameters, such as **focal length**, **sensor size**, and so on.

Use the Rotation controls to adjust the **Input** and **Output** from a single point governing **Look** position, two points going **From/To**, **Pan/Tilt/Roll**, or **Full Rotation** angles with control over rotation order.

The **Output** rotation is also controllable using an in-viewer control system. Hold down **Ctrl**/**Cmd**+**Alt** and left-click and drag to move the image around, setting the pan and tilt setting. Add **Shift** to lock into a single dimension for the movement. In a partial frame projection, use the right mouse button to set the **focal length**, in essence zooming in and out.

SphericalTransform can convert between the following projection modes:

Projection Name	Example	Description
Latlong		**Latlong**, or equirectangular, projections are the most common full 360 frame projection. Many VR pipelines use them for both ingest and export due to their simplicity and wide use. Working in latlong space can be problematic due to its unfamiliar mapping, that many compression techniques were not designed to handle, and inefficiencies towards the poles, where many pixels represent a single pixel in the output.
Cubemap		**Cubemap** projections are another full 360 projection. Each of the six faces is essentially rectilinear, so the data can be more familiar to work in. Faces can be packed in a number of ways: • **Image** - all faces are placed in a single image and packed according to the **Packing** control. The default is **LL-Cross** as shown in the example. • **Views** - each face creates a view in the stream. The view is named using the following convention: `cubemap_<direction><axis>` Where **<direction>** can be **pos** or **neg**, for positive and negative, and **<axis>** can be **x**, **y**, or **z**. • **Faces** - each face is a placed in a separate image stream, but due to Nuke's limitations, there can only be a single face output in this mode. You can choose the face to output using the **Face** control. When in input mode, you get six separate inputs, labeled according to the face they represent.
Rectilinear		**Rectilinear** projection is a partial frame, standard projection you're most likely familiar with. **Tip:** Remember when you're going to or from rectilinear, you're only able to cover part of the frame (up to virtually 180 degrees). As a partial frame projection this enables extra parameters on the **Input** and

Projection Name	Example	Description
		Output tabs (depending on if you've picked it as an input or output projection). These govern the camera parameters, such as **focal length**, **sensor size**, and so on that are not applicable in full frame projections.
Fisheye		**Fisheye** covers a number of projections, all of which emulate common optical models used in fisheye lenses. These are all partial frame projections, so enable specific camera parameters similar to the **Rectilinear** projection type. Additionally, you can select the particular model to use: • **Stereographic** is not widely used in optics, Samyang Optics being one of the few to employ it. The **Stereographic** model forms the basis of the **little planet** projection, where the center of the projection is the nadir. SphericalTransform ships with a **LittlePlanet** properties preset, which applies such a look to a latlong with a horizon line centered vertically. • **Equidistant** is the default setting and matches the zeroed model employed in the fisheye distortion estimation employed by CaraVR's C_CameraSolver node. **Equidistant** is often considered the ideal model, as its response is a balance between the curves of various models. Other tools without this level of control most likely employ this model. • **Equisolid** is the most frequently found model in practical optics. • **Orthographic** is a classical 'perfect' response model that sees little use in practical optics. It does, however, match the fisheye model in Nuke.
Mirrorball		Mirrorball produces an image of a reflective ball, scaled up so that the ±XY of the image are at the edge of the format.

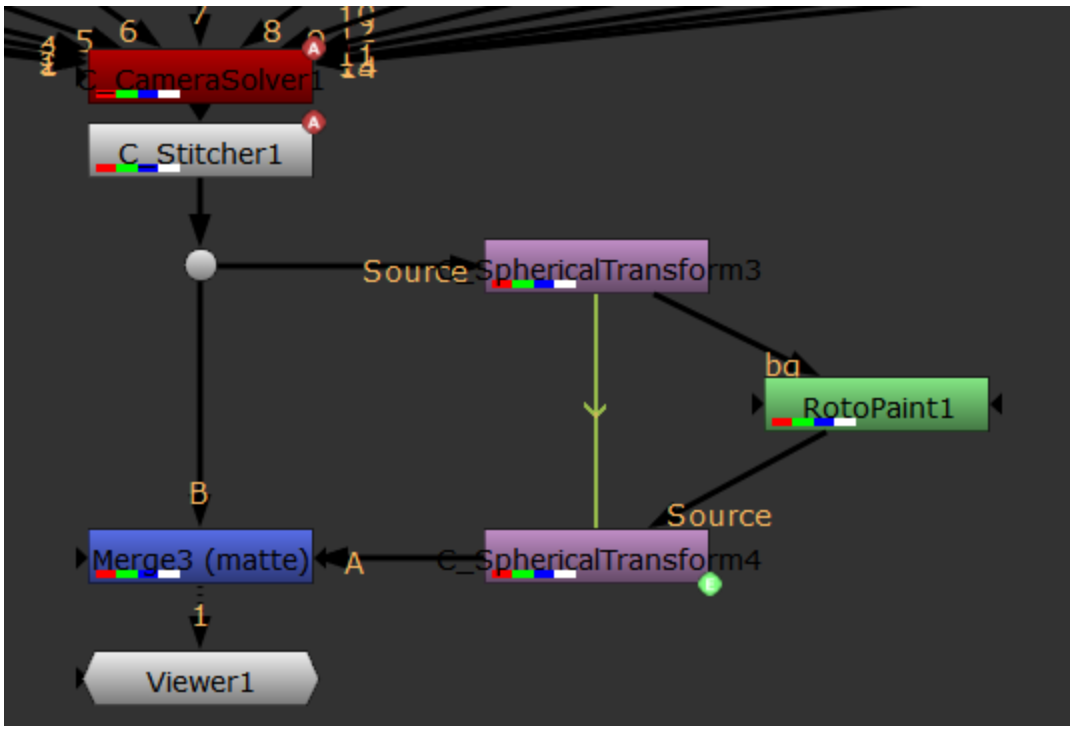
Compositing in 360 Footage

Nuke nodes often work under the assumption they are applied to rectilinear shots, and as an artist, you might prefer working with material in rectilinear space, for example when rotoing. SphericalTransform allows you to work on 360 material quickly and easily, taking care of the required projection changes for you.

Painting in Latlong Space

CaraVR, with NukeX or Nuke Studio licenses, ships with a toolset to make these conversions quick to perform, but you can recreate the node tree shown manually.

1. Select the point in the node tree you want to apply the paint to, and navigate to **CaraVR** > **ToolSets** > **Latlong_RotoPaint**.
2. The toolset is added to the Node Graph at the specified node.



The toolset contains two SphericalTransforms converting the image into a rectilinear preview, so you can add your paint, and then back into latlong space. Apply any paint required in the RotoPaint node and it'll be mapped back to 360, and then merged back into the original image.

3. Connect the Viewer to the first SphericalTransform node and open its **Properties** panel to activate the **directionInput** Viewer widget.
4. Move the **directionInput** widget to the area you're interested in to move around the image.

Tip: You can Adjust the **Angle** control to rotate the image relative to the **directionInput** widget to get the view you need.

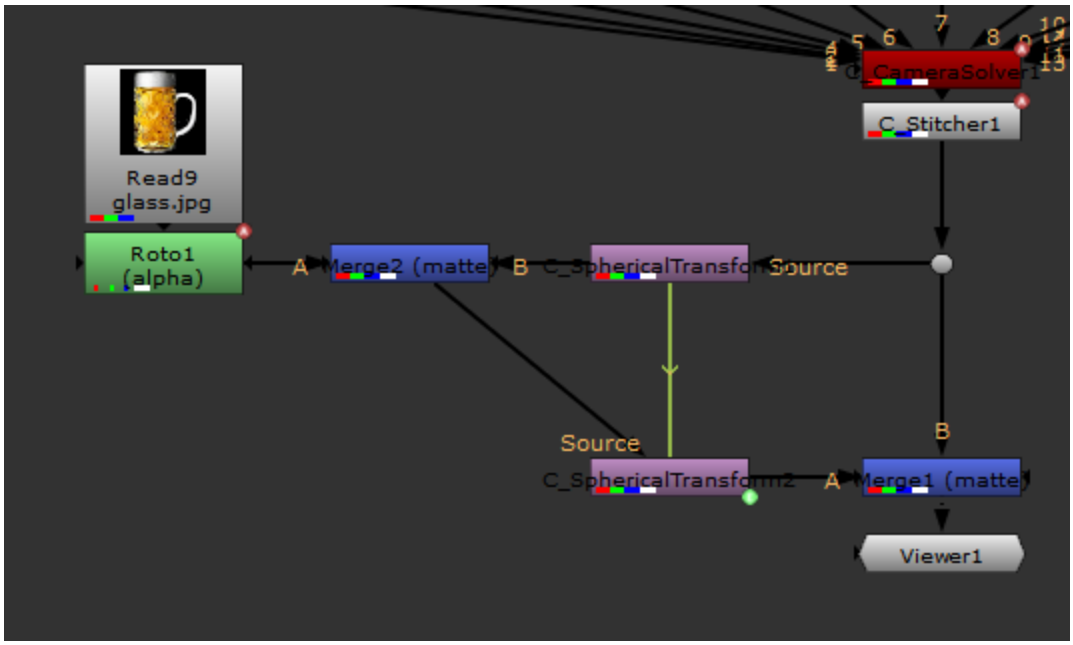


5. If necessary, you can increase the field of view of the rectilinear projection by decreasing the output **focal length** of the first SphericalTransform node.
6. Connect the Viewer to the RotoPaint node and paint as required.
7. To view you paint in latlong space, connect the Viewer to the second SphericalTransform node.

   The Merge node in the toolset uses the alpha channel you painted into to merge your paint back over the source 360 shots, so you get fewer filtering hits for the majority of the image.

Comping in Latlong Space

Similar to the RotoPaint toolset, the **Latlong_Comp** toolset in CaraVR, allows you to merge a rectilinear object into a 360 environment.



Note: Ensure the alpha channel of the rectilinear object carried through to the final merge, so that there's minimal filtering hits on the other parts of the frame.

Transforming Using Metadata

C_SphericalTransform can be very useful when transforming and rotating 360 material, such as when trying to straighten a stitch horizon line, but multiple transforms cause filter hits, degrading the image as it is transformed.

One way around this is to transform the image by passing metadata downstream without altering the image until the final transform, which results in a single filter hit. The **Metadata** control determines how and when the upstream metadata and transforms described in a node are applied:

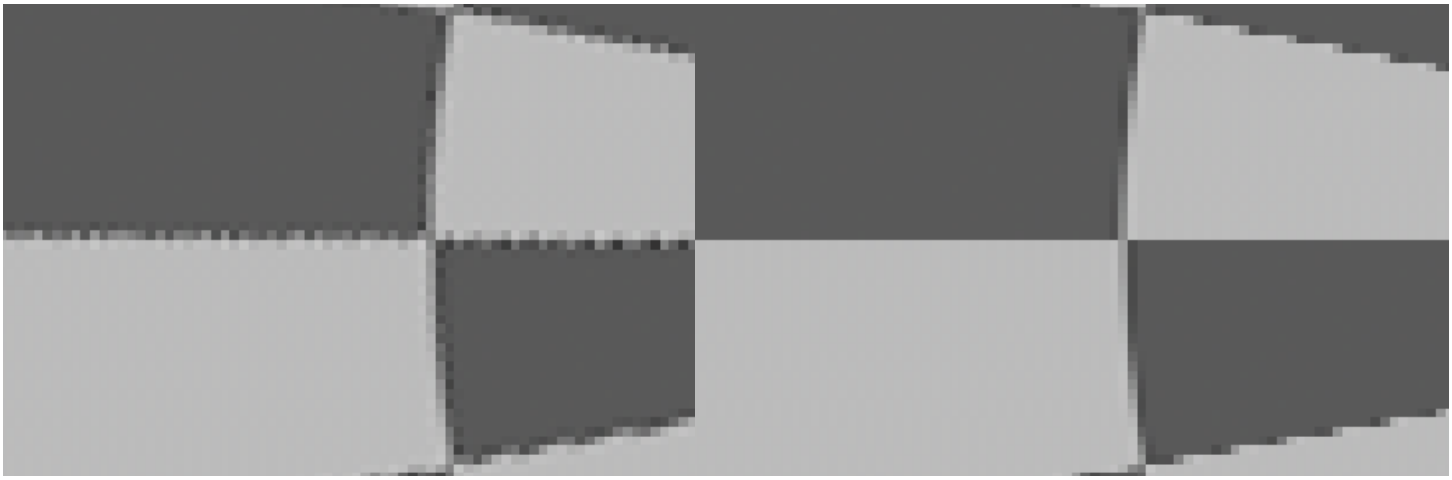
- **Ignore** - any metadata present in the stream is passed downstream unaltered.
- **Apply** - rotates the image according to the metadata in the stream and the total transform described by the controls in the C_SphericalTransform node.

Note: The Input option assumes that the incoming image is in LatLong space.

- **Transform** - sets the output metadata to the sum of the input metadata and the total transform described by the controls in the C_SphericalTransform node, but does not modify the output image.

A simple example including three C_SphericalTransforms, applied to Checkerboards for clarity, produces superior results to three separate filter hits. The first two C_SphericalTransform **Metadata** controls are set to **Transform** and the last node, which applies all the transforms at once, is set to **Apply**.

Tip: You can setup a series of transforms quickly using the **Export** > **C_SphericalTransform (Concatenated)** option. When you click **Create**, the Metadata controls in the transform nodes are configured automatically to pass only metadata down the node tree.



Three separate filter hits.

One filter hit performed at the end.

Reviewing 360 Footage

SphericalTransform is invaluable for reviewing 360 material without the use of a headset. Setting the output to **Rectilinear** allows you to pan around the scene to get a feel for how the final render might look.

1. Add a SphericalTransform downstream of your stitch.
2. Navigate to the **Output** tab in the **Properties** panel.
3. Set the **Focal** control in the range 15-50 mm or use the **Review** preset that is included with the node.
4. Click the **Rotation** dropdown, and then select **Look**.
5. Use the **directionInput** Viewer widget to pan around the scene.

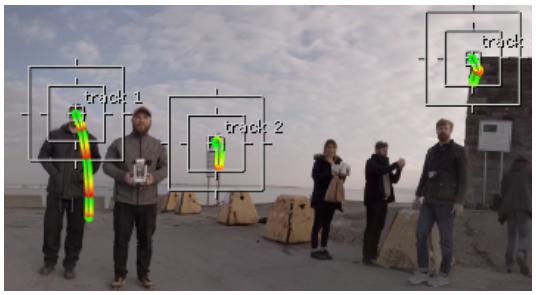


Tip: If you disconnect the SphericalTransform from the node tree and rename it to VIEWER_INPUT using the **Properties** panel, you can toggle the rectilinear projection on and off using the **IP** button (or **Alt**+**P** keyboard shortcut) above the Viewer.
If you leave the SphericalTransform **Properties** panel open, you can pan around in the lat-long projection as well.

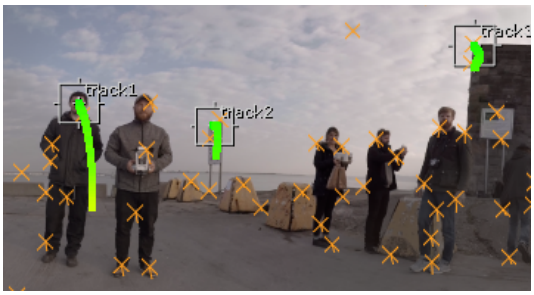
Tracking and Stabilizing

CaraVR's 2D Tracker allows you to extract animation data from the pan, tilt, and roll of a pattern. You can then apply the data directly to transform or match-move another element. Or you can invert the data values and apply them to the original element to stabilize the image.

C_Tracker differs from vanilla Nuke's tracker in that tracking anchors don't include a search area. This is because C_Tracker searches the entire image for each track. As a result, a greater number of tracks does not impact performance like Nuke's tracker. In fact, in some cases fewer tracks can take longer to calculate.



Nuke's Tracker node.



CaraVR's C_Tracker node.

Additionally, C_Tracker it can track in latlong space, meaning features are tracked as they pass the edge of the frame when the image wraps around.

C_Tracker can solve the camera using auto-tracks, user tracks, or both combined. See Automatic Tracking, Manual Tracking, Solving Cameras and for more information.

Automatic Tracking

Calculating tracks automatically uses the controls in the **Properties** panel to determine the direction and range of frames to track. Tracking forwards and backwards can produce a better track than going forwards if the pattern is clearly visible later in the clip.

To help avoid clutter in the Viewer, you can enable or disable the C_Tracker overlay by right-clicking in the Viewer and selecting **Overlay**, or by pressing **Q** to toggle between the available states:

- overlay off
- overlay on
- overlay on, no animation path

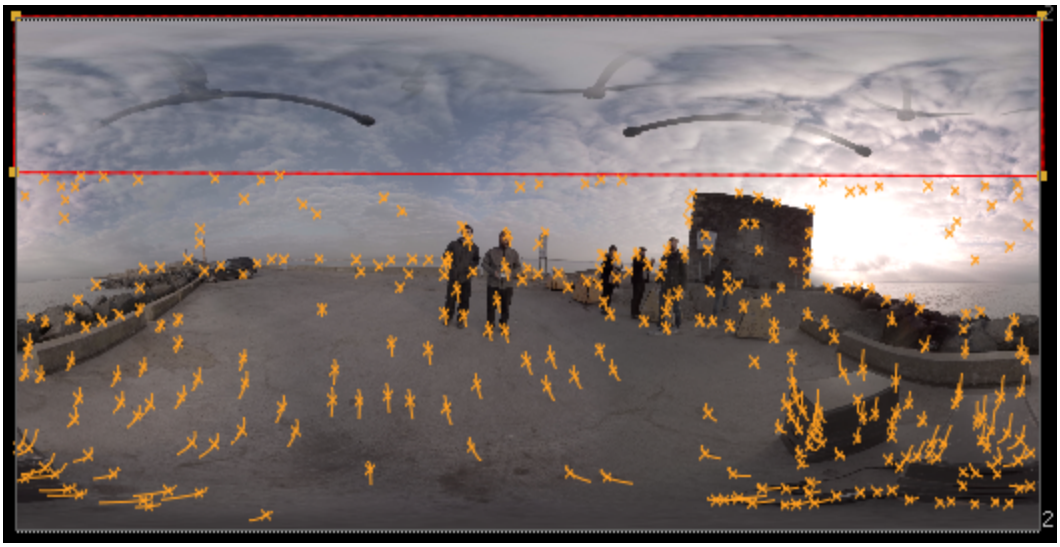
Masking Out Regions of the Image

Tracking works best on fixed, rigid parts of the scene. The solver uses these points to work out the camera path. Moving elements and rigs should be masked out before tracking.

To mask regions of your sequence, attach a matte to the **Mask** input to define image regions that should not be tracked. You can also use the source input's alpha channel as a matte.

1. If you want to use a separate matte for masking, connect a Roto node to the **Mask** input.
2. Scrub through the sequence and keyframe the roto shapes to cover the areas you don't want to track.

You don't have to be too accurate with the mask, it's only intended to cover areas that are likely to cause C_Tracker problems. For example, in the image shown, the drone rig is masked.



3. In the **Properties** panel, set **Mask** to the component you want to use as a mask:
   - **None** - track features in the whole image.
   - **Source Alpha** - use the alpha channel of the source clip to define which areas to ignore.
   - **Source Inverted Alpha** - use the inverted alpha channel of the source clip to define which areas to ignore.
   - **Mask Luminance** - use the luminance of the mask input to define which areas to ignore.
   - **Mask Inverted Luminance** - use the inverted luminance of the mask input to define which areas to ignore.
   - **Mask Alpha** - use the mask input alpha channel to define which areas to ignore.
   - **Mask Inverted Alpha** - use the inverted mask input alpha channel to define which areas to ignore.
4. Track as normal using the automated Auto Analysis **Track** button. See Auto-Tracking for more information.

Note: There is no need to mask areas of the image when tracking manually - you specify where User Tracks are placed.

Auto-Tracking

To calculate auto-tracks:

1. Render out a preview stitch from C_CameraSolver or C_GlobalWarp to work on. This step cuts down processing overheads considerably.

2. Mask out any regions of the image you don't need. See Masking Out Regions of the Image for more information.
3. Use the **Frame Range** dropdown to determine which frames are analyzed:
   - **Input** - the default value, sets the frame range to the length of the sequence attached to the **src** input.
   - **Global** - sets the tracking frame range to the range set in the **Project Settings** > **frame range** controls.

     If no frame range is defined, the frame range of the first image you read in is used as the **Global** frame range.
   - **Custom** - sets a tracking range of frames described by the **startFrame** and **endFrame** fields.
4. Click **Track** to begin analyzing the image and seeding tracks.

   When tracking is complete, you can solve the camera and then use the results for stabilzing or match-moving. See Solving Cameras for more information.

   If there appears to be too few tracks, or the distribution looks poor, see Troubleshooting Auto-Tracks for help with improving the results.

Troubleshooting Auto-Tracks

No matter how sophisticated tracking becomes, some sequences are inevitably going to cause problems. There are a number of controls on the **Settings** tab that can help achieve a better result. Try adjusting these settings and re-tracking to improve your results.

Tip: You can also add User Tracks to improve the result further. See Manual Tracking for more information.

Feature Settings

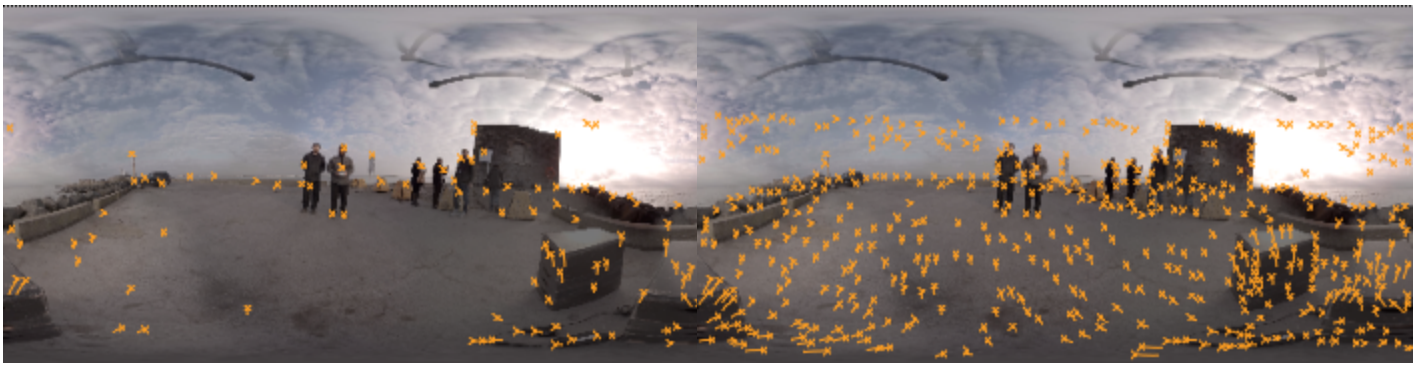
These controls determine the number and position of auto-detected features that C_Tracker analyzes per frame. Enable **Preview Features** so that you can see how altering the controls affects the positioning of features across the image.



Note: Tracking takes longer with **Preview Features** enabled, so don't forget to disable it before re-tracking.

- **Number of Features** - sets the number of features you want to track in each frame.

  In most cases, the default 300 should be sufficient, but in difficult sequences you may consider using a higher number.

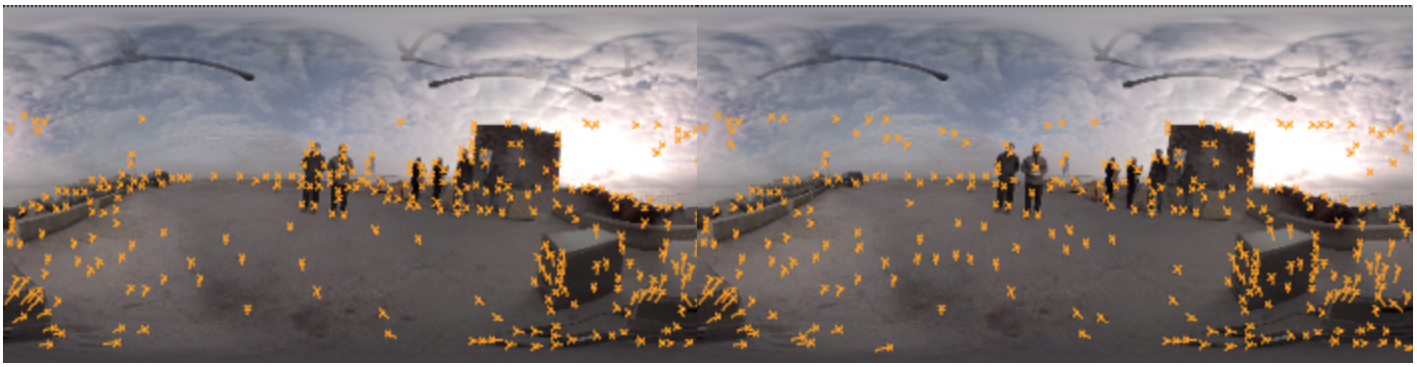


A low **Number of Features**. A high **Number of Features**.

Tip: Check **enforce** to keep the number of tracks constant, even if it means stopping some tracks and seeding new ones when **Track forwards and backwards** is enabled.

- **Feature Spread** - sets the spread of features over the image.

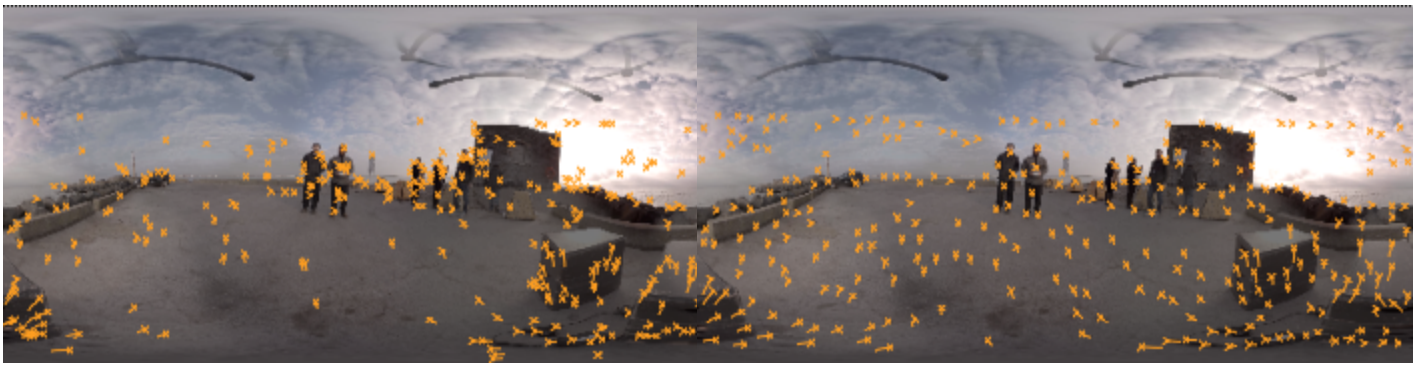
  A high value selects points that are distinct in local regions, whereas lower values select more prominent points in areas of high contrast.



A low **Feature Spread**. A high **Feature Spread**.

- **Feature Separation** - sets the distribution of features in relation to each other.

  To force feature separation and spread features evenly over the image at even distances, enter a high feature separation value.



A low **Feature Separation**.

A high **Feature Separation**.

Tip: Check **enforce** to separation, even if it means fewer overall tracks.

Check **Refine Feature Locations** to lock detected features to local corners. If you activate this, C_Tracker finds the closest corner point in your footage and locks the feature to it.

Tracking Settings

You can improve a set of feature tracks using the following controls and then retrack to improve your chances of getting a good solve:

- **Track forwards and backwards** - enabling this control can improve stabilization results, but tracks take longer to generate.
- **Patch Size** - sets the size of the pattern that C_Tracker looks for when analyzing the sequence.

  Reducing this value helps to match the pattern more closely, but higher values can produce more stable tracks.
- **Minimum Length** - sets a threshold value for the minimum acceptable track length. Tracks that fail to stick to the associated feature for this amount of frames are rejected.

  In long slow camera moves, it is best to solve from long tracks generated in the shot. If there are lots of short tracks, this can lead to noise on the calculated camera. Try removing the short tracks before solving.



A low **Minimum Length**.



A high **Minimum Length**.

- **Track Error Threshold** - C_Tracker's tolerance to change along the track is determined by this control.

  Increasing this threshold makes tracking more tolerant to image changes, potentially producing longer tracks.



A low **Track Error Threshold**.



A high **Track Error Threshold**.



Note: The **Track Error Threshold** is also applied to User Tracks by default. If you don't want User Tracks to be subject to this threshold, disable the **apply to user tracks** control.

- **Track Smoothness** - sets the threshold for smooth track generation. Adjusting this value can be useful in preventing poor tracks in complex sequences.

  Increasing the smoothness value removes tracks that fail over time.
- **Track Consistency** - sets the threshold for how inconsistent a feature track can be before C_Tracker discards it and reseeds in a different location.

  Higher values allow for less inconsistency.

When you're happy with the number and distribution of tracked features, you can solve the camera and then use the results for stabilizing or match-moving. See Solving Cameras for more information.

Manual Tracking

Calculating tracks manually uses the tools above the Viewer to control the direction and range of frames to track. Tracking backwards can produce a better track than going forwards if the pattern is clearly visible later in the clip.

To help avoid clutter in the Viewer, you can enable or disable the C_Tracker overlay by right-clicking in the Viewer and selecting **Overlay**, or by pressing **Q** to toggle between the available states:

- overlay off
- overlay on
- overlay on, no animation path

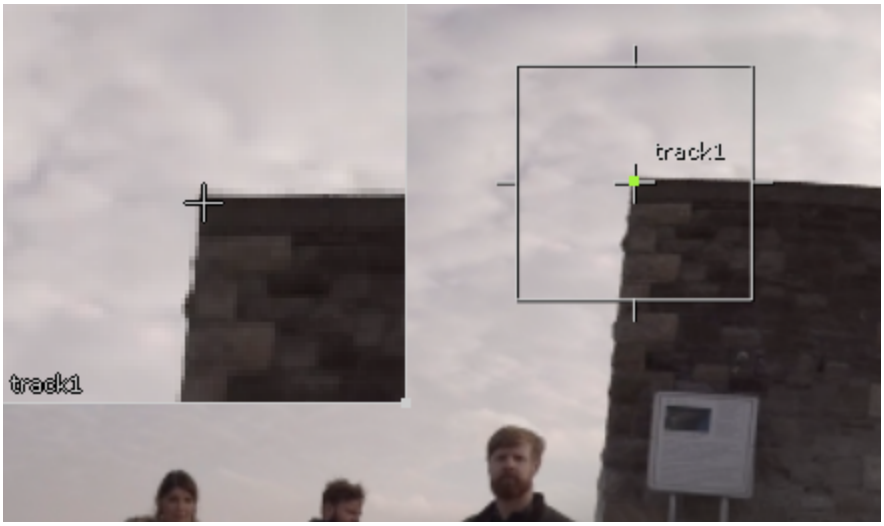
You can add as many user tracks as required, depending on which transformational components you wish to track. For example, when tracking in areas of distortion or noise, it's a good idea to add a lot of tracking anchors and then average the results to get a better overall track.

1. Render out a preview stitch from C_CameraSolver or C_GlobalWarp to work on. This step cuts down processing overheads considerably.
2. Enable the **fast-add** button  and click in the Viewer to add a user track, or click **Add track** on the **User tracks** tab in the **Properties** panel to create the required number of anchors.



Note: Holding **Ctrl**/**Cmd**+**Alt** and clicking in the Viewer also enables fast-add mode.

The **Add track** button places anchors in the center of the current Viewer. You'll also notice an anchor zoom window in the top left of the Viewer. This allows you to accurately position tracks without zooming the entire Viewer.



3. Set when the zoom window is displayed using the **Show Zoom Window** dropdown:
   - **always** - the window is always displayed.
   - **on track move** - the window is only displayed when a track anchor is moved.
   - **when tracking** - the window is only displayed during tracking.
   - **when tracking or track change** - the window is displayed during tracking and when a tracking anchor is moved.
   - **never** - the window is never displayed.
4. Set the zoom widget's size and magnification using the **Zoom Size/Mag.** dropdowns. You can select **custom** and enter values manually for greater flexibility.
5. A filter is applied to the zoom window on playback by default, but you can enable the filter permanently, or disable it, using the **Zoom Window Filter** dropdown.

Note: The filter applied is the same as that selected on the **Transform** tab, and can produce a more visually stable track. It can make track positioning more difficult, however.

6. Temporarily disable tracks using the checkbox on the left of the **User tracks** tab or remove tracks by selecting them in the **Tracks** list and clicking **Delete tracks**.
7. In the C_Tracker **User tracks** tab, select each track you wish to calculate or click **select all** to use all tracks.
8. Using the tool bar above the Viewer, click either the frame backward (X) or forward (C) buttons to move to the previous or next frame. Move through a few frames in this manner to ensure that all enabled track anchors are "sticking" to their patterns.

   If a particular track anchor doesn't stick, experiment with a different position.
9. Once all track anchors stick, click the track backward (Z) or track forward (V) buttons to analyze the whole sequence.

10. To track only a certain frame range, use the range buttons to enter the required frames.
11. Click stop , to cease tracking in either direction.

Tip: When calculating multiple tracks simultaneously, you may find that some tracks stick accurately to the pattern, while others require resetting and re-analysis. When you're happy with a given track, deselect it in the Tracks list. This protects it from recalculation, and lets you experiment with better placement for the wayward tracks.

When tracking is complete, you can solve the camera and then use the results for stabilizing or match-moving. See Solving Cameras for more information.

If the tracks don't stick to the selected feature or produce poor results, see Troubleshooting Manual Tracks for help with improving the results.

Troubleshooting Manual Tracks

Even with manually positioned tracks, some sequences are inevitably going to cause problems. There are a number of pre-tracking checks you can perform to assist C_Tracker:

- Play through the sequence before placing your tracking anchors.
- Look for features that are consistent throughout the majority of the sequence.
- Avoid occluded features where possible - see Dealing with Occlusions.

Tip: You can also add auto-tracks to improve the result further. See Automatic Tracking for more information.

To troubleshoot manual tracks:

1. First, turn on the color-coded error indicator by clicking the traffic light Viewer tool .

   Each keyframe is colored on a sliding scale from green (good match) to red (poor match).

A red keyframe doesn't necessarily mean that the tracking result is poor, only that C_Tracker couldn't reliably match the pattern from one keyframe to the next.

2. Move the tracking anchor to the first of the poor frames, just about the center of the image in the example.
3. Using the zoom window, drag the anchor to the correct location of the grabbed pattern.
4. Advance the playhead to the next poor keyframe and repeat until the track is complete.

When you're happy with the number and distribution of tracked features, you can solve the camera and then use the results for stabilizing or match-moving. See Solving Cameras for more information.

Dealing with Occlusions

C_Tracker's offset capability allows you track an obscured feature using the relative position of another feature, providing that the distance between the two points remains constant.

1. Track the pattern normally until the occlusion causes Tracker to fail.

   The zoom window helps to identify the failure point.
2. Play though the sequence to identify a likely offset point - a pattern that remains equidistant from the original pattern grab.
3. Hold down **Ctrl/Cmd** and drag the tracking anchor to the offset position.

   The offset amount is recorded in the Tracks list and highlighted in yellow in the Viewer.
4. Continue tracking as normal by clicking the track backward (Z) or forward (V) button. 

   Tracker combines the two tracks into a single continuous track.
5. Use the clear backward and forward buttons  to clear poor keyframes. Click clear all  to remove all keyframes.



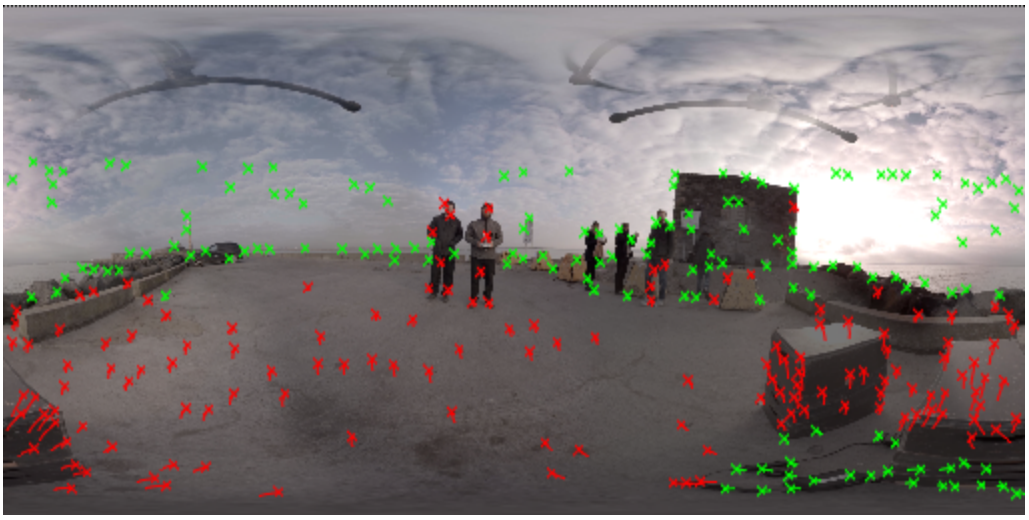
Note: You can reset tracking anchor pattern and search areas by clicking .

Solving Cameras

When you're happy with the features that you've tracked, you can solve the camera position. C_Tracker uses the tracking information to calculate the camera position, which is used for stabilizing or match-moving.

1. In the C_Tracker **Properties** panel, click the **Solve Using** dropdown and select the type of tracking data to use. If you created auto-tracks and then added some manual tracks as well, select **Combined**.
2. Click **Solve**.

When the solve is complete, the Viewer displays the tracked features with solve data applied. Green features represent good solve data and red features represent poor solve data.



The **Solve Error** is good indicator of the overall quality of the solve. If the RMS (root mean square) error is quite high, consider refining your tracking data or adding some User Tracks and Solving the camera again. See Troubleshooting Auto-Tracks and Manual Tracking for more information.

3. If too many tracks are rejected by the solve, try increasing the **Error Threshold** or refining the solve using the **Solve** controls on the **Settings** tab. See Troubleshooting Solves for more information.

   When you're happy with the solve, you can use the transform data for stabilizing or match-moving. See Stabilizing Using C_Tracker for more information.

Troubleshooting Solves

Camera solves depend on good tracking data, so the first task when troubleshooting solves is to make sure you have solid tracking data from which the solve is calculated. See Troubleshooting Auto-Tracks and Troubleshooting Manual Tracks for more information.

C_Tracker also includes controls on the **Settings** tab to refine solve data:

- **Keyframe Overlap** - sets the rate at which keyframes are created to detect camera motion.

  If your tracks include rapid changes in motion, try increasing the **Keyframe Overlap** to add more keyframes. Lowering the **Keyframe Overlap can** minimize drift.
- **Validation Threshold** - this control is linked to the tracking **Error Threshold** control on the **C_Tracker** tab. You can increase the threshold to include more rejected tracks or decrease it to reject more tracks.

Tip: Enabling **Refine camera motion** can improve the accuracy of a solve, but increases processing time.

When you're happy with the solve, you can use the solve data for stabilizing and match-moving workflows. See Stabilizing Using C_Tracker for more information.

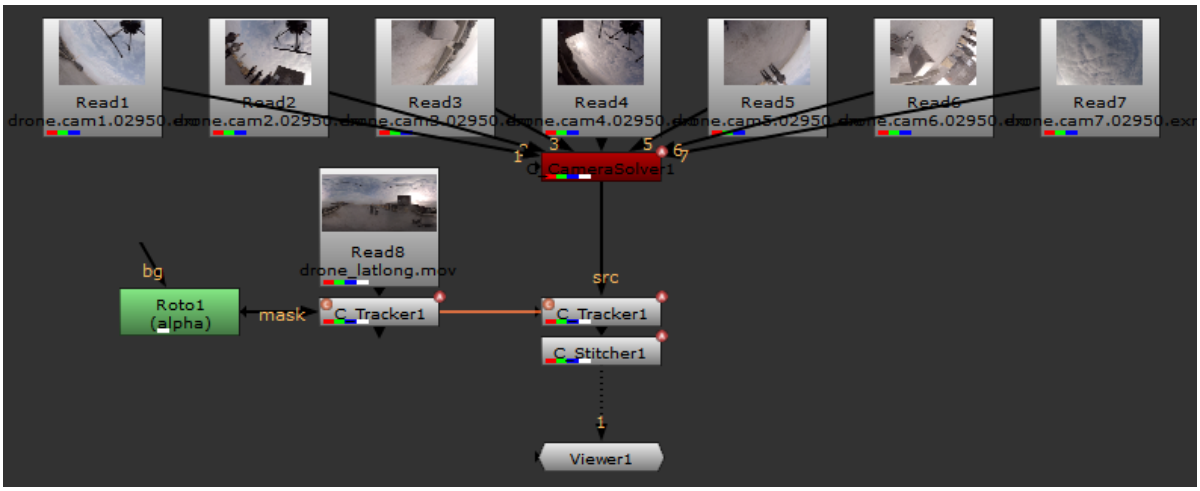
Stabilizing Using C_Tracker

Moving camera rigs inevitably produce unwanted motion in the shot, particularly on the horizon, which can prove unpleasant when viewed in VR. C_Tracker includes a stabilization workflow to help you eliminate this type of movement.

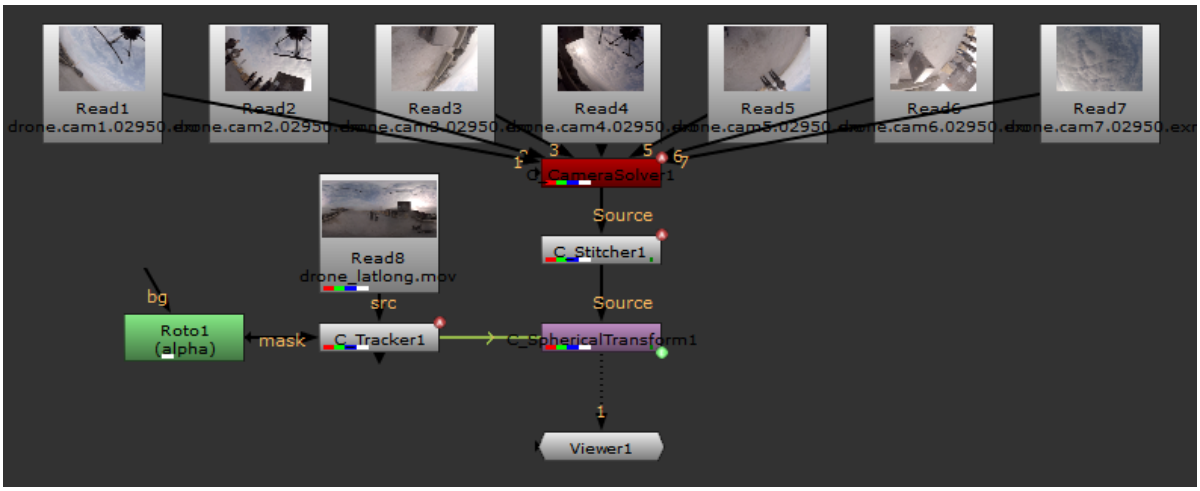
1. Track and solve the sequence as described in Automatic Tracking, Manual Tracking and Solving Cameras.
2. When tracking and solving is complete, switch to the **Transform** tab to view the average pan, tilt, and roll keyframe data.

   There are two ways you can use the data to stabilize the shot:

   • plug the C_Tracker into the stitch node tree before C_Stitcher and switch **Transform** to **Stabilise**,



   • on the **C_Tracker** tab, export a C_SphericalTransform node and plug it into the stitch node tree after C_Stitcher.



Stabilizing Using CameraTracker

If you have a NukeX license, you can stabilize footage using CameraTracker. CaraVR adds two new options to the standard CameraTracker **Export** dropdown:

- **C_SphericalTransform (stabilise)** is useful for stabilizing stitches from other packages, as well as from CaraVR, but it does add another filtering hit during processing which can affect image quality. See C_SphericalTransform Stabilization for more information.
- **C_SpericalTransform (Metadata only)** can be used for CaraVR stitches to reduce filtering hits, as it edits the metadata passed down from the solver directly. See Transforming Using Metadata for more information.

Note: You can also handle metadata using C_SphericalTransform. See Transforming Using Metadata for more information.

C_SphericalTransform Stabilization

1. Render a preview stitch from C_CameraSolver to work on. This step cuts down processing overheads considerably.
2. Attach a C_SphericalTransform node downstream of the preview stitch.

   CameraTracker was developed with standard rectilinear footage in mind, so feeding the latlong preview straight in may cause problems for the tracking process.
3. Set the Output **Projection** dropdown to **Rectilinear**.

Tip: You can reduce the time taken to track the footage by adjusting the **Width** control, which cuts down the resolution that CameraTracker has to process. For stabilization purposes, you only need rough rotation data, not pixel-perfect tracking data.

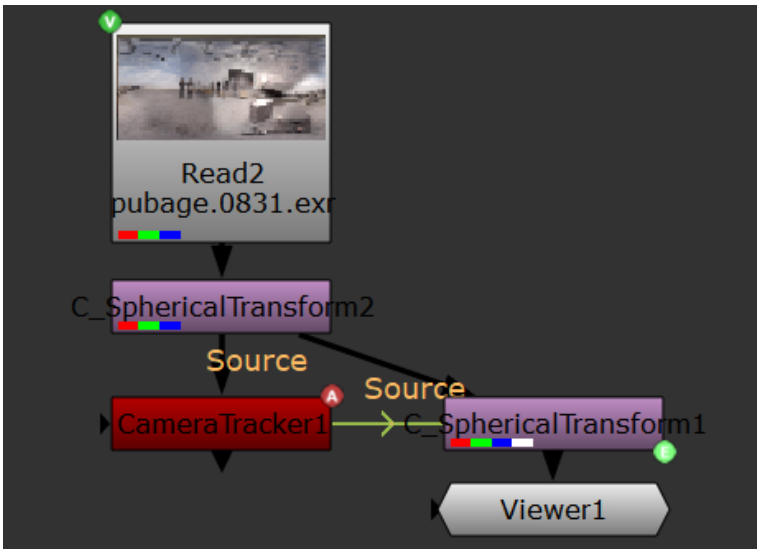
4. Click the **Output** tab and make sure that the **Focal Length** and **Sensor** controls match those in the C_CameraSolver properties.
5. Attach a CameraTracker node to the C_SphericalTransform and open its **Properties** panel.
6. Track and solve the camera as described in Camera Tracking.

   When the process is complete, the Viewer shows the computed tracks and their associated solve error. CameraTracker calculates the solve error for each view in the script, but the **main**, or **Principal View** in stereo scripts, is the important view.



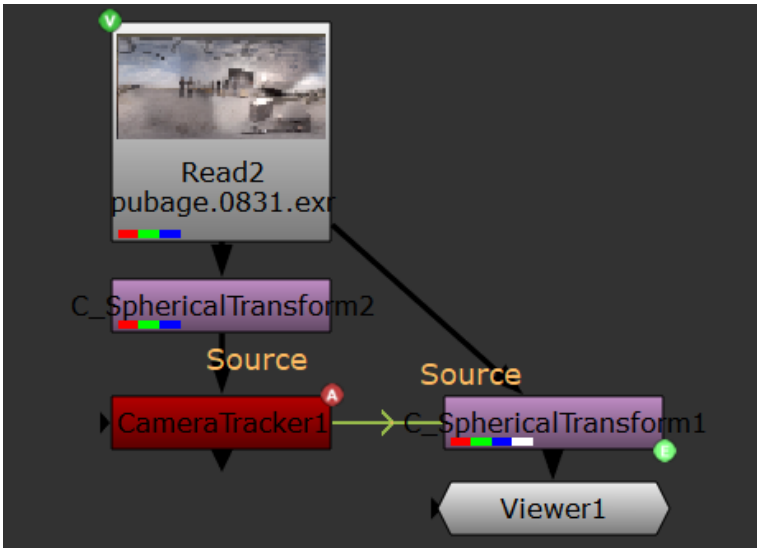
7. Click the **Export** dropdown and select C_SphericalTransform from the list.

   C_SphericalTransform is set up to invert the rotation data extracted from the camera track.
8. Click **Create** to connect the new node to the node tree.



The CameraTracker and C_SphericalTransform nodes are linked using an expression, which means that any changes made to the track and solve data in CameraTracker are automatically passed to the C_SphericalTransform node.

9. The final step is to connect the C_SphericalTransform node to the source latlong preview because the camera data was created in rectilinear space.



10. Play through the footage to check that the stabilization was successful.

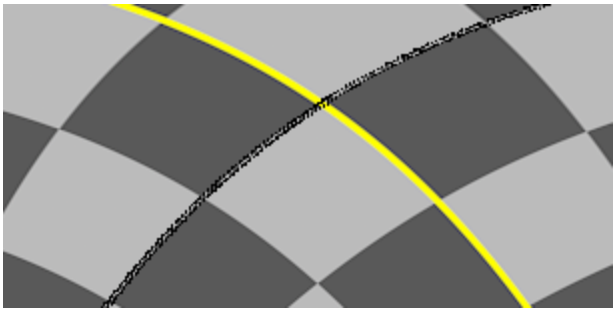
Warping Using STMaps

The C_STMap node allows you to warp the **src** input according to the stitch or ppass attached to the **map** input. C_STMap accepts **stitch_map** and **ppass_map** channels in specified layers. The **stitch_map** UV

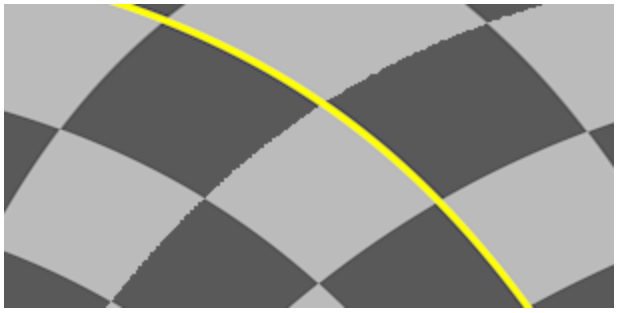
channels represent the absolute pixel positions of an image normalized between 0 and 1, whereas the **ppass_map** XYZ channels represent the normalized 3D coordinates of a latlong image ranging between -1 and 1.

Note: When you export from C_Stitcher, the UV or XYZ values are stored in the **stitch_map** or **ppass_map** layer, but you can use a Copy node to merge the channels into any layer. The selected layer must contain at least two channels (stmap) or three channels (ppass), but only the required channels are used if there are more channels in the layer.

The advantage of using a **ppass_map** over a standard **stitch_map** is that latlong continuity is naturally preserved, reducing or even eliminating hard-edge artifacts along latlong discontinuities.



Artifacts using **stitch_map**



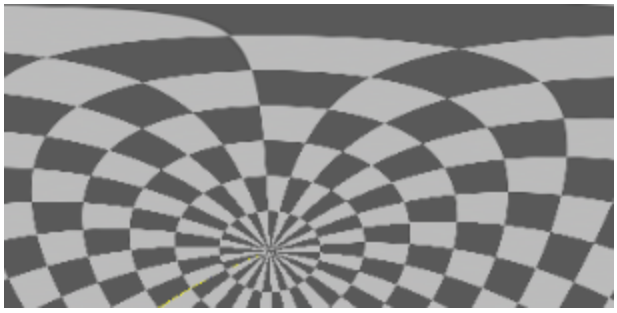
Reduced artifacts using **ppass_map**

C_STMap can produce three modes of output for both **stitch_map** and **ppass_map** inputs:

- **Warped src** - warps the **src** input using the **map** input. This option is similar to Nuke's STMap node's output.



The warp from the **map** input



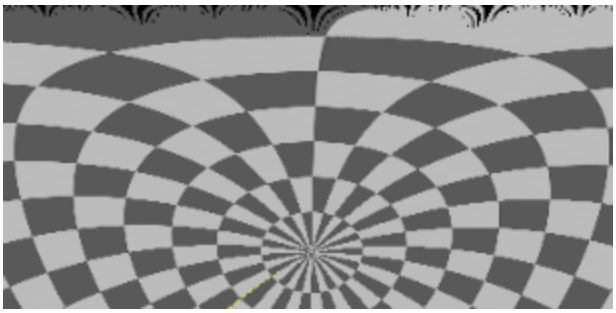
Output set to **Warped src**

- **Warped src (inverse)** - internally calculates the inverse of the **map** input and warps the **src** input using the result.

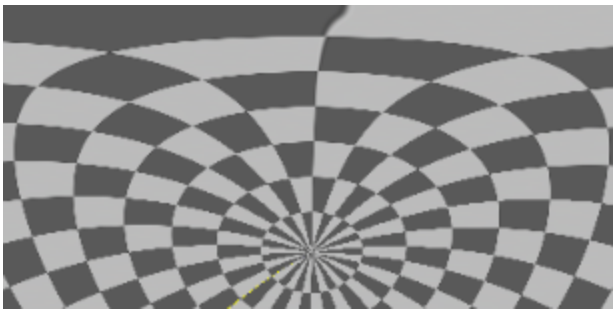
  Inverting warps typically produces output with missing values, as shown, so CaraVR provides an **interpolate** control to fill the missing areas.



The warp from the **map** input



Output set to **Warped src (inverted)**



Output set to **Warped src (inverted)**, but with **Interpolate** enabled.

The interpolation is a simple smoothing operation to produce consistent results in relation to existing neighbouring pixels.

- **Map inverse** - writes the inverse of the input map to the channels selected in the **Map** control.

  This option doesn't require a **src** input.

See Re-applying Corrections to a Warp for workflow examples.

Re-applying Corrections to a Warp

Working collaboratively can mean that work applied in parallel while a stitch is a work in progress occasionally needs to be re-applied when the warp changes. An inverse **ppass_map** can be used to re-apply corrections, such as localized GridWarp or RotoPaint, to a new version when the warp changes.

For example, if you're given a warp as shown, add paint, and then the GridWarp changes, your paint doesn't warp correctly.



The original warp.



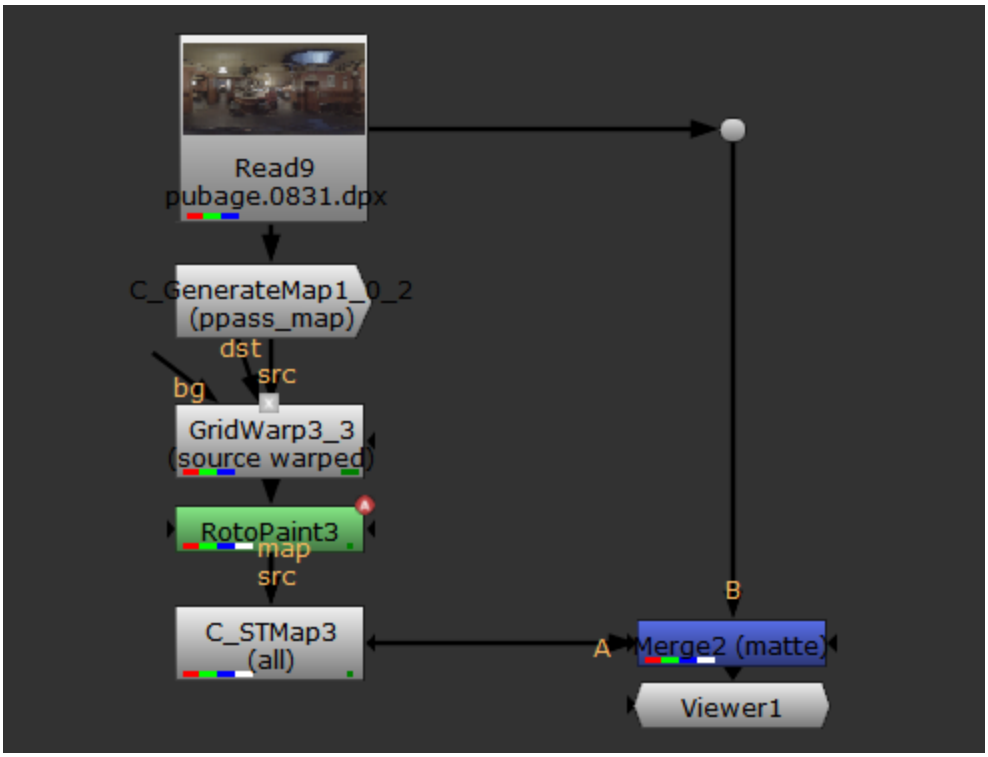
Paint applied to remove the rig.



The same paint when applied to a new warp is offset.

C_Stitcher generates **ppass_map** layers for all inputs by default, but you can create the **ppass_map** layer using C_GenerateMap if your source image is prestitched.

Placing the C_GenerateMap between the source and the warp, passes the warp down the node tree in the **ppass_map** layer, which can then connect into a C_STMap node. Make sure that the **Output** of the C_STMap node is set to **Warped src (inverse)**.



Using the **ppass_map** as the C_STMap's **map** input creates a faster and more accurate inverse result, which can be reused easily throughout a script.



The paint repositioned with the new warp.



The merged result.

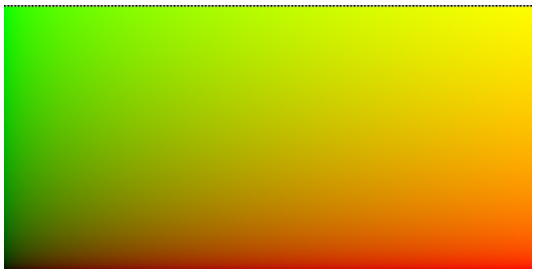
Generating Stitch and PPass Maps

C_GenerateMap allows you to generate **stitch_map** and **ppass_map** channels in specified layers. The **stitch_map** UV channels represent the absolute pixel positions of an image normalized between 0 and 1, whereas

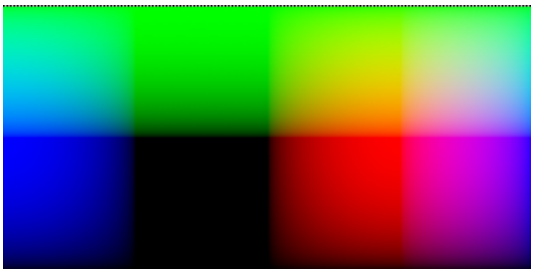
the **ppass_map** XYZ channels represent the normalized 3D coordinates of a latlong image ranging between -1 and 1.

The **Format** control sets the format of the output map when the **Source** input is not connected. If the **Source** input is connected, the format is obtained from the parent node and the **Format** control is disabled.

The standard maps produced by C_GenerateMap are shown in the images, although you need to select the appropriate layer from the **channels** dropdown in the Viewer.



Viewing the **stitch_map** layer



Viewing the **ppass_map** layer

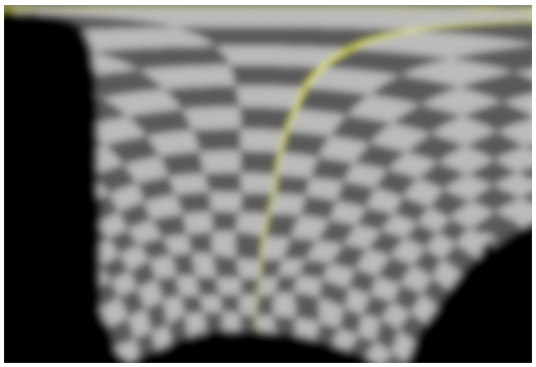
The output can then be warped and piped into a C_STMap to warp other source images or invert existing warp calculated by C_Stitcher. See Warping Using STMaps for more information.

Applying LatLong Blur

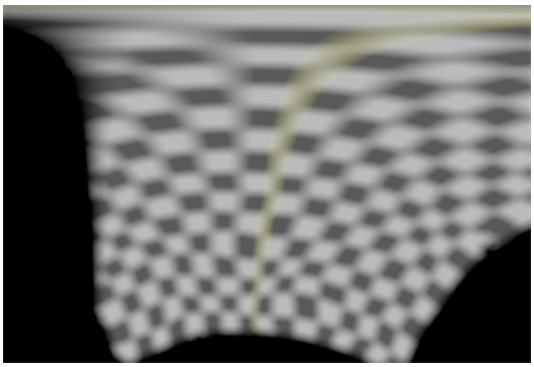
C_Blur allows you to apply blur to a latlong image and produce a sensible result across the entire frame, as if the blur was applied to a rectilinear image all around.

Certain Nuke nodes only require a single input pixel to produce a corresponding output pixel, for example ColorCorrect, Grade, and so on. These work as expected when applied to latlong material. Other nodes, such as Blur and Defocus, need a tile of pixels around a location to produce a single corresponding output pixel, and do so in a consistent manner across the entire image. If you apply these nodes to a latlong image, the output they produce looks strange when viewed in a headset, because the blur was created in latlong space, rather than in rectilinear.

C_Blur's effect is warped in-line with the view to produce a more natural-looking effect when viewed in rectilinear. The effect is easier to see if you switch the view input to a Checkerboard node. Closer to the equator, both filters produce similar results, but as you move toward the poles, you can see that C_Blur produces a greater effect.



Standard Nuke Blur.



C_Blur.

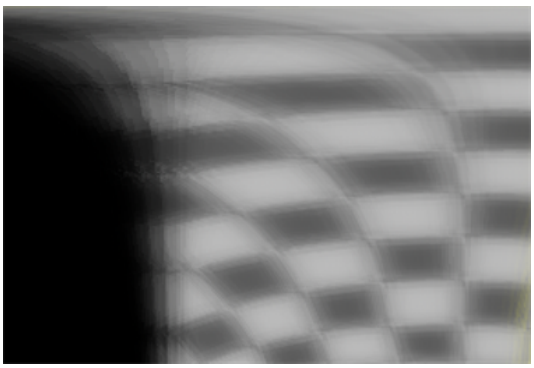
C_Blur includes additional controls, **Accuracy** and **Prescale**, that represent a trade-off between quality and speed of calculation. The **Filter** control provides a similar function, **Box** is the fastest and **Gaussian** is the smoothest.

- **Accuracy** - controls the amount of samples used to calculate the blur. Lower values are faster to calculate, but image quality can suffer.

  If you're using large C_Blur **Size** values, you can use **Accuracy** to reduce processing time with minimal image quality loss.



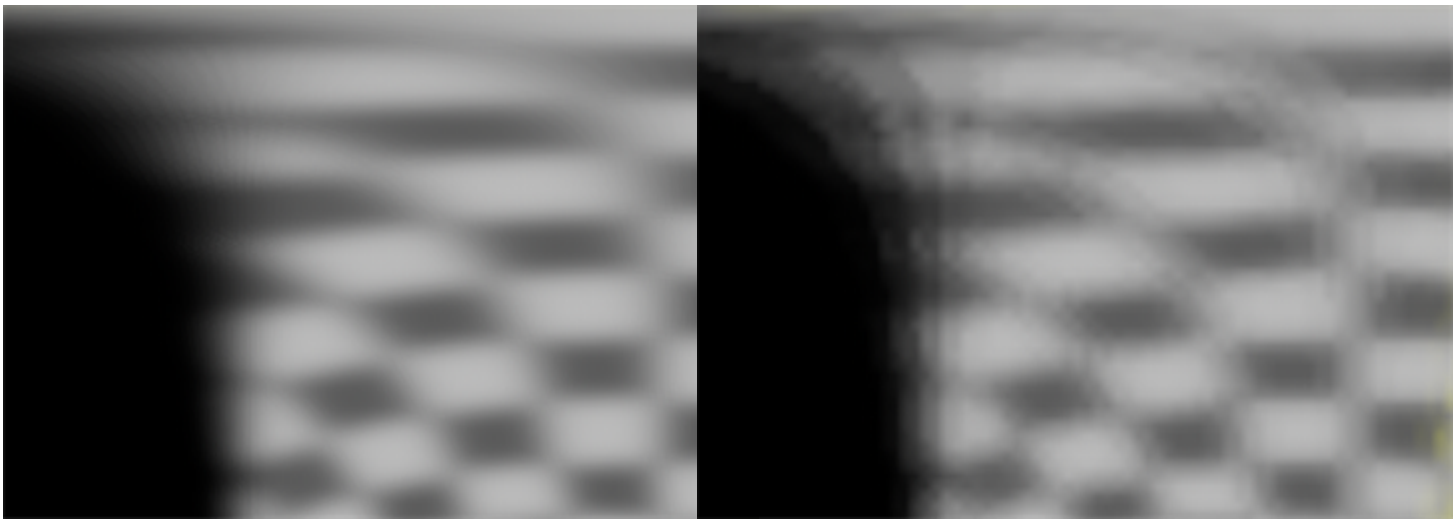
High **Accuracy** value.



Low **Accuracy** value.

- **Prescale** - controls the amount of downscaling applied to the image before the blur is applied. Again, lower values are faster to compute, but may introduce banding.

  If you're using large C_Blur **Size** values, you can use **Prescale** to reduce processing time with minimal image quality loss.



High **Prescale** value.

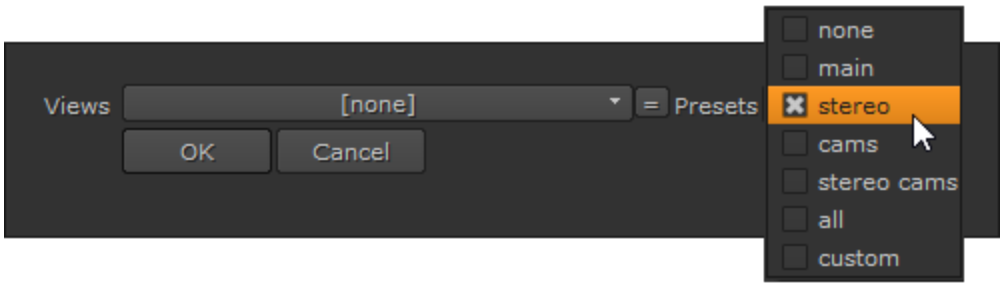
Low **Prescale** value.

Split and Join Selectively

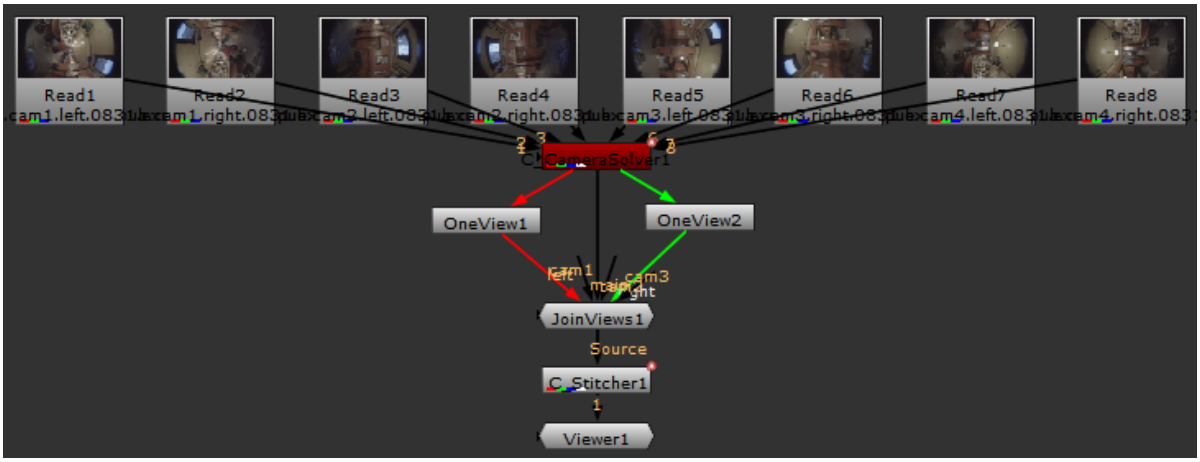
Split and Join Selectively is a combination of OneView and JoinViews nodes, just like Nuke's vanilla Split and Join node, but it allows you to select which views are affected.

Using Nuke's Split and Join after a C_CameraSolver, for instance, splits and joins all views in the script, which can add unnecessary clutter to the Node Graph.

Adding Split and Join Selectively presents you with a dialog to select the required cameras manually or choose a preset, such as **main** and **stereo**.



In this example, only the **left** and **right** views are split out along side **main**, rather than all views.



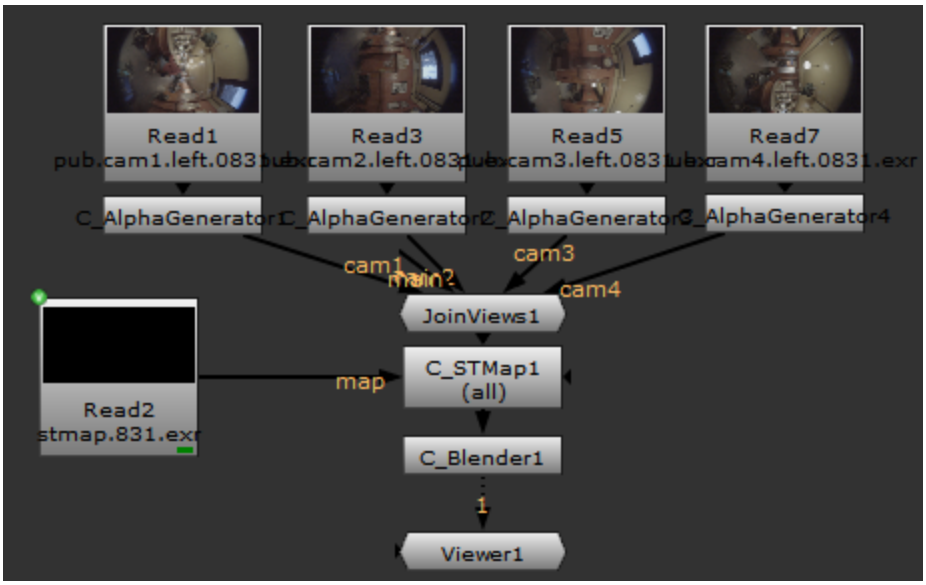
Note: You can use Split and Join Selective after C_Stitcher as well, but remember to add a C_Blender after the join so that the views are recombined into a single stream. See Blending Multiple Views Together for more information.

Generating Alpha Masks

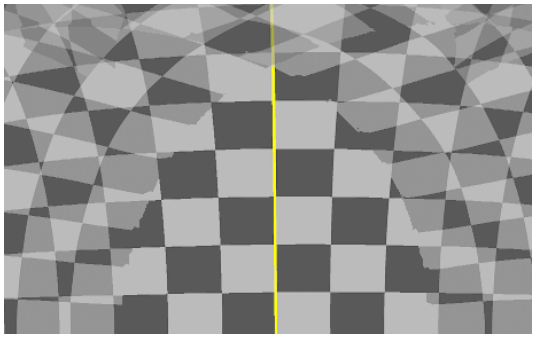
C_AlphaGenerator creates a rectangular or elliptical mask in the alpha channel. It's primarily used for manual stitching and is included in the Manual 2D and 3D Stitching toolsets included in C_CameraSolver's **Export** dropdown.

For example, if you're given an STMap describing the warp required to stitch a sequence, but don't have access to the masks created by C_CameraSolver, you can use masks created by C_AlphaGenerator to improve blending between adjacent cameras.

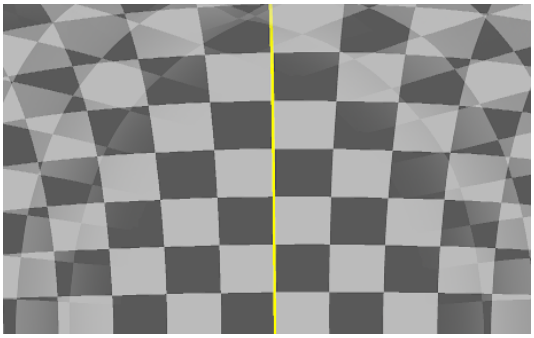
The script shows an example node tree including the C_AlphaGenerator node controlling the blending.



The alpha mask generated by the nodes smooths the hard edges from the STMap, creating a more natural blend.



Blending using just the STMap's warp.

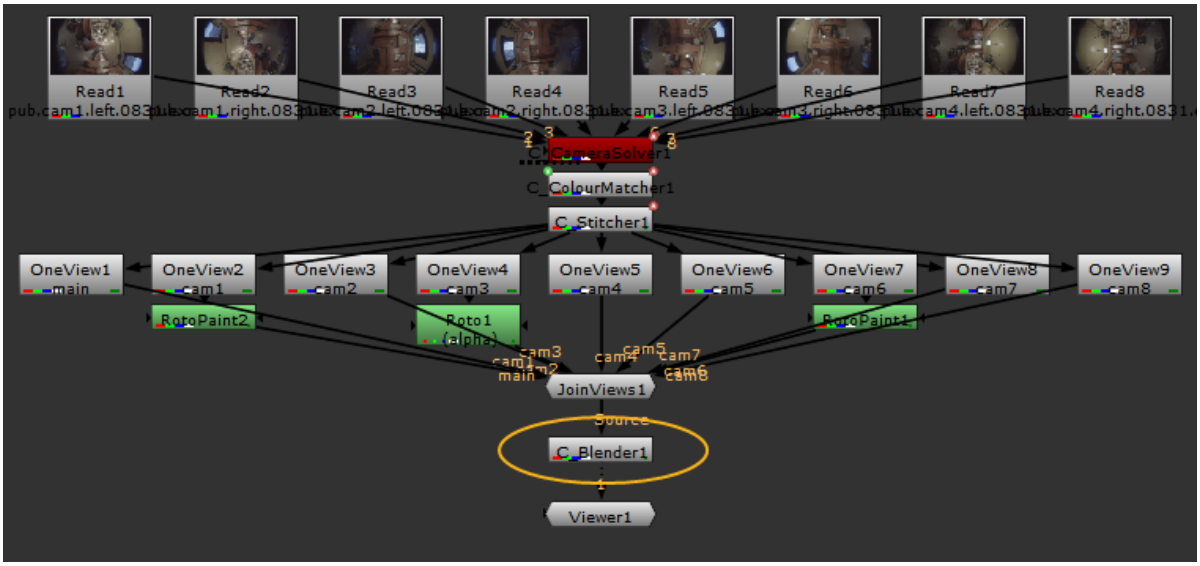


Blending using C_AlphaGenerator.

Blending Multiple Views Together

C_Blender is used as a Merge node to combine all images together when manually correcting a stitch. The C_Stitcher can output separate warped images, which can be corrected manually to remove vertical misalignment, edited to control the seam between different images, and then finally combined by C_Blender.

C_Blender only accepts one input, so make sure it's downstream of a JoinViews node.



C_Blender is a utility node, and as such, only has a few dedicated controls of its own:

- The **Input** > **Rig Views** controls allow you to blend specific views, rather that all views passed to C_Blender. Enable **Override** to activate the controls.

  Use the **Preset** dropdown to quickly set common views, rather than selecting them individually from the **Rig Views** dropdown.

> **Note:** Any set of views can be selected, provided that they match in format.

- The **Format** dropdown controls the output resolution.
- The **Blend** dropdown allows you to switch between the default **Alpha** mode, **Multi-Band**, and **Spherical Multi-Band** modes. Multi-band blending can improve stitch results by matching low frequency color changes over the course of the blend region, but can be slower to process than the default **Alpha** mode. Spherical Multi-Band blending can improve stitch results at the poles, looking up or down.

  You can use the **Suppression** control to adjust the amount of blending applied between adjacent views when using **Multi-Band** blending. Lower values can help balance color and exposure between blended images, higher values are closer to the result from the default **Alpha** blending mode.

> **Note:** If you exported C_STMaps from individual views in C_Stitcher for manual stitching or corrections, enable **Multi-Band** blending in C_Blender because multi-band blending has no effect when applied through the stitcher. However, the results will not match those output by C_Stitcher because C_Blender uses resampled input images.

Rendering Using RayRender

CaraVR adds to a **Camera** tab to reconstruct stereographic VR sequences using slit scan techniques to the vanilla Nuke RayRender node. Ray rendering is a 3D to 2D process, so there's some setup involved before you can use RayRender.

When connected to a Scene node, the RayRender node renders all the objects and lights connected to that scene from the perspective of the Camera connected to the **cam** input (or a default camera if no **cam** input exists). The rendered 2D image is then passed along to the next node in the compositing tree, and you can use the result as an input to other nodes in the script.

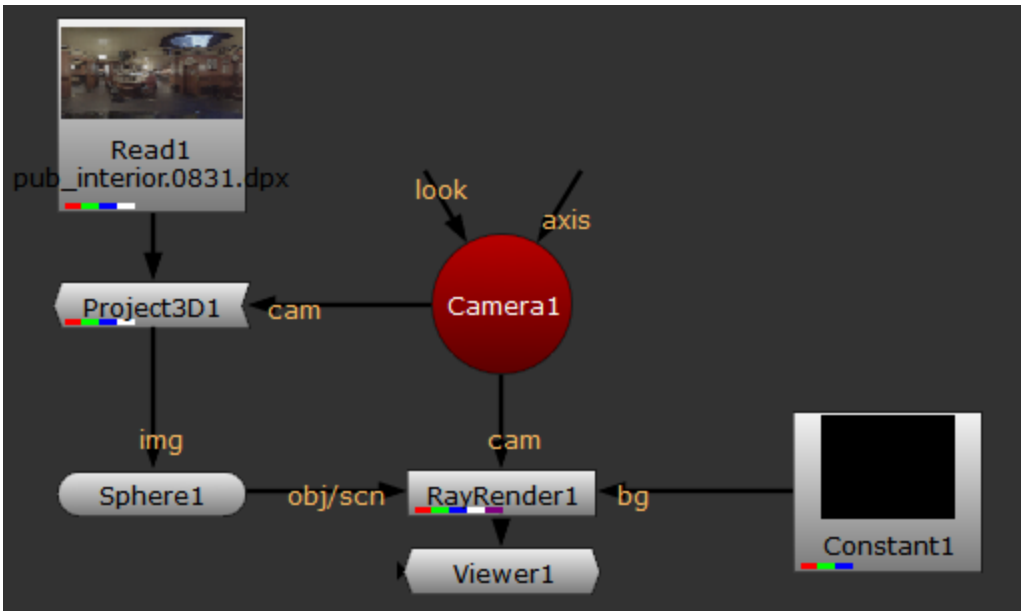
RayRender is included as an alternative to Nuke's ScanlineRender because it is particularly suited to working with latlong material. RayRender can processes full 360 renders relatively quickly, and it includes a number of controls to improve the quality of stereo views towards the poles, when you look up or down wearing a headset.

Spherical Projection with RayRender

For this workflow, you already have your stitch rendered out to a latlong and you have some geometry, or a sphere in this case, to project the latlong on to for correction in 360 space.

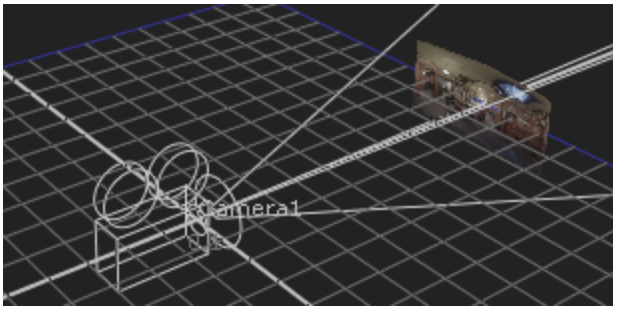
Refer to Classic 3D Compositing for more information on compositing in Nuke's 3D environment.

1. Set up a simple script in the Node Graph using a Project3D node, a Sphere, and a Camera connected into a RayRender.

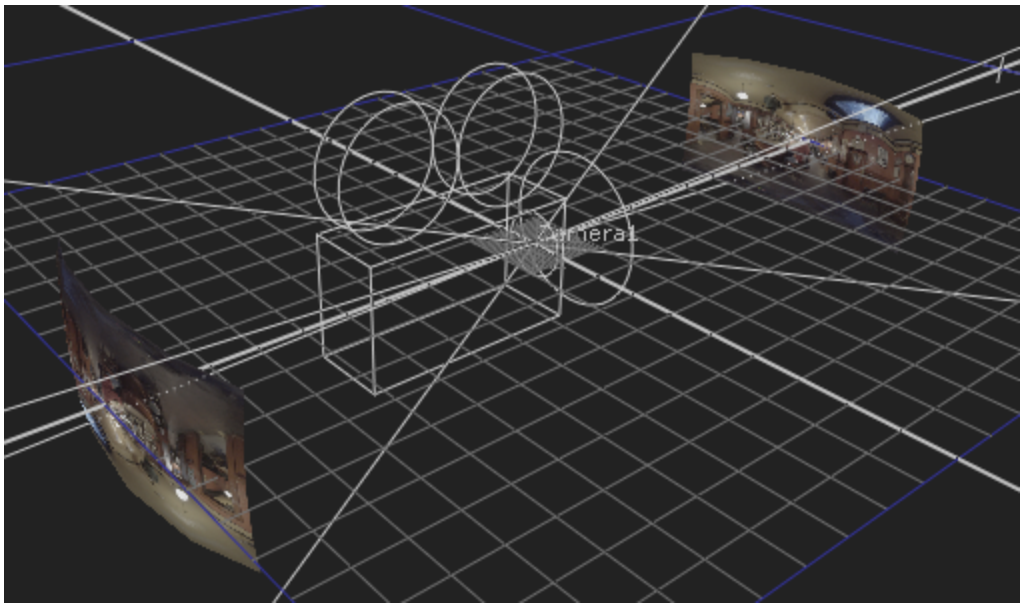


2. Switch the Viewer between 2D and 3D modes to view your output.

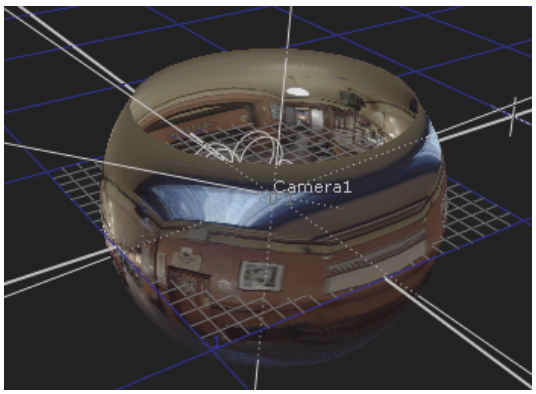




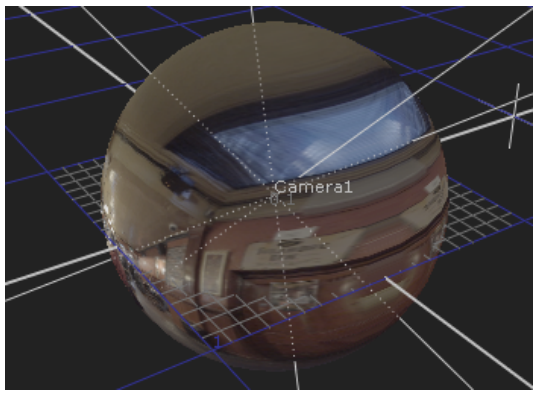
3. In the Camera's **Properties** panel, use the **projection** dropdown to select **spherical**, as we're dealing with latlong material, not rectilinear.
4. Set the Camera's **near** clipping plane to the inverse of the **far** clipping plane, so that the camera projects behind as well as in front. The default **far** clipping plane is **10000**, so set the near to **-10000**.



5. The last step is to massage the **focal length**, so that the latlong actually appears as a sphere. Decrease the focal length until the sphere is closed at the poles, and this should give you something reasonable to work with in the 2D view.



A **focal length** of 10.

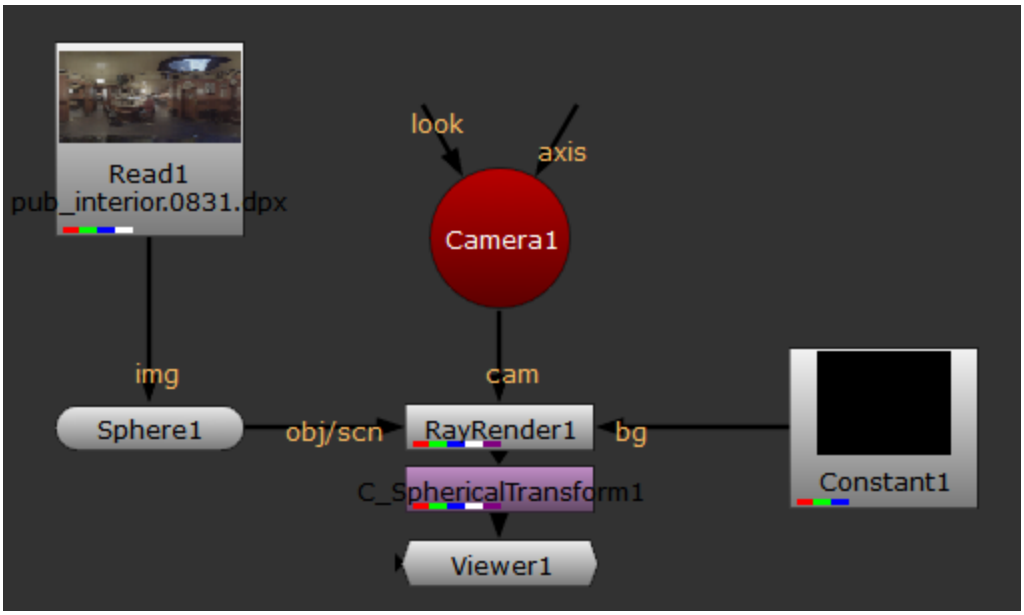


A **focal length** of 6.

Using Slit Scan with RayRender

VR headsets have trouble with stereo images at the poles of the 360 sphere, when you look up or down. This effect can be minimized by pole merging, which gradually makes the left and right eye the same towards the poles. RayRender's **Camera** tab renders this merging effect for stereo output.

1. You can examine the falloff in a scene by setting up a simple script in the Node Graph using a Sphere, and a Camera connected into a RayRender.



2. Set the RayRender **projection mode** to spherical and in the **Camera** tab, check **Stereo Scan Enable** to activate the stereo controls.
3. Add a C_SphericalTransform node and set the **Rotation** mode to **Rotation Angles**.
4. Adjust the rotation so that the view displays one of the poles, in this example, the 'top' of the scene is **z -90**.
5. RayRender's **Falloff Type** defaults to **Cosine**, but you can see the effect of other types by selecting them:
   - **None** - no IPD adjustment occurs towards the poles.
   - **Linear** - the views are merged gradually from the **Start Angle** specified toward the pole. Increasing the angle moves the start point toward the poles.

Note: Setting the **Start Angle** to 90 disables pole merging.

   - **Cosine** - the views are merged smoothly toward the poles. Reducing the falloff shifts the transition in depth towards the poles.

Note: Setting the **Separation Falloff** to 0 disables pole merging.



Falloff Type set to **None**.



Falloff Type set to **Linear**.



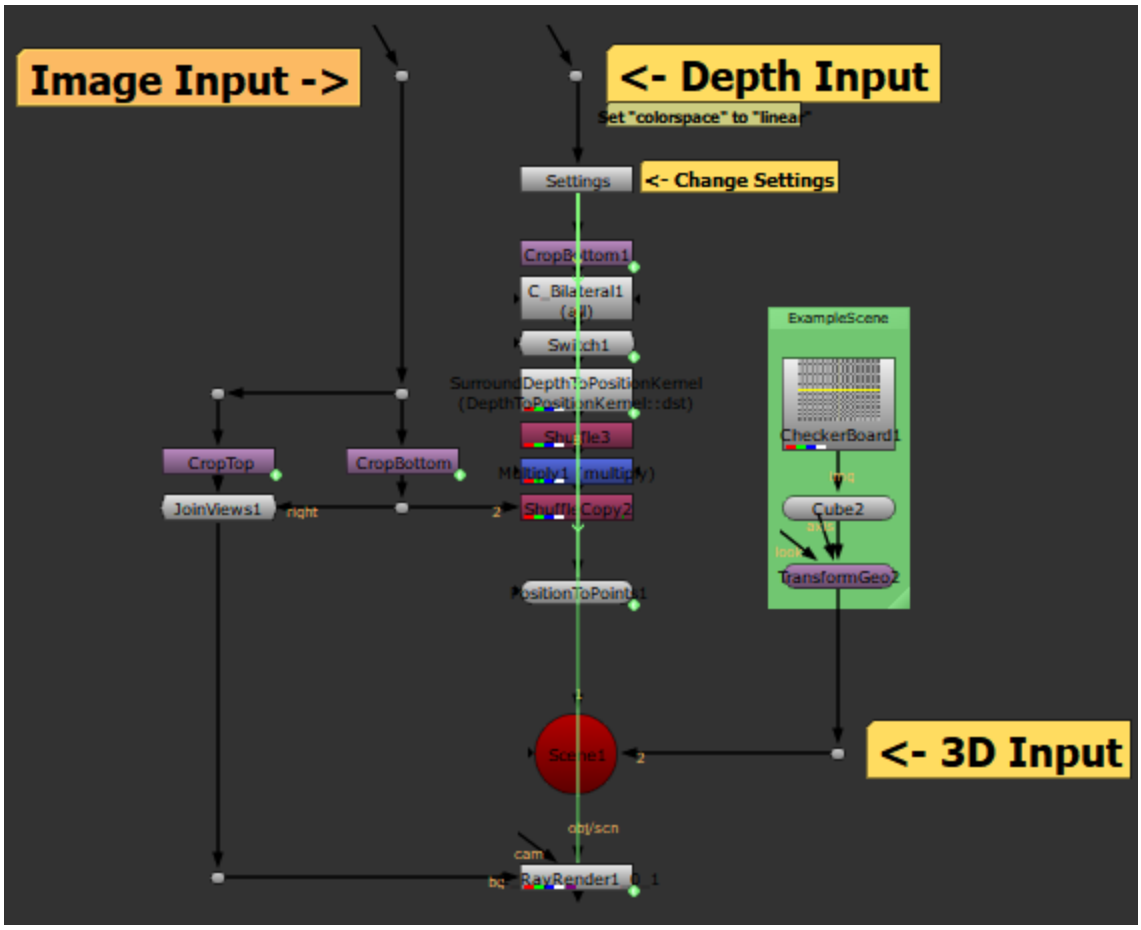
Falloff Type set to **Cosine**.

Compositing Using Facebook Surround Data

CaraVR's Facebook Surround toolset allows you to quickly extract depth information to construct point clouds for use with depth-dependent workflows. The depth data is particularly useful for positioning 3D elements accurately and then rendering into 2D through RayRender.

To set up the toolset:

1. Navigate to **CaraVR** > **Toolsets** > **Facebook_Surround_Depth_To_Points**.
   The preset node tree is added to the Node Graph.



2. Nuke prompts you to create left and right views, if they don't exist in your script.
3. Connect the source image to the **Image Input** and the depth information to the **Depth Input**.

Note: Ensure that the **Depth Input** Read **colorspace** is set to **linear**.

4. Connect a Viewer to the Read node to view depth data. Darker areas are closer to the rig and lighter areas are farther away.



In the example, the characters are in the foreground and the set is in the background.

5. Attach a Viewer to the RayRender node to view the scene.

   The depth data and camera data are then processed by a BlinkScript kernel, converted to a point cloud, and then passed into RayRender.

   The default toolset adds a cube to the scene, using the injected data to position it accurately. You can swap out the example geometry to test other objects within the scene.



6. You can also examine the scene in the 3D Viewer by pressing **Tab**. Navigate around the 360° scene using the standard Nuke navigation controls: **Ctrl**/**Cmd** to rotate the camera and **Alt** to pan.





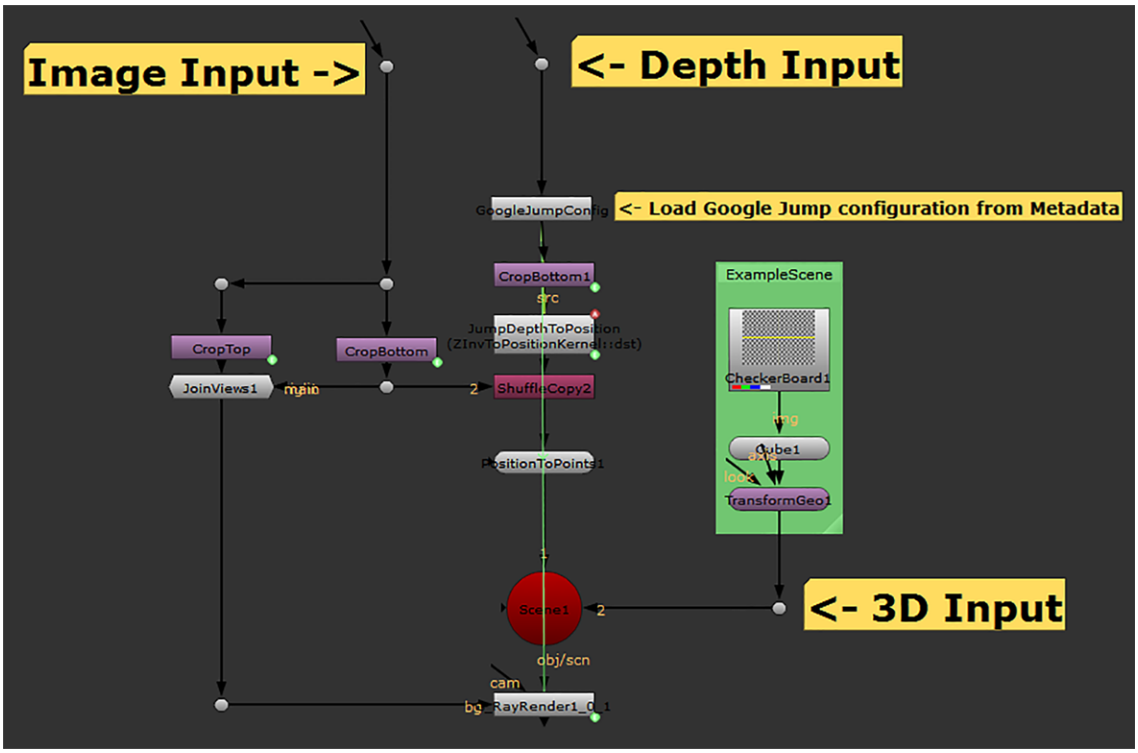
7. Adjust the values in the Settings node to control the point cloud's appearance:
   - **Eye Separation** - determines how far apart the two views are (in meters), from a viewer's perspective. If you set the Eye Separation, or interpupillary distance (IPD), too low, objects in the scene appear crushed horizontally, but raising it too high can leave holes in the stitch.
   - **Point Details** - controls the number of points in the point cloud. High values take longer to render.
   - **Cutoff** - sets the depth in meters beyond which points are omitted from the point cloud. Reducing the value removes points farther from the rig.
   - **Filter Depth Map** - when enabled, a bilateral filter is applied to the depth map in order to prevent stepping artifacts.
   - **ZInv Depth Encoding** - when enabled, use ZInv depth encoding to reverse the way depth is represented in the image, providing smoother depth maps. The inversion treats smaller values in the image as greater depth, rather than the default where small values equal less depth.

Compositing Using Google Jump Data

CaraVR's Google Jump toolset allows you to quickly extract depth information from Google Jump metadata to construct point clouds for use with depth-dependent workflows. The depth data is particularly useful for positioning 3D elements accurately and then rendering into 2D through RayRender.

To set up the toolset:

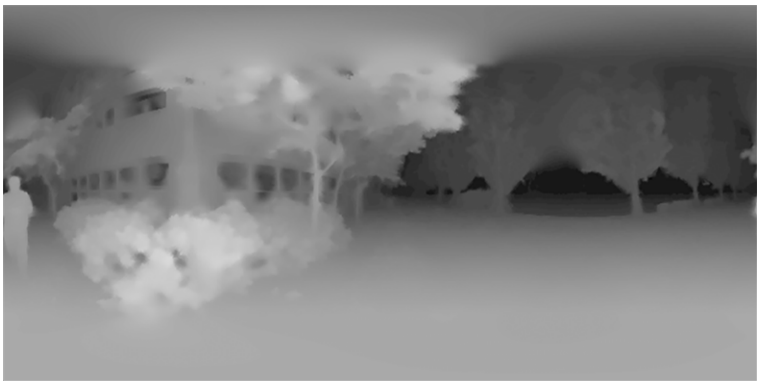
1. Navigate to **CaraVR** > **Toolsets** > **Google_Jump_Depth_To_Points**.
   The preset node tree is added to the Node Graph.



2. Nuke prompts you to create left and right views, if they don't exist in your script.
3. Connect the source image to the **Image Input** and the depth information to the **Depth Input**.

> **Note:** Ensure that the **Depth Input** Read **colorspace** is set to **linear**.

4. Connect a Viewer to the Read node to view depth data. Brighter areas are closer to the rig and darker areas are farther away.



In the example, the bushes to the left of the image are in the foreground and the trees to the right are in the background.

5. Attach a Viewer to the RayRender node to view the scene.

   The toolset reads the camera metadata from the **.json** file produced by the rig using the custom GoogleJumpConfig node.

Note: If you update the metadata, click **Load from Jump Metadata** to update the script.

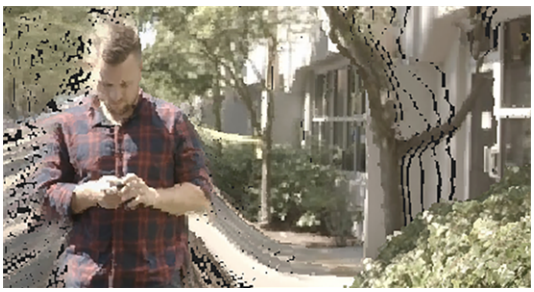
The depth data and camera data are then processed by a BlinkScript kernel, converted to a point cloud, and then passed into RayRender.

The default toolset adds a cube to the scene, using the injected data to position it accurately. You can swap out the example geometry to test other objects within the scene.



6. You can also examine the scene in the 3D Viewer by pressing **Tab**. Navigate around the 360° scene using the standard Nuke navigation controls: **Ctrl**/**Cmd** to rotate the camera and **Alt** to pan.



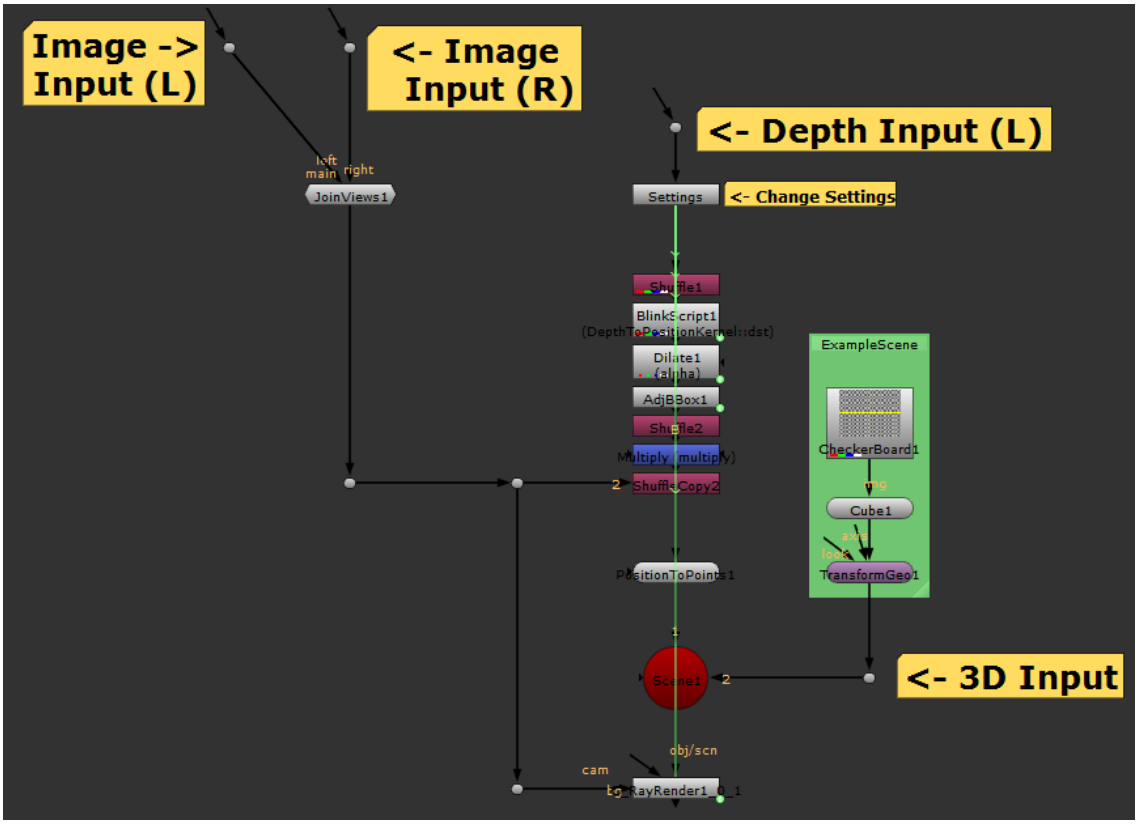


Compositing Using Nokia OZO Data

CaraVR's Nokia OZO toolset allows you to quickly extract depth information from OZO metadata to construct point clouds for use with depth-dependent workflows. The depth data is particularly useful for positioning 3D elements accurately and then rendering into 2D through RayRender.

To set up the toolset:

1. Navigate to **CaraVR** > **Toolsets** > **Nokia_OZO_Depth_To_Points**.
   The preset node tree is added to the Node Graph.



2. Nuke prompts you to create left and right views, if they don't exist in your script.
3. Connect the source images to the **Image Inputs** and the depth information to the **Depth Input**.

Note: Ensure that the **Depth Input** Read **colorspace** is set to **linear**.

4. Connect a Viewer to the depth Read node to view depth data. Darker areas are closer to the rig and brighter areas are farther away.



In the example, the trees to the left of the image are in the background and the bear in the center is in the foreground.

5. Attach a Viewer to the RayRender node to view the scene.

The depth data and camera data are processed by a BlinkScript kernel, converted to a point cloud, and then passed into RayRender.

The default toolset adds a cube to the scene, using the injected data to position it accurately. You can swap out the example geometry to test other objects within the scene.



6. You can also examine the scene in the 3D Viewer by pressing **Tab**. Navigate around the 360° scene using the standard Nuke navigation controls: **Ctrl**/**Cmd** to rotate the camera and **Alt** to pan.





7. Adjust the values in the Settings node to control the point cloud's appearance:
   - **Cutoff** - the depth in meters beyond which points are omitted from the point cloud. Reducing the value removes points farther from the rig.
   - **Dilate** - points near the edges of the cut off depth can create artifacts. The **Dilate** control adjusts the alpha mask at the edges to help reduce these artifacts.

Reviewing on VR Headsets (HMD)

CaraVR provides a monitor out plug-in for the Oculus CV1 and HTC Vive/Vive Pro, which work in a similar way to Nuke's existing SDI plug-ins.

Note: HMD devices are not enabled for Monitor Out by default. To add headsets to the Monitor Out **Device** dropdown, go to the **Preferences** under **Panels** > **Viewer (Monitor Out)** and check **Enable support for HMD devices**. You must restart Nuke for this change to take effect.

Supported Headsets

The following table shows the headsets supported by Nuke and the drivers required, by operating system:

Headset	Windows	macOS	Linux
Oculus Rift CV1 and Oculus Quest 2	Oculus SDK 1.3 or later		
HTC Vive/Vive Pro	SteamVR		

Note: HMD performance on Linux using OpenHMD is limited to the refresh rate of your main monitor, as set in your GPU driver preferences. For example, if your monitor refreshes at 60 Hz, the HMD cannot refresh faster than that and you may experience performance issues.

Reviewing in Nuke's Node Graph

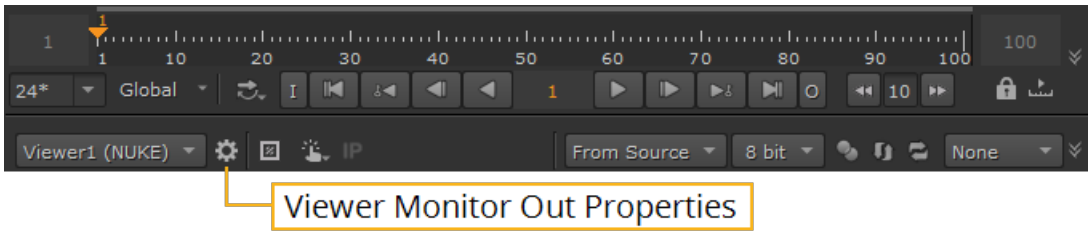
You can review your VR work from the Node Graph environment using Nuke's Monitor Out as follows:

Article: See Knowledge Base article Q100455 for information on how to configure headsets on Linux operating systems.

1. Ensure that your headset is plugged in and active on your OS before launching Nuke.

Note: If you're using a headset on multi-GPU machines, ensure that the device is connected to the GPU that handles the monitor out, not the internal computation.

2. Navigate to **Workspace** > **Monitor Out** or press **Shift**+**F7** on your keyboard to display the **Monitor Out** controls.
3. Click the cog icon in the bottom-left of the **Monitor Out** panel to display the **Properties** panel.



4. Select the required headset from the **Device** dropdown.

   The image is expected to be in latlong, so for mono output 2:1 lat long and for stereo output 4:1 side by side.

   For stereo view support, either:

   - write out the left and right views into a single frame comprised of two latlongs horizontally, with left eye to the left and right eye to the right, or
   - use the SideBySide node to display the left and right views from a multi-view **.exr** or **.sxr** file.

Tip: If you're playing back footage in Nuke's Viewer, you can right-click in the Viewer and select **No incomplete stereo**, for a more comfortable experience.

See Reviewing Your Work with Monitor Out in Nuke (NDI - SDI/HDMI) for more information on Monitor Out in Nuke.

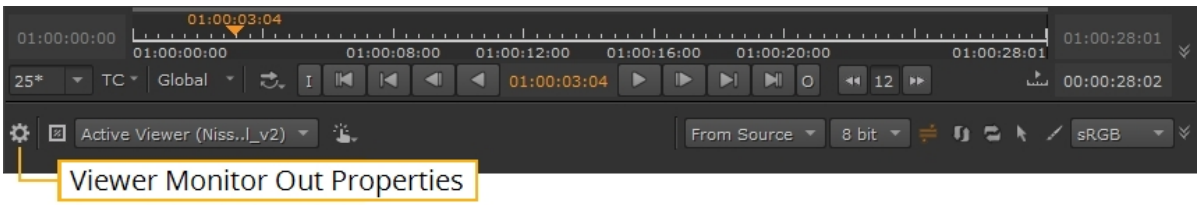
Reviewing in Nuke Studio's Timeline

You can also review your VR work from the timeline environment using Nuke Studio's Monitor Out as follows:

1. Ensure that your headset is plugged in and active on your OS before launching Nuke Studio. See Installation and Licensing for more information.

Note: If you're using a headset on multi-GPU machines, ensure that the device is connected to the GPU that handles the monitor out not the internal computation.

2. Navigate to **Workspace** > **Monitor Out** or press **Shift**+**F7** on your keyboard to display the **Monitor Out** controls.
3. Click the cog icon in the bottom-left of the **Monitor Out** panel to display the **Properties** panel.



4. Select the required headset from the **Device** dropdown.

See Monitor Output in Nuke Studio (NDI - SDI/HDMI) for more information on Monitor Out in Nuke Studio.

Setting the Field of View

You can override the default field of view (FOV) setting in headsets using the FN_CARA_HMD_FOV environment variable. The variable accepts a numeric value representing the angle you want to set as the FOV in the headset. There's no minimum, but values over 110° can produce unexpected results.

For more information on setting environment variables, see Setting Environment Variables.

Written Tutorials

Welcome to the written tutorials!

Introduction

If you've gone through the Getting Started section - which we highly recommend - you already know something about Nuke. These tutorials show how to pull everything together through a series of practical examples.

You can also go through these tutorials using Nuke Non-commercial, but some features used in the tutorials may be disabled in the non-commercial version. For more information, see About Nuke Non-commercial.

The Projects

These tutorials include the following projects:

- Tutorial 1: Compositing Basics explains the Nuke user interface, project workflow, and basic compositing tasks.
- Tutorial 2: 2D Point Tracking demonstrates how to track image patterns, stabilize footage, lock down images for clean plates, and match-move.
- Tutorial 3: Keying and Mattes shows you how to pull mattes with standard keying tools and Nuke's own image-based keyer.
- Tutorial 4: 3D Integration shows how you can use Nuke's 3D workspace to help your 2D compositing.

Installing the Project Files

Before you continue, download the tutorial project files from our website and move them to a directory you'll create, called "Nuke_Tutorials". It's up to you where you put your tutorial files, but here's our recommendations below depending on your operating system. Whatever you do, you'll need to remember where you put these files.

Tip: If you're using a Mac or Linux system, log in under administrator privileges to avoid issues with permissions when installing the files.

To Create the Tutorial Directory (Windows)

1. On the Windows desktop, double-click the **My Computer** icon to open a file browser.
2. Double-click on **Local Disk (C:)** and navigate to where you would like the content to reside.
3. Click the right-mouse button over the displayed directory and choose **New > Folder**.
4. Name the folder: **Nuke_Tutorials**.

To Create the Tutorial Directory (Mac or Linux)

1. Open a shell or terminal window.
2. At the command line, enter **mkdir ~/Nuke_Tutorials** to create the tutorial directory under your user or "home" directory.

To Download and Install the Project Files

1. Click Nuke Tutorials to download the project files to your local computer.
2. Extract the downloaded files and move (or copy) them to the **Nuke_Tutorials** directory you created earlier.

   You're now ready to start the first tutorial with Nuke.

Tutorial 1: Compositing Basics

Hello! This tutorial is your introduction to Nuke, where you'll create a simple composite and breeze through most of the windows, on-screen controls, and other user interface items.

Introduction

We've heard rumors that many people would rather march through icy rain than review an introductory tutorial on digital compositing. Certainly, that's not you. When you finish this lesson you'll have a good understanding of the Nuke workflow and should feel confident about approaching the other tutorials.



Your first composite in Nuke.

Before you get into the project, we have some administrative things to do - such as defining a few application preferences and project settings. We know this sort of thing is not terribly exciting, but it is terribly important, so please be patient and we'll get through it as quickly as possible.

Note: If you haven't already downloaded and moved the tutorial project files to the Nuke_Tutorials directory you created, turn to Installing the Project Files for instructions.

Starting Nuke

The Nuke icon may appear on your desktop. If so, double-click it to launch the application. Otherwise, start Nuke with one of the methods described below, assuming you have installed Nuke to the default location.

To Launch Under Windows

• From the **Start** menu, choose **All Programs > The Foundry**, and then select **Nuke17.0v3**.

To Launch Under Mac

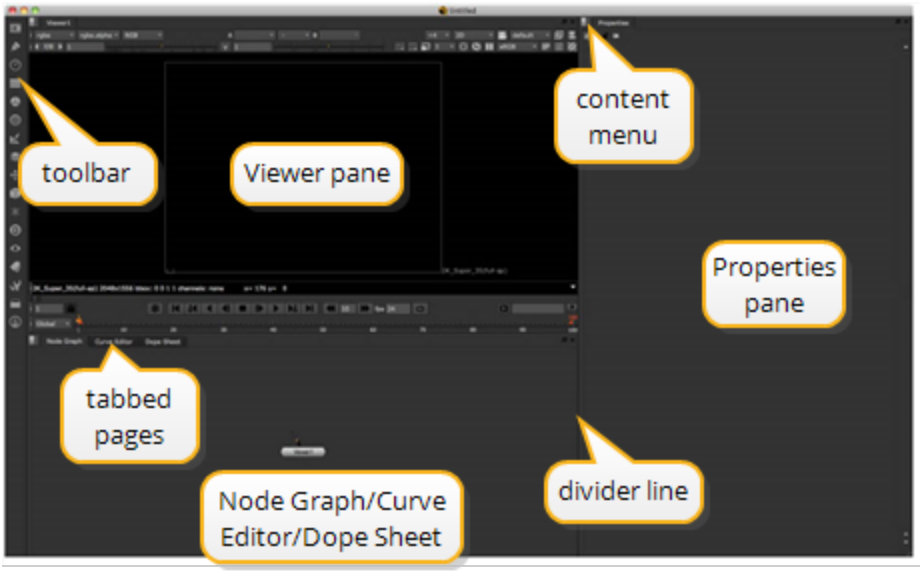
• Open the /**Applications/Nuke/** folder and double-click the **Nuke17.0v3** icon.

To Launch Under Linux

• Open the **/usr/local/Nuke17.0v3/** folder and double-click the **Nuke17.0v3** icon.

Tip: If you're operating under Linux, you can also launch Nuke from the command line of a terminal. Simply navigate to the Nuke directory and enter the name of the Nuke application.

A clean copy of the main Nuke window appears. Divider lines organize the window into different panes. Each pane has one or more pages of content, separated by tabs at the top of the pane. The Toolbar appears at the left edge of the main window.



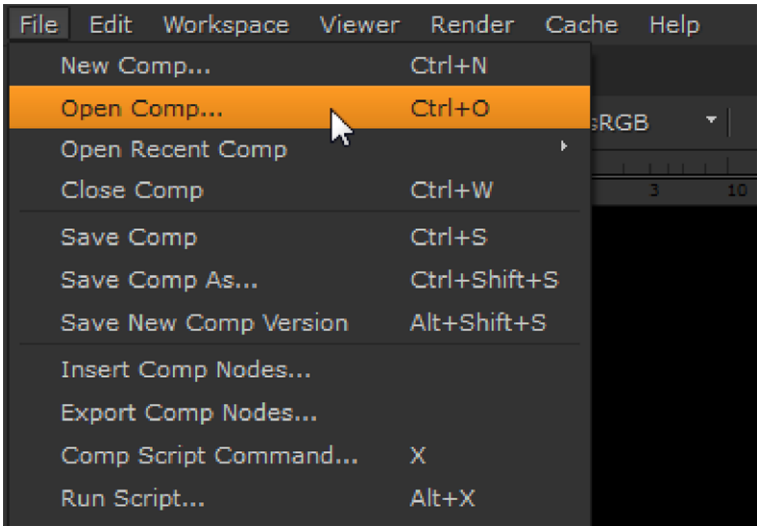
By default, the panes are setup to display the Viewer, the Node Graph/Curve Editor, and Properties. You'll create the script for this project inside the Node Graph page on the Node Graph/Curve Editor pane. We'll talk about each of these on-screen controls when you need them for the project.

Using the Toolbar

The Toolbar includes the options you can use to build your project, such as importing images, layering images, drawing shapes and masks, applying color correction, and so on. Each Toolbar icon displays a menu of *operators* or *nodes* that you can select. Roll the mouse pointer over the Toolbar and you'll see pop-up tool tips that identify each icon.

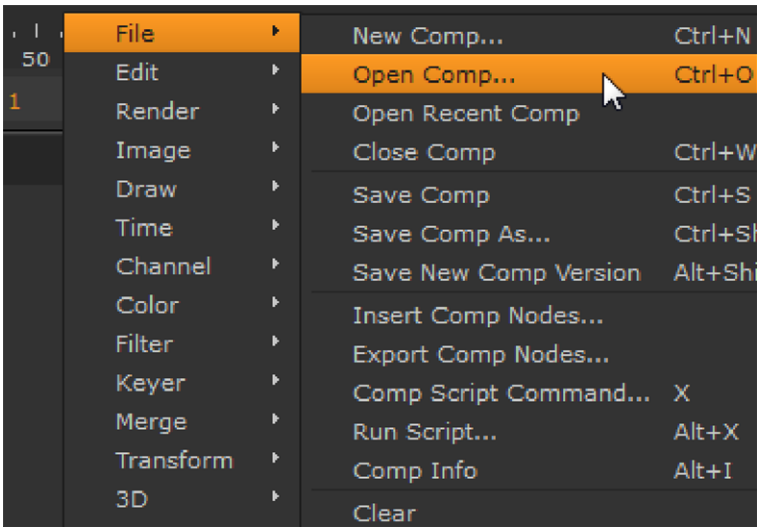
Using the Menus

The Nuke menu bar appears at the top of your screen, outside the main window. This menu begins with the options **File**, **Edit**, **Workspace**, and so on. When instructed to do so, make selections from the menu bar, or click the right mouse button to choose from a pop-up version of the menu bar.



The Nuke menu bar.

The "right-click" menu is highly contextual. Its options change according to the location of the mouse pointer. Right-click over the Node Graph, for example, and you'll see the options from the menu bar and the nodes you can insert from the Toolbar. Right-click over the Viewer pane and you'll see a menu of Viewer options.



The "right-click" menu.

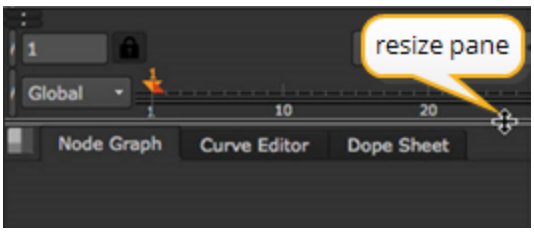
Try the right-click menu when you can't find appropriate controls or menu options. Many features are hidden in the menu until you're in the situation where you need to use them.

Note: Nuke's menu bar, at the top of the screen, is organized a little differently between the operating systems, but the right-click menu contains the same options, regardless of the system you're using to run Nuke.

Customizing Your Workspace

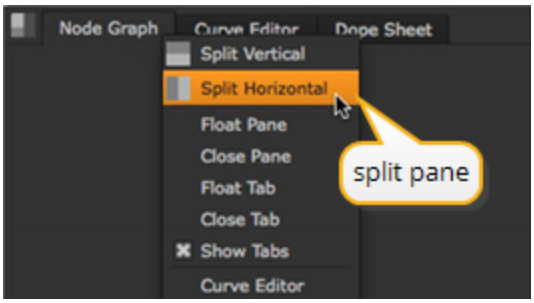
Nuke gives you several options for customizing the window layout. It's time for you to claim your copy of Nuke and make it your own! You don't need to customize the layout for this lesson, but why not try it now for your own personal amusement? Here are some things you can do to reorganize the window layout:

• Drag a divider line between panes to change the size of the panes.



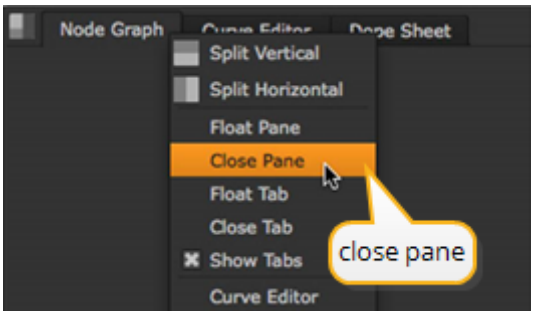
Resizing a pane.

- To divide a pane, click on the content menu (the checkered box at the upper-left corner of each pane), and choose **Split Vertical** or **Split Horizontal**.



Splitting a pane.

- To discard a pane, click on the content menu and choose **Close Pane**.



Closing a pane.

- To add a new tabbed page to a pane, click on the content menu and choose one of content options, such as **New Viewer** or **Curve Editor**.
- Click on the "x" inside a tab to discard a tabbed page.



Closing a tab.

- To move a tabbed page, drag the tab to another pane inside the main window.
- To tear-off a page as a floating window, drag the tab outside the borders of the main window, or simply **Ctrl**+click (Mac users **Cmd**+click) on the tab name.
- Drag a floating window into a pane, inside the main window, to convert it to a tabbed page.
- From the menu bar, choose **Workspace > Save Workspace** to save the current layout. Choose **Workspace > Restore Workspace** *x* to apply a previously-saved layout.

Saving Files and File Backup

We assume you already know how to save files (Hint: choose **File > Save Comp As**). In addition, Nuke includes an autosave feature, which helps recover project files after a system failure. Yes, we know that *never* happens to you, but in the unlikely event that it does, you won't lose your work when you have autosave enabled.

Defining File/Saving Options

1. Click the right mouse button over the Node Graph pane, and choose **Edit > Preferences**.

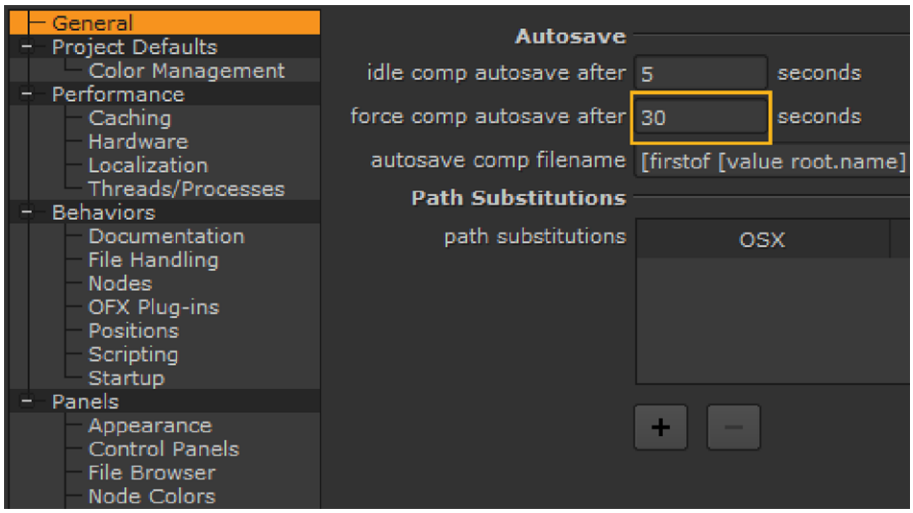
   Notice the **autosave filename** directory is set to:

   [firstof [value root.name] [getenv NUKE_TEMP_DIR]/].autosave

   You don't need to make a change; this simply tells Nuke to store automatic backup files in the same directories as your project files or the path supplied by the NUKE_TEMP_DIR environment variable (for more information on environment variables, see Configuring Nuke).

   Now, how often would you like Nuke to generate an automatic backup while you're working? Every five minutes?

2. Change the **force comp autosave after** option to **300** seconds, to generate an automatic backup every five minutes.



3. Click **Save Prefs** to keep the changes and then **Close** to return to the main window.

   If you close this dialog box without clicking the Save button, then the changes affect only the current session of Nuke.

Recovering Back-Up Files

You may ask, "How do I recover a back-up file in the event of a system or power failure?" Good question! When you relaunch Nuke, you'll see a message that asks if you want to recover the **.autosave** file for the project that was last open. Click **Yes** and Nuke opens the back-up file.

Tip: The **.autosave** files can still be useful, even when you properly exit Nuke, because they are not deleted from the directory. You can, for example, rename an **.autosave** file to create an archive of the previous version of your project file.

Sometimes you may see the recovery message even though you have not experienced a system failure. This happens when you exit Nuke without saving the changes to a project file, and Nuke recognizes that the time stamp on the **.autosave** file is later than the Nuke project file you're trying to open. In this case, you decide which version of the project file you want to open.

Turning off Automatic Back-Up

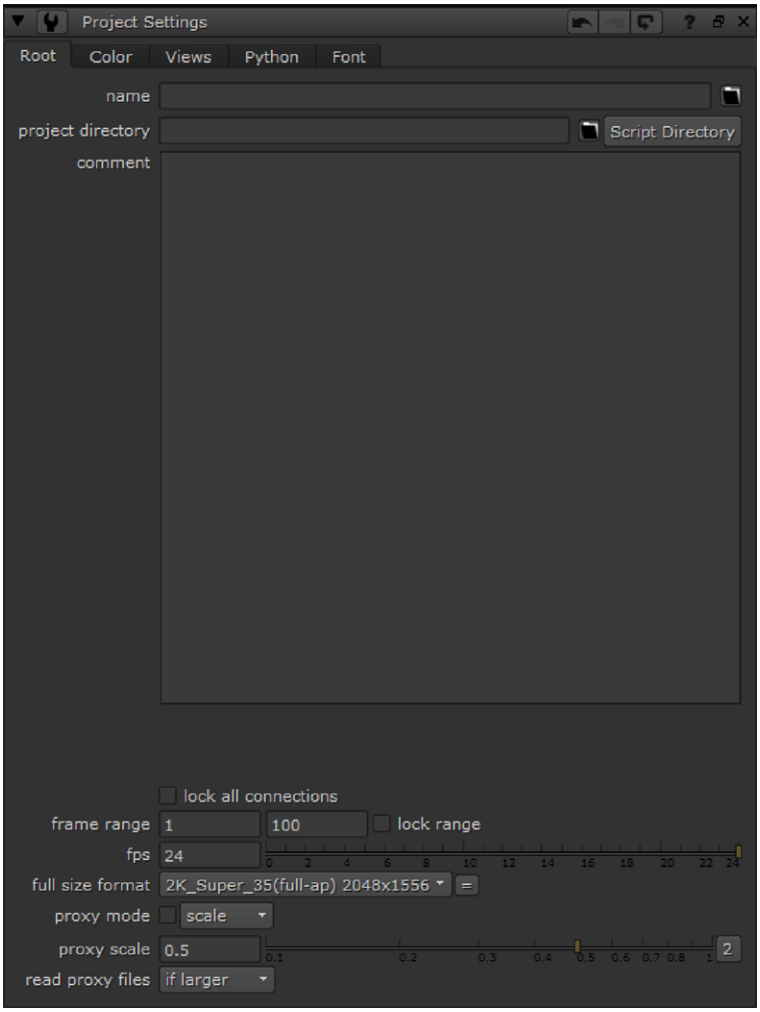
Okay. You're reading this, so we assume you're a freewheeling rebel who possibly enjoys the risk of losing your work. It's an adrenaline thing. Or perhaps you prefer to do everything yourself, manually, and you have a secret obsession for saving your files. Whatever the reason, you can disable the autosave features by setting the intervals for both "autosave idle" and "force autosave" to zero seconds. That's all you need to do. Good luck.

Setting Up the Project

When you start a new project, you need to define project settings for length or frame range, the number of frames per second for playback, and the output format. These options appear on the **Project Settings** dialog box.

To Set Up Your Project

1. Click the right mouse button over the Node Graph, and then choose **Edit** > **Project Settings**.



2. In the **frame range** fields, enter a range of **1** to **28**. This is the length of the shot we create for the project.
3. Enter **24** as the frames per second (fps).
4. Click the **full size format** dropdown menu and choose **PC_Video 640 x 480**.
5. Close the **Project Settings** control panel.

Note: On the **Project Settings** control panel, the **Color** tab includes options that ensure color integrity for your display and output devices. You don't need to change the LUT for these tutorials, but we recommend that you research and set these options for your own projects.

Until now, everything you've done is standard procedure for a new project. You used the menu bar to access several features during the setup process, and now you'll use the Nuke toolbar to insert nodes and create a compositing tree.

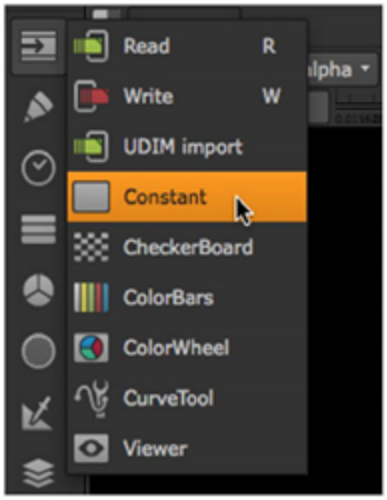
Working with Nodes

A *node* is simply one of the building blocks for the list of operations you want to complete. A *node tree* is a diagram that shows the order in which the operations are performed. Do the following to add a few nodes and start your node tree. The result creates the background for the project.

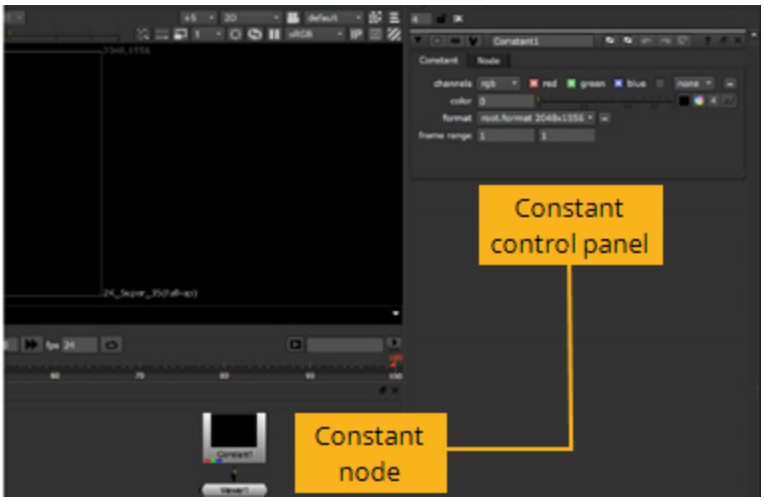
Inserting Nodes

To insert nodes:

1. On the Toolbar, click the first icon to display a menu for nodes that are in the **Images** category.
2. Select **Constant** from the menu to insert this node into the Node Graph pane.



When you insert a new node, its control panel also displays with parameters that let you define what the node produces. In this case, the Constant node creates a solid color backdrop.



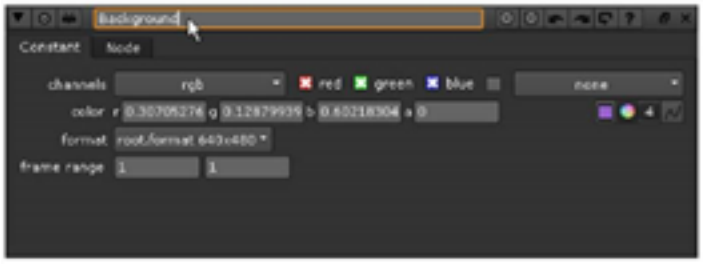
3. In the Constant control panel, click on the color wheel to open the **Color Picker**.



4. Drag the color sliders and the cursor inside the wheel to choose a light color, something appropriate for the "horizon" of the composite background. Then, close the color wheel window.

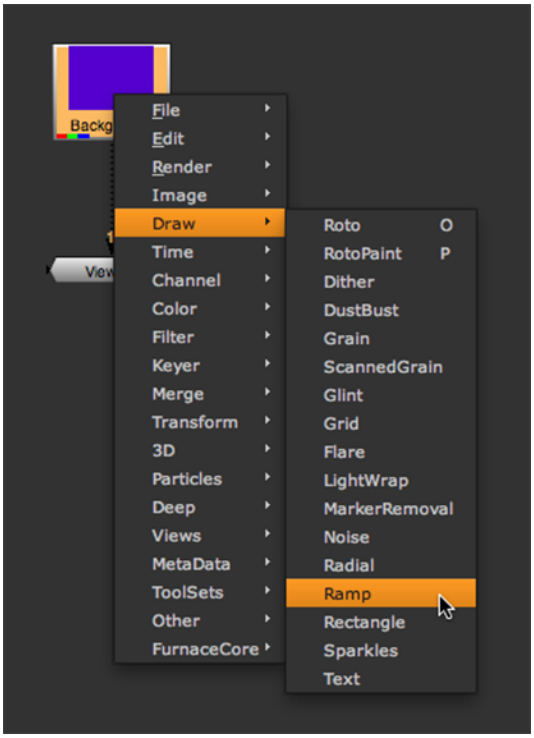
   At this point, you should probably rename "Constant" to something more descriptive.

5. Inside the control panel, click on the **Constant** name. You can now edit the name, so type **Background** and press **Enter**.

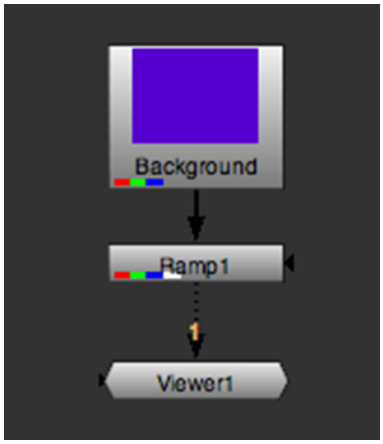


From here onward, we'll call this node the "Background" node.

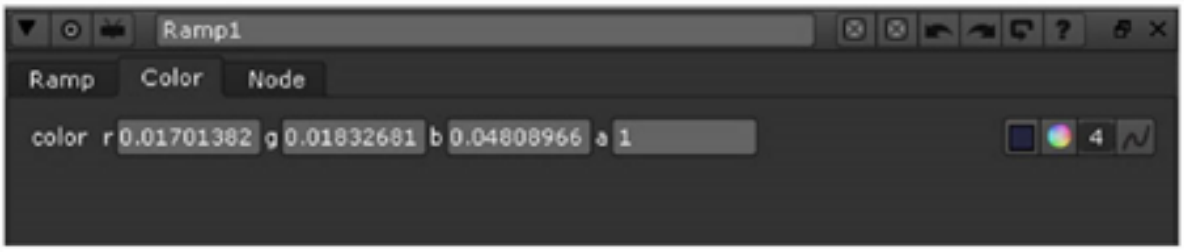
6. Close the control panel for the Background node. When you need to reopen it, just double-click the node and the control panel reappears.
7. Click on the **Background** node to select it. Then, click the right mouse button and choose **Draw > Ramp**.



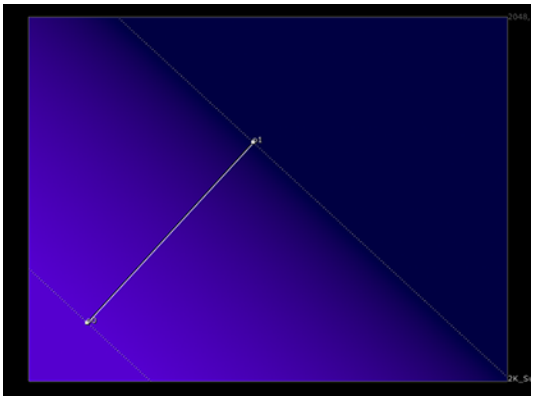
8. Drag the tail of the arrow from the **Viewer1** node to the center of the **Ramp1** node. You'll see the output of the Background node and the ramp controls displayed in the Viewer window.



9. Click the **Color** tab inside the control panel for **Ramp1**. Then choose a dark color that blends well with the color you selected for the Background node.



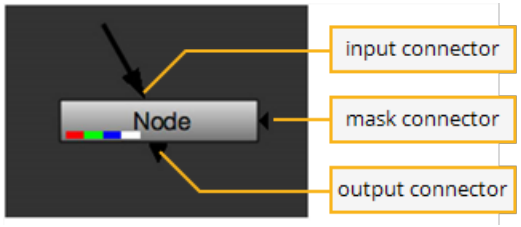
10. Click the Ramp tab in the control panel to reactivate the overlay controls. Then, drag the **p0** and **p1** control points to adjust the spread and angle of the ramp over the background.



11. When you're happy with the results, close the Ramp1 control panel to remove the overlay.

Connection Tips

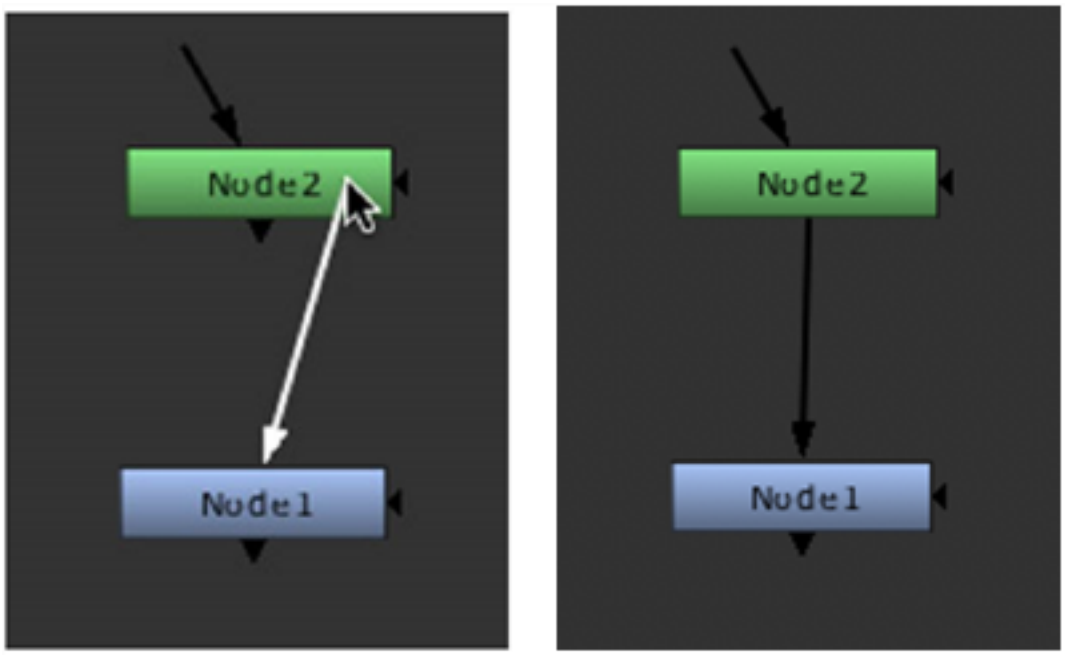
Most nodes have input and output connectors that are used to establish the order in which the operations are calculated.



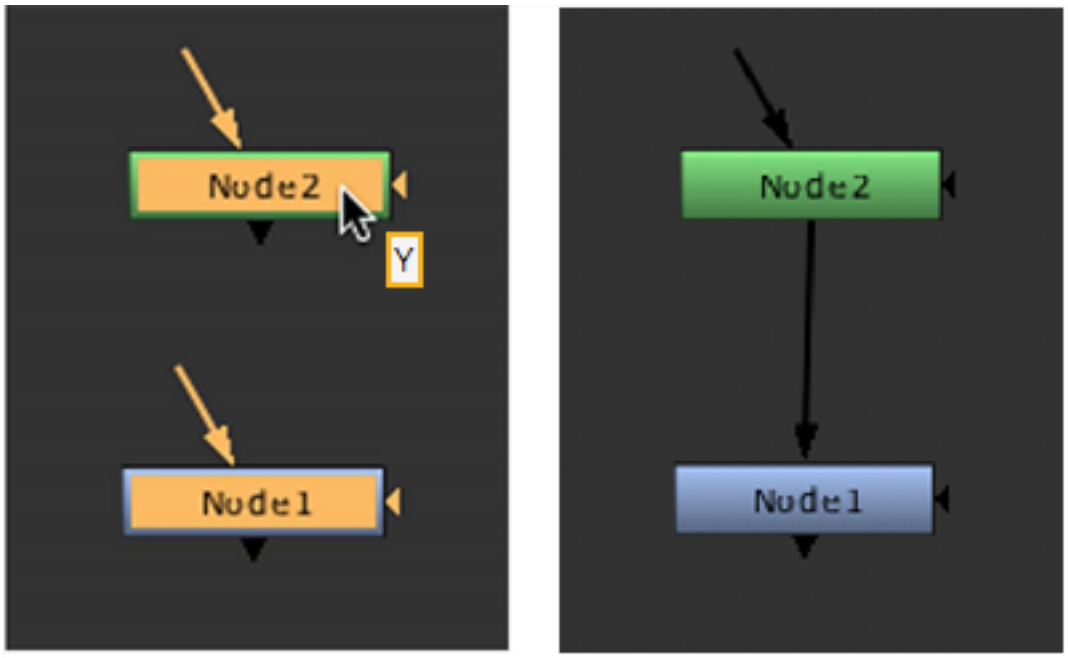
Connectors on a node.

Try the following to connect nodes after you insert them into the Node Graph:

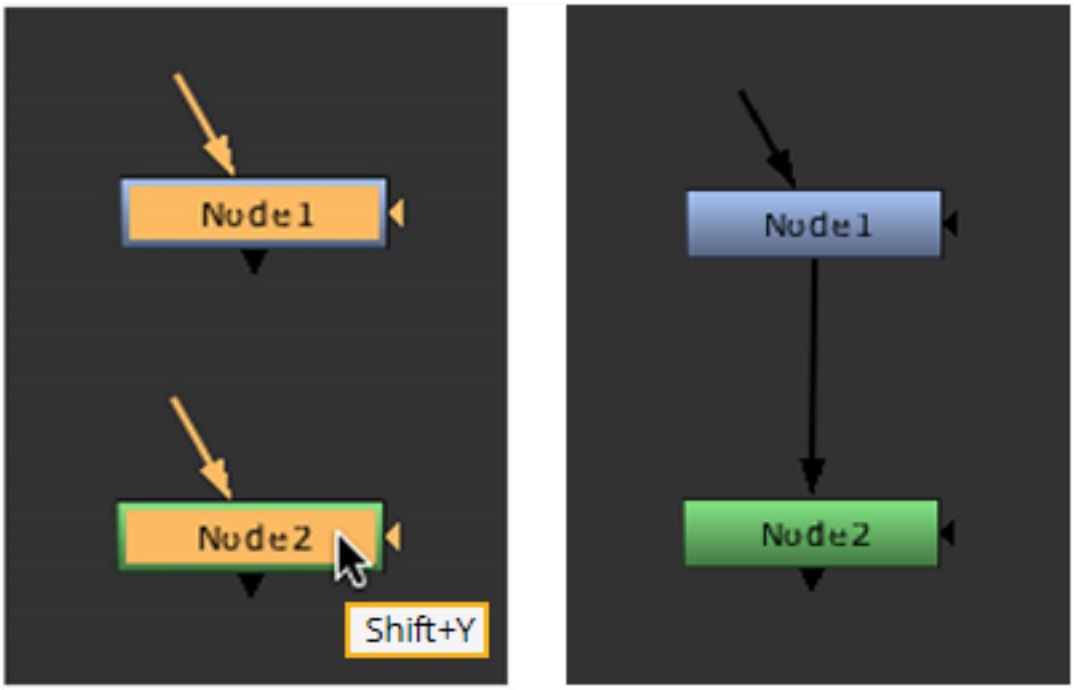
• Drag an input or an output connector onto another node to establish a connection.



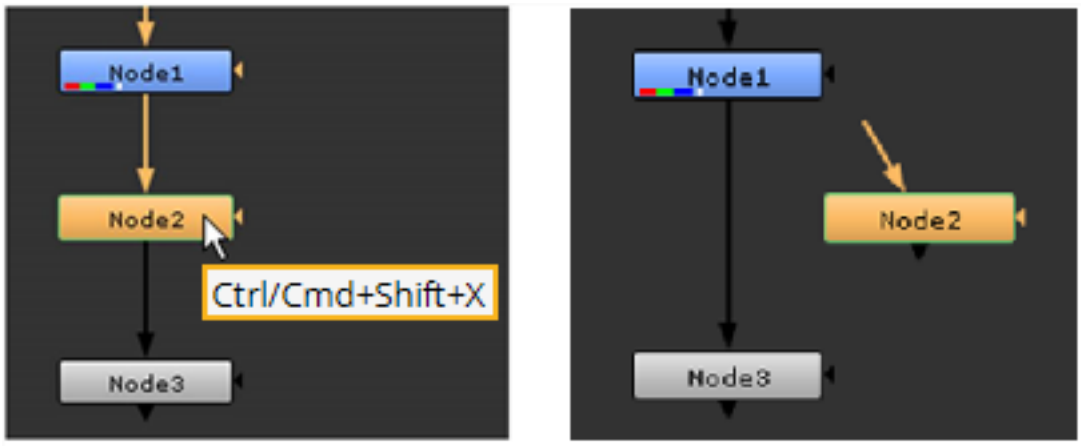
- Select a node, press the **Shift** key and select a second node. Then press **Y** to connect the first node to the output of the second node.



- Select a node, press the **Shift** key and select a second node. Then press **Shift**+**Y** to connect the second node to the output of the first node.



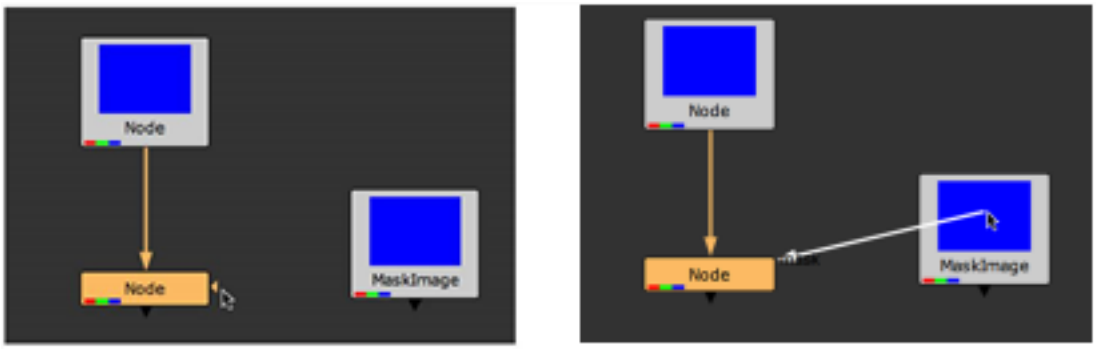
• Select a node and press **Ctrl/Cmd+Shift+X** to extract the selected node from the tree.



• For nodes that have two inputs, select the node and press **Shift+X** to swap the A/B inputs.



• Drag the mask connector to the node that provides the image you want to use as the mask for the selected node.



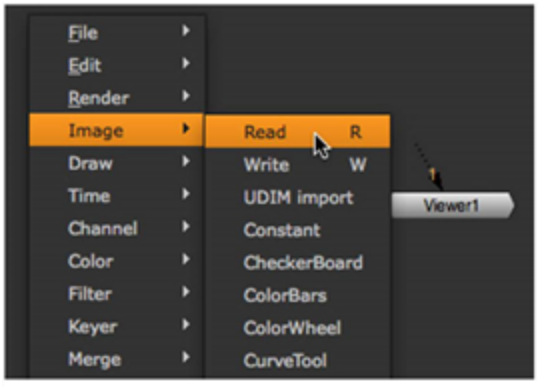
Importing Image Sequences

For this project, you need to import a few image sequences for the foreground elements and a background plate.

To Read the Images

1. Click on a blank space in the Node Graph. This ensures none of the nodes are selected.
2. Click the right mouse button over the Node Graph and choose **Image** > **Read** (or press **R** over the Nuke window).

Tip: Pressing **R** with an existing Read node selected, opens the file browser at the location specified by that node.

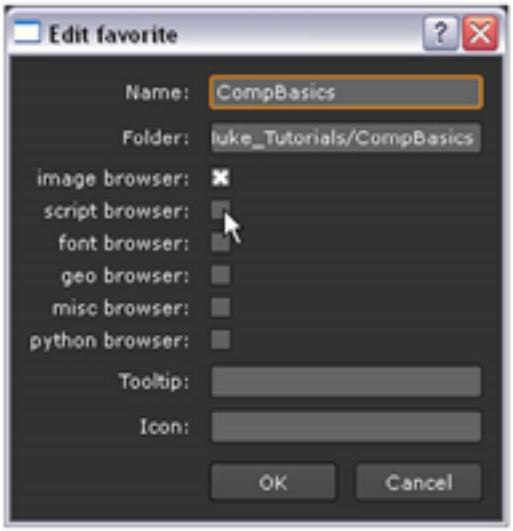


A file browser appears. This is where you select the image file you want to import. When you browse through your directories from this window, Nuke displays sequentially-numbered files as one item in the directory.

3. Browse to the **Nuke_Tutorials/CompBasics/** directory.
4. Add a bookmark to this directory. Right-click over the list, on the left side of the file browser window, and choose **Add** from the menu.



5. Type a name for the bookmark or keep the default, which is the directory name. Then click **OK**.



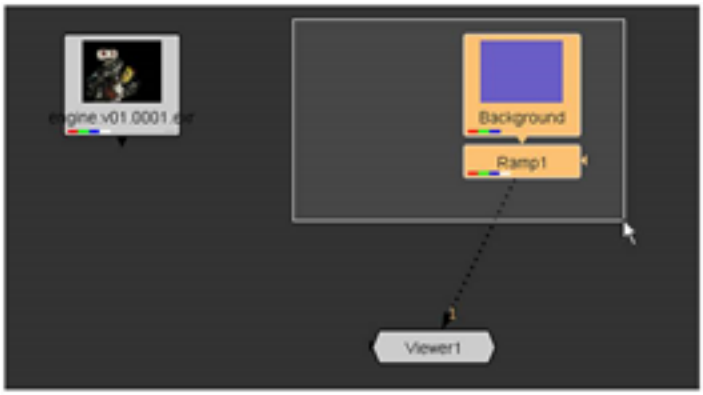
6. Open the **engine_rgba** directory, select the **engine.v01.####.exr** image sequence, and click **Open**.



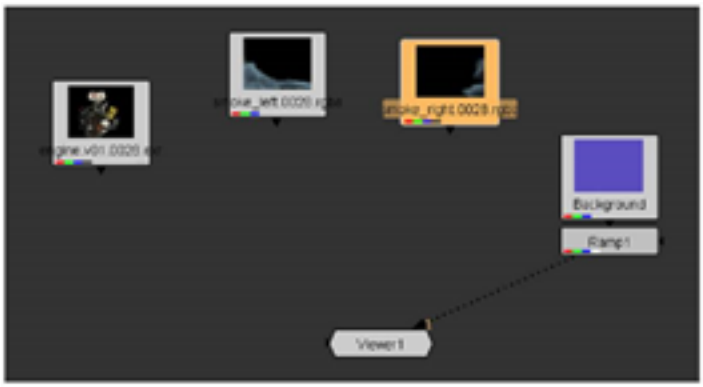
Nuke retrieves the image sequence and displays it as a thumbnail on the node. The Read control panel displays the resolution and the frame range for the image.

Note: Nuke reads images from their native format, but the Read node outputs the result using a linear color space. If necessary, you can change the **Input Transform** option in the Read node's control panel, or insert a **Color** > **Colorspace** node to select the color scheme you want to output or calculate.

7. Drag a marquee (hold down the left mouse button while dragging) around the Background and Ramp nodes to select them. Then drag them to the right to make room for additional nodes.



8. Choose **Image** > **Read** from the right-click menu to import another image sequence. Use the file browser to select the image sequence stored in **Nuke_Tutorials/CompBasics/smoke_left.wh/smoke_left.####.rgba**.
9. Add one more Read node and retrieve the image sequence stored in **Nuke_Tutorials/CompBasics/smoke_right.wh/smoke_right.####.rgba**



10. Arrange the nodes, as shown above, to allow some room to create the connections for the node tree.

Navigating Inside the Windows

The Node Graph panel can seem very small, especially when your node tree grows. True, you already know how to resize and tear-off the windows, but sooner or later you may run out of display real estate. It's time to learn some navigation controls that can help you work in the Node Graph (and other windows) in Nuke. Try the following navigation controls:

Panning Your View

- Windows/Linux: While pressing the **Alt** key and the left mouse button, drag the mouse pointer across the Node Graph.
- Mac: While pressing the **Option (alt)** key and the left mouse button, drag the mouse pointer across the Node Graph.

  As you drag the mouse, you pan your view of the Node Graph.

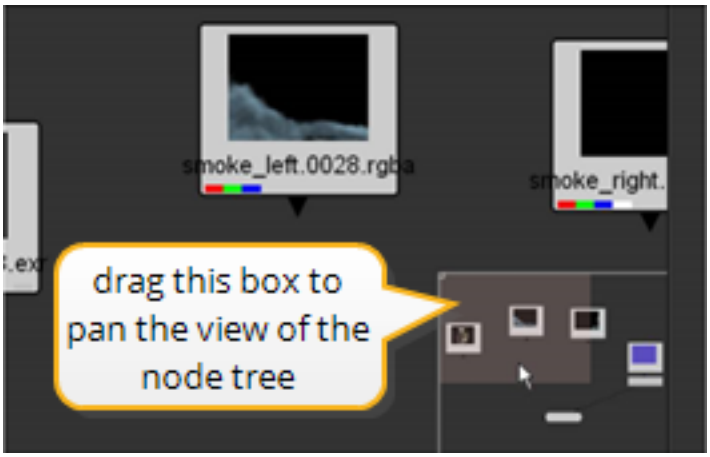
Zooming or Magnifying Your View

- Windows/Linux: While pressing **Alt** and the middle mouse button, drag the mouse pointer across the Node Graph.
- Mac: While pressing **Option (alt)** and the middle mouse button, drag the mouse pointer across the Node Graph.

  Drag to the right and you'll zoom-in. Drag to the left and you'll zoom-out.
- Keyboard zoom-in/out. Tap the plus (**+**) key to zoom-in. Tap the minus key (**-**) to zoom-out.

Using the Node Graph Overview

- When the node tree extends beyond the borders of the window, a navigation box appears in the lower-right corner of the Node Graph. Drag the shaded rectangle inside the box and you'll quickly pan to another view of the node tree.



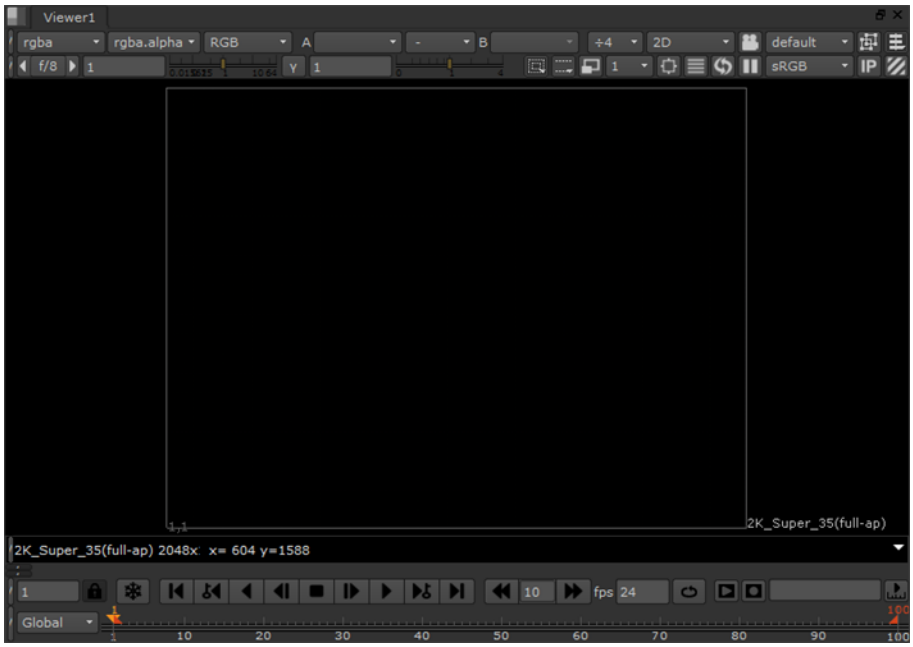
Framing the View in the Window

- Press the letter **F** on your keyboard to fit the entire contents of the node tree within the borders of the Node Graph.

The navigation controls for the Node Graph also work inside the next window on our agenda, the Viewer.

Working with Viewers

The *postage stamps* on the nodes - those little pictures, often called thumbnails - show what each node passes onto the next node in the tree. Although quite lovely, they won't do for real compositing work. You need to open a Viewer window to see the full picture.



You can open several Viewers at once. In addition, you have up to 10 pages, or buffers, for each Viewer window; these allow you to toggle between different views along the node tree.

When you start Nuke, you see a default Viewer node in the Node Graph. You can easily drag the connection arrow from a node onto the Viewer to display the node's output. You can open additional Viewers by choosing **Viewer** > **Create New Viewer** from the menu bar or by pressing **Ctrl**+**I**.

Displaying the Images in a Viewer Window

To display the images in a Viewer window:

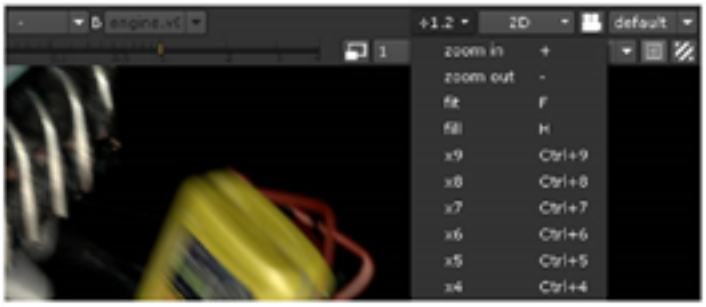
1. Drag the connector from the Viewer node onto the Read node for the **engine.v01** clip.



Here's an alternate method: Select the **engine.v01** clip node and then press **1** to connect to the Viewer node. Nuke displays the node's output in the Viewer window.



2. Press the **Alt** key (Mac users press **Option**) and the left mouse button, and drag the mouse pointer across the Viewer window to pan.
3. Press **Alt** (Mac users press **Option**) and the middle mouse button, and drag to zoom in/out. You can also use the "zoom" dropdown menu at the top of the Viewer to magnify the view.



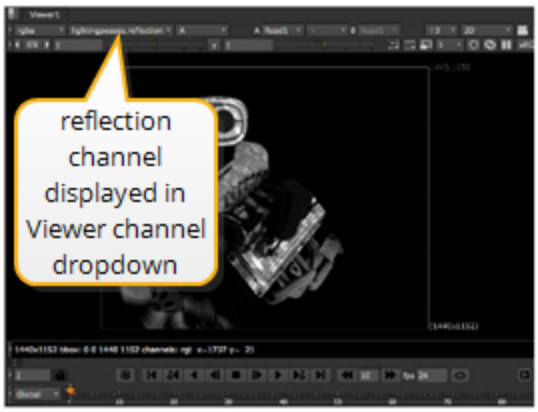
4. Press **F** to fit the current image into the borders of the Viewer window.

   This image has different *channels* of information you can view. The "RGB" label appears at the top because the Viewer now shows the result of the red, green, and blue channels.
5. To view individual color channels, press **R** (red), **G** (green), **B** (blue) or **A** (alpha). As you press each keyboard shortcut, the label at the top of the Viewer reflects the displayed channel.



6. Press one of the channel keyboard shortcuts again to return to the "RGB" display, or choose **RGB** from the Viewer's channel dropdown menu.

   In addition to the standard color channels for red, green, blue, and alpha, this image also includes channels for specular highlights, reflections, and other masks.
7. To view additional channels, press **A** to display the alpha channel, and then select the **lightingpasses.reflection** channel from the Viewer channel dropdown menu.



   You now see the reflection mask from the image file.
8. Select **rgba.alpha** from the Viewer channel dropdown menu to reset this as the preferred channel when you press the A key.
9. Press **A** again to toggle the display and show all color channels.

Viewing Multiple Inputs

To view multiple inputs:

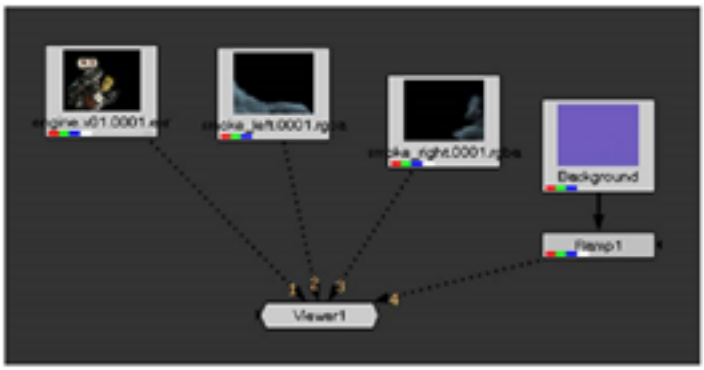
1. Select the Read node for the **smoke_left** clip, and press **2** at the top of your keyboard or on the numeric key pad.

This creates a second connection to the Viewer from the selected node. When the cursor is over the Viewer, you can press a number on the keyboard to pick the connection you want to view.

2. Move the mouse pointer over the Viewer and press **1** to display the **engine.v01** clip. Press **2** to display the result of the **smoke_left** node.

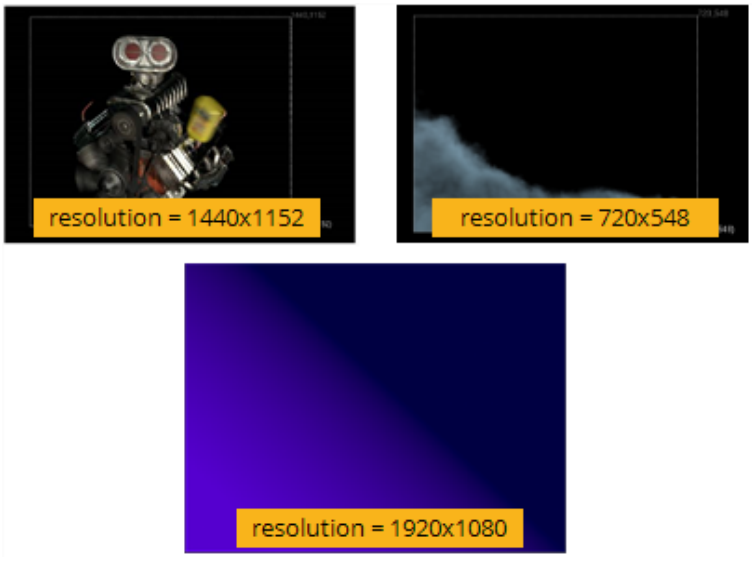
   In this manner, you can connect multiple images to the same Viewer and then switch between the images.

3. Select each of the other nodes and press a number to establish a connection to the Viewer.



4. Move the mouse pointer over the Viewer and press the numbers on your keyboard to display each of the connected nodes.

   As you switch between the different views, the images may appear to be the same size. However, if you look in the lower-right corner of the Viewer, you'll see the images have different resolutions.



These images have different resolutions.

Nuke allows multiple resolutions in one composite, but you need to conform these images to match the project resolution. This allows the elements to be properly aligned in the composite.

Reformatting Images

Elements created within the Nuke script, such as Background and Ramp, automatically inherit the global format and that's how you want it for this project. The imported images, however, do not conform to the project settings and must be reformatted.

To Conform Images to the Project Format

1. Click the Read node for the **engine.v01** clip to select it.
2. Click the right mouse button and choose **Transform > Reformat**.
3. Repeat steps 1 and 2 for all the Read nodes in the Node Graph.
4. Move the mouse pointer over the Viewer, and press the keyboard numbers (**1**, **2**, and **3**) to switch between the connected images.

   Each image should now conform to the project format.

If you change the delivery format in the project settings, then all elements set to "root.format" also change to the new project settings. If you neglect to reformat images when you read them into the project, the images retain their original format, independent of the project settings.

Using Proxies and "Down-res"

Proxies are low-resolution versions of the final image you intend to create. For many compositing tasks, the low-res version can help you work faster. Then, when you're ready to create the final output, switch proxy mode off and return to the full-res version.



Full resolution.



Proxy resolution.

Nuke can generate proxies on-the-fly, according to the *scale* or *format* of your images. You select the method under **Edit** > **Project Settings**.

To toggle the proxy resolution defined under Project Settings, you use the "proxy" button on your Viewer. Alternatively, you use the "down-res" button to lower the display resolution of individual Viewers. The down-res button works both in the full-res and proxy mode.

To Activate Proxy Mode

1. Click the right mouse button over the Node Graph and choose **Edit** > **Project Settings**.
2. Make sure the Viewer window is open.
3. Press the keystroke to toggle Proxy mode, **Ctrl**+**P**.

   A label inside the Viewer indicates that you are now in proxy mode.
4. Move the mouse pointer over the Viewer, and press the plus (+) key several times to zoom-in.
5. Press **Ctrl**+**P** a few times to toggle between hi-res and proxy mode.
6. Before you continue, press **Ctrl**+**P** to switch back to full resolution.

To Activate "Down-Res"

1. Choose **1:4** from the "down-res" dropdown menu to change the display resolution to 25% of full resolution.

   With a reduced resolution, Nuke requires less time to calculate and display your images.
2. Change the "down-res" setting back to **1:1**, which is 100% of the active resolution.

   If you turned off proxy mode, you should be back to full resolution. If proxy mode is turned on, the display resolution is 100% of the proxy.

Compositing Images

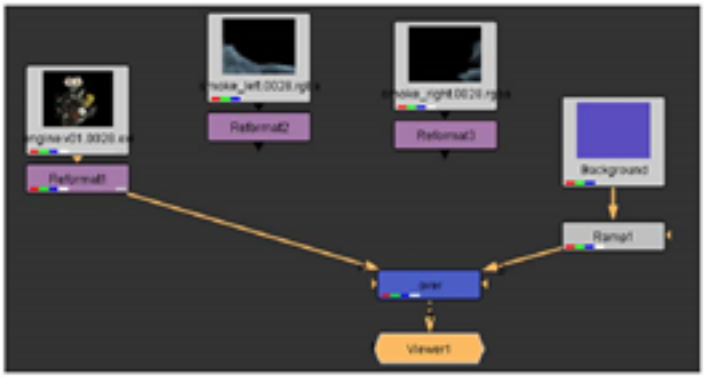
The Merge nodes create a composite with two or more images, using various compositing algorithms. In this example, we'll do a very simple "A over B" composite to layer the foreground image over the background.

You can insert a compositing node from the Toolbar or menus, but we'll show you a keyboard shortcut that bypasses both of these. The trick is the select both nodes you want to composite and then press a keyboard shortcut to assign a compositing node.

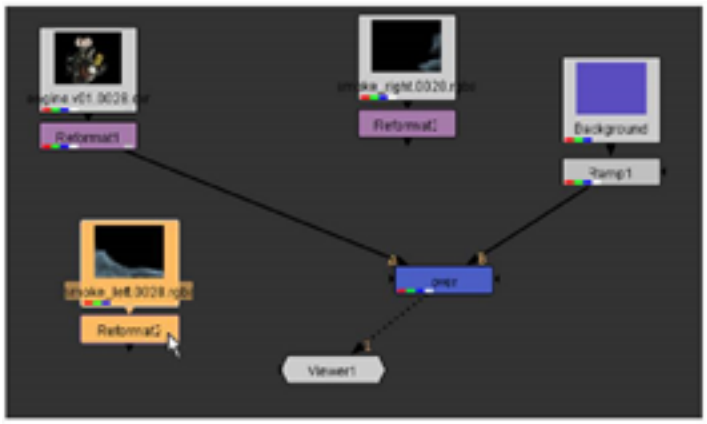
To Composite Two Nodes

1. Select the **Reformat1** node, attached to **engine.v01**. This provides the foreground image for the first compositing operation.
2. Press the **Shift** key and select the **Ramp1** node. Both "engine.v01" and "Ramp1" nodes should be selected.
3. Press the letter **M** to insert a Merge node.

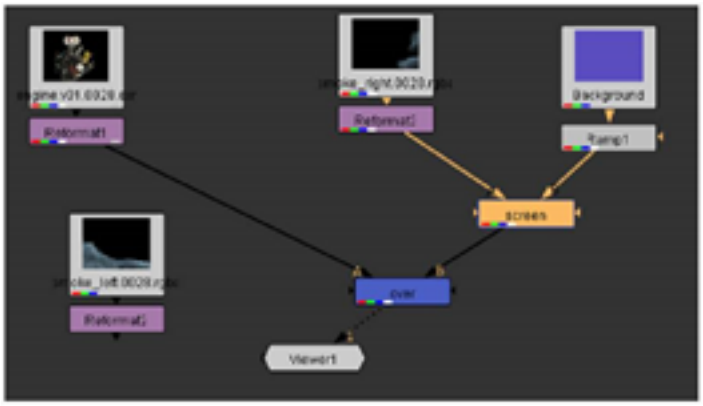
   The first node you selected is attached to the A input on the Merge node, as the foreground input. The second node you selected is attached to B, the background input. If necessary, you can swap the A and B inputs of a merge node by pressing **Shift**+**X**.



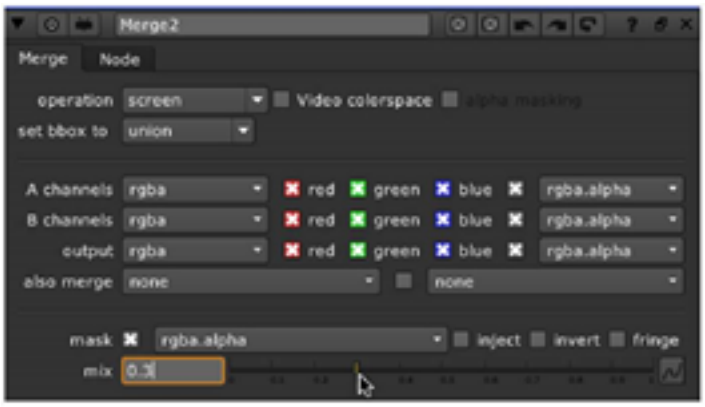
   In the Merge node control panel, the **operation** parameter determines the compositing algorithm used to generate the result of the two inputs - the selected operation becomes the name of the node in the Node Graph.
4. Rearrange the nodes, so that the node tree looks similar to this:



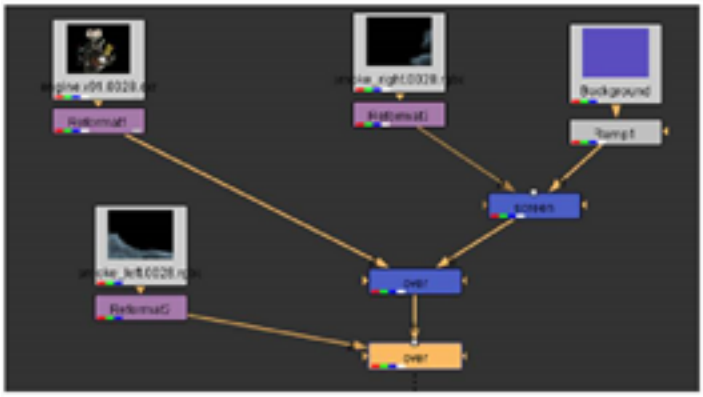
5. For the next layer, select the **Reformat3** node, attached to **smoke_right**. Then hold down the **Shift** key and select **Ramp1**.
6. Press **M** to insert a Merge node and composite one image over the other. This composites the "smoke_right" image over the background.



7. The default compositing algorithm, "Over," isn't what we need here. In the Merge2 control panel, click on the **operation** dropdown menu and select **screen**.
8. In the Merge2 properties panel, drag the **mix** slider and change its value to **0.30** to reduce the amount of the image supplied by the **A** input.



9. An additional Merge node is required. Select **Reformat2** for **smoke_left**. Hold down the **Shift** key and select the **Over** node (the first Merge node you inserted).



10. Press **M** to composite the two nodes. In the Merge3 control panel, change the **mix** slider to **0.75**.

    The result of your composite should look similar to the example below.



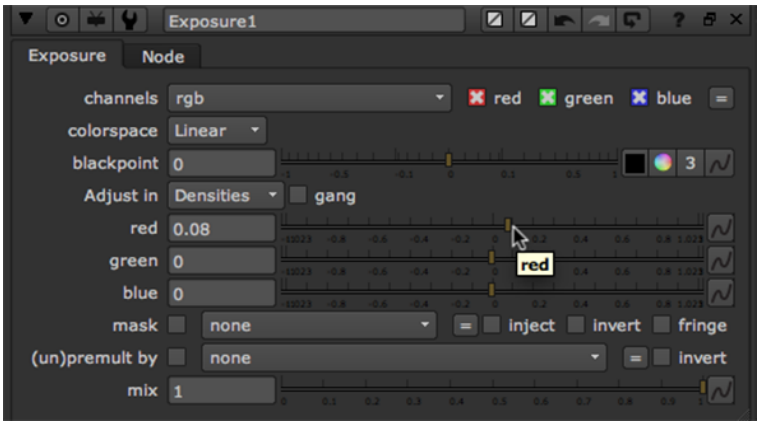
Color-Correcting Images

Color-correction and filters can help you integrate the elements for a better composite. In our example, you want to limit the correction to the foreground element only, so you'll insert a color correction node before the Merge nodes.

1. Select the **Reformat1** node. Then, right-click over the Node Graph and choose **Color > Exposure**. This inserts the Exposure1 node.



2. Suppose you want to adjust the value of the red color channel. Move the mouse pointer over the Viewer window and press **R** to display the red channel.
3. In the Exposure1 control panel, uncheck the box for **gang** sliders. This allows you to adjust individual color channels.



4. Drag the **red** slider to adjust the color values. When you are finished, press **R** over the Viewer to display all channels.

   The Exposure node worked as expected, but the result is less than spectacular. The color change is too uniform. If only there were a way to limit - or, in fact, *mask* - the color correction, perhaps we'd see a better composite. Hmm...

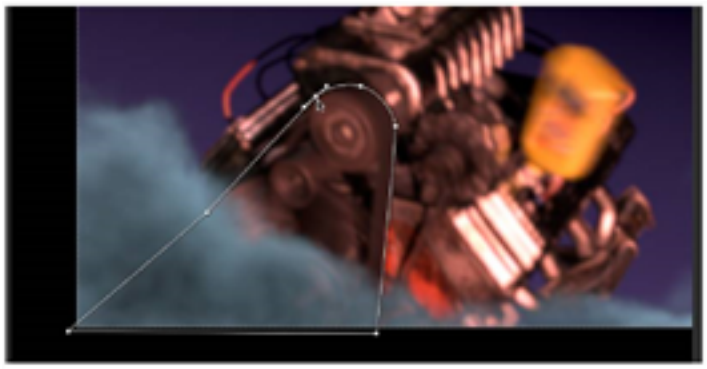
Masking Effects

You can apply masks to limit how each of these nodes affects the images. The following shows how to create a Bezier mask to limit color-correction.

To Create and Apply a Bezier Mask

To create and apply a Bezier mask:

1. Click on a blank space in the Node Graph, so that nothing is selected in the node tree.
2. From the Toolbar, choose **Draw > Roto** to insert a Roto node.
3. Click inside the Viewer window to draw a Bezier shape over the image, like this:



4. To refine the shape, click on a point to select it and then drag to adjust its position.
5. To create sharp corners, select a point, right-click and choose **Cusp**.
6. To add points to the shape, simply select the **Add Points** tool and click on the shape's outline.
7. When you're satisfied with the shape, drag the **mask** connector from the **Exposure1** node to the output of the **Roto** node.

   In the Exposure1 control panel, the mask channel option is now set to the **rgba.alpha** channel of the node that is connected to the **mask** input. In this case, this is the alpha channel of the Roto node.

Creating Flipbook Previews

On the Viewer window, the timeline buttons let you play the project, but if you pay attention to the frames-per-second (FPS) field at the top of the Viewer window, you may notice that Nuke doesn't provide real-time playback. This is because Nuke renders on-the-fly to display images in the Viewer. It's fast, but also limited by the amount of memory and computer-processing power available to you.

The Flipbook feature provides better real-time preview, because it is prerendered for the Flipbook viewer. Keep in mind that the Flipbook feature renders a preview that matches the active resolution; if you're in proxy mode, for example, that's the resolution you'll get in the flipbook.



Note: The Flipbook feature renders temporary files in the directory you specified for disk cache under **Nuke** > **Preferences**. You'll also find an option there that allows you to limit the amount of disk space the flipbook feature uses.

To Generate a Flipbook

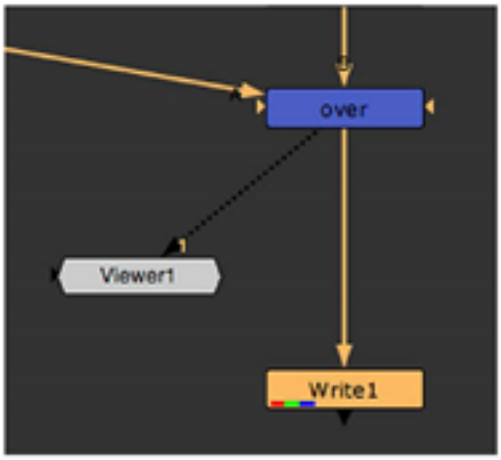
1. Select the **Over** node at the bottom of your node tree.
2. From the menu bar, choose **Render** > **Flipbook selected**.
3. Enter **1-28** as the number of frames to preview and click **OK**.
4. When the flipbook is ready to view, a Flipbook Viewer is launched. Click the **Play** button to view the results.
5. Close the Flipbook Viewer to return to your project.

Rendering Final Output

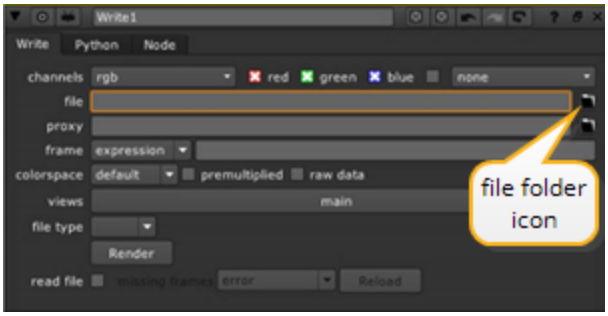
When you're ready to render the results of your composite, you insert a Write at the bottom of the node tree, and specify the path name for the rendered images. Although we'll use just one here, you can place several Write nodes in your script, anywhere you like, to render output from different places in the tree. When the render order is important, use the **render order** option in the Write nodes to specify the order in which multiple renders should be executed.

To Render the Result of Your Composite

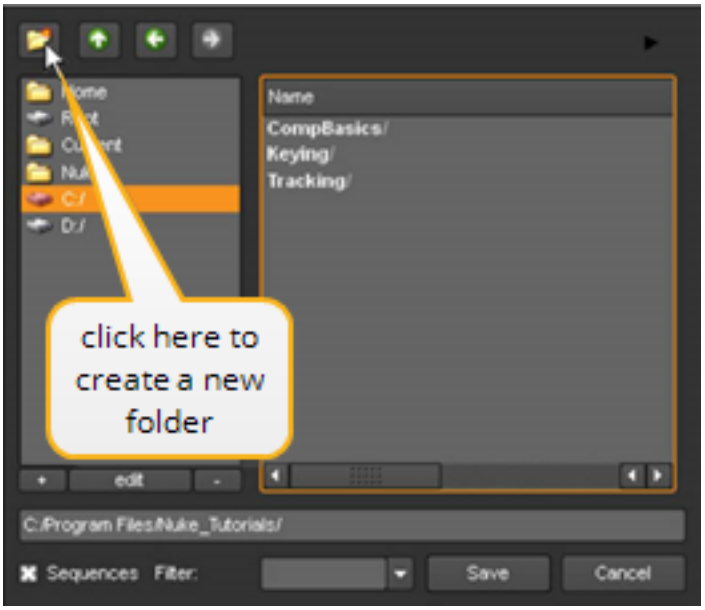
1. Select the last **Over** node at the bottom of the node tree.
2. Right-click and choose **Image** > **Write** to add a node for output.



3. In the control panel for the Write node, click the **file** folder icon.



4. Browse to the **Nuke_Tutorials** directory.



5. Click the "new folder" icon, in the upper-left corner of the browser, and type **Rendered** as the name for the new folder. Click **OK**.
6. Select the folder you just created.

   You should see the "Nuke_Tutorials/Rendered/" path name displayed at the bottom of the browser.

7. At the end of the "Nuke_Tutorials/Rendered" path name, type **first_comp.####.exr** as the name for the rendered image sequence, and then click **Save**.
8. Choose **Render** > **Render All** to render the images, or simply click the **Render** button inside the Write control panel.
9. Nuke prompts you to specify the frames to render. Enter **1-28** as the frame range and click **OK**.

A status window appears that shows the progress of your render. When the render is complete, you'll find the sequential images in the "Nuke_Tutorials/Rendered" directory. To check the results, simply insert a new Read node, point to the new image sequence, and then generate a flipbook with the Read node selected.

Using the Nuke Frame Number Variable

What's that "####" bit in the filename, you say? That's the variable that tells Nuke where to place the sequential numbers or frame numbers. You only type one name to represent the image sequence, but Nuke creates one image file for each frame in your shot.

So, in this case, you entered "first_comp.####.exr" but Nuke renders these files for frames 1 through 5: "first_comp.0001.exr," "first_comp.0002.exr," "first_comp.0003.exr," "first_comp.0004.exr," and "first_comp.0005.exr." You can change the number of hash marks in the variable - ##, ###, ##### - to change the number of padded digits for the frame numbers.

An alternative way of marking frame numbers is the Printf (%0d) notation. In this case, the same frame numbers would look like this: "first_comp.%04d.exr". Instead of the number of hash marks, with the printf notation you would change the number before d to adjust the number of padded digits, for example "%03d" or "%05d". You can choose which style you want to use by setting **sequence display mode** option on the **Appearance** tab of the **Preferences**.

Image Formats

If you don't specify a file format inside the Write node control panel, Nuke uses the format specified by the file name extension you typed. For example, in this tutorial, you used the ".exr" extension to tell Nuke to save the images as OpenEXR files.

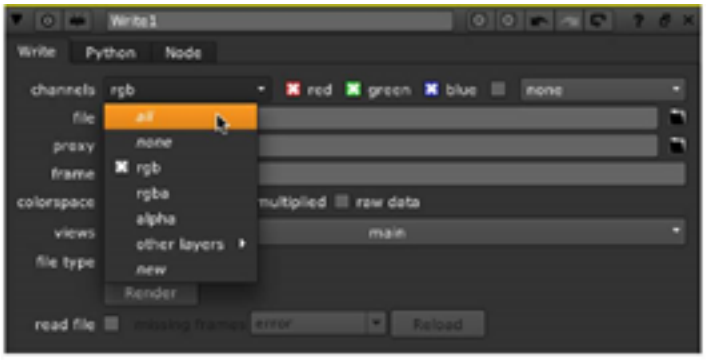
Rendering with the Active Resolution

When you execute a render or a flipbook, Nuke assumes you want to render the active resolution. When you're in full-res mode, for example, Nuke renders full-resolution images to disk. When you're in proxy mode, Nuke assumes you want to render the proxy resolution - defined in the **Project Settings** window - to the path and file name you specified as the **proxy** file name in the Write node. If the proxy field is empty or pointing to an invalid path, Nuke returns an error.

It's easy to toggle to proxy mode and then forget your images are rendered in the lower resolution. Before you execute a render, it's always a good idea to check which resolution is active. In the Viewer, the label at the lower-right corner of your image indicates whether you are in full-res or proxy.

Rendering Multiple Channels

When you insert a Write node, Nuke assumes that you need only the RGB channels in the final render. In many cases, this is acceptable because you won't need the alpha channel or other channels from the node tree when you deliver final shots to your clients. However, sometimes you need to render intermediate files - such as mattes, projection elements, or subcomps - and include all the channels in your node tree.



For example, rather than manage several elements for an animated character, you could combine the character animation, the lighting passes, alpha channel, and a depth mask in one image sequence on disk. This makes it easier to manage elements in the final composite and simplifies the artist's workflow.

To output all channels, change the Write node's **channels** dropdown menu from **rgb** to **all**, select the OpenEXR file format, and then execute the render. Currently the OpenEXR format (.exr) is the only file format that supports unlimited channels.

Epilogue

In this tutorial, you set up a new project and created a simple composite. You learned how to use (or at least, locate) practically every Nuke window and tool, and you rendered out the result of your composite. You're finished! Go home!

Well... there might be a few more things you want to know. After this tutorial, you should feel comfortable with the Nuke user interface, so put on your explorer hat and review the other tutorials. There's no specific order from here, so look through the following pages until you find what interests you.

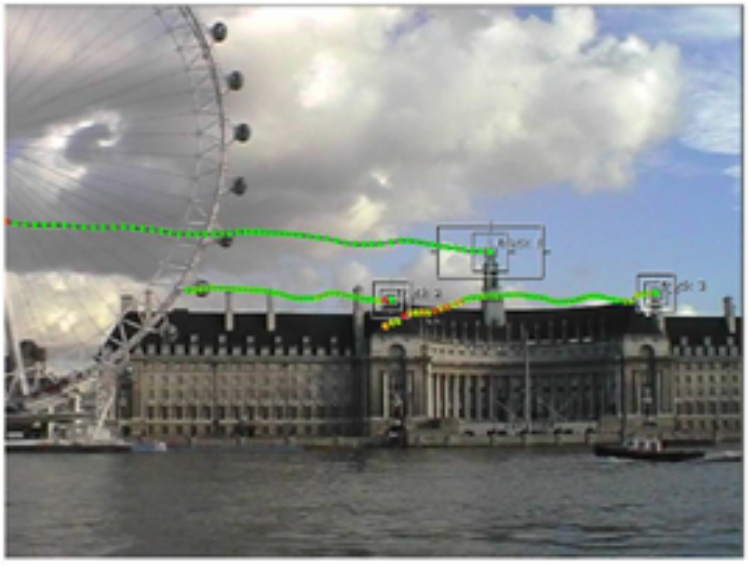
Tutorial 2: 2D Point Tracking

This tutorial teaches you how to use Nuke's Tracker node for tracking, stabilizing, and match-moving.

Introduction

Every filmmaker knows the challenges of putting together a vision. You may not have the money to build post-apocalyptic Montreal, but you might have enough to create it in post. You may have brilliant performances by your actors - but not together in the same shot. Fortunately, you can composite the best takes. Your battle sequence with 5 A-list actors, 100,000 extras and 57 elephants, comes back from the lab with scratches on the negative. You can fix it. You can. A savvy production team knows how to leverage digital technology to make it possible, and Nuke's tracking tools are indispensable for these situations.

As you may know, tracking is the process of recording the location of features as they move through the scene. The result is stored as 2D coordinates on the image plane. Once you have the tracking data, you can use the movement to perform a variety of useful tasks, such as stabilizing the footage, applying the movement to other elements in your composite, and improving the accuracy of roto mattes.



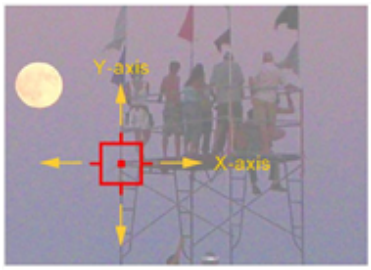
Tracking image features.

An important aspect of the tracking process involves carefully reviewing your footage *before* you attempt to track. Play through your sequences several times and look at the direction of movement for the features you want to track. Note potential problems with motion blur, obscuring objects, or frames where the features are hidden or move off screen.

Tip: Nuke can often compensate for problem footage, but tracking works best when you can identify distinct features throughout the length of the shot.

One-Point, Two-Point, Three-Point, Four

Before we get into the first example, let's review a few tracking concepts. You can track as many features or patterns as required with the Tracker node in Nuke. How do you decide whether to track one, two, or more features? It depends on what you want to do with the data and the level of accuracy you need in the result. Here are some general guidelines:



One track:
X and Y position only.



Two tracks:
X, Y, and Z-rotation.



Three tracks:
X, Y, Z-rotation, & scale.

- **One-point tracking** - Track one feature's horizontal (x-axis) and vertical (y-axis) position, with little or no perspective change on the image. You can apply this information to move other elements in the composite or apply the inverse to stabilize the image.
- **Two-point tracking** - Track horizontal and vertical position for two features. The feature positions, relative to each other, indicate whether the image is rotating clockwise or counter-clockwise (z-axis rotation). In some cases, two tracking points are sufficient to calculate the scaling of the features as well.
- **Three-point tracking** - Track horizontal and vertical position for three features. Provides all the benefits of two-point tracking with an additional set of tracking data for more accuracy on z-rotation and scaling.
- **Multi-point tracking** - Again, all the benefits of fewer tracks with additional sets of tracking data. Three-point is usually sufficient for most 2D tracking needs, but multi-point makes it possible to distort and match-move another element into the points of the features you track, for example, using four tracks and a CornerPin2D node.

Open the Tutorial Project File

In this tutorial, you work from a project file that already includes the node trees. Each tree is setup for the examples that follow.

To Open the Project File

1. Launch the Nuke application and choose **File > Open Comp** from the menu bar.
2. In the file browser, navigate to your **Nuke_Tutorials/Tracking/** folder, select the **tracking_tutor.nk** project file and click **Open**.
3. It should show some nodes in error. Don't worry! It can't find the tutorial files. So before doing anything else you have to tell this script where to find these tutorial images. Double-click on the NoOp node in the top left corner of your Node Graph. It's called Tutorial_Path. Double clicking brings up a Properties panel on the right. Enter the path to the tutorial files in the Tutorial Project Directory. Use the file browser as that's often easier than typing it in. You should then see tutorial images appear.
4. Move the mouse pointer over the Node Graph, and press **F** to frame the entire contents of the project file.

The examples in this project file are grouped with colored boxes, called *backdrops*, and each contains a node tree for the tutorial examples that follow.

Tip: Backdrops let you organize groups of nodes, like those shown in this project file. Select **Other** > **Backdrop** from the Toolbar and drag the backdrop title bar to move it. Drag the backdrop corner to resize it. Any nodes surrounded by the borders of the backdrop move with the backdrop when you drag its title bar.

Tracking a Single Feature

In this first example you'll learn how to set up a tracking anchor and then track a single feature, which is the most basic 2D tracking operation. After you achieve a solid track for one feature, you can build on that and track other features as needed.

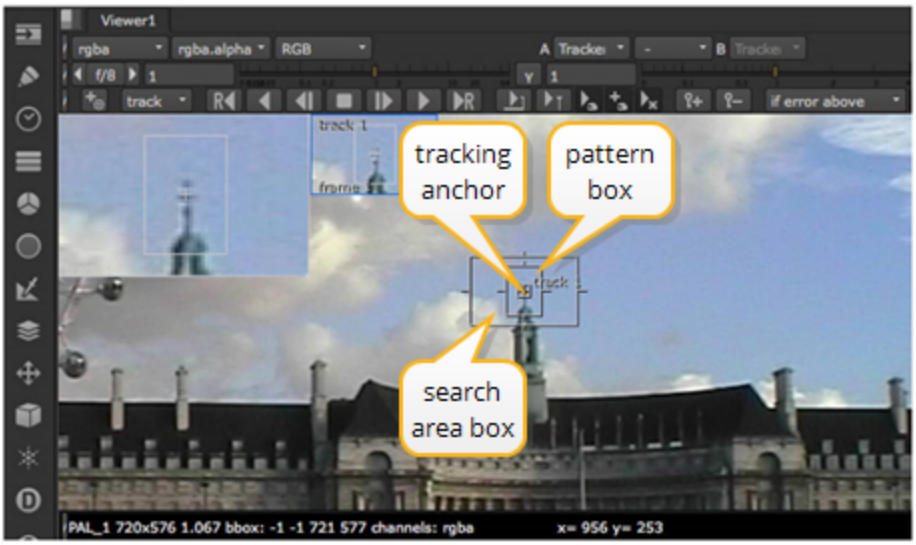
Setting a Tracking Anchor

1. In the project workspace for the **tracking_tutor.nk** file, locate the node tree labeled **Tracking an Image**.
2. Click on the **LondonEye** Read node to select it.
3. Play through the sequence several times, using RAM cache or a Flipbook of your choosing, to review the footage.
4. Look at the features in the image and notice the amount and direction of movement as the clip plays.
5. Choose **Transform** > **Tracker** from the Toolbar to attach a new Tracker node to the **LondonEye** Read node.



6. Connect the Viewer node to the Tracker node, as shown.
7. In the Viewer, scrub the time slider to frame 1, to make sure you're at the beginning of the shot.
8. Double-click the Tracker node to display its Properties panel.
9. Click **add track** to create a tracking anchor in the Viewer.

10. Drag the tracking anchor over the tower spire, as shown. Use the zoom window in the Viewer to help you with positioning.



11. Click on the pattern box (inner box) of the tracking anchor, and adjust its size to contain the feature.
12. Click the search area (outer box) of the tracking anchor, and adjust its size to enclose the amount of space you think the feature may move between frames.

    Large search areas require more calculation time, so keep it as small as possible. However, when the search area is *too* small, the feature may move outside the box and you'll lose the track. If you aren't sure how large to make the search area, go back and review the flipbook of your image.

Auto-Tracking vs. Keyframe Tracking

After placing a tracking anchor, you're ready to calculate your track. Nuke's Tracker provides two calculation methods:

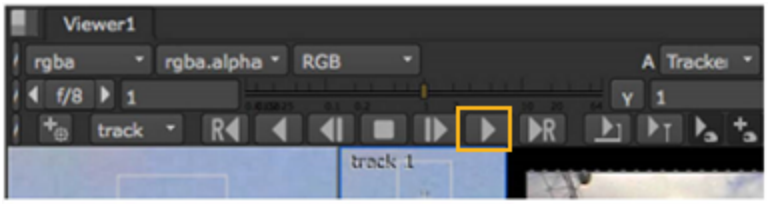
- **Automatic tracking** - ideal for simple tracks, there are no extra preparation steps once you've set your tracking anchors.
- **Keyframe tracking** - a more involved method, requiring you to set keyframes on the sequence in order to calculate tracks. Keyframe tracking may be the better option for more complicated patterns and movement.

Using Auto-Tracking

1. Enable **show error on track paths** by clicking the traffic light icon in the Viewer tools.

   This color codes tracks showing their pattern matching error values, green for a good match through to red for a poor match.
2. In the Tracker Properties panel, select the track you wish to calculate in the **Tracks** list.

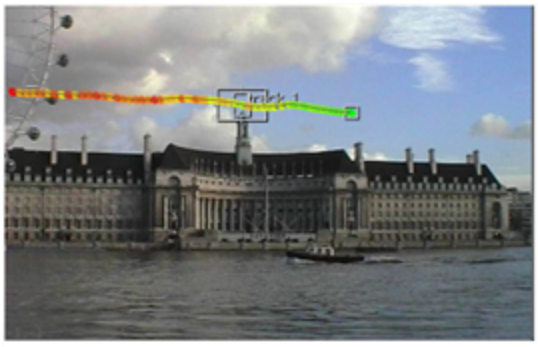
3. Select the type of movement the track is intended to output: translation, rotation, or scaling. In this simple example, you only need to select **T**ranslation.
4. At the top of the Viewer, click the **track forward** button to generate the track.



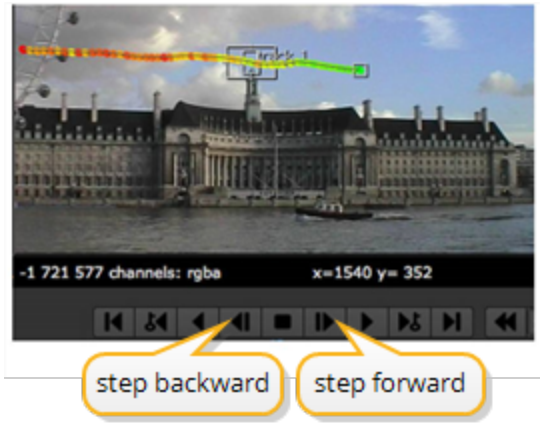
If you only need a certain frame range, use the  button and enter the required range.

When Tracker has finished, you'll see the track curve with color-coded points along the curve.

Note: If tracking fails, try resizing the pattern or search boxes, as described in Setting a Tracking Anchor and retracking.



5. Use the **next frame** and **previous frame** buttons on the timeline to step through the track to verify its accuracy.

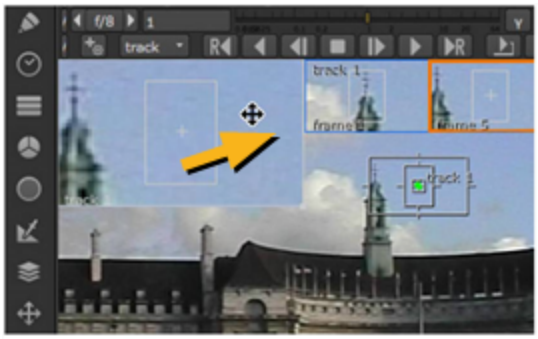


The track is fairly solid in this example. However, some images don't track as easily as this one. What are your options? You could retrack Using Keyframe Tracking as described below, or you could edit your existing track in the Curve Editor, see Editing Track Data.

Note: Bear in mind that a red keyframe doesn't necessarily mean that the tracking result is poor, only that Tracker couldn't reliably match the pattern from one keyframe to the next.

Using Keyframe Tracking

1. In the Tracker Properties panel, select the track you wish to calculate in the **Tracks** list.
2. Select the type of movement the track is intended to output: translation, rotation, or scaling. In this simple example, you only need to select **T**ranslation.
3. Scrub through the sequence a few frames and adjust the position of the tracking anchor by dragging the anchor to the location of the pattern. You can use the zoom window to fine-tune your positioning. Continue on through the sequence as required.



At each frame, a new keyframe window is added to the right of the zoom window. The keyframe closest to the current playhead frame is highlighted in orange.

It's a good idea to place more keyframes around areas of complexity or greater movement and fewer on straight forward translation. Generally speaking, a greater number of keyframes produces a better track, but at the expense of processing time.

4. When you're satisfied with your keyframes, make sure your track is selected in the **Tracks** list and then click  to track all keyframes.

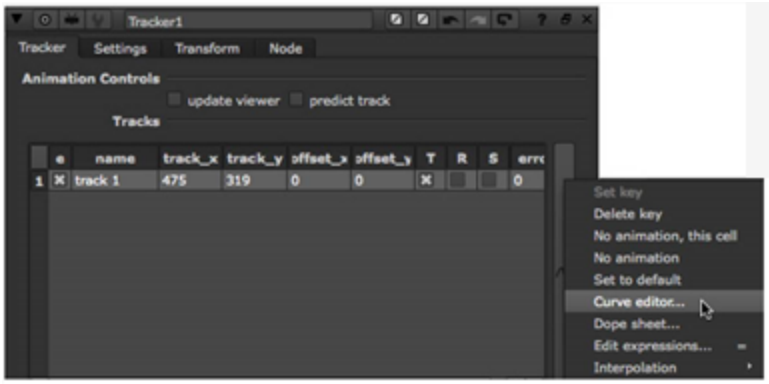
   You can also force the selected tracks to recalculate between the two nearest keyframes by clicking  in the Viewer toolbar.
5. Use the **next frame** and **previous frame** buttons on the timeline to step through the track to verify its accuracy.

Note: Bear in mind that a red keyframe doesn't necessarily mean that the tracking result is poor, only that Tracker couldn't reliably match the pattern from one keyframe to the next.

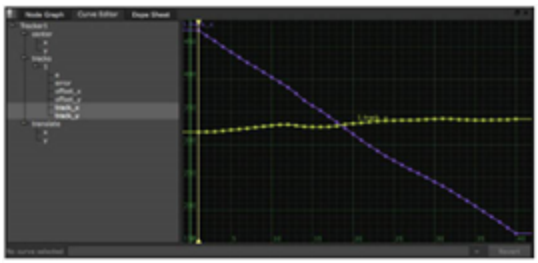
6. Proceed to Editing Track Data if the track is still not satisfactory.

Editing Track Data

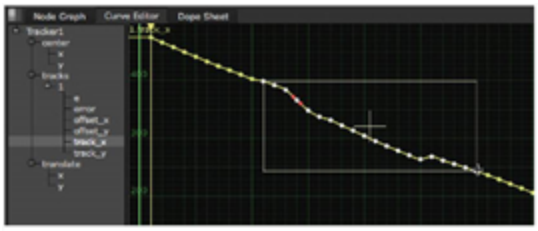
1. In the Tracker Properties panel, select the tracks you want to view.
2. Click the animation button next to the **Tracks** list, and choose **Curve editor**.



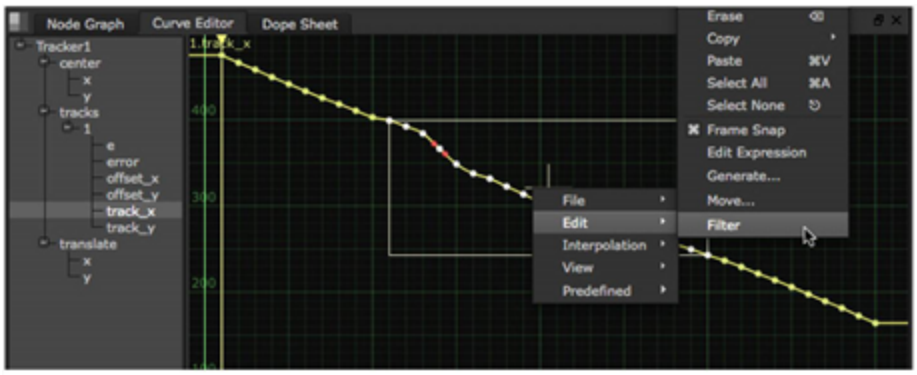
3. Click the **track_x** and **track_y** items in the Curve Editor tree and you'll see values recorded for each of the parameters during the tracking process.



4. Select both curves by holding down the **Shift** key and clicking the **track_x** and **track_y** curves, under **tracks.1**.
5. Press the **F** key to "frame" the curves in the Curve Editor.
   - To adjust a value, select a point and drag it up or down.
   - To change the frame for a particular point, select it, hold down the **Ctrl** key and drag the point left or right.
6. Let's assume you want to smooth a curve by applying a filter to the values. Draw a marquee - drag while pressing the left mouse button - around a section of the curve to select multiple points.



7. Click the right mouse button to display a menu of Curve Editor options. Choose **Edit** > **Filter** and enter **2** as the number of times to filter the key frames.



Nuke averages the location of each point based on the values of the surrounding points, smoothing the curve.

8. Close the Curve Editor window and then play the result in the Viewer.

Those are the basics for tracking and editing the results for a single feature. In the next example, we'll make it a little harder - tracking a feature that moves out of view.

Tracking Obscured Features

At the end of the previous example, you may have noticed the track was dropped at frame 58 when the feature moved off the screen. When features move out of frame, or become obscured by other elements in the image, you can use the track offset feature to pass the tracking operation to another feature in the image. Nuke then attempts to continue the track along its current course.

To Track a Feature That Moves off Screen

Note: This example uses auto-tracking, but the offsetting principle is the same for keyframe tracking.

1. In the project workspace, locate the node tree **Tracking Obscured Features**.
2. Double-click the Tracker2 node to open its control panel. This node tracks one of the chimneys in the clip you used from the previous example.
3. Attach a Viewer to the Tracker2 node and scrub the timeline until you see the tracked feature move out of frame.



As you can see, track1 accurately tracks its feature through most of the clip - until it moves off the screen at frame 44. This is where the problem starts.

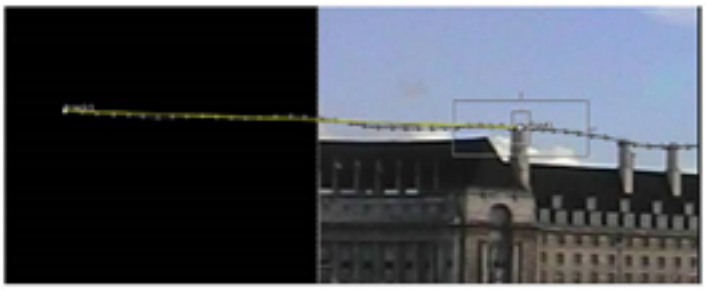
4. Press the plus key (**+**) on your keyboard a few times to zoom in on the Viewer.

   Examine the sequence to find an alternate feature that stays in view during the length of the clip.
5. At frame 44, press the **Ctrl/Cmd** key and drag the **track1** anchor to the first chimney on the right of the building.



A line connects the new feature to the original feature indicating the offset, and the **Tracks** list is updated to show the x and y offset values.

6. In the **Tracker** control panel, press the  button to continue the off screen track using the offset feature.



How is this useful? Well, now you can use the track data to matchmove an element - a trail of chimney smoke, for example - that locks to the feature even after it moves off the screen.

7. The track is now complete, so you can clear the offset by clicking  in the Viewer tools.
8. Deselect **track 1** in the **Tracks** list to prevent it from being recalculated.
9. Before you continue, close all Tracker control panels that are currently open.

The offset doesn't change the track location. Instead, it allows Nuke to continue the track with the assumption that the offset feature remains at the same relative distance to the original feature. Later in this chapter, you'll see how to use this tracking data to composite another element to match the background plate.

Stabilizing Elements

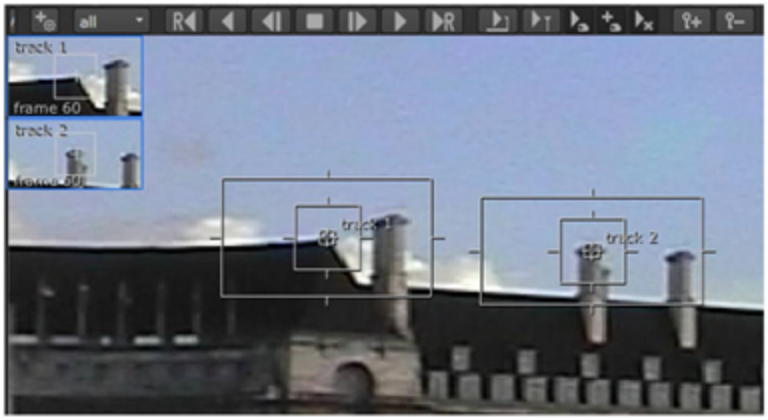
Stabilization is the process of removing motion - camera-shake, for example - and locking down the element for your composite. A one-point track provides enough information to stabilize horizontal and vertical motion along the image plane. A two-point track lets you stabilize horizontal and vertical motion, and remove rotation in the image, as well.

To Track and Stabilize

Note: This example uses auto-tracking, but the stabilizing principle is the same for keyframe tracking.

1. Locate the node tree labeled **Stabilizing Elements**.
2. You'll see a copy of the same **LondonEye** Read node that we've been using for the other examples. Click on it to select it.
3. Choose **Transform** > **Tracker** and then attach a Viewer to the new Tracker3 node. Double-click the node to open up the Properties panel.

4. In the Properties panel, click **add track** twice to create two tracking anchors.
5. For each track in the **Tracks** list, check the boxes for **T** (translate), **R** (rotate) and **S** (scale).
6. In the Viewer, scrub to the end of the sequence and adjust the size and position of each tracking marker for the features shown in the image below.



7. Select both tracks in the **Tracks** list.
8. At the top of the Viewer, click the **track backward** button to generate the tracks.



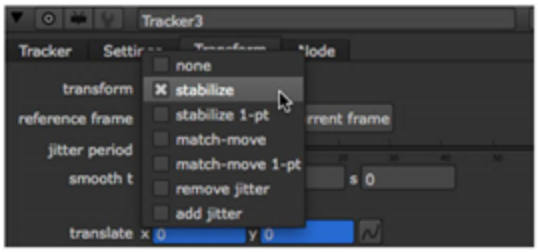
Now you have positional data for two tracks, and you can use this information to remove the unwanted movement in the image.



9. In the Tracker3 Properties panel, click the **Settings** tab and open up the Auto-Tracking sub-menu.
10. Set the **warp type** to **Translate/Rotate/Scale** so that Tracker expects all three transforms.

Note: If you're using keyframe tracks, there's no need to set the **warp type**.

11. Click the **Transform** tab and choose **stabilize** from the **transform** list.



Note: When you're tracking simple translation using a single track, you can use **stabilize 1-pt** for faster calculation.

12. In the Viewer, click play to see the results.



As the clip plays, you'll see the features remain locked to the same position within the compositing frame.

Tip: After you track and stabilize footage, you can add a **Transform** > **Transform** node after the Tracker3 node to adjust the position and the rotation of the stabilized image for a final composite.

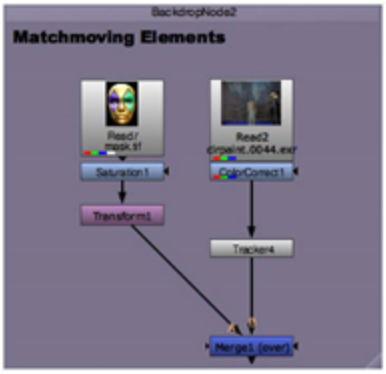
Match-Moving Elements

Match-moving is the opposite of stabilization. The intent is to record and use the motion in an image and apply it to another element. In the following example, you'll use the tracker to match-move and composite a mask image onto the performer in a background plate.

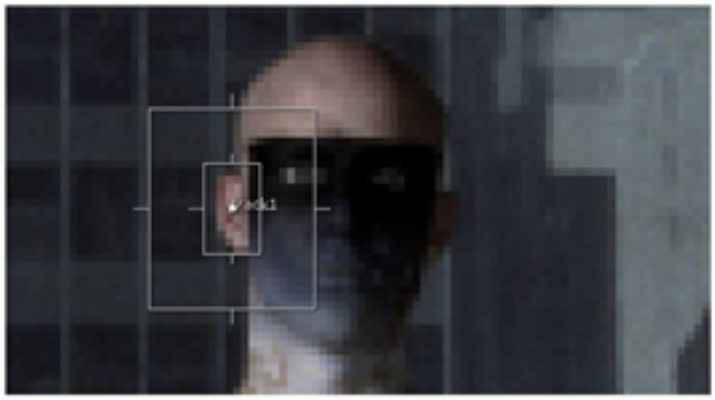
To Match-Move an Element

Note: This example uses auto-tracking, but the match-move principle is the same for keyframe tracking.

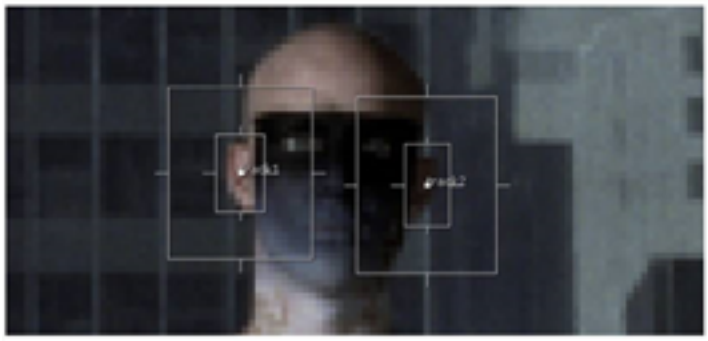
1. Find the node tree labeled **Matchmoving Elements**.
2. Drag the time slider to the beginning of the timeline. Select the **ColorCorrect1** node and then choose **Transform > Tracker**.



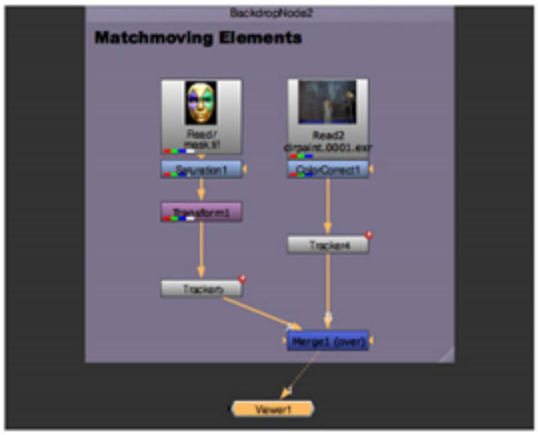
3. Attach a Viewer to the new **Tracker4** node, create a tracking anchor, and position the **track 1** anchor over the performer's right ear.



4. Adjust the size of the pattern box and the search area as shown.
5. In the Properties panel, check the boxes for **T** (translate), **R** (rotate) and **S** (scale) for **track 1**.
6. Adjust the size and position of the second tracking anchor, as shown below.
7. Create another tracking anchor, **track 2**, and check the boxes for **T** (translate), **R** (rotate) and **S** (scale) on this track.



8. Select both tracks in the **Tracks** list and click track forward at the top of the Viewer to generate the tracks. Edit the tracks as described in Tracking a Single Feature if necessary.
9. Once you have two solid tracks on the performer, make a copy of the Tracker4 node by selecting it and pressing **Ctrl+C** to copy it.
10. Select the **Transform1** node and press **Ctrl+V** to paste the Tracker node copy (Tracker5).
11. Connect the Viewer to the **Over** node. Your node tree should now look similar to this:



12. In the Tracker5 Properties panel, click the **Transform** tab and choose **match-move**. Then close the Tracker5 Properties panel.
13. Click play in the Viewer or render a flipbook and you should see the Mardi Gras mask transform to match the movement of the performer.

If you see jitter in the movement, you can edit the track data in the Curve Editor to smooth out the data. You can also add values to the smooth T, R, and S controls on the **Transform** tab to filter the tracks.

Epilogue

In this tutorial, you worked with several examples for the Tracker node. You learned how to record the locations for multiple features and you applied the tracking data for other tasks in the composite, such as

stabilization and match-moving.

Tutorial 3: Keying and Mattes

This tutorial introduces you to keying in Nuke. You will learn how to use the Primatte and IBK nodes.

Introduction

Keying is one of those fundamental compositing skills. You can't composite anything until you have mattes pulled for the elements you want to layer together. It's nice to say you could just push a button to complete this task, but as you probably know, one keying operation seldom produces an acceptable matte. Image quality, lighting conditions, subject motion, colors - even camera moves - affect the steps required to get a clean matte for your composite.



Keying Footage in Nuke.

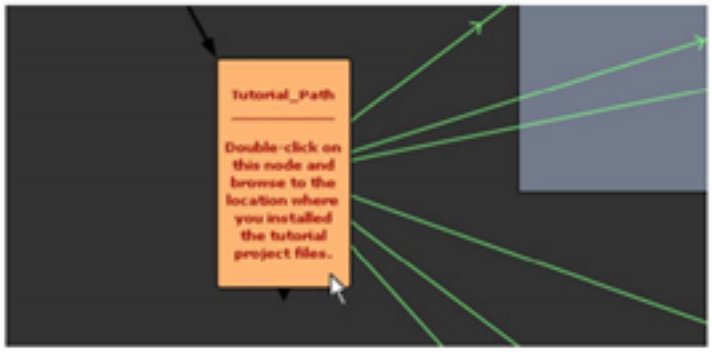
So how do you get a clean matte in Nuke? The best approach is to understand the strengths of each keying tool and combine them as needed. This tutorial shows how to pull keys in Nuke and how to layer the results with channel operations, merge nodes, and roto shapes.

Open the Tutorial Project File

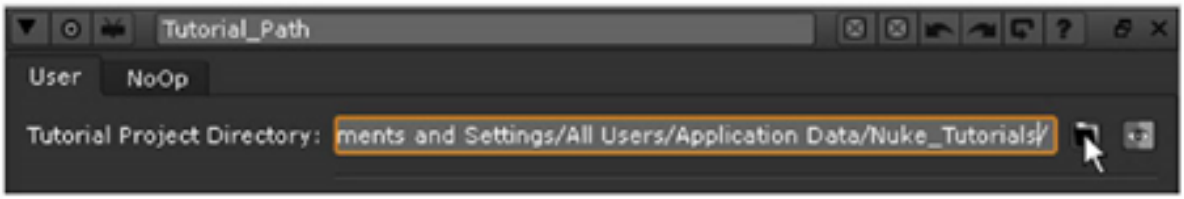
The project file for this tutorial includes several node trees for the keying operations described in this chapter.

To Open the Project File

1. Launch the Nuke application and choose **File > Open Comp** from the menu bar.
2. In the file browser, navigate to your **Nuke_Tutorials/Keying/** folder, select the **keying_tutor.nk** project file and click **Open**.
3. Double-click on the **Tutorial_Path** node, located on the left side of the script, to open its control panel.



4. In the **Tutorial_Path** control panel, click the "file folder" button. Browse to the location where you installed the tutorial project files, and then click **Open** to select the location.



   After you select the correct path, the error messages should clear from the Read nodes, and the thumbnails in the script update with the correct images.
5. Close the **Tutorial_Path** control panel. Then, choose **File > Save Comp As** to save a copy of the project file.
6. Move the mouse pointer over the Node Graph, and press **F** to frame the entire contents of the project file.

   The green arrows (lines) show the links between the Tutorial_Path node and the Read nodes.
7. If you wish, press **Alt+E** to hide the expression arrows.

   The Tutorial_Path node saves the location of the project files on your computer, so you don't need to repeat this for future sessions.

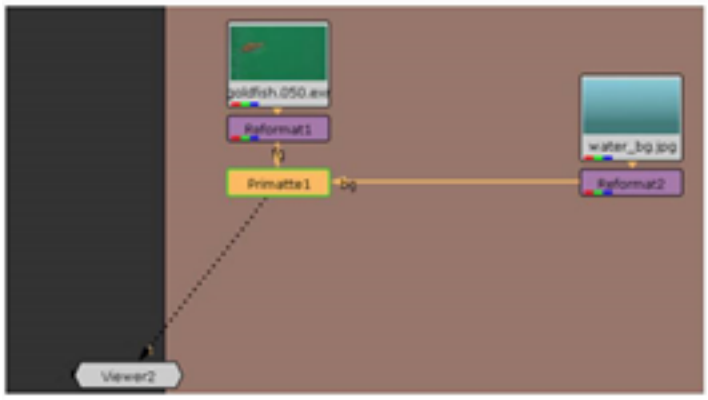
Keying with Primatte

The Primatte keyer includes a quick "Auto-Compute" option that evaluates your image and determines a good baseline key. From there, you can easily tweak the settings and generate an acceptable matte.

The two examples in this section show how to pull a key with the Auto-Compute option (method 1), and also how to manually sample a color from the screen background and build your key from there (method 2).

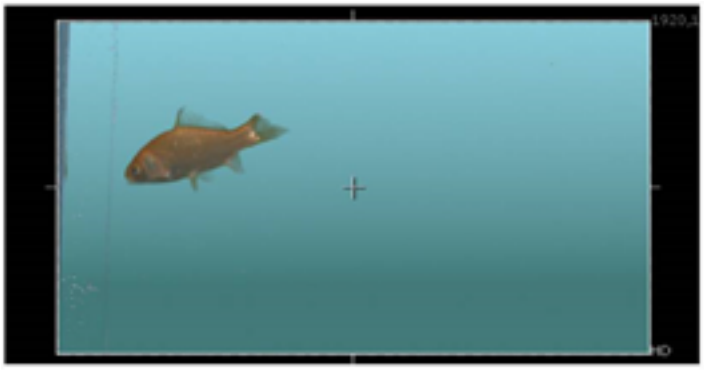
To Pull a Key with Primatte (Method 1)

1. In the project file, locate the node tree labeled "**Keying with Primatte**," and make sure a Viewer is attached to the **Reformat1** node.
2. Choose **Keyer** > **Primatte** to insert the keyer between the foreground image and the Viewer.



3. Drag the **bg** connector from **Primatte1** to the **Reformat2** node, which supplies the background image for this example. The **fg** connector should be attached to **Reformat1**.
4. Move the time slider to frame **50**, and click the **Auto-Compute** button inside the **Primatte1** control panel.





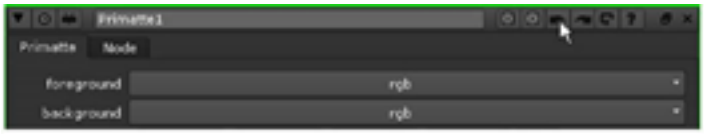
That's it. You're done... well, nearly done. We need a "free-floating" goldfish, but the reflections in the aquarium glass clearly indicate "captivity."

A garbage matte easily removes the reflections, and you'll learn how to do that later in the section on rotoscoping. For now, let's keep working with Primatte.

As you've seen, Primatte's auto-compute option can quickly pull keys on certain images. However, you should also know how to pull and tweak keys manually. You might, for example, need more control over the transparency of the fins on the goldfish.

To Pull a Key with Primatte (Method 2)

1. Continuing from the previous example, open the **Primatte1** control panel.
2. Click the "undo" button at the top of the control panel to step back to the previous state of the **Primatte1** node. Or, you can also delete the current **Primatte1** node and insert a new one.



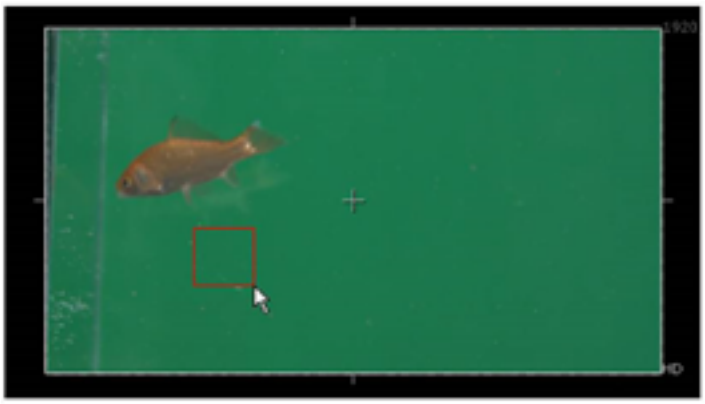
3. Scroll down through the Primatte options and set the keying **operation** to **Select BG Color**.



4. The **current color** chip should display the eyedropper icon. If it doesn't, click on the color chip to toggle the eyedropper.

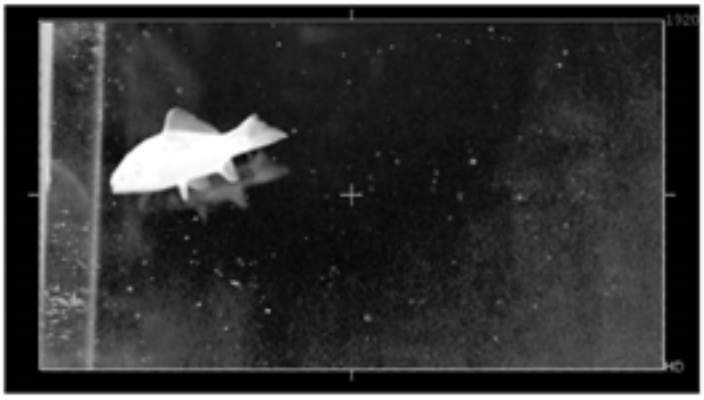


5. Hold down the **Ctrl**+**Shift** keys (Mac users, hold down **Command**+**Shift**) and drag - or scrub - over a portion of the greenscreen in the image displayed in the Viewer.



This returns an average color-pick of the sampled pixels. If you want a color pick from a single pixel, press **Ctrl** or **Command** and click once over the greenscreen. After you pick, you can clear the red square by **Ctrl**- or **Command**-clicking again.

6. Press **A** over the Viewer to toggle to the alpha channel display. Looks like the aquarium is not as clean as we thought. Our color pick gave us a fairly noisy key, so let's clean it up.

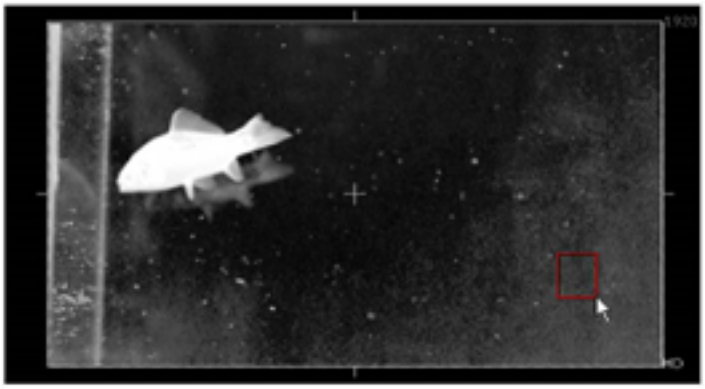


Now you'll sample a few areas of the image to "push" selected pixels to one of three areas: the transparent matte, the opaque subject, or the semi-transparent part of the matte.

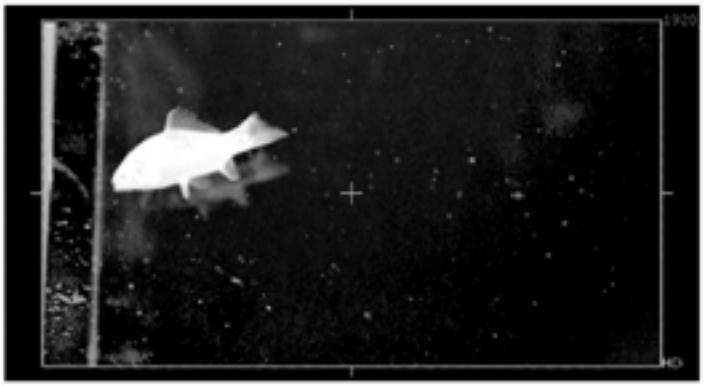
7. In the **Primatte1** control panel, change the keying **operation** to **Clean BG Noise**.



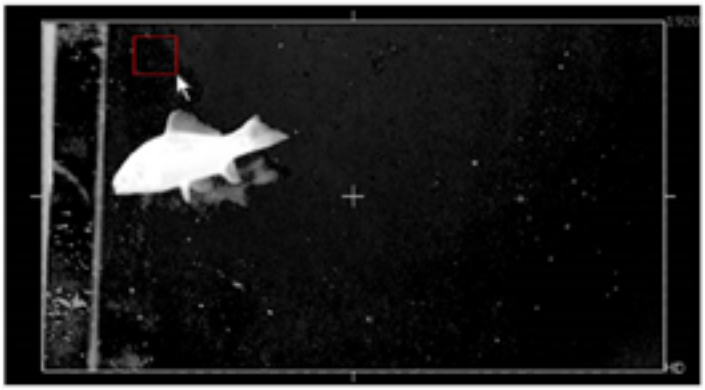
8. Press **Ctrl**+**Shift** or **Command**+**Shift** and drag a small square over the dark area in the lower-right corner of the image.



This second color sample cleans the background by “pushing” the selected pixels into the transparent area of the matte. You probably need a few more samples to get a better key.



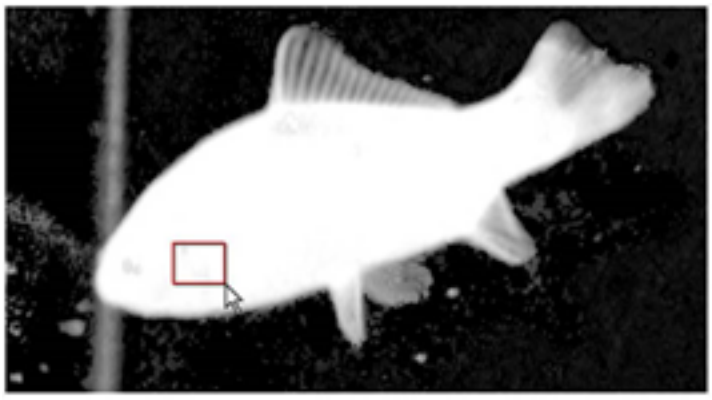
9. Scrub a few small areas in the background, focusing on the gray pixels until the matte improves.



The background doesn’t need to be solid black. We’re just trying to get a good separation between our foreground subject and the greenscreen background.

10. Change the keying **operation** to **Clean FG Noise**. This time, sample areas of gray pixels inside the goldfish.

    One or two small samples should be enough. The color pick pushes the selected pixels to the opaque part of the matte.



You want to keep the gray pixels inside the fins to retain a semi-transparent matte in these areas. If you go too far, you can always press the undo button in the control panel to step back to the previous action.

11. Press **A** again over the Viewer to toggle to all color channels. Your image should look similar to the example shown below. You may see some detail dropping out from the fins.



12. Change the keying **operation** to **Restore Detail**, and scrub over the fins to bring back some of the edge detail.



You may get different results than those shown here, depending on the pixel values you sample from the image.

Use Restore Detail to push the selected pixels back toward the opaque part of the matte. Use the **Make FG Transparent** operation to fine-tune the semi-transparent area.

You could go back and forth, between cleaning the background and foreground, but this usually produces a matte with "crunchy" edges. The goal is to find the balance between foreground and background that produces an acceptable matte for your subject.

Later in this chapter, you'll use the rotoscoping tools to clean-up this matte and combine this with the image from the next example.

Image-Based Keying

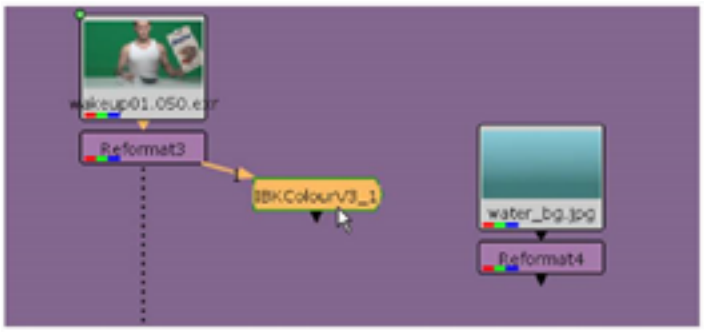
Many keying tools, like Primatte, use a color-pick as the baseline for the matte extraction process and then require the artist to tweak the matte from that baseline. Nuke's image-based keyer (IBK) uses the pixel values of the compositing images, instead of a color-pick, to generate the best matte for the image you want to extract. It works by generating a processed screen image that preserves the color variations of the blue- or greenscreen and using this - rather than a single color - to pull the key. This generally gives good results and speeds up the keying process when working with uneven blue- or greenscreens.



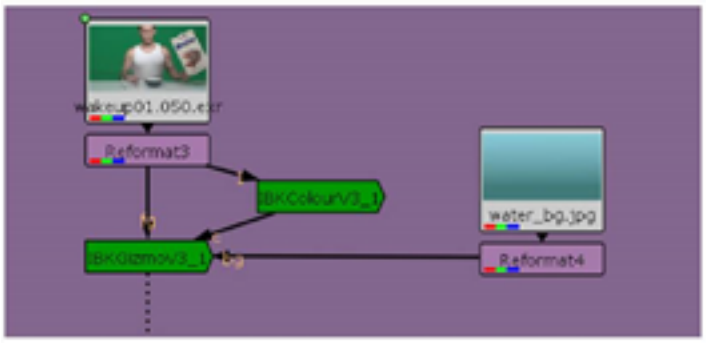
Image-based keying requires two nodes in Nuke. First, you insert an IBKColour node to process the screen image, which is preset to work with either greenscreen or bluescreen. This node generates the processed screen image that preserves the color variations in your blue- or greenscreen. Then, you insert an IBKGizmo node to generate the matte using the processed screen image, the original image, and also the background image for the composite.

To Pull a Key with IBK

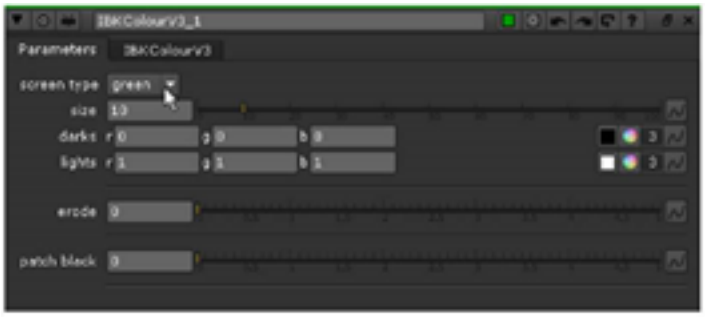
1. In the **keying_tutor.nk** project file, locate the node tree labeled, "**Image-based Keying**".
2. Right-click over the **Reformat3** node and choose **Keyer** > **IBKColour**. Drag the **IBKColourV3_1** node to the right.



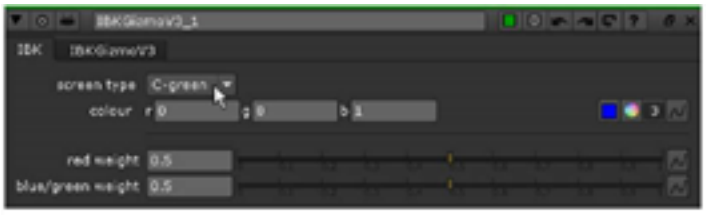
3. Click an empty spot in the Node Graph to deselect all nodes. Then, right-click and choose **Keyer** > **IBKGizmo**.
4. From **IBKGizmoV3_01** node, connect **fg** (foreground) to the **Reformat3** node. Connect **c** (color screen) to the **IBKColourV3_1** node.
5. Connect **bg** from **IBKGizmoV3_1** to the **Reformat4** node, which supplies the background for the comp.
6. Connect the Viewer to the **IBKGizmoV3_1** node, and your node tree should look similar to this:



7. Open the control panel for **IBKColourV3_1** and change the **screen type** to **green**.



8. Open the control panel for **IBKGizmoV3_1**, and change its **screen type** to **C-green**.



You should see an acceptable matte, shown in the screen capture below, on frame 50.

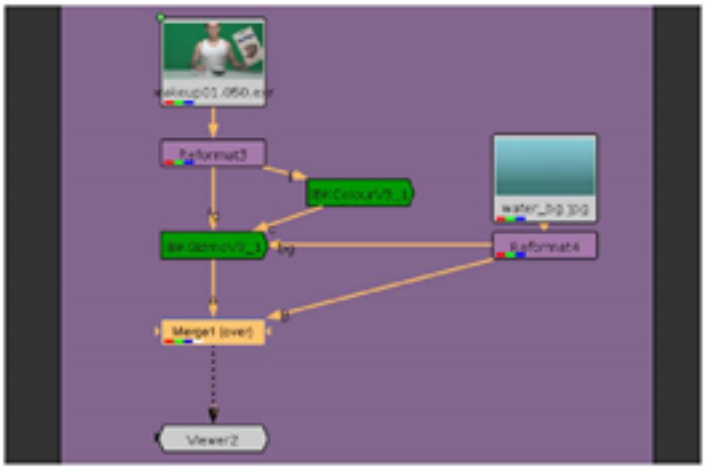


This is a very good start for this image.

9. Connect the Viewer to the **IBKColourV3_1** node. You'll see the processed screen image, which is essentially a Gaussian-filtered high-contrast key.



10. Choose **Merge** > **Merge** (or press **M** over the Node Graph) to insert a **Merge (over)** node.



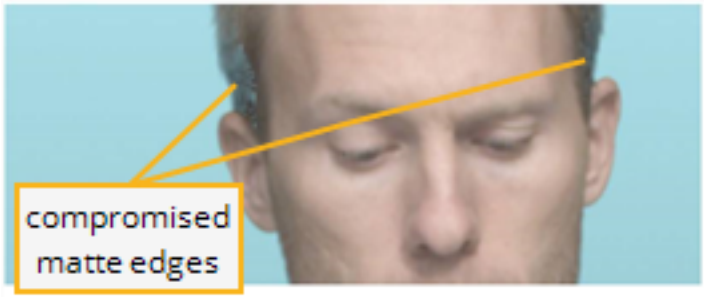
11. Connect **IBKGizmoV3_1** to the **A** input of the **Merge (over)** node. Then connect the **B** input to the **Reformat4** node.



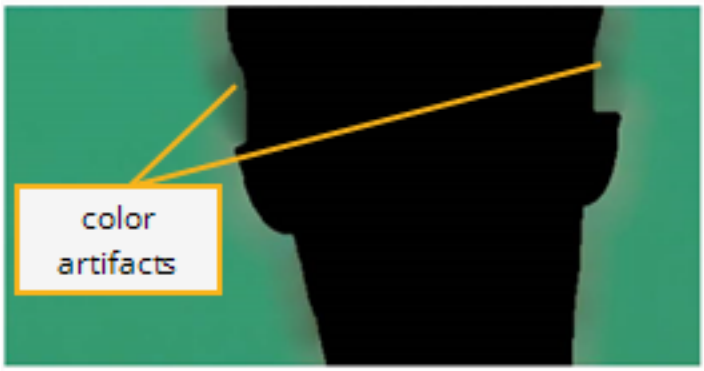
The color of this greenscreen is completely out of the region of the acceptable industry standard, but IBK does a good job anyway, by smoothing the screen and using the result to essentially create a difference matte with the foreground.

Tip: IBK has presets for green and blue screens, but you can also do a color-pick for any screen color inside the IBKGizmo node.

If you zoom-in on the image, you'll see small areas near the subject's hair, where the matte is compromised.



12. Connect the Viewer to **IBKColourV3_1** and you'll see color artifacts around the edges of the matte.



When you look at the smoothed screen produced by IBKColour, you should see only values of your screen color and black.

13. In the **IBKColourV3_1** control panel, lower the **darks**, **g** (green) setting to **-0.08**. (If you were keying a bluescreen image, you would lower the "b" value for "darks.").



This fixes most of the problems at the hairline, but destroys the good key for the lower portion of the image.

Tip: The artifacts in the IBKColour image appear as specific color shades: light green, dark green, light red, dark red, light blue, and dark blue. To remove these, simply adjust the appropriate controls for the artifacts you want to remove: lights/g, darks/g, lights/r, darks/r, lights/b, and darks/b.

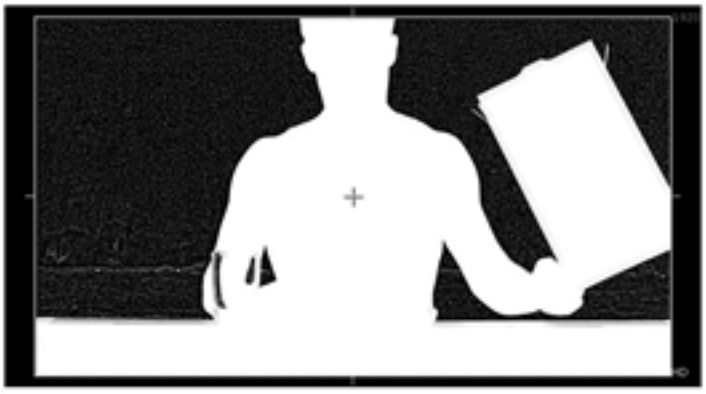


14. In **IBKColourV3_1**, raise the **darks**, **g** value to **-0.03**. Then change the **lights**, **g** value to **0.75**. This corrects the artifacts of the screen image.
15. Now, change the **patch black** setting to **1.5** to restore the edge detail of the hairline.





16. Connect the Viewer to the **IBKGizmoV3_1** node. Press **A** and you'll see the current alpha channel produced by the IBK system.



The displayed alpha image shown is correct for the IBK. If the intensity of the noise in your alpha channel is greater than the example show above, you may need to adjust - in very small increments - the **dark** and **light** values for the color channels in the IBKColour node.

17. Press **A** again over the Viewer to toggle back to display all color channels, and scrub through the timeline to check the matte at various points in the clip.



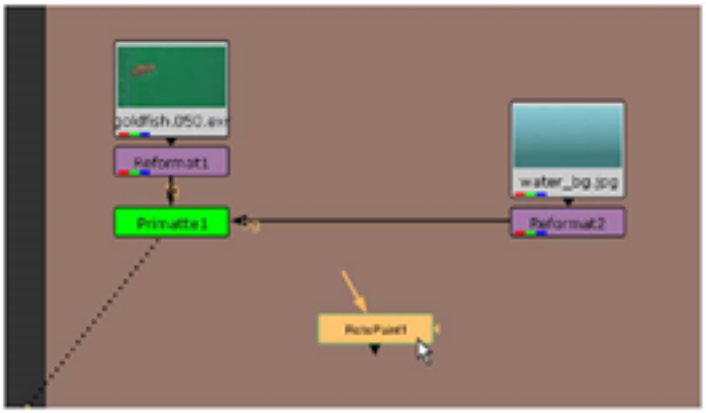
18. If you haven't already done so, save your project under a new file name to save the changes you've made to project.

Rotoscoping

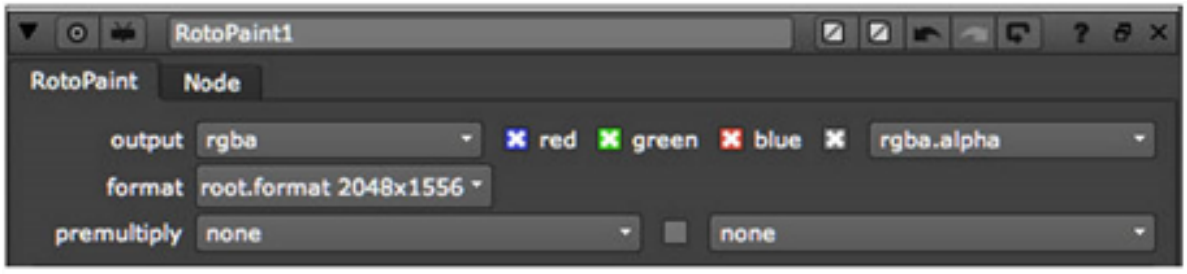
In this example, we'll return to our first keying example to apply a garbage matte and clean-up the aquarium image.

To Draw a Garbage Matte

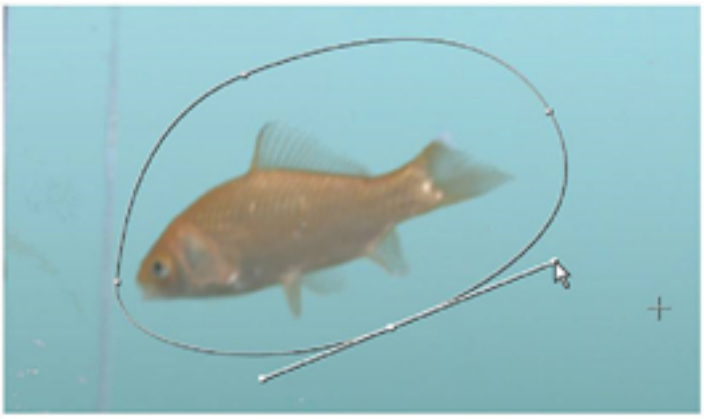
1. Go back to the node tree from the first example, and connect the Viewer to the **Primatte1** node. Drag the time slider to frame **50**.
2. Click an empty spot on the Node Graph to deselect all nodes. Then, right-click and choose **Draw > RotoPaint**.



3. At this point, you don't need to connect the **RotoPaint1** node to anything, but its control panel must be open, and the first tab, **RotoPaint**, should be active.



4. Inside the Viewer, you'll see the goldfish image. Click the **Bezier** tool  in the RotoPaint toolbar on the left side of the Viewer. Then in the Viewer, click four points around the goldfish to create a roto shape. You can drag+click to draw a point and adjust its curve at the same time.



Tip: As long as the RotoPaint1 control panel is open, you can view and edit the roto shape. You can press **Q** over the Viewer to toggle the display overlay, if necessary. Click the right mouse button over any point to select options for the roto shape.

Because this is a garbage mask, we want to edit the shape to remove elements from the glass aquarium.

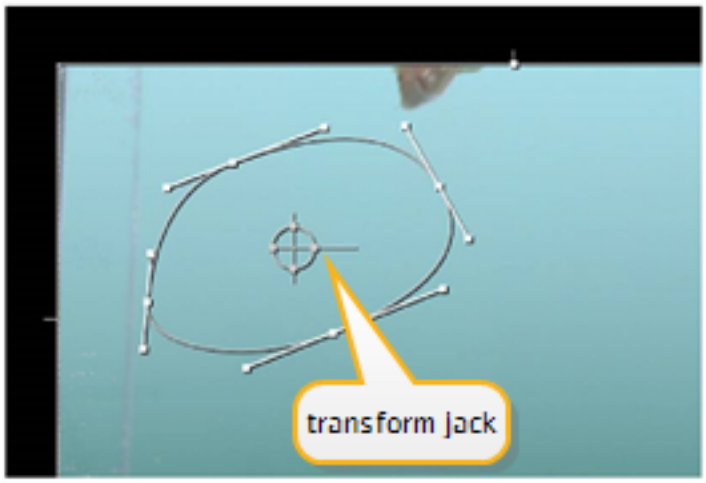
5. Drag the points and adjust the tangents - the handles on each of the points - to refine the roto shape.

Now we need to animate the garbage matte to follow the motion of the fish.

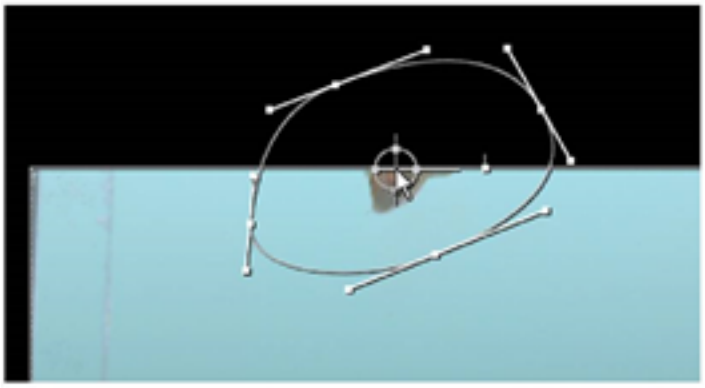
6. In the RotoPaint tool settings panel, on top of the Viewer, the **autokey** option should be active. If not, click the box for this option.



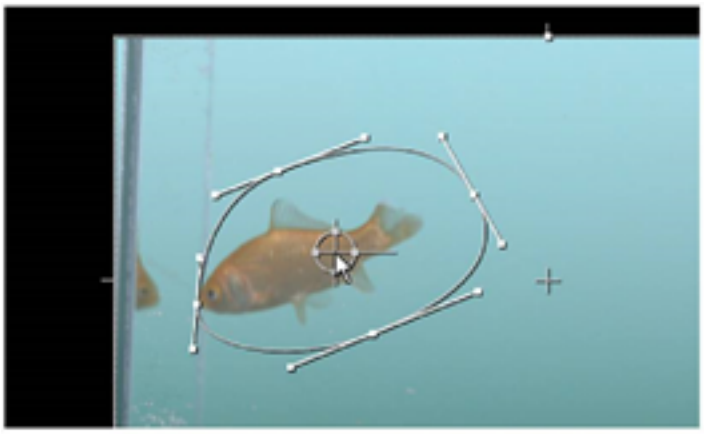
7. Move the time slider to frame **1** and click the **Transform** tab in the RotoPaint control panel. Then select the entire Bezier shape in the stroke/shape list (at the bottom of the control panel) or by clicking one of the points in the shape using the **SelectAll** tool. A transform jack appears.



8. Drag the center point of the transform jack, and move it over the current position of the goldfish.



9. Go to end of the timeline, to frame **60**. Drag the shape once more to adjust for the movement of the goldfish.



If your Bezier shape is similar to the one shown above, then you probably don't need more than the three key frames at frames 1, 50, and 60.

However, you may want to scrub through the timeline and make adjustments.

10. Scrub to frame **60** on the timeline and you'll see the roto gets a little close to corner-line that we want to remove from the aquarium glass.
11. Click on an empty spot in the Viewer to deselect all points. Then, press **Ctrl/Cmd** and click on the point near the goldfish's nose, to temporarily break the point's tangent handle.
12. Adjust the handles to create a peak at the fish's nose.



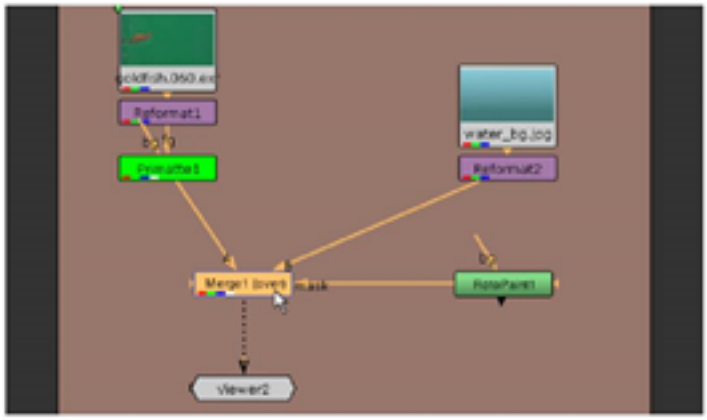
Now, for good measure, let's create a feathered edge for this particular point.

13. With the point selected, drag the feather handle away from the fish to create a feathered edge for this point, at this frame.

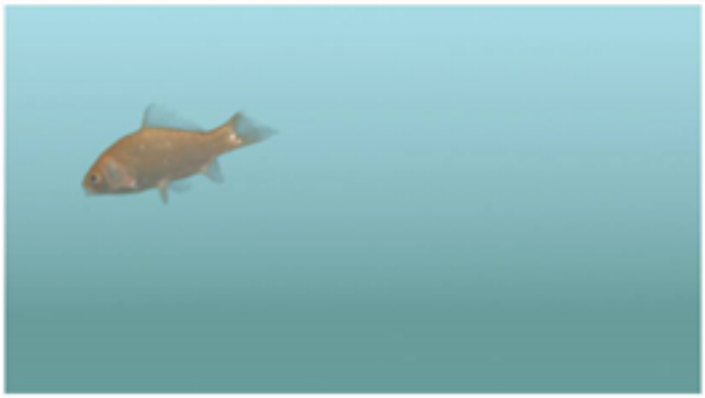


So you've drawn and animated the roto shape. Let's wire it into the node tree to mask out the "garbage."

14. Drag the **bg** connector off the **Primatte1** node to disconnect it from the **Reformat2** node.
15. Choose **Merge > Merge** from the right-click menu. Connect **Primatte1** to the **A** input on the over node. Connect **Reformat2** to the **B** input.



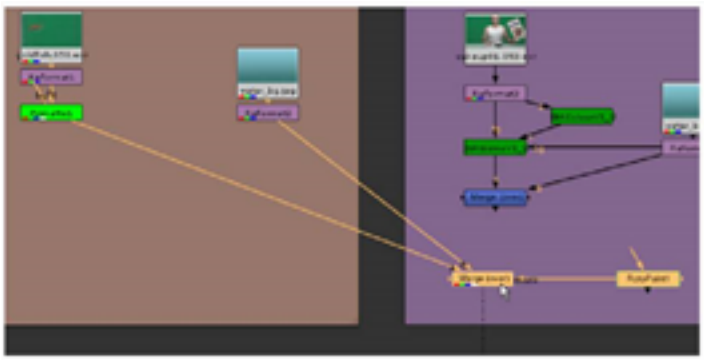
16. Connect the **RotoPaint1** node to the mask connector on the over node. This effectively removes the aquarium reflections in the image.



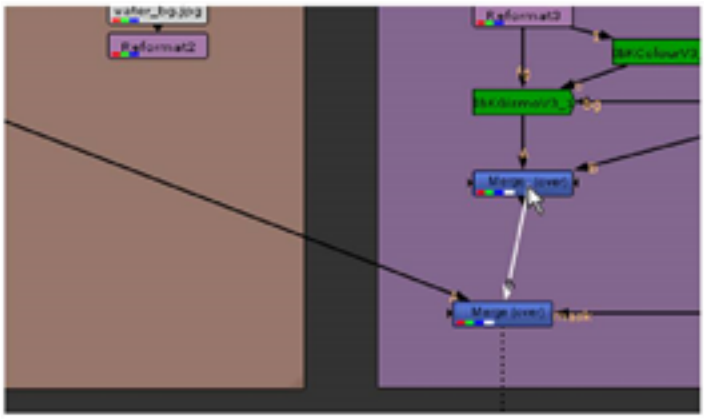
You might want to scrub through the timeline to see if there are places where you need to adjust the roto shape.

If you want to take this a little further, you can now add the goldfish to the composite from the second example.

17. Select and drag the **Merge (over)** node and the **RotoPaint1** node below the node tree you used for the IBK example.



18. Drag the **Viewer** node over, as well, and keep it connected to the **Merge (over)** node.
19. Drag the **B** connector from the **Reformat2** node and connect it to the over node in the IBK node tree.



The Viewer shows the result. Of course, you might want to add a **Transform** node after the first **Merge (over)** node, to size and position the goldfish. Otherwise, this project is completed.



Epilogue

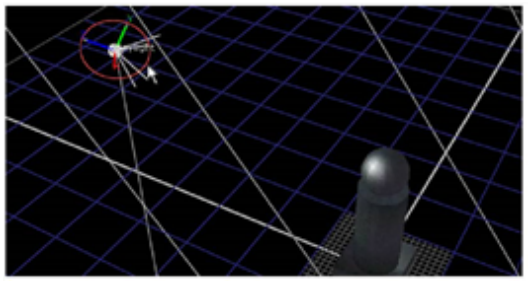
Keying is rarely a simple matter of picking the screen color you want to remove. To get the very best mattes, you often need to combine several techniques and you've learned several in this chapter. You've pulled mattes with Primatte and Nuke's Image-based Keyer, and you've used the rotoscoping tools to cleanup a matte and control the parts of the image you want to use in the composite.

Tutorial 4: 3D Integration

This tutorial teaches you the basics of using Nuke's 3D workspace.

Introduction

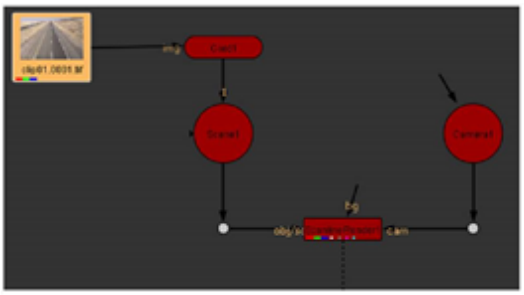
Nuke's 3D workspace creates a powerful compositing environment within your project script. This workspace combines the advantages of cameras, lighting, and a three-axis (x, y, and z) environment, with the speed of node-based compositing. You pipe 2D images into the 3D space, setup a camera, animate your scene, and then render the results back to the 2D composite.



Compositing in the 3D workspace.

The Basic 3D System

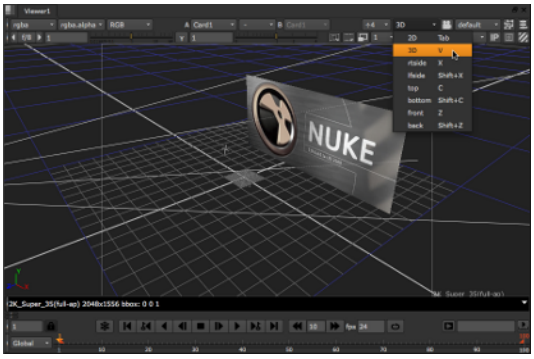
The 3D workspace is defined by a group of nodes in your script. The most basic setup includes a Camera node, a Render node, a Scene or Geometry node, and nodes that provide the 2D images you want to pipe into the 3D compositing space.



The basic 3D node tree: 2D image, geometry, scene, render, and camera.

The 3D Viewer

Once you have the 3D node structure, you can use any Viewer in Nuke as a gateway to the 3D compositing space. Choose **3D** from the view dropdown menu, or press the **Tab** key over the Viewer to toggle between the 2D and 3D views.



On the view dropdown menu, you'll also see orthographic views - rtside, lfside, top, bottom, front, back - which provide non-perspective views into the scene. In three-quarter perspective, it can be difficult to accurately place objects on the axes, and the non-perspective views make it easier to line things up.

The Geometry or Scene Node

Every 3D system needs a piece of geometry - a card, a sphere, a cube, something - to receive an image or clip that the camera can "see." One is all you need, but you can setup complex systems with a large amount of 3D data. When you have two or more objects for a 3D system, you need a Scene node to create a "place" where the camera (and the ScanlineRender node) can see all the objects at once.

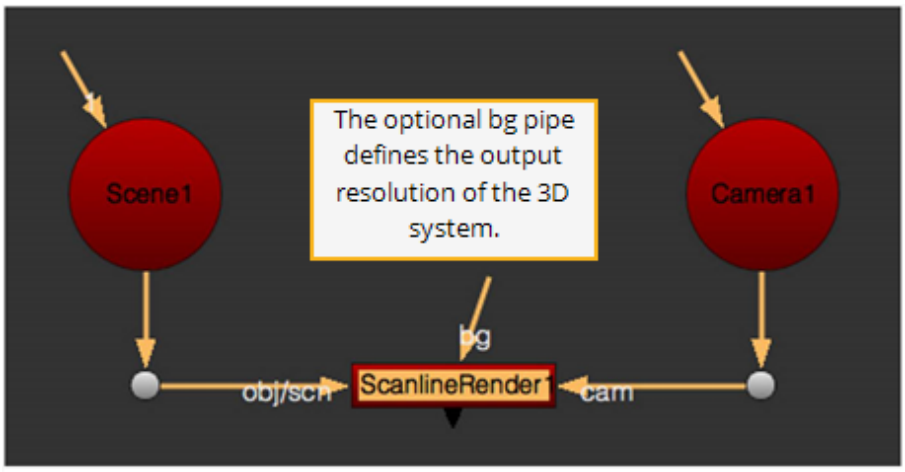
The Camera Node

The Camera node creates your view into a scene. It has several controls to help you match the properties of a physical camera. You can animate its position or import animation or tracking data to matchmove your 3D scene with a background plate. A 3D system can have multiple cameras connected to the Scene node, to create different views on a 3D scene.

Tip: Only one camera can be connected to the ScanlineRender node to generate the output, but you can insert multiple ScanlineRender/Camera node pairs to generate output from various perspectives.

The ScanlineRender Node

The last node, ScanlineRender, sends the results of your 3D scene back into your composite as a 2D image. It's always 2D in, 3D manipulation, and then 2D back out, which is why this is often called "2-and-a-half-D."



The scanline render node converts the image back to 2D.

The image created by the ScanlineRender node is the same resolution as your project settings. When you need to render a specific resolution, use the optional **bg** pipe. Connect a Constant node with the resolution you want and that defines the output of the ScanlineRender node.

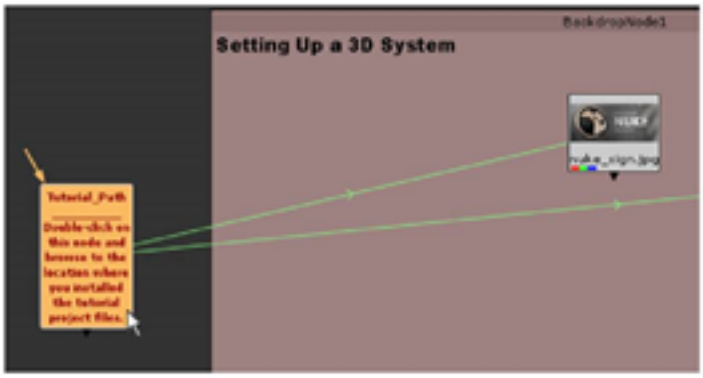
So now you know the basic 3D setup for your compositing script. Let's take a test drive.

Open the Tutorial Project File

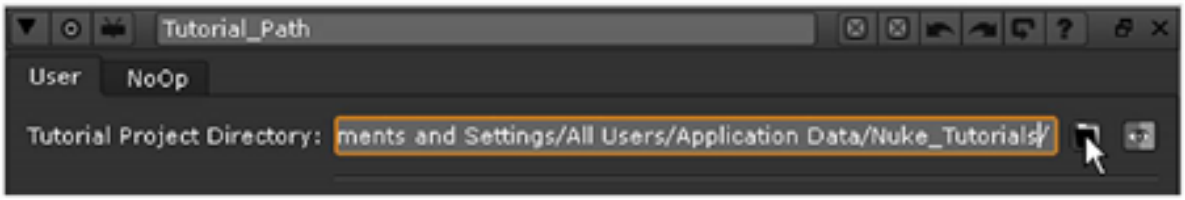
The "3Dinteg_tutor.nk" project file includes the node trees for the first part of this chapter.

To Open the Project File

1. Launch the Nuke application and choose **File** > **Open Comp** from the menu bar.
2. In the file browser, navigate to your **Nuke_Tutorials/3DInteg/** folder, select the **3dinteg_tutor.nk** project file and click **Open**.
3. Locate the **Tutorial_Path** node, on the left side of the script, and double-click it to open its control panel.



4. Click the "file folder" button. Browse to the location where you installed the tutorial project files, and then click **Open** to select the location.



   After you select the correct path, the error messages should clear from the Read nodes, and the thumbnails in the script update with the correct images.
5. Close the **Tutorial_Path** control panel. Then, choose **File > Save Comp As** to save a copy of the project file.
6. Move the mouse pointer over the Node Graph, and press **F** to frame the entire contents of the project file.

   The green arrows (lines) show the links between the Tutorial_Path node and the Read nodes.
7. If you wish, press **Alt+E** to hide the expression arrows.

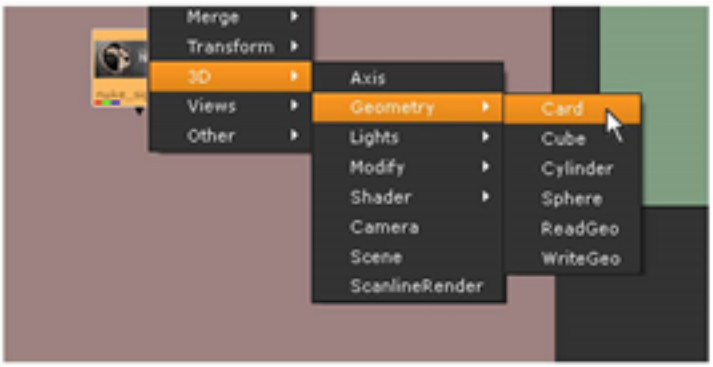
The Tutorial_Path node saves the location of the project files on your computer, so you don't need to repeat this for future sessions with this project file.

Setting Up a 3D System

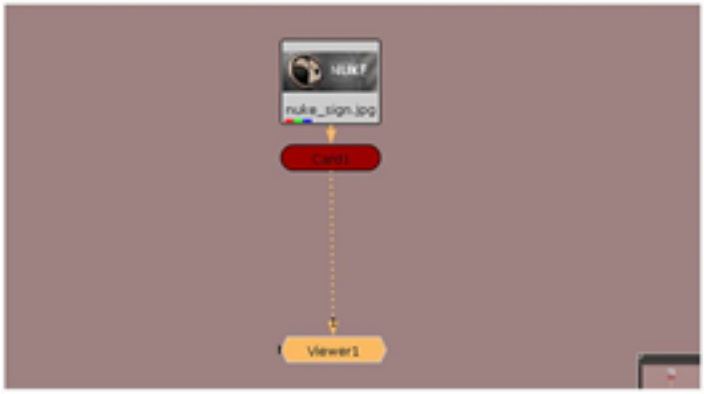
Let's start with the basics. In this first example, you'll create a basic 3D node tree, map an image to a 3D card, manipulate it, and then render the result back out to the 2D composite.

To Set Up a 3D Node Tree

1. In the "3Dinteg_tutor.nk" project file, locate the backdrop node labeled "**Setting Up a 3D System**." You'll see a Read node with the image you'll use for this example.
2. Right-click over the **nuke_sign.jpg** node, and choose **3D > Geometry > Card**.

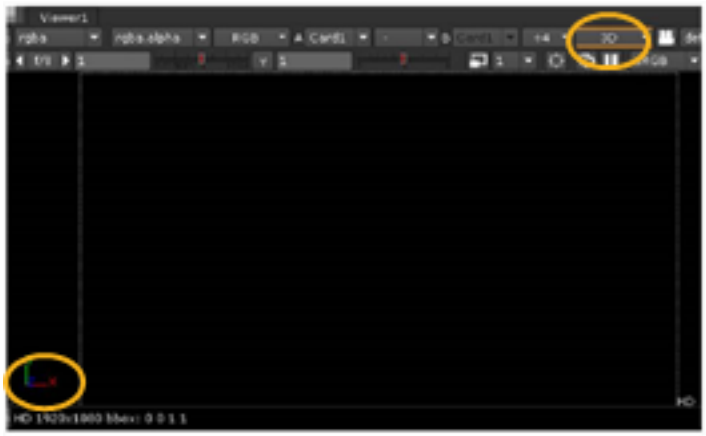


This attaches a "Card1" node. Let's see what it looks like in 3D.



3. Attach a Viewer to the **Card1** node and Nuke switches the Viewer to 3D.

Wow, that's amazing. It looks exactly like the 2D Viewer. How can anyone tell the difference? Check the lower-left corner of the Viewer and you'll see an orientation marker for the three axes in 3D. You'll also see that "3D" is displayed on the view dropdown menu.



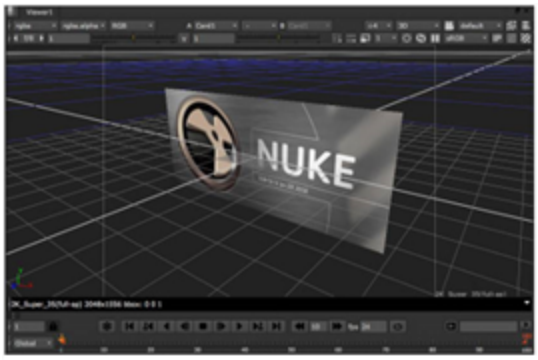
That sign is a little darker than expected, isn't it? Actually, you can't see the image yet because the default view of the 3D workspace is at the origin or center of the space. Perhaps zooming-out may improve the view.

4. Press the **Alt** key (Windows /Linux) or the **Option** key (OS X), and drag with the middle mouse button to zoom or "dolly." Drag to the left and you'll zoom out.



Hey, look. There's the Nuke emblem. In the 3D Viewer, the "pan" and "zoom" controls are exactly the same as what you've used for the node tree and the 2D Viewer, but let's try "tumbling" to get a better view.

5. **Alt-** or **Option**-drag with the right mouse button to rotate around the origin point of the 3D workspace. You now see the 3D grid and the image mapped to the card.

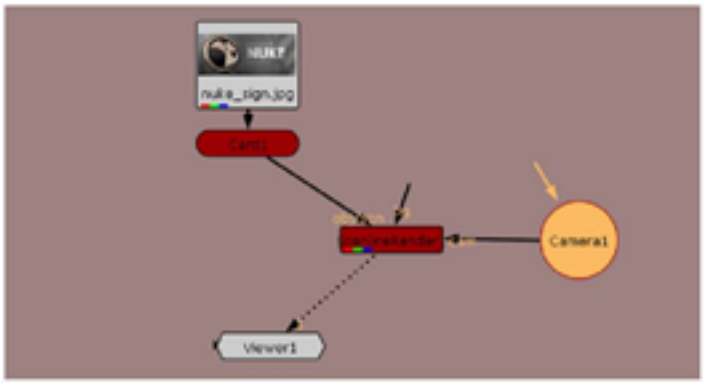


When an image is connected directly to a Card node like this, it is applied as a flat or "planar" map. The size of the card adjusts to the dimensions of the image.

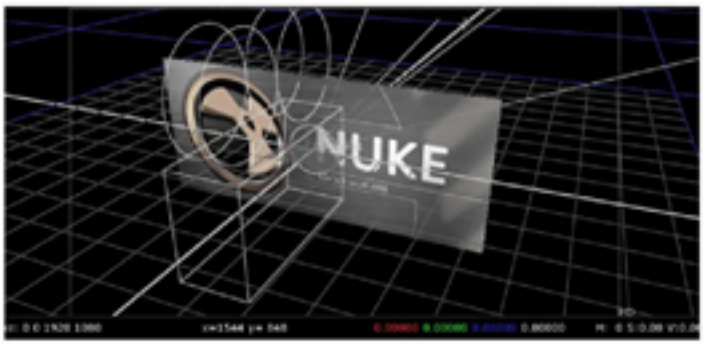
6. Click on the card and you select the node in the node tree and also the card inside the 3D workspace.
7. Use the mouse (and the **Alt** key) to navigate through the workspace. Go ahead, pan, dolly, and rotate at will. Then, press **F** over the Viewer to frame the 3D view.

Tip: If you don't like the standard navigation controls, open the **Preferences** control panel (**Shift+S**), select the **Viewers** tab and change the **3D control type** to **Maya**, **Lightwave**, or **Houdini**.

8. Click on an empty spot in the Node Graph to deselect all nodes. Let's add the other nodes you need.
9. Right-click on the Node Graph and choose **3D > Camera**. Keep its control panel open so you can manipulate the camera in the Viewer.
10. Right-click and choose **3D > ScanlineRender** to insert a render node, and then connect the nodes as shown below.

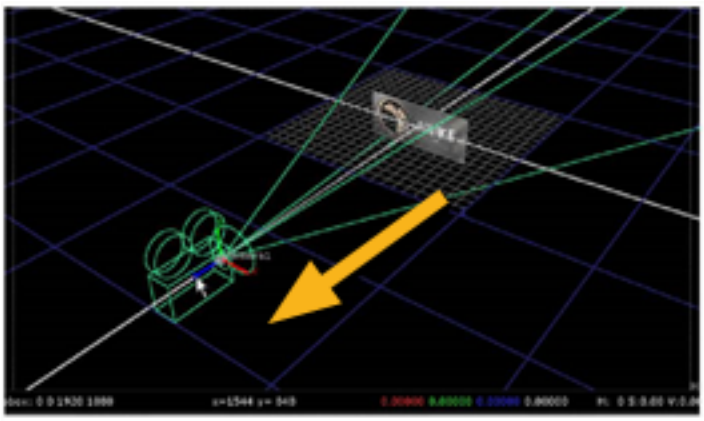


11. Connect the Viewer to the **ScanlineRender** node, and you have the most basic 3D system in Nuke.
12. Press **Tab** over the Viewer to change to the 2D view. You won't see the Nuke emblem - hey, where did it go? We saw it before.
13. Press **Tab** again to switch back to 3D. You'll see the default camera position is too close to view the card. Let's move things around to get an image for 2D.

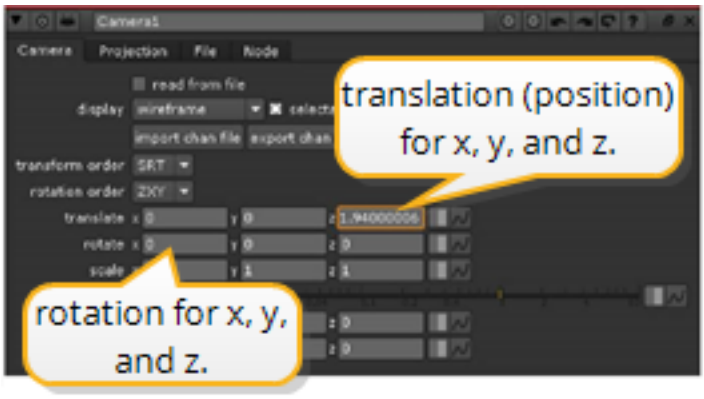


To Position Objects in the Scene

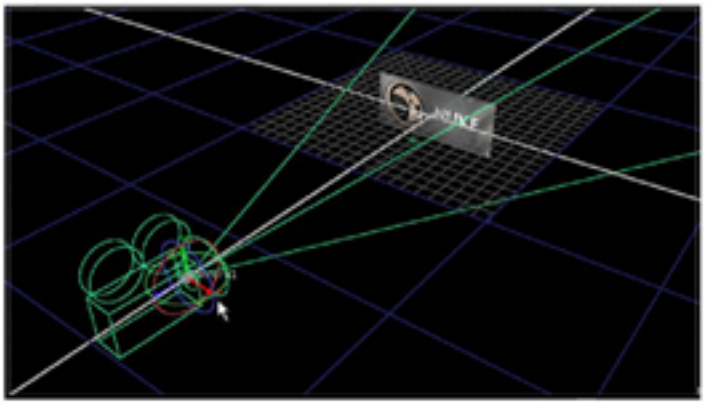
1. **Alt-** or **Option-**drag with the middle mouse button to dolly out and show more of the 3D workspace.
2. Select the camera. You can do this by clicking the camera object in the Viewer or clicking the **Camera1** node in the Node Graph.
3. Drag the transform handles to move the camera away from the card, along the z-axis.



As you drag the camera, look at the camera's control panel. You'll see the x/y/z transform values reflect the camera's current position.



4. Press and hold **Ctrl** (Mac users press **Command**) over the Viewer and the transform handles change to rotation rings.



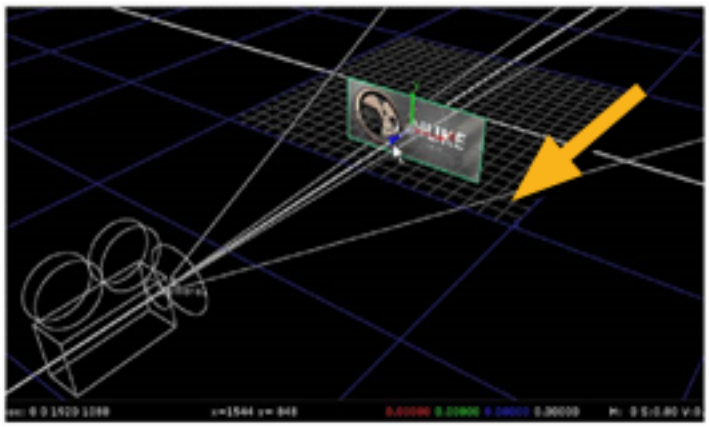
5. Drag the green ring to rotate the camera around the Y-axis. Notice the x/y/z rotation values in the control panel reflect the angle of the rotation.

   The blue handle "rolls" or rotates on the Z-axis, and the red handle rotates on X.

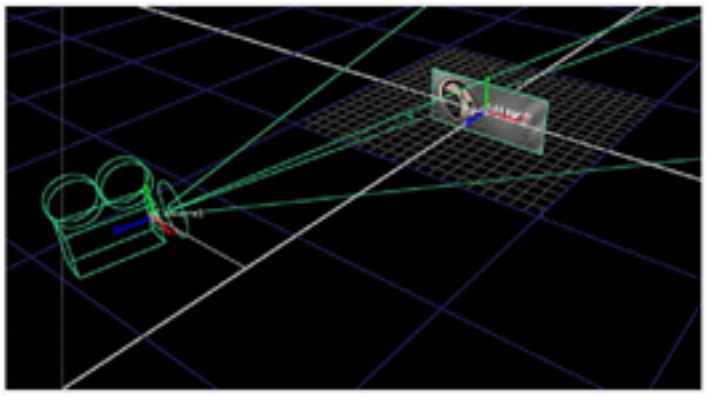
6. Now, select the card object and move it away from the camera.

   Keep the control panel open for the Card node. As with the Camera node, the transform handles disappear from the Viewer when you close the control panel.

7. Drag the card's transform handles to position it in the 3D workspace. If you wish, press the **Ctrl** key (Mac users press **Command**) over the Viewer and rotate the card.



8. Press **Tab** over the Viewer to switch between the 2D and 3D views to see the image the ScanlineRender node produces.





9. Before you continue to the next example, close all the control panels that are currently open.

   In this example, it doesn't matter where you move the camera or card. In reality, however, you often need to use specific values, which you can enter directly in the control panels.

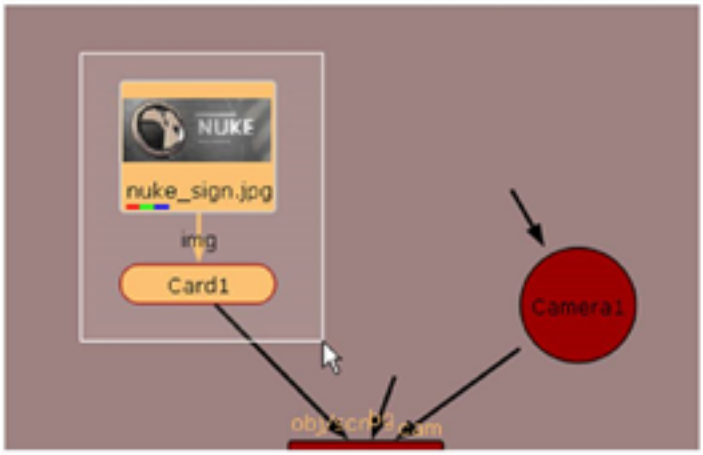
   You can also import camera data or animation curves - did you notice the **import chan file** button in the camera's control panel? - and apply them to the objects in the workspace.

Making a Scene

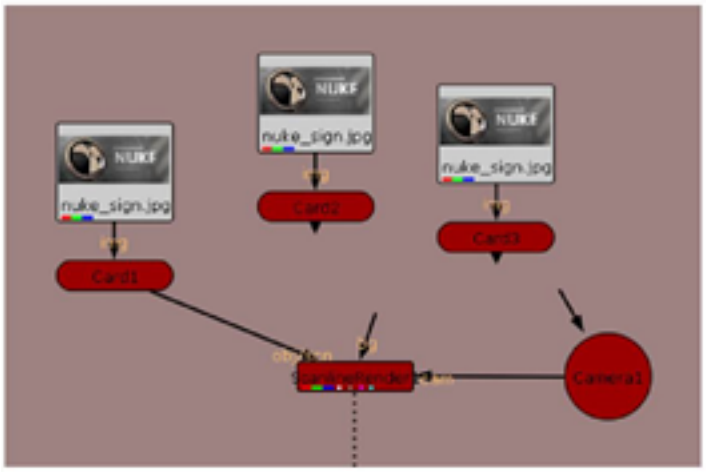
We mentioned earlier the Scene node creates a place where multiple objects may be seen by the camera and the render node. If you only have a single object, you don't need a Scene node, but where's the fun in that? Scene nodes make it possible to really tap into Nuke's ability to handle huge amounts of 3D information, and you should know how to use it.

To Set Up a Scene

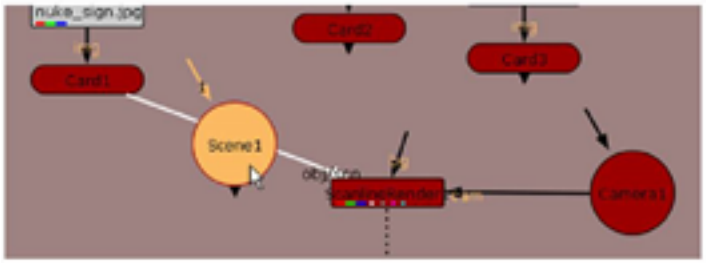
1. Inside the "Setting Up a 3D System" node tree, drag a selection around the **nuke_sign.jpg** node and the **Card1** node to select them.



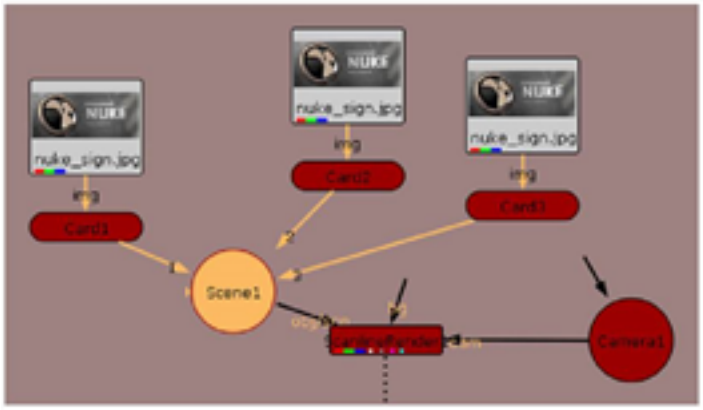
2. Press **Ctrl+C** (Mac users press **Cmd+C**) to copy the selected nodes or choose **Edit > Copy** from the right-click menu.
3. Press **Ctrl+V** (Mac users press **Cmd+V**) or choose **Edit > Paste** to insert a copy of the nodes. Press **Ctrl+V** or **Cmd+V** again to insert a second copy, and then arrange the nodes as shown below.



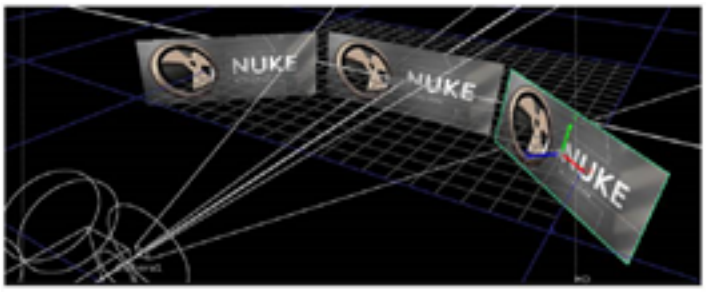
4. There are multiple cards now, and you need a Scene node to create a space where the rendering node can see all cards at once.
5. Click on an empty space in the Node Graph to deselect all nodes. Right-click and choose **3D > Scene** to insert the "Scene1" node.
6. Drag the **Scene1** node onto the **obj/scn** connector to insert the Scene1 node between **Card1** and **ScanlineRender1**.



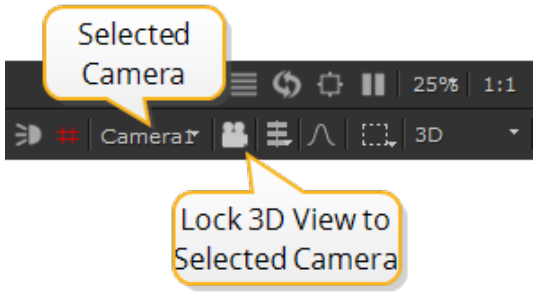
7. Connect the **obj/scn** connector from the **ScanlineRender1** node to the **Scene1** node. Connect each **Card** node to the **Scene1** node.



8. Double-click on the **Card1** node to open its control panel. In the Viewer, you'll see the transform handles for the first card.
9. Move and rotate the card to a different position. For this example, it doesn't matter where you place it.
10. Open the control panel for **Card2** and move its card to a different place in the scene.
11. Open the **Card3** control panel and move that card, also.



You could switch to the 2D view to see the result, but why not just look through the camera? Next to the view dropdown menu, you see the button that locks the 3D view to the camera.



12. From the view dropdown menu, choose **3D** (**V**) to switch to a 3D perspective view. Then click the "lock view to 3D camera" button.

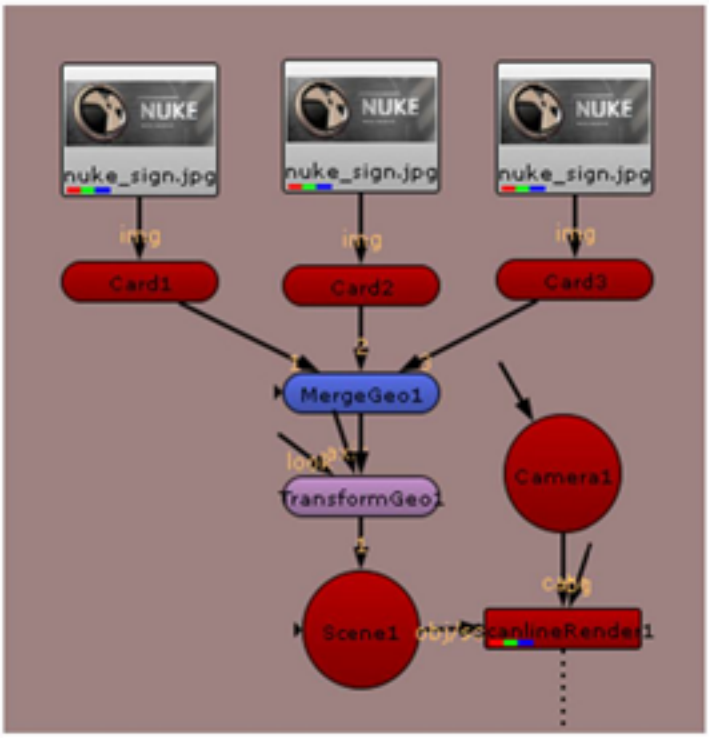


13. Turn off the “lock 3D view to selected camera” button. You won’t need it during the rest of this tutorial.

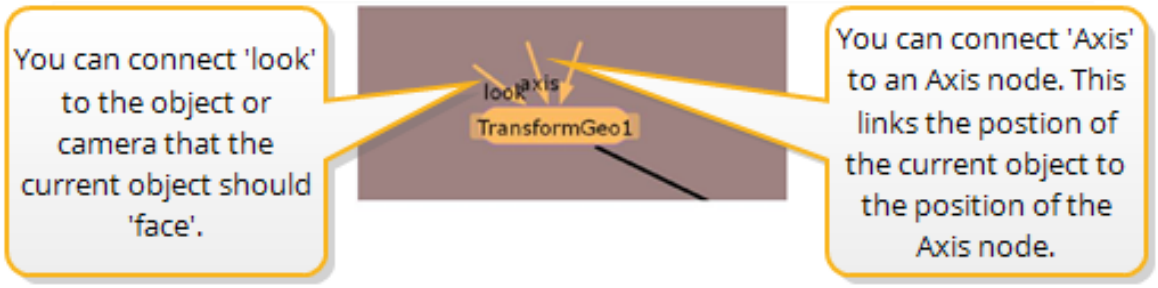
Merging and Constraining Objects

You can merge objects and move them together as a group. To do so, you need to insert MergeGeo and TransformGeo nodes after the objects. The MergeGeo node first merges the objects together, after which you can use the controls of the TransformGeo node to move the merged objects in the 3D space. You can also use the TransformGeo node to constrain objects, as you may notice later in this tutorial.

To merge the three card objects together, right-click on the Card1 node and select **3D** > **Modify** > **MergeGeo**. This inserts a MergeGeo node between Card1 and Scene1. Disconnect the Card2 and Card3 nodes from the Scene node and connect them into the MergeGeo node. Then, right-click on the MergeGeo node and select **3D** > **Modify** > **TransformGeo**. Your node tree should now look like the following:



On the TransformGeo nodes, you see multiple connectors. The connector without a label should be attached to a geometry object or a MergeGeo node. The other connectors act as constraints on the connected object's position.



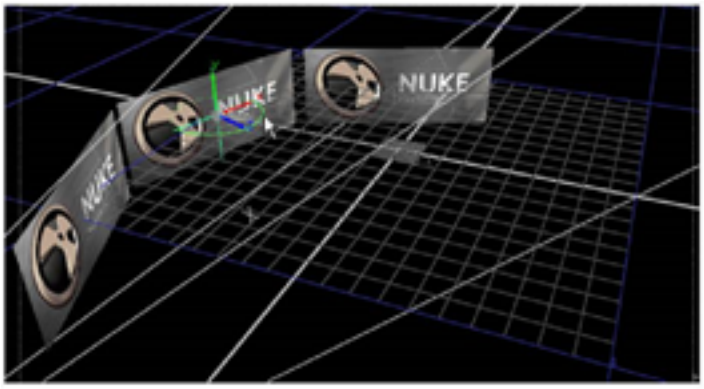
When a camera or object is connected to the optional **look** connector, the TransformGeo node adjusts the rotation so that the object's z-axis always "points" to the camera or object.

The **axis** connector can be used to link the current object to the position, rotation, and scale of a special 3D object called the Axis node. If you've worked with other 3D applications, you know the Axis node as a "null" or "locator" object.

You are still working with the "Setting Up a 3D System" node tree. The following steps show how you can move the merged nodes, and also how to make objects "look" at the camera and other objects.

To Move the Merged Objects Together

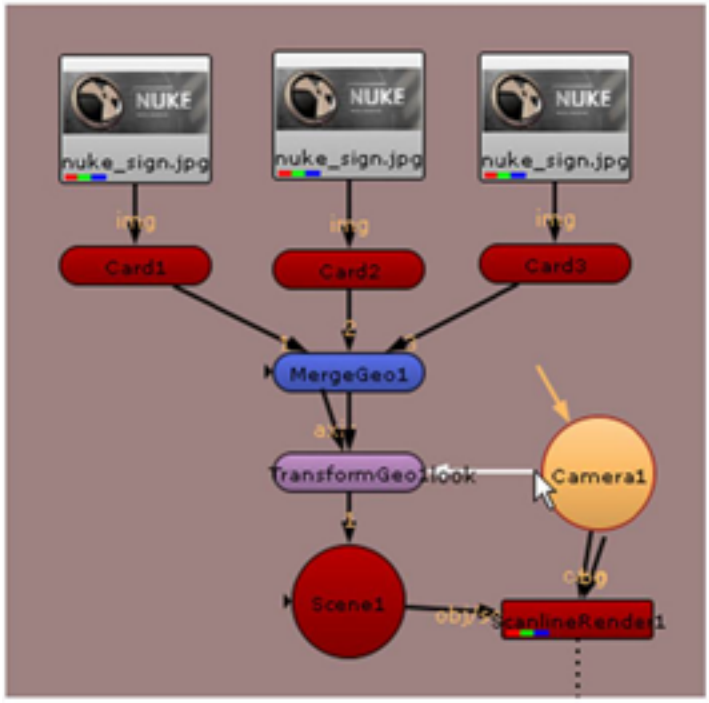
1. Click on the TransformGeo1 node to select it. Its control panel should also be open and you'll see its transform handles in the Viewer.
2. Drag the handles to move all the cards merged with the MergeGeo node.
3. Press the **Ctrl** or **Command** key and drag the rings to rotate the cards as a group.



4. In the **TransformGeo1** control panel, drag the **uniform scale** slider to increase the size of the entire group of cards.

To Make Objects ‘Look’ at the Camera

1. Drag the **look** connector from the **TransformGeo1** node onto the **Camera1** node.



   In the Viewer, you’ll now see the TransformGeo1 node is constrained to the location of the camera.

2. Select and move **Camera1** in the Viewer window. As you do so, the three cards controlled by the **TransformGeo1** node rotate to “look at” the camera location.

   Why is this useful? Let’s assume you have a 2D matte painting mapped to a card in your scene. The “look” option ensures that the plane of the painting always faces the camera, regardless of the camera position, and maintains the illusion depicted by the painting.

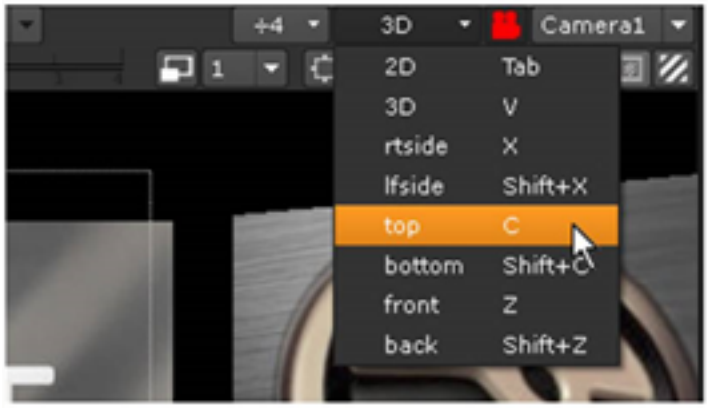
Before you move on, disconnect the TransformGeo node’s **look** connector from the camera.

Animating a Scene

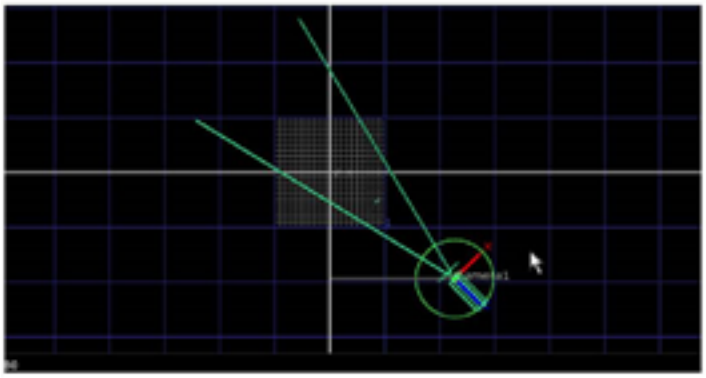
The little scene you’ve created would be more interesting with a camera move. You can animate both cameras and objects; in each control panel, you’ll see an Animation button next to each parameter you can animate over time.

To Animate the Camera

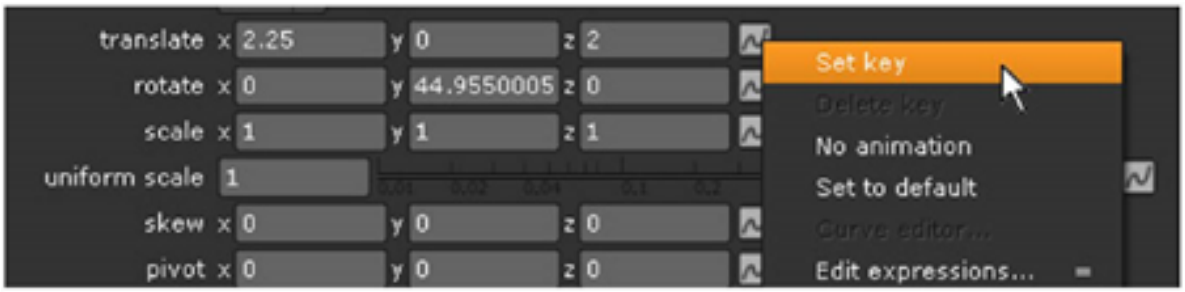
1. In the Viewer, drag the time slider to frame **1** on the timeline.
2. Let's switch to an overhead view to move the camera. Choose **top** from the view dropdown menu.



3. Double-click on the **Camera1** object (either inside the Viewer or on the Node Graph) to open its control panel.
4. Move the camera to the right and rotate it to "look" at the center of the 3D workspace.

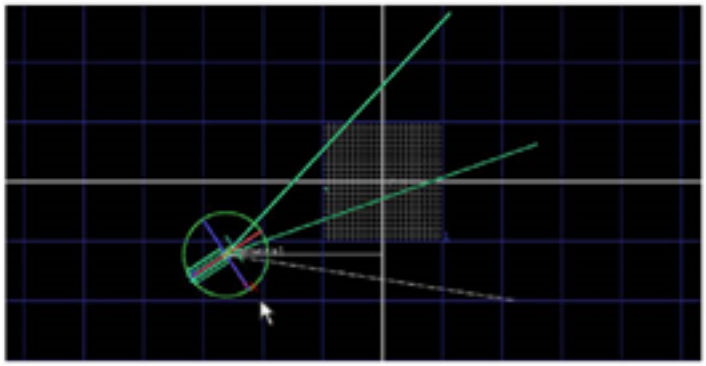


5. Click on the animation button next to the camera's **translate** parameters, and choose **Set key**.



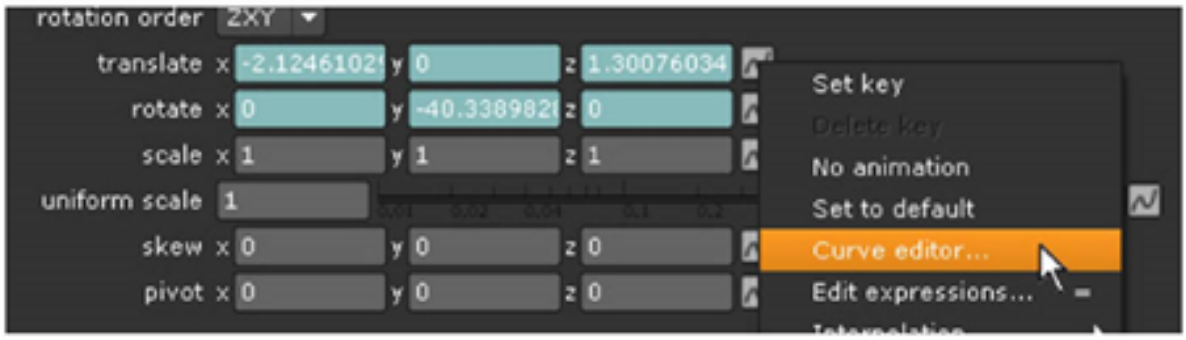
The background of the parameter boxes change color to show that a keyframe is now set for these values at the current frame in the timeline. Now, you need to set a keyframe for the rotation values.

6. Next, click the animation button next to the camera's **rotate** parameters, and choose **Set key**.
7. In the Viewer, scrub to the end of the timeline. Then, move the camera to the left side and rotate it to face center. This automatically sets keys for the **translate** and **rotate** parameters, for the last frame.



8. Drag through the timeline and the camera moves between the positions recorded by key frames.

   Yawn. With only two key frames, the camera moves in a straight line, from start to finish. Let's edit the animation curves to make this more interesting.
9. Click the animation button next to the camera's translate parameters and choose **Curve editor**.



This opens the Curve Editor window in the same pane as the Node Graph. The outline at the left side of the Curve Editor displays a list of the parameters you've animated, and the curves show the values plotted over time. Each dot shows the value recorded for a value on a keyframe.



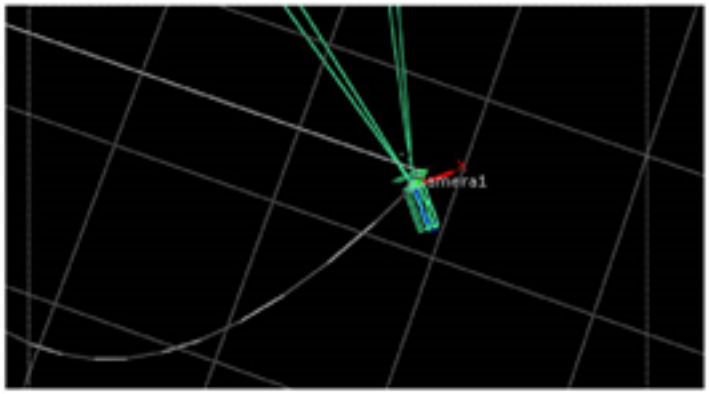
Yes, we know. They don't look like curves - yet. You only set two key frames, remember? You can press **Ctrl+Alt** (Mac users press **Cmd+Alt**) and click on any line to add more key frames. However, let's assume you want to create a smooth arc between the first and last recorded position of the camera. Rather than set more key frames, let's just change the shape of the curve.

10. You want to control the shape of the camera path along the z - the distance between the origin point and the camera, so click the **translate / z** parameter in the Curve Editor.

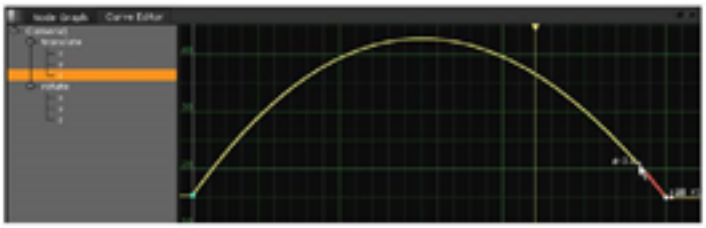
    Click on the first point of the translate/z curve to select it, and drag the tangent handle upward, as shown below.



    From this point forward, the curve increases the distance of the camera on the z-axis, which you'll now see in the Viewer.

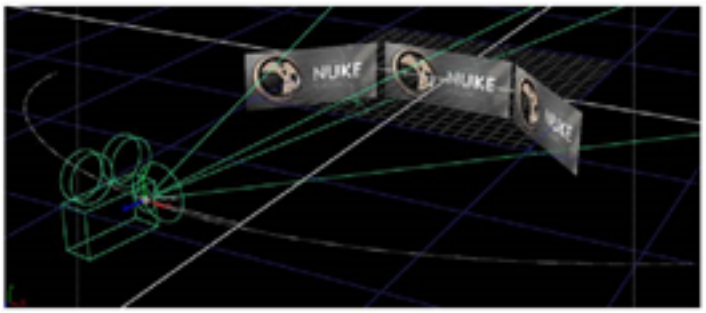


11. Click on the last point of the translate/z curve to select it, and drag it upward also to finish the desired shape. This eases the distance back toward the value of the keyframe at the end of the timeline.



    Select the camera in the Viewer and you should see the gradual slopes of the curve create a more interesting arc for the camera move.

    Switch to the **3D** (V) perspective view and scrub through the timeline to see the new camera move.



Your version may look a little different than this, depending on the positions and rotations you defined, but you get the idea.

12. If you wish, you can set additional key frames to refine the camera path. Hold down **Ctrl+Alt** or **Command+Option** and click on the z-axis curve in the Curve Editor, to add new points to the curve, and then adjust their positions.
13. Before you continue, click the **Node Graph** tab to hide the Curve Editor and return to the node trees.
14. Close all the control panels that are currently open.

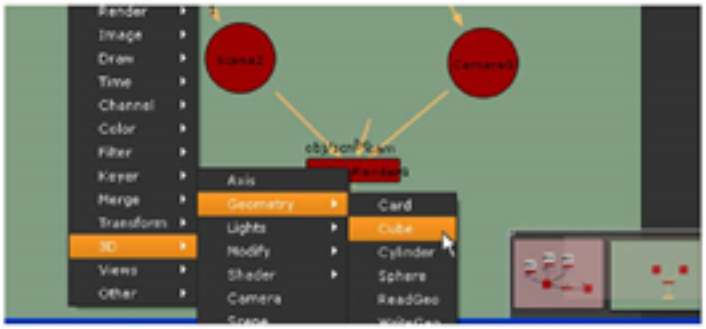
Working with Geometry

In the previous example, you worked with the card object. Nuke also includes primitive geometry, which can be used as set-extension geometry or placeholders for other elements you plan to add to scene.

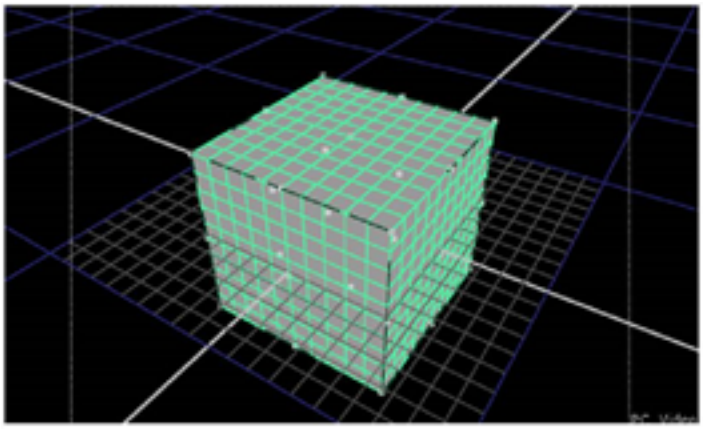
To Add Primitive Objects to the Scene

1. In the "3Dinteg_tutor.nk" project file, locate the node tree labeled "**Working with Geometry**."

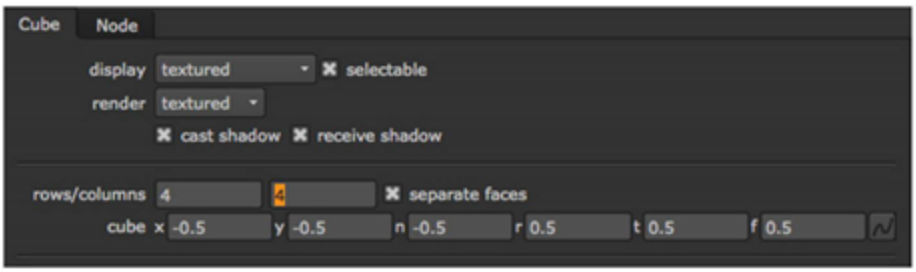
   We've already supplied the 3D node tree with a camera for you, so you need to add the geometry objects, and also create a "scene" where they can co-exist.
2. Right-click over the Node Graph and choose **3D** > **Scene**. Connect **Scene2** to **ScanlineRenderG**.
3. Connect a Viewer to the **ScanlineRenderG** node and switch to the **3D** perspective view.
4. Right-click and choose **3D** > **Geometry** > **Cube**.



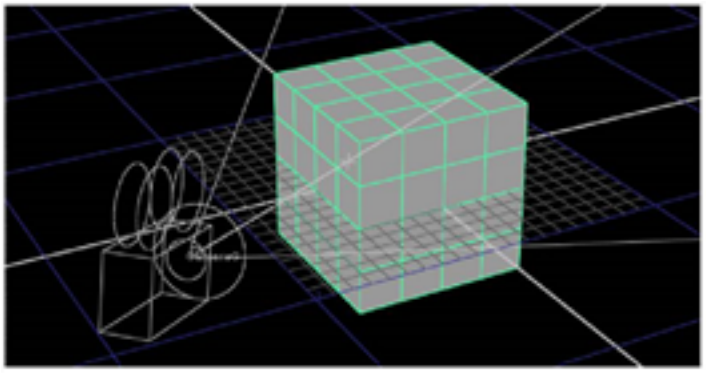
The default cube primitive appears at the center of the 3D workspace. Let's reduce the number of subdivisions on the cube.



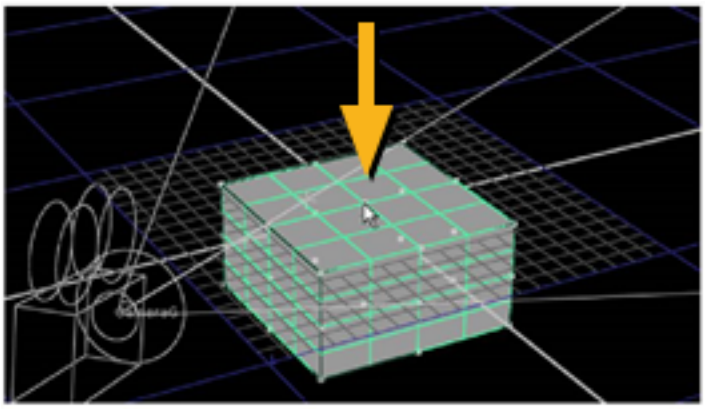
5. In the Cube1 control panel, change the **rows** parameter to **4**. Change the **columns** parameter to **4**, also.



6. Connect the **Cube1** node to the **Scene2** node. Now let's adjust the shape of the cube.



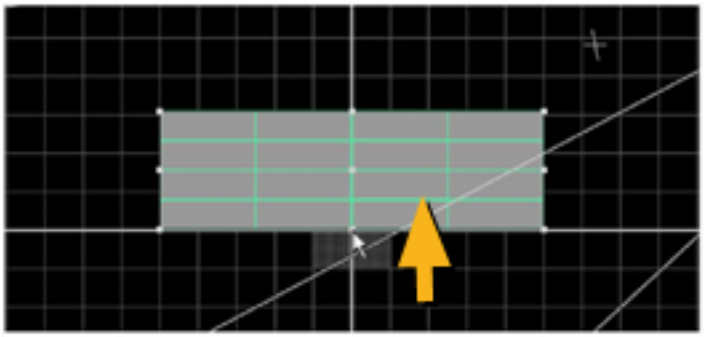
7. Reduce the height of the cube by dragging the top-center point down.



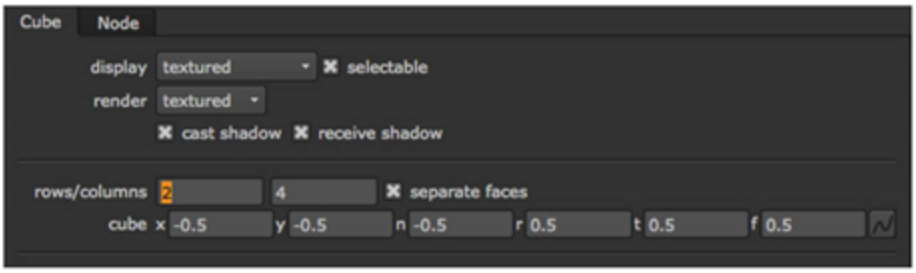
8. From the view dropdown menu, choose the **front** view to see a non-perspective view of the cube. These non-perspective views can help you size and position objects with more accuracy than you might get in a perspective view.

   Mm... the cube is actually below the x-axis. Let's move it up, but check the values in the **Cube1** control panel.

9. Drag the top of the cube until the **t** (top) value in the **Cube1** control panel is about **0.3**. Drag the bottom of the cube to align it with the x-axis.

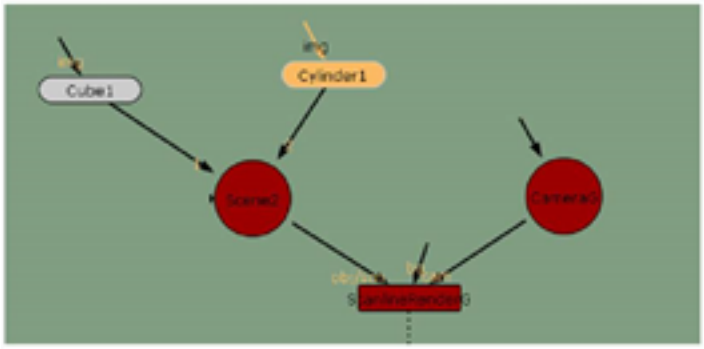


10. It looks like you don't need 4 divisions on the sides of the cube, so change the number of rows to **2** in the **Cube1** control panel.

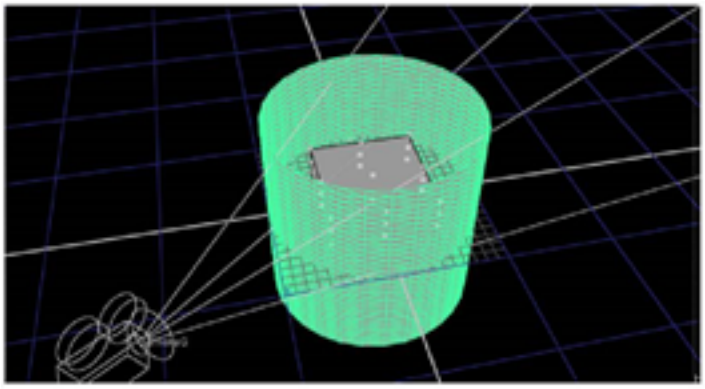


   Now let's add a few more primitives - a cylinder and a sphere.

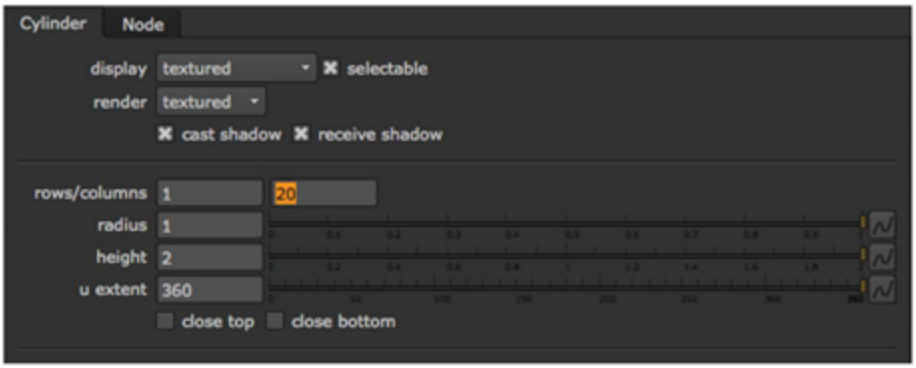
11. Right-click on the Node Graph and choose **3D** > **Geometry** > **Cylinder**. Connect the **Cylinder1** node to the **Scene2** node.



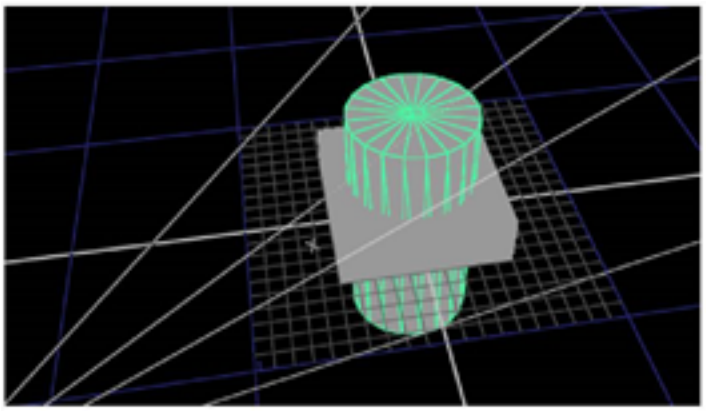
12. Change the view to **3D** (V) and zoom out a little to see the whole cylinder.



13. In the Cylinder1 control panel, set the **rows** to **1**, the **columns** to **20**.

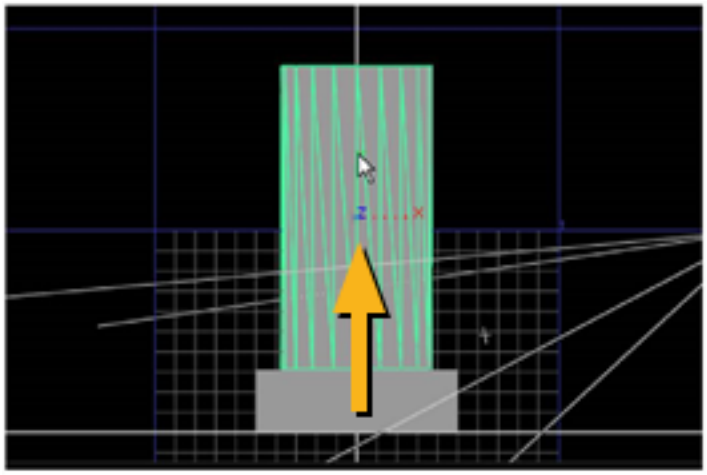


14. Set the **radius** to **0.35** and the height to **1.5**. Also check the box for **close top**.

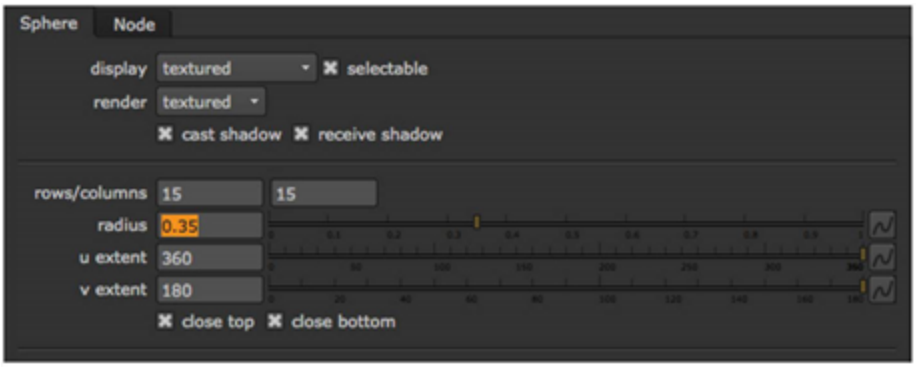


So now you have a cylinder in the scene.

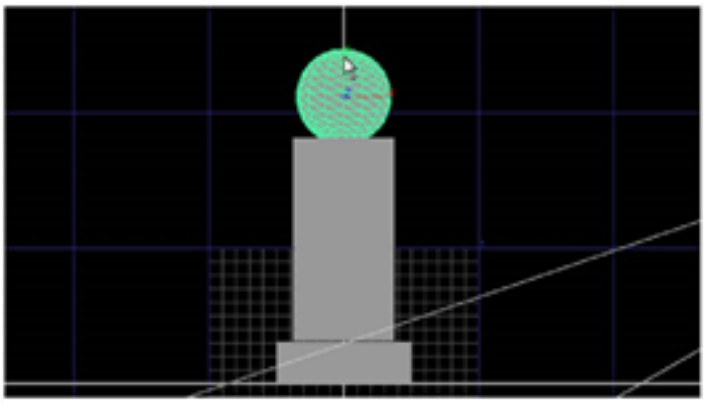
15. Choose front from the view dropdown menu and move the cylinder up to rest on top of the cube.



16. Now add a sphere. Choose **3D** > **Geometry** > **Sphere**. In the Sphere1 control panel, set both the **rows** and **columns** to **15**, and change the **radius** to **0.35**.



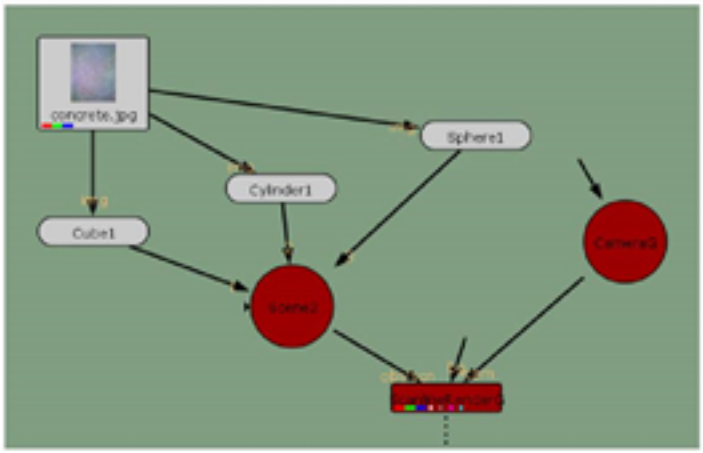
17. Make sure the Sphere control panel is open and move the sphere object to cap the top of the cylinder.

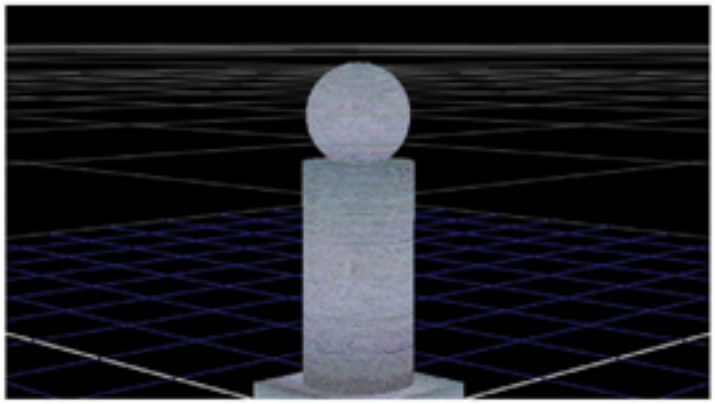


18. Select the **3D** from the view dropdown menu and rotate the view around the objects in your scene.

    At this point, they have no surface properties, so you'll need to connect a 2D image from the Node Graph to each object.

19. In the Node Graph, connect the **concrete.jpg** to each of the objects.

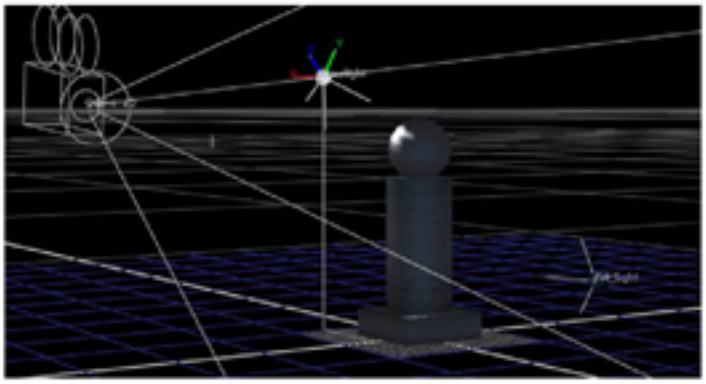




Lighting and Surface Properties

Nuke includes lighting tools to enhance the existing lighting in the plates and images you include in a 3D scene. Also included are fundamental surfacing tools to control the attributes of the objects in the 3D workspace.

These tools are not designed to replace the use of true 3D lighting and surfacing, but they can definitely help you punch up the scene and quickly tweak the settings without sending elements back to the 3D application.



Nuke's lighting objects introduce lighting and surface attributes into your scene. When the scene has no lighting objects, then all surfaces are illuminated with the same properties and level of "brightness."

In the following steps, you'll first add nodes that define surface properties for the objects, and then you'll add the light objects to illuminate them.

To Define Surface Attributes to Objects

1. Click on an empty place in the Node Graph to deselect all nodes. Then, choose **3D > Shader > BasicMaterial**.
2. Drag the **BasicMaterial1** node onto the connector between concrete.jpg and **Sphere1**.



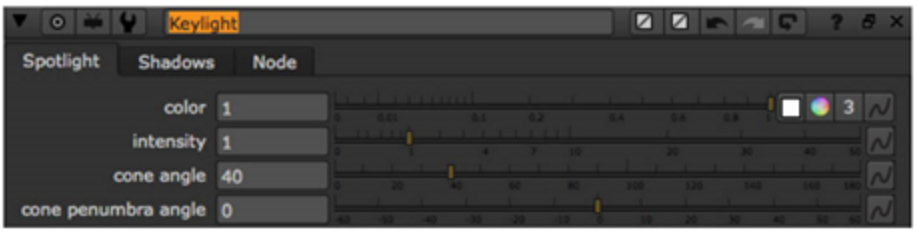
In the Basic Material control panel, you can see parameters to define the amount of light emission, diffusion, and specular properties of the surface. You can mask these properties by connecting images to the **mapS** (specular), **mapE** (emission), and **mapD** (diffuse) connectors, but this is not required for this example.



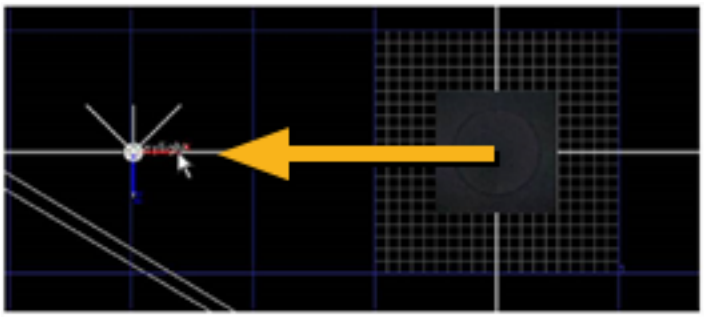
3. Set the light **emission** control to **0.25**. Set **diffuse** to **0.18**, and **specular** to **0.75**.
4. Adjust the **min shininess** and **max shininess** values to adjust the quality of the specular highlights. Once again, nothing seems to happen! That's because you haven't yet added a light into the scene. All surfaces are illuminated with the same level of brightness, so there are no specular highlights or other controllable light properties visible. It's not very exciting, but don't worry - you get to add a light into the scene soon.
5. Make two copies of the **BasicMaterial1** node and attach a copy before **Cylinder1** and before **Cube1**.

To Add Light Objects to a Scene

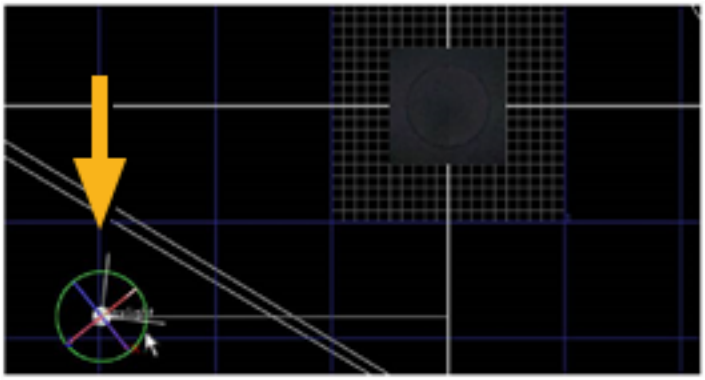
1. Choose **3D > Lights > Spot** and connect the light node to the **Scene2** node.
2. In the **Spotlight1** control panel, rename the light to **Keylight**.



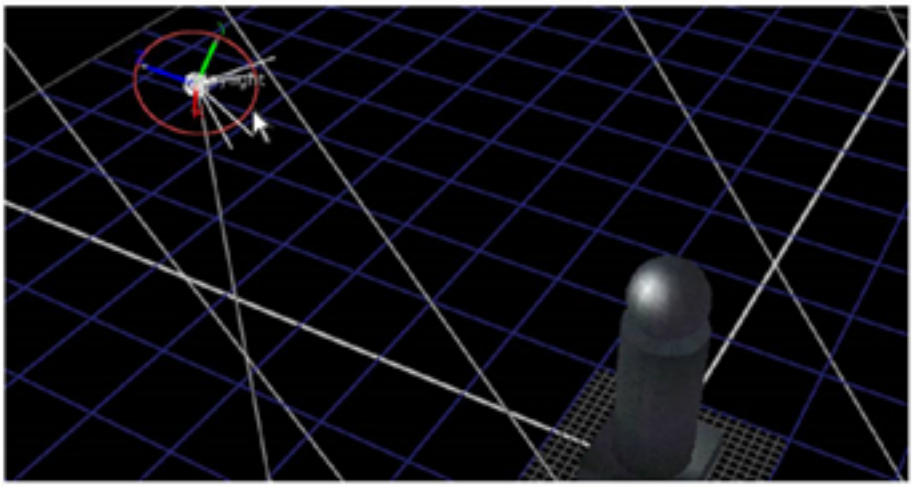
3. Switch to the **top** view and drag x-axis handle (red) to move the light to the left.



4. Drag the z-axis handle (blue) to move the light closer to the bottom edge of the screen. Then, press the **Ctrl** or **Command** key and rotate the light to face the pillar object.



5. Switch to the **3D** (V) view and rotate the view so you can see both the Keylight and the pillar geometry.
6. Drag the y-axis handle (green) to move the light up above the pillar.
7. Press the **Ctrl** or **Command** key and rotate the light down to shine on the pillar.



That's the basic setup for lighting and surfaces, but there are other tools for more complex setups. Refer to Classic 3D Compositing for more information on the 3D lighting and surfacing tools.

Epilogue

In this chapter, you learned how to setup a 3D workspace for your composite, and how to work with 3D geometry and lighting. These are all prerequisite skills for more advanced topics, such as camera projections, match-moving, and set replacement.

Appendices

This section contains supplemental reference information that you may need when using Nuke.

Organization of the Section

The section consists of the following appendices:

- Appendix A: Preferences lists the preferences supported by Nuke.
- Appendix B: Keyboard Shortcuts lists the keyboard shortcuts you can use for quicker and easier access to Nuke's features. You can also open a list of keyboard shortcuts from the application by selecting **Help** > **Timeline** or **Comp Keyboard Shortcuts**.
- Appendix C: Supported File and Camera Formats lists the image, video, and audio file formats Nuke supports.
- Appendix D: External Software lists third party libraries used in Nuke.

Appendix A: Preferences

The Available Preference Settings

The **Preferences** dialog is divided into the following sections:

General	Settings for auto-saving and path substitutions.
Project Defaults	General project settings, and settings for color management.
Performance	Settings for caching, hardware, localization, and threads/processes.
Behaviors	Settings for start up, file handling, export options, scripting, node behaviors, and more.
Panels	Settings for the interface appearance, file browser, control panels, nodes, Viewers, script editors, and scopes.

General

General	
Autosave	
force project autosave after <300> seconds	Set the number of seconds after which to automatically save your project. Disable this by setting it to zero.
idle comp autosave after <5> seconds	Define how long (in seconds) Nuke waits before performing an automatic backup after you have left the system idle (that is, haven't used the mouse or the keyboard). If you set the value to 0, automatic backups are disabled.

General	
force comp autosave after <30> seconds	Define how long (in seconds) Nuke waits before performing an automatic backup regardless of whether the system is idle. If you set the value to 0, forced automatic backups are disabled.
autosave comp filename	Sets the file name of the autosaved project. If this is not set, it defaults to **[firstof[value root.name] [getenv NUKE_TEMP_DIR]/].autosave**
Path Substitutions	
path substitutions	Allows you to remap file paths in order to easily share projects across different operating systems. When the application encounters a file path, any text in the **Mac/Linux** column is replaced with the text in the **Windows** column, or vice versa. For example, if you enter **/Volumes/networkmount** in the **Mac/Linux** column and **Z:** in the **Windows** column: • On Mac and Linux, any file paths that start with **Z:** are converted to start with **/Volumes/networkmount**. • On Windows, any file paths that start with **/Volumes/networkmount** are converted to start with **Z:**. To be able to enter text in either column, you need to click on the **+** button below to add a row to the table. **Note:** Path substitution is invaluable for Sync Review sessions if you're using local source files rather than files from a central server. See Sync Review for more information.
+	Adds a row under **path substitutions**.
-	Deletes the selected row(s) under **path substitutions**.

Project Defaults

Note: You must restart the application for changes to **Project Defaults** preferences to be applied.

Channel Management	
Channel Management	
Channel Warning Threshold	Sets the total number of channels required in a script to trigger the Channel Warning. Nuke only supports 1023 uniquely named channels per script. The **Channel Count** is displayed in the bottom-right of the interface, next to the Localization Mode indicator.  **Note:** Nuke does not remove unused channels until you close and reopen a script, so the **Channel Count** does not decrease when you remove Read nodes from the Node Graph.

Color Management	
Default Color Management	
Default Color Management	Sets the default color management system to use in new projects.
OpenColorIO config	
OpenColorIO config file	Sets the OpenColorIO configuration to use, if you don't intend to use the **nuke-default** settings. If you select **custom** from the dropdown, enter the file path of the configuration file or click **Choose** to use the browser.  **Note:** Nuke also includes an environment variable method for setting a config file. See Environment Variables in the Nuke Online Help for more information.
Default Color Transforms	
working space	Sets the colorspace files should be converted to, on import, and from, during render - it's the colorspace used by Nuke under the hood.

Color Management	
viewer	Sets the default LUT applied to Viewers.
thumbnails	Sets the default LUT applied to thumbnails when ever they are generated.
Monitor Out	Sets the default LUT applied to external monitors.
8 bit files 16 bit files log files floating point files	Sets the default LUT applied to the specified ingested file type.
OCIO Roles	
Allow OCIO Roles in Colorspace Knobs	When enabled, OCIO roles are available in all colorspace controls.
Prioritize OCIO Roles	When enabled, OCIO roles are created in the main menu of cascading dropdowns, with **Colorspaces** in a sub-menu. When disabled, roles are demoted to a sub-menu called **Roles**. You can also control this preference using an environment variable. See Nuke Environment Variables for more information.
HDR (macOS only)	
Enable macOS HDR Color Profile (Display P3) (Beta)	When enabled, running under macOS with a screen capable of displaying values greater than 1 allows you to display HDR images in the Display P3 colorspace correctly.
Nuke Script Project Settings	
color management	Sets whether Nuke uses the LUTs read from the configuration specified or the **Nuke** native LUTs during export. Selecting **OCIO** makes the relevant OCIO LUTs available to the Read and Write nodes in scripts on a per project basis. All configurations except **nuke-default** automatically switch this control to **OCIO**.

General - these preferences only apply to new scripts and projects. To affect the current project, use the **Project Settings**.	
Project	
project directory	Sets the project directory used by new projects. You can change the project directory for the current project in the **Project Settings**.
Hrox Directory	Click to set the project directory to the location of the **.hrox** file using the **[python {nuke.script_directory()}]** expression.
export directory	Sets the directory used by timeline exports: • **Use Project Directory** - use the directory specified by the **project directory** preference. • **Use Custom Directory** - use the directory specified by the **custom export directory** control.
custom export directory	Sets the export directory used by new projects when the export directory control is set to **Use Custom Directory**.
Sequence	
output resolution	Use this to set the output resolution in the Timeline environment for new projects. By default, clips in the sequence are not reformatted to fit this format, they retain the source clip resolution. You can adjust the reformatting for new clips added to a sequence using the **Clip** settings below or by selecting clips in the timeline and adjusting the settings in the **Properties** tab.
frame rate	Select the frame rate for new projects in the Timeline environment.
start timecode	Use this to define the start timecode for new projects. For shots, this overrides the timecode defined in the media.
time display	You can use this to select the display format for times. You can select either **Timecode** or **Frames**.
drop frame	Use this to choose whether timecodes from this sequence are displayed in drop frame times or not. Drop Frame is a timecode display option that leaves out two frames from the 30 fps timecode sequence every minute (except every 10th minute) so that long running NTSC sequences are accurate to a real-time clock (NTSC frame rate is 3000/1001, or approximately 0.01% slower than 30fps).

General - these preferences only apply to new scripts and projects. To affect the current project, use the **Project Settings**.	
	Note: Enabling Drop Frame is a Timecode display feature only - the source media remains a continuous stream of frames.
Clip	
clip reformat	Sets how new clips added to a sequence are formatted. The default, **None**, displays shots at the source clip resolution, whereas **To Sequence Resolution** reformats the shot to the **output resolution** using the **resize type** control.
resize type	When **clip reformat** is **To Sequence Resolution**, determines how the shot is resized to fit the output resolution. This control is disabled when **clip reformat** is set to **None**.
center	When **clip reformat** is **To Sequence Resolution**, disabling **center** places the clip at the bottom-left of the **output resolution**, rather than the center.
Link bin and track item versions	When enabled, selecting a new version for a clip in a bin or a shot on the timeline updates all instances of that media. When disabled, you can version clips and shots independently, even if they reference the same media.
Poster Frame	
poster frame	Sets a preset poster frame for all new source clips or select **Custom** to select a relative frame offset as the poster frame: • **First** - new source clips display the first frame of the file on disk as the poster frame. • **Middle** - new source clips display the middle frame of the file(s) on disk as the poster frame. • **Last** - new source clips display the last frame of the file(s) on disk as the poster frame. • **Custom** - the poster frame number is derived from the number of frames in the clip starting at 0. Use the **frame offset** control to set the relative offset. For example, a **.dpx** sequence **myClip.####.dpx 1001-1500** can have a custom poster frame between **0** and **499**.

General - these preferences only apply to new scripts and projects. To affect the current project, use the **Project Settings**.	
frame offset	When poster frame is set to **Custom**, enter the relative **frame offset** from the first available frame in the file(s) on disk.

Sync Session	
Default Appearance	
name	Sets the name and color you want to display on participant sessions. Click the color swatch to open the color picker.
Default Behavior	
open sync panel on sync session startup	When enabled, automatically open the **Sync Session** panel when a session is started.
Project Saving	
save directory	Sets the default directory where all Sync Session projects are saved.

Views - these preferences only apply to new scripts and projects. To affect the current project, use the **Project Settings**.	
Views	
+/-	Click to add and remove views from the views list.
↑ ↓	Click to move views up and down the views list. Moving a view changes the position of the corresponding Viewer button.
View	Lists the views in the script or project and the color associated with the view when **Use color in UI** is enabled.
hero	Sets the principal view selected on script or project load.
Set up views for stereo	Click to automatically add **left** and **right** views to scripts or projects.

Views - these preferences only apply to new scripts and projects. To affect the current project, use the **Project Settings**.	
Use colors in UI?	When enabled, the colors specified in the views matrix are applied to the interface. For example, if you click **Set up views for stereo** and enable this control, any UI items representing the left and right views are colored red and green.

Performance

Caching	
Timeline Disk Caching	
directory path	The directory where all **.exr** files from timeline disk caching are stored by default. The caching directory should be on a local disk, preferably with the fastest access time available. It's also important to leave enough space for the maximum disk cache size (defined below).
limit to (GB)	This allows you to set the maximum amount of space (in GB) to use for the timeline disk cache. Set to zero for unlimited size. Values lower than zero, leave that amount of space free. The **currently in use** and **currently free** fields display how much free cache remains from the total specified. If this limit is reached during caching, a dialog displays options to free up disk space.
EXR compression	Sets the type of compression applied when writing files to the disk cache: • DWAB • Zip (1 scanline)  **Note: DWAB** compression produces smaller cache files more quickly, but can be lossy compared to **Zip** compression.
clear cache	**Clear All** removes all cached files from the root directory specified in the **directory path** control. A dialog displays a list of files for confirmation before the files are deleted.

Caching	
default disk caching mode	Sets the default dick caching mode for new sequences: • **Manual** - the default mode. When frames are cached, any changes made to soft effects on the cached frames are not automatically re-cached. • **Automatic** - when frames are cached, any changes to soft effects made after the initial cache operation automatically triggers a re-cache of the affected region of your cached timeline. This means that your timeline is always cached and up to date in the selected regions, regardless of changes made.  **Note:** You can override this preference in the **Sequence** panel using the **Disk Caching Mode** dropdown.
Comp Disk Caching	
temp directory	The comp disk cache saves all recent images displayed in the Viewer for fast playback. Using this control, you can specify where you want Nuke to save these images. Select a local disk (for example, C:/temp), preferably with the fastest access time available. It's also important to leave enough space for the maximum disk cache size (defined below).
comp disk cache size (GB)	Specifies the size of Nuke's disk cache, independent of the playback cache. It is the maximum amount of disk that Nuke can allocate for caching. When the limit is reached, Nuke attempts to free disk space before using any more. The environment variable NUKE_DISK_CACHE_GB overrides this setting.  **Note:** Setting this control to 0 GB also disables Nuke's RAM cache (the orange bar in the timeslider) as both caches rely on files written to disk.
rotopaint cache size (GB)	Specifies the size of Nuke's RotoPaint tile cache. It stores tiles for the output image of each RotoPaint node, enabling you to paint on top of existing RotoPaint without having to re-render the strokes underneath. If you run out of RotoPaint disk cache, response times may suffer when painting onto tiles containing lots of strokes.

Caching	
Memory Caching	
playback cache size (% of system RAM)	Specifies the percentage of system RAM used for the timeline Viewer playback cache. The entire amount is allocated, even if you've only got a few frames in the Viewer. Recently used frames are retained in the memory to avoid relying on the disk buffering system. The cache is freed when you switch to the compositing Viewer and reallocated when you switch back to the timeline Viewer. **Tip:** On low-end machines, minimizing this may improve application responsiveness at the expense of smooth playback.
free timeline playback RAM cache when switching to the node graph	When enabled, any frames cached to RAM (the white bar in the timeline Viewer) are discarded when you switch to the Node Graph within Nuke Studio, freeing the RAM for use in Nuke.  **Note:** When you switch back to the timeline, the cached files are re-cached, which can be time consuming.
comp cache size (% of system memory)	Specifies the percentage of system memory available for comp caching. It is independent of the playback cache and is the maximum amount of memory that Nuke can use for caching. When the limit is reached, Nuke attempts to free memory before using any more.
comp playback cache size (% of comp cache)	Specifies the percentage of the comp cache available for comp playback. This cache holds data displayed in Nuke's compositing Viewer, the result from the node tree you're viewing. Results from further up the tree are sometimes cached inside the comp cache as well, such as the output from a node which is needed by more than one downstream node.
comp paint cache size (% of comp cache)	Specifies the percentage of the comp cache available for comp paint. The **comp playback cache** and **comp paint cache** are limits for the total memory that is used, so it's possible to have a combined size greater than 100%. How the memory is shared between the two depends on what you're doing, and you won't necessarily be filling both up at the same rate. For example, if you're

Caching	
	doing a lot of paint work, you might want to allow the paint cache to fill up more than 50% of the available memory if the playback cache isn't using its full share.
Audio Waveforms	
waveform memory (MB)	Sets the amount of memory available for storing timeline audio waveforms.
Application in Background	
pause timeline Viewer when the application goes to the background	When enabled, pause timeline Viewer caching when the application is in the background.
clear timeline Viewer cache when the application goes to the background	When enabled, the timeline Viewer cache is cleared when the application goes into the background.  **Note:** This preference is only available when **pause timeline Viewer when the application goes to the background** is enabled.
pause comp Viewer when the application goes to the background	When enabled, pause comp Viewer caching when the application is in the background.
clear comp Viewer cache when the application goes to the background	When enabled, the comp Viewer cache is cleared when the application goes into the background.  **Note:** This preference is only available when **pause comp Viewer when the application goes to the background** is enabled.
Undo Caching	
undo history size	Allows you to set the amount of RAM, in MB, to use for the undo history. If this limit is exceeded, older items are discarded.
minimum undo events	Use this to set the amount of undo events. This setting always applies, even if it breaches the **undo history size** limit.

Expression	
Expressions Re-Evaluation	
Mode	Sets when expressions in Nuke scripts are re-evaluated: • **Always** - expressions in the script are re-evaluated after every change to the Node Graph. This can cause performance issues in large, expression-heavy scripts. • **Lazy** - expressions are only re-evaluated when required for GUI or render updates. This option can speed up interactive performance in large, expression-heavy scripts.

Hardware	
Audio	
audio device	The **audio device** control allows you to select an existing audio device for playout from a list of automatically detected devices. You can disable playout on a device by selecting **Disabled**.
GPU	
expand 3 to 4 channels	You can use this to expand images cached for playback from 3 to 4 color channels per pixel. Some graphics hardware performs better at loading images to video memory with 4 channels per pixel, than it does with 3. Enabling this option improves playback performance on such hardware, at the expense of reducing the number of frames that it's possible to cache. If you are seeing poor playback performance, enabling this option may help. However, if you are seeing acceptable playback performance with this option disabled, then leaving it disabled increases the number of frames that may be cached for smooth playback. **Note:** You must restart Nuke for this option to take effect.
enable vsync	When enabled, synchronize new timeline Viewer's playback frame rate with the refresh rate of the monitor. Enabling vsync limits the playback fps to the monitor refresh rate and may reduce performance.

Hardware	
	When disabled, timeline Viewer performance is unaffected but you may see tearing in timeline Viewers. **Note:** This preference is only shown in Nuke Studio and Hiero sessions and only affects new timeline Viewers. Close and reopen any existing Viewers after enabling this option.
GPU texture cache size (MB)	Use this to set the maximum amount of GPU memory to use for caching textures.
default blink device	You can use this to set the default blink device to use the CPU, or choose which GPU to use for GPU accelerated plug-ins, such as, ZDefocus, Convolve, Denoise, BlinkScript, Kronos, MotionBlur, and so on. Any changes to this setting only take effect after you have restarted the application.
disable blink cache	When enabled, nodes that support GPU caching pass on the GPU data to the last node that supports GPU caching and then transfer all the data back to the CPU at once. GPU caching is supported by all CaraVR nodes, the updated Bilinear and SphericalTransform nodes, and the BlinkScript node. When disabled, Nuke behaves in the same way as legacy versions.
blink cache size	Controls the size of the GPU memory and the percentage of that memory available for the Blink cache. **Note:** Increasing the percentage of memory used for caching does not necessarily decrease processing time, and can in some cases have a negative effect. The default 40% is preferred in most instances.
enable multi_GPU support	If have multiple GPUs of the same type installed, you can enable this preference to share work between the available GPUs for extra processing speed. This is a global preference and is applied to all GPU enabled nodes. **Note:** For more information on GPU support, see the Release Notes for your version of Nuke.

Localization	
System	
mode	Sets the overall localization mode: • **on** - checks for updates to source clips and Read nodes with **localization policy** set to **On** or **From auto-localize path** and localizes those files automatically. • **manual** - checks for updates to source clips and Read nodes with **localization policy** set to **On Demand** and prompts you to update them manually. • **off** - no source clips or Read nodes are localized, regardless of the their localization policy.  **Note:** The current localization **mode** is displayed in the status bar at the bottom-right of the interface.

read source files when localized files are out of date	When enabled, source clips and Read nodes referencing out of date localized files automatically revert to reading the entire sequence from the original source files. Source files are read in the following circumstances:
	• With Localization **mode** set to **on**:
	  • A localized source clip or Read node with **localization policy** set to **on demand** is detected to be out of date.
	  • A localized source clip or Read node with **localization policy** set to **on** or **from auto-localize path** is detected to be out of date and it is queuing to be automatically localized.
	• With Localization **mode** set to **manual**:
	  • A localized source clip or Read node with **localization policy** set to **on**, **on demand**, or **from auto-localize path** is detected to be out of date.
	
	When disabled, the out of date localized files are read until you update them manually.

hide out of date progress bar	When **read source files when localized files are out of date** is enabled, you can enable this control to hide the progress/status bar on Read nodes that are reading from the original source files.
pause localization on	When enabled, localization does not start automatically when you open a script

Localization	
script/project open	or project. Enabling this option can help to open scripts and projects more quickly.
Inputs	
localization policy	Sets the localization policy for all new source clips and Read nodes: • **on** - always localize source clips and Read nodes with this policy automatically. • **from auto-localize path** - localize these source clips and Read nodes automatically if they reside in the **auto-localize from** directory. • **on demand** - only localize these source clips and Read nodes when you manually update them. • **off** - never localize these source clips or Read nodes.
Paths	
auto-localize from	Enter the location of the files you need automatically localized, unless otherwise specified in the Read node's **cache locally** control or in the bin right-click, **Localization Policy** menu. Commonly this would be your remote working folder. If you leave this field blank, automatic local file caching doesn't take place.
localize to	Enter the file path where all the localized files are automatically stored. Localizing files allows for faster reloading for files that are stored in a location that is slow to access (such as a network drive). On Windows, files saved to the **localize to** directory replace **** (double back slashes) and **:** (colon drive signifiers) with underscores so that the file path works as expected between operating systems. For example: **\\windowspath\to\my\network\file.dpx** is saved as __ **windowspath\to\my\network\file.dpx** **t:\my\network\path\file.dpx** is saved as **t_\my\network\path\file.dpx**
Storage	
limit to (GB)	This allows you to set the maximum amount of space (in GB) to use for localized files. Set to zero for unlimited size. Values lower than zero, leave that amount of space free. The **currently in use** and **currently free** fields display how much free storage remains from the total specified.

Localization	
	If this limit is reached during localization, a dialog displays options to free up storage space.
Network	
check for updated files every	When files are localized, specifies the time interval (in minutes) before Nuke checks for updated versions of the files.
Appearance	
progress bar	Sets the colors used to represent the localization states of source clips and Read nodes.

Threads/Processes	
Playback	
default number of threads per reader	Sets the number of threads to use per reader. If your source files are located on high performance local drives, increasing the number threads can significantly improve read times. CPU intensive operations, such as **.jpg** decoding, can also be improved by increasing the number of threads per reader.
override number of threads per reader	Allows you to override the default number of decode threads used, dependent on file format. Use the plus button to add an entry to the table and then select the file format using the dropdown menu. Double click the Number of threads column to set the required number of decode threads for that format.
OpenEXR helper threads to use	Sets the number of helper threads to use for OpenEXR only. The default, zero, automatically sets the number of helper threads used.
Rendering	
render using frame server (Nuke)	When enabled, the Frame Server is always used for rendering.

Threads/Processes	
	 **Note:** Local Frame Server processes use ports 5558-5662.
frame server render timeout	Allows you to increase the number of minutes a render process can stay unresponsive before ending. If you're experiencing **Render application timed out** messages with process-heavy scripts, you can try increasing this value.
focus background renders	When enabled, rendering using the Frame Server automatically opens the **Background Renders** panel, or if it is already open, shifts focus to the panel.
frame server processes to run	Set the number of worker Nuke processes to run for the frame server.  **Note:** You must restart Nuke for this setting to take effect.
export renders	You can select from several render options: • **limit renderer (more responsive ui)** – Select this option to make the user interface more responsive during transcodes. It tells Nuke to use 2 threads to transcode and to use 25% of RAM to cache. Using this option is likely to result in slower transcodes. • **no renderer limits (fastest transcoding)** – Select this option to ensure that transcodes happen as quickly as possible. This option may result in a less responsive user interface during transcodes. • **customize render limits** – Select this option to manually configure the **number of threads** used and **cache memory** available when transcoding files.  **Note:** You must restart Nuke for this setting to take effect.
number of threads	Sets the number of threads that Nuke uses when transcoding. Lower numbers allow the Timeline environment's interface to be more responsive. Higher numbers allow faster transcodes. This setting is passed to Nuke using the **-m** option.
cache memory (GB)	Use this to set the number of gigabytes of RAM that Nuke uses for its cache settings. Lower numbers may improve the Timeline environment's interface responsiveness, while higher numbers may improve the speed of the

Threads/Processes	
	transcodes. This setting is passed to Nuke with the **-c** option.
background renders	Sets when background renders occur: • **don't auto-start background renders** – Comps on the timeline are not rendered in the background automatically. • **start background renders on Comp save** – Comps on the timeline are rendered in the background automatically when they are saved. • **start background renders on Comp create, Comp save and Comp version change** – Comps on the timeline are rendered in the background automatically when first created, when saved, and when a new version is selected.
Downsize Filtering	
8-bit images 10-, 12- and 16-bit integer images 16-bit float images 32-bit images	Customizes the downsize filtering behavior by bit-depth. The default (1x) retains the original image size. You can select 2x to halve the image size, or 4x to quarter the image size. The Viewer image quality dropdown affects the decode rate and resolution of clips displayed in the Viewer. Lower resolutions decode faster and vice versa.

Behaviors

Documentation	
documentation source	Sets the help source for the **Properties** ? button: • **local** – use Nuke's built-in HTML help server. Nuke's local help server also searches the NUKE_PATH, .nuke, and .nuke/Documentation directories for HTML files with the same name as the requested node, such as Blur.html. • **foundry** – This uses Foundry's Online Help, the most up-to-date version of the documentation. • **custom** – Select this to point to your own help server.
auto port	When enabled, assign a free port automatically.

Documentation	
local port	Specify a local documentation server port manually. This is usually >= 1024. You can also set this to 0 to automatically assign the port.
range	Specify a range of ports to attempt with the **local** documentation server.

File Handling	
scan for file sequence range on drop into Bin view	When enabled, identify and import the file range of media that is dropped into the bin. When disabled, no range is detected and only a single frame is ingested. (This does not affect container formats, such **.mov** and **.r3d**.)
automatically rescan versions when moving off end of the version list	When enabled, incrementing a source clip or shot's version past the end of the previously discovered versions list, forces a rescan to update the versions list. See Using Versions for more information.
frame number style	Sets the sequence display mode to be used in the file browser.
assume fixed width frame numbers in file sequences	When enabled, assume frames have a fixed width number. With this selected, frame numbers need to be padded with zeros to a fixed length, otherwise frame numbers without preceding zeros are assumed to belong to sequences with no padding. This is important as the sequence identifier specifies a unique file name for each and every frame. For example:

Files	**With fixed frame width assumed**	**With fixed frame width NOT assumed**
sequence 1.18.exr, sequence 1.19.exr, and sequence 1.20.exr	sequence 1.##.exr / sequence 1.%02d.exr	sequence 1.#.exr / sequence 1.%d.exr

alembic files	
always load abc files as all-in-one	When enabled, all **.abc** files are imported as a single node, without displaying the Alembic import scenegraph.
3D geometry files	
block geoimport popup	Block the geoimpot popup from appearing when loading a geo file.

Nodes	
new Merge nodes connect A input	When enabled, inserting a new Merge node automatically connects the **A** input.
autokey roto shapes	When enabled, keyframes are added automatically to Roto shapes when they are adjusted.
when Viewer is closed delete its node	When enabled, Viewer nodes are deleted when you close the associated Viewer.
Double click the Viewer node to open the Viewer settings	Open the Viewer settings by doing any of the following: • Double-click on the Viewer node in the Node Graph. • FN_CTRLKEYNAME+click on the Viewer node in the Node Graph. • Select the Viewer node in the Node Graph and press Return. • To open the Viewer settings in a floating window, FN_CTRLKEYNAME+double-click or FN_CTRLKEYNAME+Alt+dick on the Viewer node.
Tab Search Menu	
Weighting	When enabled, nodes in the Tab menu are weighted so that more commonly used nodes appear at the top of the list. When disabled, the node list is ordered as it appears in the nodes toolbar or alphabetically if you start typing.
Favorites	When enabled, nodes that you have marked as favorite in the Tab menu appear at the top of the list. When disabled, the node list is ordered as it appears in the nodes toolbar or alphabetically if you start typing.
Clear Weighting	Click to reset any weighting information collected by Nuke.
Clear Favorites	Click to reset any nodes marked as favorite.

OFX Plug-ins	
allow trial mode in OFX plugins	When enabled, OFX plug-ins that offer a trial mode render in that mode, if a license cannot be found. When disabled, OFX plug-ins that can't get a license appear in an error state.

Positions	
show menus with previous item under the cursor	When enabled, opening contextual menus positions them with the most recently used item under the pointer.

Scripting	
script command dialog defaults to Tcl	When enabled, the dialog that appears when you press **X** in the Node Graph defaults to Tcl, rather than Python.

Startup	
startup workspace	Sets which workspace to display on startup. You can choose from **Compositing**, **Conforming**, **Editing**, **Finishing**, **Reviewing**, and **Timeline**. You can also choose to save a customized workspace, which would also be available from this list.
show splash screen at startup	When enabled, display the splash screen on startup.
show startup dialog	When enabled, display the dialog on startup.
restore workspace when opening projects	When enabled, restore the selected saved workspace when opening projects.

Timecode	
R3D file timecode	Sets the source timecode for RED files. You can choose from **Default From File**, **Absolute Timecode**, or **Edge Timecode**.
other media timecode	Sets the timecode source for file-per-frame media (such as **.dpx**). You can choose from **File Header** or **Frame Number**. If **File Header** is selected and a timecode exists, then the timecode is used. Otherwise it defaults back to using the frame number from the file name.
max valid timebase (fps)	Sets the maximum image header timebase above which the value is clamped. Image files are often created with application specific timebase values in the header description. This can lead to reading in spuriously high frame rates and the clamp aims to prevent this from happening. If your clips do have extremely high frame rates, increase this value as necessary to avoid clamping.
EDL style spreadsheet timecodes	When disabled, the srcOut and dstOut values in the spreadsheet use the film convention, representing the last frame of the cut. When enabled, the srcOut and dstOut values in the spreadsheet use the video convention, representing the frame directly after the cut.

UI Scaling	
 **Note:** You must restart the application to apply changes to **UI Scaling** preferences.	
UI Scaling Mode	Sets the scaling mode for the UI on Windows and Linux. macOS handles UI scaling automatically. • **off** - no scaling is applied to the UI, regardless of screen resolution. • **auto** - the UI attempts to scale appropriately dependent on screen resolution. • **all** - enables the **Scale All** dropdown, which allows you to scale all screens by the selected factor. • **per display** - enables the **Per Display** table, which allows you to set the scaling applied per screen.

UI Scaling	
Scale All	When **UI Scaling Mode** is set to **all**, allows you to scale all screens by the selected factor.
Per Display	When **UI Scaling Mode** is set to **per display**, allows you to set the scaling applied per screen.

Panels

Appearance	
Font	Change the type, weight, angle, and size of the font used on Nuke's user interface.
UI Colors - right-click on any color button and select **Set color to default** to revert changes.	
Background	Change the background color of most user interface elements (menus, toolbars, panes, properties panels, Viewers, and pop-up dialogs).
Base	Change the color of input fields, the input pane of the Script Editor, and the left side of the Curve Editor.
Highlight	Change the color of the highlighting that appears when you hover the cursor over a control, select a file or folder in the File Browser, or scrub to a new frame on the timeline.
Highlighted Text	Change the color of any highlighted text (for example, text you select in node properties).
Label	Change the color of labels and text on the application interface. Note that this does not set the color of the labels on nodes in the Node Graph.
Button	Change the color of buttons and dropdown menus.
Animated	Change the color that indicates a control has been animated.
Keyframe	Change the color that indicates a keyframe has been set.
Predicted	Change the color of the keyframes on the frame slider.

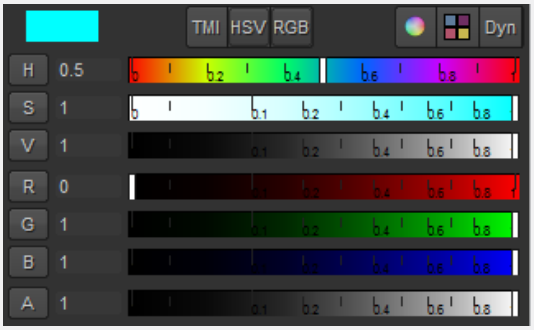
Appearance	
Disk cached frames	Change the color of the disk cached frames on the Viewer timeline.
RAM cached frames	Change the color of the RAM cached frames on the Viewer timeline.
Playhead	Change the color of the frame marker on the Viewer timeline.
In/Out Markers	Change the color of the in and out frame markers on the Viewer timeline.
Clip Annotation Markers	Change the color of the clip annotation markers.
Sequence Annotation Markers	Change the color of the sequence annotation markers.
Curve Editor / Dope Sheet - right-click on any color button and select **Set color to default** to revert changes.	
no. of curves visible	Sets the maximum number of curves visible in the Curve Editor.
background	Change the background color of the Dope Sheet tab.
unselected key	Change the color used for an unselected key on the Dope Sheet.
part-selected key	Change the color used for a part-selected key on the Dope Sheet.
selected key	Change the color used for a selected key on the Dope Sheet.
timeline	Change the color used for the timeline on the Dope Sheet.
control text	Change the color used for the control text on the Dope Sheet. These indicate the frame number of a key when you select one.
control text shadow	Change the color used for the shadow of the control text on the Dope Sheet.
time label	Change the color used for the time labels on the Dope Sheet. These indicate frame numbers.
current frame	Change the color used for the current frame on the Dope Sheet. This is a vertical line that indicates the current frame on the timeline.
project frame range	Change the color used for the project frame range on the Dope Sheet. These are two vertical lines indicate your frame range.

Contact Sheet	
show border	Toggle the shot border on or off.
border color	Change the border color of shot items.
Pick Mode Fill Color	Change the fill color of shots in the sequence viewer.
Pick Mode Border Color	Change the border color of shots in the sequence viewer.
default resolution	Defines the default output resolution of the Contact Sheet.
default max row count	Sets the default maximum number of rows on a Contact Sheet Page.
default max column count	Sets the default maximum number of columns on a Contact Sheet Page.

Control Panels	
max nodes in properties bin	Use this to set the maximum number of panels that can be open in the **Properties** pane.
reopen acts like new panel	When this is enabled, double-clicking a node that has been open before, places the panel in the same place as a new panel. If this is disabled, the panel appears in its previous position.
expand / collapse panels in Properties bin to match selection	If this is enabled, the node selection in the Node Graph determines which control panels are expanded (all unselected nodes automatically have their panels collapsed). This does not apply to floating control panels.
input button action	Use this to define node input button action, which is located in the top-left of the node properties panel. For example, you can set this to center a selected input of a node in the Node Graph.
max items channel menu	Use this to set the maximum number of channels or layers that are displayed in a single sub-menu of the main channel control.
Color Panel	
color picker button	Sets the type of color picker displayed when you click a color picker button in

Control Panels	
opens	the properties panel: • **in-panel color picker** - opens a color wheel and sliders in the properties panel. • **floating color picker** - opens a color wheel and sliders in a floating panel. **Tip:** Holding **Ctrl/Cmd** and clicking the color picker button opens the alternate color picker to the one specified in the **Preferences**.

File Browser	
start file browser from most recently used directory	When enabled, new file browsers open at the last location used. When disabled, new file browsers open at the current working directory.

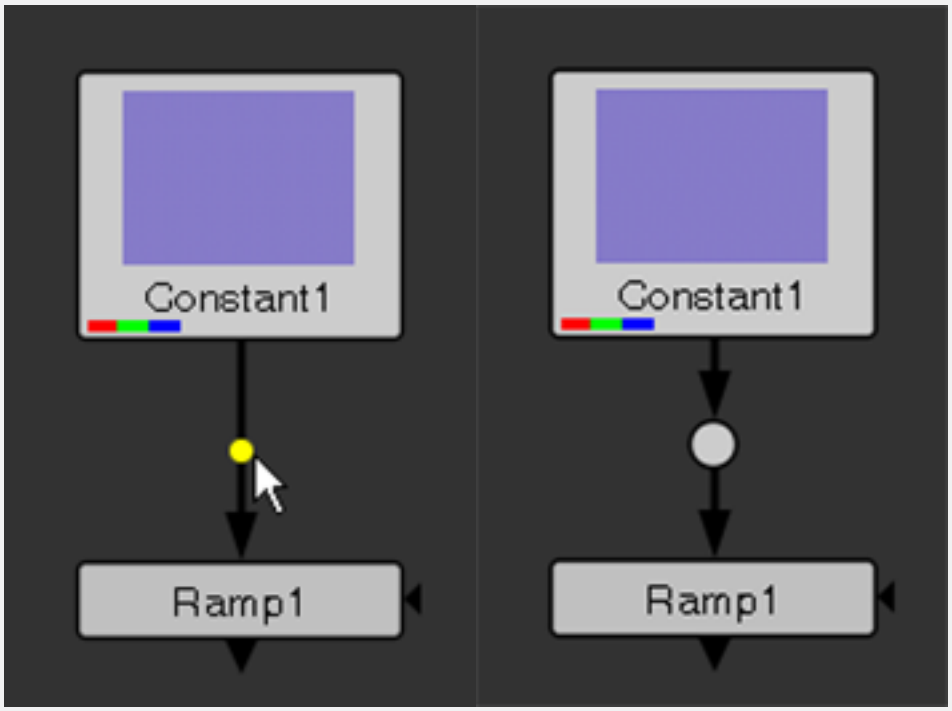
Node Colors	
autocolor	Deselect this checkbox to ignore individual node/soft effect color settings and, instead always use the **All Others** color.
Shade Nodes	Select this checkbox to apply a slight gradient shading to nodes.
<node name or type>	The nodes listed here have been given a default color. You can change this by clicking the assigned color to open the color menu, and selecting a new one. 

Node Colors	
All Others	Use this to select the color to use as default for all nodes not otherwise specified above, or all nodes if **autocolor** is disabled.
Text	Use this to select the color used for node label text.
Selected	Use this to choose the highlight color applied to any selected nodes.
Selected Input	Use this to choose the color used for node inputs' label text when selected.
GL Color	Use this to select the color to draw nodes' Viewer controls in. For example, the Ramp node's gradient control.

Node Graph	
autolabel	When disabled, nodes only show the filename or node name - most of the code in **autolabel.py** is disregarded. For example, the Blur node does not display the affected channels when this control is disabled.
highlight running operators	When enabled, highlight any nodes whose output is currently being calculated.
postage stamp mode	When displaying a thumbnail render of the node's output on its surface (either using the PostageStamp node or the **postage stamp** control on the **Node** tab of each node), you can select one of two modes: • **Current frame** - The postage stamp is always updated to match the current frame. • **Static frame** - The postage stamp displays a fixed frame. To specify the frame to use, open the node's controls, go to the **Node** tab, and adjust **static frame**.  **Note:** If the frame number you use is outside the frame range for the node, it is clamped to the first or last frame in the range accordingly.
node name background	When a node is selected and the node's name is too long to fit inside the node, a background is drawn behind the name to improve legibility. Use this control to set the intensity of the background, from 0 (no background) to 1 (fully opaque

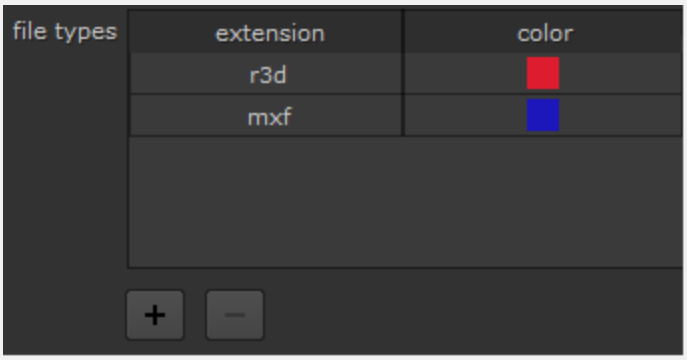
Node Graph	
	background). 
label font	Sets the font for labels. You can use the **B** and *I*, to the right of the font dropdown, to bold or italisize the selected label font.
tile size (WxH)	Sets the size of nodes in the Node Graph using the width and height.
snap to node	When enabled, nodes snap into positions (while dragging them) that line up horizontally and vertically with their inputs and outputs.
grid size (WxH)	Sets the grid size using width and height.
snap to grid	When enabled, nodes snap into positions (while dragging them) that line them up with the grid.
show grid	When enabled, display the grid using the overlay color.
snap threshold	When **snap to grid** is enabled, use this to set the maximum number of pixels to jump by, when snapping nodes to the grid or other nodes.
Colors	
Node Graph	Sets the color of the Node Graph background.
Overlay	Sets the color of the selection marquee when you lasso nodes.
Elbow	Sets the color of the dots created when you 'elbow' a connection pipe by holding **Ctrl**/**Cmd**.
Bounding Box Warning	
Highlight	When the bounding box warning is enabled, sets the color of the warning displayed on nodes that exceed the threshold.
Line	When the bounding box warning is enabled, sets the color of the dotted line around the **Highlight** color on nodes that exceed the threshold.
enable	When enabled, nodes that force the bounding box past the format size are

Node Graph	
	marked in the Node Graph: • **red rectangle with dotted stroke** - the indicated node creates a bounding box greater than the format. • **dotted stroke without the red rectangle** - the bounding box size is greater than the format at the indicated node, but the bounding box size has been set by an upstream node.
threshold %	Sets the threshold past which the bounding box warning is displayed.
Arrow	
<directional arrows>	You can select one of the directional arrows (up, down, left, or right) to change the color it is displayed in. Click the arrow to open the color menu and select a new color.
deep arrows	Sets the color of arrows carrying deep data. Click the button to open the color menu and select a new color.
expression arrows	Sets the color of expression arrows, if enabled. Select the **enable** checkbox to display expression arrows.
link knob arrows	Sets the color of arrows indicating that a node contains linked knobs, that is, knobs that are in use in another node's **Properties** panel. Select the **enable** checkbox to display link knob arrows.
clone arrows	Sets the color of clone arrows, if enabled. Select the **enable** checkbox to display clone arrows.
<arrow components>	Sets arrow and arrow head, lengths and widths. You can also use the numeric field next to each component to enter a specific value.
allow picking of connected arrow heads	Select this checkbox to be able to pick up and move connected arrow heads.
allow picking of arrow elbows to create Dots	When enabled, press **Ctrl** (**Cmd** on a Mac) on the Node Graph to display yellow “elbows” on the Node Graph arrows and then click on these to insert Dot nodes.

Node Graph	
	 If you **Ctrl**/**Cmd**+**Shift**+click on an elbow, the new Dot node is branched off to a new arrow rather than inserted in the existing arrow. When disabled, adding Dot nodes in this manner is not possible.
drag-to-insert only work near middle of arrows	Select this checkbox to restrict the arrow hotspot, for inserting nodes, to the middle of the arrow.
size of dots	Use this slider to set the size of the Dot nodes. You can also enter a specific value in the numeric field to the left of the slider.
Disconnect Nodes	
shake to disconnect nodes	When enabled, click and hold a node or selection of nodes, and then shake the pointer to disconnect the selected nodes.
shake sensitivity	Determines the sensitivity of the **shake to disconnect nodes** feature. Higher values mean less shaking to disconnect nodes.
Group View	

Node Graph	
Global Group View Disable	Turns off Group View across your whole project. When this is checked, the ability to show group views will be disabled and all currently visible group views will become invisible.
Zoom	
Middle Click Zooms to Fit	When this is checked, clicking the middle mouse button on the Node Graph will zoom the Node Graph to fit all Nodes.

Project Items	
shade project items	When enabled, additional shading is applied to source clips and shots in the Project bin and timeline.
Item Labels	
project bin	Click to change the color of labels in the **Project** and timeline panels. A color wheel displays allowing you to select the required color.
timeline	
auto-adjust contrast	When enabled, label colors are automatically adjusted if a potential color-clash is detected.
Item States	
offline	Click the buttons to change the color of shots and comps in the timeline panel. A color wheel displays allowing you to select the required color.
error	
freeze	
comp not rendered	
comp out of date	
comp rendered	
Item Colors	
display in project panel	When enabled, the specified item colors, or defaults, are displayed in the **Project** panel.

Project Items	
display in sequence panel	When enabled, the specified item colors, or defaults, are displayed in the timeline panel.
spreadsheet color rows	When enabled, the specified item colors applied to rows in the spreadsheet.
project bin sequence source audio comp	Click the buttons to change the color of items in the **Project** and timeline panels. A color wheel displays allowing you to select the required color.
file types	Allows you to add custom color-coding by file extension. Click the + button and then select a file type from the **extension** dropdown. Double-click the color swatch to display a color wheel allowing you to select the required color. Any source clip or shot with that extension is then colored accordingly in the interface. 
Item Behavior	
Keep the source folder structure when pasting between projects	When enabled, pasting bin items between projects creates the same folder structure as the original project. When disabled, pasting bin items between projects adds the items to the root folder of the project.

Scopes	
black point	Use the slider or the entry box to select the black point value.
white point	Use the slider or the entry box to select the white point value.
luma/chroma encoding	Use this to select the video standard to use when converting from RGB to luma and chroma for scope display.
Include viewer color transforms	Select this checkbox to include applied Viewer color transforms (gain, gamma, and LUT) in scope data. If this checkbox is disabled, all Viewer transforms are ignored.  **Note:** If disabled, rendering may become slow as image calculation may be needed.
Force full frame	Select this checkbox so that the Viewer always requests full frame data when a scope is displaying data for that Viewer. If this checkbox is disabled, the scopes only display data for the current area requested by the Viewer, rather than the whole image.

Script Editor	
font	Use this to select the font to use in the Script Editor.  **Note:** This control also changes the font in the BlinkScript Kernel Source field.
indent	You can use this control to set the indent value to use when scripting.
save and restore script editor history	Disable this checkbox if you prefer that the contents of the Script Editor is not saved and restored between sessions of Nuke.
echo python comments to output window	Select this checkbox to print any Python commands executed by Nuke itself to the Script Editor output window.

Script Editor	
	Note: Note that not everything you do results in a command being echoed, because many of Nuke's internal functions are not executed using Python commands.
clear input window on successful script execution	Disable this checkbox if you want the most recent script to remain in the input window after execution.
BlinkScript	
Default Library Files Directory	Set the default path to look for Blink library files.

Timeline	
Highlight clones of selected clone(s)	When enabled, selecting a clone in the timeline highlights all clones associated with the selected clone.
Playhead	
Show frame end marker	When enabled, an extra line is drawn on the timeline to the right of the playhead, indicating the end of the current frame.
Show buffer labels	When enabled, A/B labels are displayed on the buffer indicators at the playhead if the buffers are enabled and have shots assigned.
Visible range follows playhead	When enabled, the timeline scrolls with the playhead, constantly updating the view. When disabled, the playhead is allowed to move off screen.
Use the Shift key to insert a new track between existing tracks while dragging	Disable this to drag to insert a new track without pressing the Shift key. Inserting a new track on the top of the timeline does not require the Shift key.
Audio Tracks	
half waveforms	When enabled, audio tracks on the timeline display only the rectified waveform. When disabled, the full waveform is displayed.

Timeline	
Selection	
Auto-selection mode enabled	Toggles whether the auto-selection feature is on or off by default. Setting an environment variable NUKE_DISABLE_SOFT_SELECTION overrides this preference.

Viewer (Comp)	
Defaults	
default renderer	Currently, only one Hydra delegate is available in Nuke, the **GL** renderer, which handles the USD-based geometry. Classic 3D system geometry is still drawn with Nuke's classic renderer, regardless of which Hydra render delegate you choose. In future releases, more render delegates will be made available.
new Viewers go to own window	When enabled, new Viewers are placed in their own window instead of docking in existing windows.
prevent auto zoom for new Viewers	When enabled, new Viewers are not auto-zoomed to the current Viewer's zoom level.
apply LUT to color channels only	When enabled, look-up tables (LUTs) are only applied to the red, green, and blue channels. When disabled, LUTs are applied to all channels.
viewer buffer bit depth (byte)	Use this to select the OpenGL buffer depth, and enable use of the GPU for the Viewer process and input process. • **byte** – Converts to 8-bit with error diffusion. • **half-float** – Converts to 16-bit (half) float. In this mode, the GPU can be used to apply Viewer effects like gamma and the LUT in a Viewer process. • **float** – Uses a full 32-bit floating point texture. This may be slow on selected cards. In this mode, the GPU can be used to apply Viewer effects like gamma and the LUT in a Viewer process. You can choose a default value for this setting in the **Preferences**, or by using

Viewer (Comp)	
	knobDefault() in a startup script.
use GPU for Viewer when possible	When this is checked, the Viewer applies its effects (such as the Viewer Process node) in the GPU when possible. However, in some cases, like when monitor out is enabled or **gl buffer depth** is set to **byte** in the Viewer settings, effects (such as gain and gamma) must still be computed on the CPU.
use GPU for inputs when possible	Normally, the Viewer only attempts to run its own effects (such as the Viewer Process node) on the GPU. However, when this is checked, any nodes connected to the Viewer are also computed on the GPU when possible. Note that this cannot be done for all nodes because not all nodes have a GPU implementation. If nodes are computed on the GPU, the color values displayed in the Viewer are inaccurate. This is because they show the color from the last node computed in the CPU prior to transferring the image into the graphics card.
disable GPU Viewer dithering	Check this to disable dithering in the Viewer when you're using half-float depth. Uncheck to allow dithering at all times.
dithering algorithm	Sets the frequency of dithering applied to GPU enabled compositing Viewers, so that you can be sure you're working in a consistent environment: • **high frequency** - the default setting, applies a high frequency dithering algorithm in screen space, after any viewer rescaling is applied. As you pan and zoom the Viewer, dithered pixels remain the same scale. • **low frequency** - applies a low frequency dithering algorithm in image space, before any viewer rescaling is applied. As you zoom in the Viewer, dithered pixels get bigger. • **no dithering** - disables GPU dithering.
 **Note:** When the Viewer AB mode is changed to wipe or stack, the state of the GPU acceleration controls is stored, GPU acceleration is turned off, and GPU dithering is disabled. When the Viewer AB mode is changed back to default, GPU acceleration is re-enabled and the state of the GPU dithering controls is restored.	
no incomplete stereo for new viewers	When this is checked, the Viewer only displays one view of a stereo project until both views have been rendered. This is to prevent disorienting effects when watching the results.

Viewer (Comp)	
	When this is not checked, the Viewer displays both stereo views, even if the render of either is incomplete.
show hardware stereo warning	When enabled, a warning message is displayed in the Viewer when you switch to **OpenGL Stereo mode**.
Settings	
flip stereo interlaced views	Select this checkbox to flip the left and right stereo views when the Viewer is set to **Stereo Mode** > **Interlaced** in the right-click/context menu.
texture size	Use this to define the OpenGL texture size in Nuke's Viewers. It affects the 2D Viewer preview texture size, such as the preview displayed when dragging transform or SplineWarp handles, and all textures in the 3D Viewer. You can choose from **256x256**, **512x512**, **1024x1024**, and **2048x2048**.  **Note:** The larger the texture size, the sharper it is, but larger texture sizes need more time and memory. Setting this preference to a high value can impact 3D Viewer performance.
texture mode	Use this to choose how textures are handled in the Viewer. You can choose from the following options: • **Multiframe** – Cache each frame of a texture; this gives animated textures and fast playback after each frame is cached, but uses a lot more memory. It also enables you to have multiple frames of a texture visible at one time (for example, on particles). • **Classic** – Textures are not updated in the Viewer during playback; this gives the fastest playback speed.

Viewer (Monitor Out)	
use video legal range for monitor out	When enabled, automatically limit monitor out to the legal range when swapping between the Timeline environment and Compositing environment.
always keep the floating window on top	When enabled, Nuke's internal floating monitor out window always remains on top of other open application windows.

Viewer (Monitor Out)	
Enable support for HMD devices	When enabled, VR headsets such as the Oculus Rift and HTC Vive are added to the **Monitor Out** > **Device** dropdown. Choose the required headset from the dropdown to pass the monitor out to the selected device. See Reviewing on VR Headsets (HMD) for information on supported headsets.
Enable global monitor out mode	Makes all viewers share the same monitor output, including node graph and timeline viewers (Nuke Studio). Requires a restart to take effect.
Mouse Pointer Scale	Sets the size of the pointer displayed on the monitor when the **Show Pointer** control is enabled for Nuke Studio monitor out.
Background	
[background]	Sets the background color used to fill Nuke's floating window when the output is not full-frame: • Black • Gray - use the slider to the right to set the brightness of the plain background. • Checkerboard - use the slider to the right to set the brightness of the checkerboard background.
NDI	
Enable NDI	When enabled, you can stream Viewer output over the local network from the Monitor Out controls using NDI from NewTek.

Viewer (Sequence) or (Flipbook)	
default flipbook	Sets the default flipbook mode, Nuke's **Flipbook Viewer** or **HeiroPlayer**. • **Flipbook Viewer** opens a new Viewer inside Nuke containing the flipbooked content. • **HieroPlayer** launches an instance of the application containing the flipbooked content, but requires either a Nuke Studio or HieroPlayer license.
playback mode	Use this to set the Viewer playback mode: • **Play All Frames** - the default setting, plays all frames in real-time (dependent on hardware).

Viewer (Sequence) or (Flipbook)	
	• **Skip Frames** - plays frames in real-time skipping where necessary to maintain the frame rate. • **Play All Frames, Buffering** - plays all frames by buffering and playing frames back as they become available.
guides	You can use this to choose to show overlays in the image area. You can choose from: • **Title Safe** – Indicates where text should be entered to be visible. • **Action Safe** – Indicates the area in which to place actions so that they are visible. • **Format** – Displays the size of the format over the Viewer.
fullscreen display	Use this to select which display to use for fullscreen mode. This setting takes effect the next time fullscreen mode is entered.
see through missing media	Select this checkbox to see through missing media in the timeline, displaying the first displayable media in the underlying tracks.
background	Use this to select the Viewer background. You can select black, or gray (using the slider to determine the grayscale), or checkerboard (using the slider to determine the brightness).
frame increment	Use this to set the default number of frames skipped by the Viewer skip controls, and the timeline **Nudge More** commands.
filtering mode	Use this to determine the filtering used during rendering in the Timeline environment. You can select **Auto**, **Nearest neighbour**, or **Linear**. **Auto** uses ths same automatic selection as in the Compositing environment. This does not affect exports or rendering in the Compositing environment.
Audio	
default latency adjustment (ms)	Use this to adjust the default timing offset (in milliseconds) between audio and video to apply to new Viewers. Positive values make audio play earlier relative to video; negative values make audio play later. To convert from video frames to ms, divide 1000 ms by the video frame rate. For example: • at 25fps, a video frame is 1000/25 = 40ms, or • a 1.5 video frame delay = 1.5 * 40ms = 60ms.
default volume	Use the slider or numeric field to set the default volume.

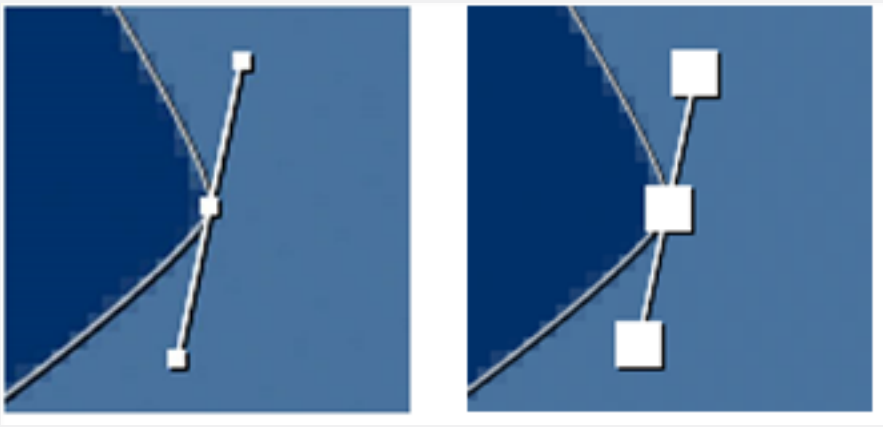
Viewer Handles	
Colors	
2D	• **bg** – Change the background color of the 2D Viewer. • **fg** – Change the color of borders and text in the 2D Viewer.
3D	• **bg** – Change the background color of the 3D Viewer. • **fg** – Change the color of borders and text in the 3D Viewer. • **sel** –Change the color of the selected vertices or faces of an object in the 3D viewer.
Splines	
line width	Sets the width of splines drawn in RotoPaint and SplineWarp.
draw shadows	When enabled, shadows are added to splines drawn in RotoPaint and SplineWarp to make the overlay easier to see.
general	• **Expression color** – Change the default color of the control points when an expression is set. • **Focus color** – Change the default color of the control points when focused.
roto	• **Points** - Change the default color of the points on RotoPaint shapes and strokes. • **Curves** - Change the default color of the rotoshape and stroke curves in RotoPaint and Roto. • **Transform** - Change the default color of the RotoPaint transform jack. • **Locked** - Change the default color of RotoPaint points and curves when locked or otherwise unmodifiable.
splinewarp	• **A Sourcecolor** - Change the default color of SplineWarp's A source curves. • **B Sourcecolor** - Change the default color of SplineWarp's B source curves. • **draw source stippled** - Check to change source curves from solid to stippled. • **A Destinationcolor** - Change the default color of SplineWarp's A destination curves. • **B Destinationcolor** - Change the default color of SplineWarp's B destination curves. • **draw destination stippled** - Check to change destination curves from solid to

Viewer Handles	
	stippled. • **Correspondencecolor** - Change the default color of the SplineWarp correspondence lines. • **Boundarycolor** - Change the default color of the SplineWarp boundary curves. • **Hardboundarycolor** - Change the default color of the SplineWarp hard boundary curves.



Note: To set a color back to its default, right-click on the button and select **Set Color to Default**.

Controls	
middle button pans	Check this to use the middle-mouse button to pan in the Viewer, Node Graph, the Curve Editor, and the Dope Sheet.
left-middle to zoom	Check this to use the left and the middle-mouse button together to zoom in the Viewer, Node Graph, the Curve Editor, and the Dope Sheet.
show transform preview	Check this to disable the OpenGL preview when manipulating the handles of 2D nodes, such as Transform and CornerPin.
3D hotkeys	Select the hotkeys control scheme you want to use in the 3D Viewer: Katana/Maya, Houdini/Flame, Blender, or Nuke Classic.
3D navigation	Select the navigation control scheme you want to use in the 3D Viewer: Nuke, Maya, Houdini, Lightwave, or Modo.
3D transform handle size	Sets the size of the transform handles in Nuke's 3D Viewer.
3D handle size	Adjust the size of the square control handles that appear when you're, for instance, selecting vertices on a 3D object in the 3D view. By default, this value is set to 5. You can also set the pickable area size of the square control handles in the numeric field or slider to the right of the **3D handle size** control.
2D handle size	Adjust the size of the square control handles that appear on the Viewer for some operations, such as transformations, warps, and Bezier and B-spline shapes.

Viewer Handles	
	 By default, this value is set to 5. You can also set the pickable area size of the square control handles in the numeric field or slider to the right of the **2D handle size** control.
icon size	Adjust the size of the 2D transformation overlay, 3D camera, 3D object normals, and 3D axis on the Viewer. By default, this value is set to 50.
icon scaling	Adjust how much the scale of display affects the size of the 2D transformation overlay, 3D camera, and 3D axis. When this is set to 0, these controls are always drawn the same size, regardless of the zoom level. When the value is set to 1, the controls scale with the displayed image or 3D scene when you zoom in or out. Intermediate values mix this so that the controls do scale, but not as much as the image does. This gives an optical illusion that you are zooming in or out without making the controls unusably small or large.
object interaction speed	Set how fast mouse movements rotate and translate 3D axis and cameras. The lower the value, the finer the movements. The default value is 1.
camera interaction speed	Set how fast mouse movements tumble and roll the 3D view in the Viewer. The lower the value, the finer the movements. The default value is 1.

Appendix B: Keyboard Shortcuts

Keyboard shortcuts, or hotkeys, provide quick access to the features of Nuke. The following tables show these keystrokes.

Conventions

The following conventions apply to instructions for mouse-clicks and key presses.

- LMB means click or press the left mouse button.
- MMB means click or press the middle mouse button
- RMB means click or press the right mouse button.
- When you see the word "drag" after a mouse button abbreviation (i.e., **"MMB drag"**), this tells you to press and hold the mouse button while dragging the mouse pointer.
- Keystroke combinations with the **Ctrl**, **Alt**, and **Shift** keys tell you to press and hold the key and then type the specified letter.

For example, "Press **Ctrl+S**" means hold down the **Ctrl** key, press **S**, and then release both keys.



Note: On Mac, replace the **Ctrl** key with the **Cmd** key.



Note: Keystrokes in the tables appear in upper case, but you do not type them as upper case. If the **Shift+**modifier does not appear before the letter, just press the letter key alone.



Note: This section assumes you are using the default keyboard and mouse-button assignments. If the mouse buttons do not work for you as described here, try resetting the mouse control type back to the standard Nuke setting (**Preferences** > **Panels** > **Viewer Handles** > **3D control type** > **Nuke**).

Global

Keystroke(s)	Action
Backspace/Delete	Delete selected clips or folders
F12	Clear buffers and playback cache
MMB drag	Virtual slider (number fields)
Space bar (short press)	Expand the focused panel to the full window
Space bar (long press)	Raise the right-click menu
Alt+S	Make the application or floating window fullscreen.
Alt+`	Show Curve Editor.
Ctrl+A	Select all
Ctrl+C	Copy selected item(s)
Ctrl+D	Duplicate selected item(s)
Ctrl+F#	Save current window layout. The # represents a function key number, F1 through F6
Ctrl+LMB on panel name	Float panel
Ctrl+N	Create a new project or script, depending on environment
Ctrl+O	Open a project or script, depending on environment
Ctrl+Q	Exit the application
Ctrl+S	Save current project or script, depending on environment
Ctrl+T	Cycle through tabs in the current pane. Note that this does not work if the focus is on the input pane of the Script Editor
Ctrl+V	Paste the contents of the clipboard

Keystroke(s)	Action
Ctrl+W	Close the current project or script, dependent on environment
Ctrl+X	Cut selected item(s)
Ctrl+Z	Undo last action
Shift+Esc	Close the current tab
Shift+F1..F6	Change workspace
Shift+S	Open the Preferences dialog
Alt+Shift+1..6	Open a recent project or script, depending on environment
Ctrl+Shift+A	Select none
Ctrl+Alt+`	Goto next pane
Ctrl+Shift+[	Goto next tab
Ctrl+Shift+]	Goto previous tab
Ctrl+Shift+S	Save current project or script, depending on environment, and specify name (Save As)
Ctrl+Shift+Z	Redo last action
Ctrl+Alt+Shift+`	Goto previous pane

Nuke Studio's Timeline Viewer

Keystroke(s)	Action
-	Zoom Out
+	Zoom In
A	Alpha
B	Blue

Keystroke(s)	Action
C	Razor selected
E	Clipping Warning
End	Go to End
F	Zoom to Fit
F12	Clear playback cache
G	Green
H	Zoom to Fill
Home	Go to Start
I	Mark In point
J	Play Backward
K	Pause
L	Play Forward
Left Arrow	Frame Backwards
O	Mark Out point
PgDown	Next Layer
PgUp	Previous Layer
Q	Show Overlays
R	Red
Return	Swap A/B Inputs
Right Arrow	Frame Forwards
V	Display the version selector
W	Wipe
Y	Luma

Keystroke(s)	Action
Alt+# (1, 2, etc.)	Zoom out by 100%, 200% and so on
Alt+Down Arrow	Version down the selected clip or shot
Alt+I	Clear In Point
Alt+O	Clear Out Point
Alt+Shift+Left Arrow	Previous Tag
Alt+Shift+Right Arrow	Next Tag
Alt+U	Clear In/Out Points
Alt+Up Arrow	Version up the selected clip or shot
Ctrl+# (1, 2, etc.)	Zoom-in by 100%, 200% and so on
Ctrl+/	Show Timeline Editor
Ctrl+F	Full Screen
Ctrl+LMB	Color picker (single pixel)
Ctrl+RMB	Deselect sampled pixels
Ctrl+S	Save current project
Shift+C	Razor all under playhead
Shift+I	Go to In Point
Shift+Left Arrow	Skip Backwards
Shift+O	Go to Out Point
Shift+Right Arrow	Skip Forwards
Ctrl+Alt+Down Arrow	Go to min version
Ctrl+Alt+Up Arrow	Go to max version
Ctrl+Shift+1	Zoom to Actual Size
Ctrl+Shift+2	Zoom to Half Size

Keystroke(s)	Action
Ctrl+Shift+F	Full Quality 1:1
Ctrl+Shift+LMB	Color picker (region of pixels)
Ctrl+Shift+P	Ignore Pixel Aspect
Ctrl+Shift+S	Save current project and specify name (Save As)

Nuke Studio's Timeline

Keystroke(s)	Action
, (comma)	Nudge selected shot(s) left, where space is available
. (period)	Nudge selected shot(s) right, where space is available
1, 2, 3, 4, 5, 6, 7, 8, 9, 0	• Selects playheads 1-10 if the playhead is already on. • Jumps to that playhead's position on the timeline. • If not yet activated, creates a new numbered playhead at the current timeline position.
A	Display in the A input buffer
B	Display in the B input buffer
Alt and drag	Ripple and duplicate the dragged shot
Alt then drag	Duplicate the dragged shot
D	Enable or disable the selected shot(s)
Down Arrow	Next Edit
drag then Alt	Activate Ripple mode while dragging shot
E	Cycles between the Slip Clip and Slide Clip tools
Enter (numeric keypad)	Edit playhead time

Keystroke(s)	Action
F12	Clear playback cache
F5	Render all Comp containers
F7	Render selected Comp containers
LMB	Select a shot including any linked tracks
F	Zooms to view the selected shot(s) on the timeline
H	Zooms to view the whole timeline
M	Insert the contents of a source Viewer into the timeline at the current playhead position overwriting existing shots
N	Insert the contents of a source Viewer into the timeline at the current playhead position and ripple existing shots downstream to accommodate the change
Numeric keypad arrows	Change selection between shots
Q	Cycles between the available move tools: Multi, Move/Trim, and Select
R	Cycles between the available edit tools: Ripple, Roll, and Retime
T	Cycles between the available razor tools: Razor, Razor All, and Join
U	Mark shot
Up Arrow	Previous Edit
V	Display the version selector
W	Cycles between the available selection tools
Shift and drag shot	Insert new track
Alt+, (comma)	Nudge selected shot(s) up, overwriting any shots on the tracks above
Alt+. (period)	Nudge selected shot(s) down, overwriting any shots on the tracks below
Alt+C	Insert a ColorCorrect soft effect on any shot selections or at the playhead
Alt+D	Display metadata for the selected track item(s)

Keystroke(s)	Action
Alt+Down Arrow	Version down the selected shot
Alt+G	Insert a Grade soft effect on any shot selections or at the playhead
Alt+LMB	Select a shot, ignoring linked tracks (for example, audio only)
Alt+T	Insert a Transform soft effect on any shot selections or at the playhead
Alt+Up Arrow	Version up the selected shot
Ctrl+numeric keypad arrows	Nudge selected shot(s), where space is available
Ctrl+A	Select all shots
Ctrl+I	Import files
Ctrl+Return	Open selected shot in the Viewer
Ctrl+T	Add a dissolve between two selected shots
Shift+, (comma)	Nudge selected shot(s) left by the increment amount set under the Viewer, where space is available
Shift+. (period)	Nudge selected shot(s) right by the increment amount set under the Viewer, where space is available
Shift+A	Sends all tracks to Input Buffer A
Shift+B	Sends all tracks to Input Buffer A
Shift+Backspace	Ripple delete
Shift+Ctrl and drag shot	Disable snap to transition when dragging shots
Shift+U	Mark selection
Alt+Shift+/	Rename shots
Ctrl+Alt+A	Select all in track
Alt+Shift+Down Arrow	Go to min version

Keystroke(s)	Action
Alt+Shift+Up Arrow	Go to max version
Ctrl+Alt+Return	Open selected shot in a new Viewer
Ctrl+Shift+A	Deselect all shots
Ctrl+Shift+E	Open the Export dialog
Ctrl+Shift+I	Import folders
LMB then Shift+LMB	Select all shots between the left-clicks
Shift+Alt+LMB	Ignore linked tracks during selection
Ctrl+Alt+Shift+I	Import EDL, OTIO, XML, or AAF
PgUp	Select track(s) and press PgUp to jump to next edit in the selected track(s).
PgDown	Select track(s) and press PgDown to jump to previous edit in the selected track(s).

2D Compositing Viewer

Keystroke(s)	Action
-	Zoom Out
, (comma)	Decrease Gain
; (semi-colon)	Previous view (multi-view)
. (period)	Increase Gain
' (apostrophe)	Next view (multi-view)
[	Toggle left toolbar
]	Toggle right toolbar
` (backtick)	Toggle floating viewers

Keystroke(s)	Action
+	Zoom In
1, 2, etc.	Set Viewer inputs for A buffer
A	Toggles between the Alpha channel and RBG
B	Toggles between the Blue channel and RBG
Down Arrow	Switch to the previous input for the A buffer
End	Go to the out point in the time slider
F or MMB	Zoom to Fit
G	Toggles between the Green channel and RBG
H	Zoom to Fill
Home	Go to the in point in the time slider
I	Mark In point
J	Play Backward
K	Pause
L	Play Forward
Left Arrow	Step backward one frame
M	Toggles between the Matte channel and RBG
Numeric pad left/right arrow keys	Nudge one frame backward or forward
O	Mark Out point
P	Pause Viewer refreshing
PgDown	Next layer
PgUp	Previous layer

Keystroke(s)	Action
Q	Toggle overlays
R	Toggles between the Red channel and RBG
Return	Swap A/B input buffers
Right Arrow	Step forward one frame
S	Open Viewer settings **Note:** You can't use the **S** keyboard shortcut to open Viewer properties when Roto or RotoPaint properties are open.
Tab	Toggle 2D/3D
U	Update the Viewer
Up Arrow	Switch to the next input for the A buffer
W	Toggle wipe tool
Y	Toggles between the Luminance channel and RBG
Alt+# (1, 2, etc.)	Zoom out by 100%, 200% and so on
Alt+G	Go to a specific frame
Alt+I	Clear In point
Alt+LMB	Pan
Alt+MMB and drag	Zoom in/Out at the pointer location
Alt+O	Clear Out point
Alt+P	Toggle Input Process
Alt+U	Clear In and Out points
Alt+W	ROI - Activate new region

Keystroke(s)	Action
Alt+Z	Toggle lock/unlock zoom level
Ctrl+# (1, 2, etc.)	Zoom-in by 100%, 200% and so on
Ctrl+Left Arrow	Move the playhead backwards by 1/2 the distance from the playhead to the in point (subsequent key strokes recalculate the distance before applying the move)
Ctrl+LMB	Color picker (single pixel)
Ctrl+P	Toggle Proxy
Ctrl+Right Arrow	Move the playhead forwards by 1/2 the distance from the playhead to the out point (subsequent key strokes recalculate the distance before applying the move)
Ctrl+RMB	Deselect sampled pixels
Ctrl+S	Save current script
Ctrl+U	Toggle previewing output on an external broadcast video monitor
LMB + MMB and drag	Zoom in and out with the click point set as the center of the Viewer
MMB and RMB	Toggle between the last specified range (MMB and drag) and **Visible** mode
MMB and drag (time slider)	Zooms the time slider to the range specified by the drag operation
MMB and drag (Viewer)	Pans in the Viewer
MMB scroll	Zoom in and out of the time slider
Shift+[	Toggle top toolbar
Shift+]	Toggle bottom toolbar
Shift+1, 2, etc.	Set Viewer inputs for B buffer
Shift+Down Arrow	Switch to the previous input for the B buffer
Shift+Left Arrow	Move the playhead backward by the amount of frames specified in the skip frames control

Keystroke(s)	Action
Shift+numeric pad left arrow	Nudge the playhead backward by one frame
Shift+numeric pad right arrow	Nudge the playhead forward by one frame
Shift+Right Arrow	Move the playhead forward by the amount of frames specified in the skip frames control
Shift+Up Arrow	Switch to the next input for the B buffer
Shift+W	Enable or disable ROI
Alt+Shift+S	Save script and increment version number The script name must include **_v#** for the increment to work as expected. For example, **myScript_v01.nk**
Ctrl+Shift+LMB	Color picker (region)
Ctrl+Shift+P	Toggle pixel aspect ratio
Ctrl+Shift+S	Save current script and specify name (Save As)

3D Compositing Viewer

Keystroke(s)	Action
C	3D Top view
PgDown	Next layer (color channel display)
PgUp	Previous layer (color channel display)
N	Toggle soft selection
N+LMB	Mouse drag to change soft selection radius
S	Display Viewer Settings

Keystroke(s)	Action
Tab	Toggle 2D/3D viewer
V	3D Perspective view
W	Toggle the Wipe tool
X	3D right-side view
Z	3D front view
Alt+LMB	Translate viewer on y,z axis
Alt+MMB	Zoom in/out (drag left/right)
Alt+RMB or Ctrl+LMB	Rotate viewer on x,y axis
Ctrl+L	Toggle Unlocked/Locked/Interactive Camera or Light
Shift+C	3D Bottom view
Shift+X	3D Left-side view
Shift+Z	3D Back view
Ctrl+Shift+LMB	Rotate viewer on z axis

Node Graph

Keystroke(s)	Action
–	Zoom-out
. (period)	Insert Dot node
/	Search by node name or class
\	Snaps all nodes to grid
1, 2, 3, etc.	With a node selected in the Node Graph, create/connect Viewer inputs With no nodes selected in the Node Graph, cycle through connected views

Keystroke(s)	Action
+	Zoom-in
B	Insert Blur node
Backspace/Delete	Delete selected nodes
C	Insert ColorCorrect node
D	Disable/Enable node
Down Arrow	Next node in tree
F	Fit selected nodes to Node Graph panel or, if no nodes selected, fits all nodes to Node Graph panel
F5	Render all Write nodes
F7	Render selected Write nodes
G	Insert Grade node
I	Display selected node information
J	Jump to bookmarked node
K	Insert Copy node
L	Auto-place selected nodes
M	Insert Merge node
MMB	Fit all nodes to Node Graph panel
MMB and drag	Pan
LMB (hold) + RMB	Cycle between a node's available input connections
N	Rename selected node
Numeric pad arrow keys	Navigate around the node tree in the direction specified. For example, pressing the left arrow key (4) selects the next node to the left of the current node in the tree.
O	Insert Roto node

Keystroke(s)	Action
P	Insert RotoPaint node
Q	Show named script info
R	Insert Read node
Return	Open properties for selected node(s)
S	Display Project Settings
T	Insert Transform node
Tab	Tab node search menu
U	Splay first (splay selected nodes to first selected node)
Up Arrow	Previous node in tree
W	Insert Write node
X	Command entry mode
Y	With two or more nodes selected, splay the inputs of the first node selected to the outputs of subsequent node selections upstream **Note:** If more than two nodes are selected that don't have multiple inputs, **Y** ignores all nodes except the first two selections.
Alt+# (1, 2, etc.)	Zoom-out %
Alt+B	Duplicate and branch selected nodes
Alt+C	Duplicate selected node(s)
Alt+D	Toggles whether a node is always displayed in Dope Sheet or not
Alt+Down Arrow	Version down the current Read/Write node file name
Alt+E	Toggle expression links on or off
Alt+F	Generate Flipbook for node

Keystroke(s)	Action
Alt+H	Hide node inputs when not selected
Alt+I	Display script information, such as the node count, channel count, cache usage, and whether the script is in full-res or proxy mode
Alt+K	Clone selected node(s)
Alt+N	Create StickyNote
Alt+P	Toggle postage stamp on or off
Alt+U	With two or more nodes selected, splay first selected node to input A of all subsequent selections
Alt+Up Arrow	Version up the current Read/Write node file name
Alt+X	Run a script from the file browser
Ctrl+# (0, 1, 2, etc.)	Zoom-in %
Ctrl+A	Select all nodes in Node Graph
Ctrl+B	Toggle node buffer for selected nodes **Tip:** When enabled, the output from the nodes is cached so that it can be re-read quickly. A yellow line displays under the nodes to indicate that caching is enabled.
Ctrl+C	Copy selected node(s)
Ctrl+create node	Replace selected node with new node, such as Ctrl+B to replace the target with a Blur node
Ctrl+D	Disconnect upstream node
Ctrl+Down Arrow	Move selected node downstream
Ctrl+F7..F10	Save locations 1 through 4
Ctrl+G	Nest selected nodes in Group
Ctrl+I	Open new Compositing Viewer

Keystroke(s)	Action
Ctrl+K	Copy as clone
Ctrl+L	Collapse selected nodes to a LiveGroup or, if no nodes are selected, create an empty LiveGroup
Ctrl+LMB on node	Highlight all upstream nodes
Ctrl+P	Toggle Proxy
Ctrl+Return/Enter	Open a Group's sub-graph
Ctrl+Up Arrow	Move selected node upstream
Ctrl+V	Paste node(s) from the clipboard
Ctrl+X	Cut selected node(s)
LMB + MMB and drag	Zoom in and out with the click point set as the center of the Node Graph
Shift+\	Snap selected node to grid
Shift+# (1, 2, 3, etc.)	Connect selected node to Viewer B buffer
Shift+A	Insert AddMix node
Shift+create node	Create node in new branch, such as Shift+B to create a Blur node in a new branch
Shift+drag	Duplicate selected arrow
Shift+F7..F10	Restore locations 1 through 4
Shift+U	With two or more nodes selected, splay first selected node to input B of all subsequent selections
Shift+X	Swap A/B inputs on node
Shift+Y	With two or more nodes selected, splay the outputs of the selected nodes to the inputs of the last node selected downstream  **Note:** If more than two nodes are selected that don't have multiple inputs, **Shift**+**Y** ignores all nodes except the last two selections.

Keystroke(s)	Action
Alt+LMB drag	Pan
Alt+MMB drag	Zoom-in/out
Alt+Shift+K	Declone nodes
Alt+Shift+U	With two or more nodes selected, splay last selected node to input A of all previous selections
Alt+Shift+Up Arrow	Version to up to the latest file (Read nodes only)
Ctrl+Alt+A on node	Select all nodes connected to the target node's tree
Ctrl+Alt+G	Replace Group node with nested nodes
Ctrl+Alt+LMB on node	Open node properties in floating window
Ctrl+Alt+V	Paste knob values to a node of the same class as the copy operation
Ctrl+Shift+/	Search and Replace (Read and Write nodes)
Ctrl+Shift+B	Toggle selected node(s) bookmark on or off
Ctrl+Shift+C	Change node color
Ctrl+Shift+G	Copy gizmo to group
Ctrl+Shift+V	Paste a copied node into a new branch from an existing node
Ctrl+Shift+LMB on node	Select all upstream nodes
Ctrl+Shift+N	Create a new script in a new session
Ctrl+Shift+P	Create Precomp from selected nodes
Ctrl+Shift+X	Extract selected nodes from tree
Ctrl+Alt+Shift+G	Create Group from selected nodes
Ctrl+Alt+Shift+K	Force clone

Project/Tags/Versions Bin

Keystroke(s)	Action
D	Hide version(s) of a clip. You can only hide versions when a clip is opened in the Versions Bin
F8	Refresh clips
V	Display the version selector
Alt+D	Display metadata for the selected shot(s)
Alt+Down Arrow	Version down the selected clip
Alt+F5	Rescan clip range
Alt+Up Arrow	Version up the selected clip
Ctrl+B	Create a new bin
Ctrl+N	Create a new sequence
Ctrl+Return	Open selected clip in the Viewer
Ctrl+Y	Create a new tag. You can only add tags to the **Tags** panel
Alt+Shift+Down Arrow	Go to min version
Ctrt+Alt+Return	Open selected clip a new Viewer
Alt+Shift+Up Arrow	Go to max version

Properties Panel

Keystroke(s)	Action
\	Snaps all nodes to grid

Keystroke(s)	Action
/	Search by node name or class
. (period)	Create Dot node
1, 2, 3, etc.	With a node selected in the Node Graph, create/connect Viewer inputs. With no nodes selected in the Node Graph, cycle through connected views.
B	Insert Blur node
Backspace/Delete	Delete selected nodes
C	Insert ColorCorrect node
D	Disable/Enable node
Esc	Closes currently active or last selected Properties panel
F5	Render all Write nodes
F7	Render selected Write nodes
G	Insert Grade node
I	Display selected node information
J	Jump to bookmarked node
K	Insert Copy node
L	Auto-place selected nodes
M	Insert Merge node
O	Insert Roto node
P	Insert RotoPaint node
Q	Show named script info
R	Insert Read node
Return/Enter	Chooses selected menu item
S	Display Project Settings

Keystroke(s)	Action
T	Insert Transform node
Tab	Move to next control in the Properties panel
Up/Down Arrow	Increment control values
U	Splay first (splay selected nodes to first selected node)
W	Insert Write node
X	Command entry mode
Y	With two or more nodes selected, splay the inputs of the first node selected to the outputs of subsequent node selections upstream  **Note:** If more than two nodes are selected that don't have multiple inputs, Y ignores all nodes except the first two selections.
LMB and drag	Copy current value from one control to another
MMB and drag	Adjust control values using virtual slider (regular)
Alt+B	Duplicate and branch selected nodes
Alt+C	Duplicate selected node(s)
Alt+D	Toggles whether a node is always displayed in Dope Sheet or not
Alt+E	Toggle expression links on or off
Alt+F	Generate Flipbook for node
Alt+H	Hide node inputs when not selected
Alt+K	Clone selected node(s)
Alt+LMB on close (x)	Close all open Properties panels
Alt+N	Create StickyNote
Alt+P	Toggle postage stamp on and off for selected Properties panel

Keystroke(s)	Action
Alt+U	With two or more nodes selected, splay first selected node to input A of all subsequent selections
Alt+X	Run a script from the file browser
Ctrl+A	Select all nodes in Properties panel
Ctrl+B	Toggle node buffer for selected nodes **Tip:** When enabled, the output from the nodes is cached so that it can be re-read quickly. A yellow line displays under the nodes to indicate that caching is enabled.
Ctrl+C	Copy selected node(s)
Ctrl+D	Disconnect upstream node
Ctrl+F7..F10	Save locations 1 through 4
Ctrl+G	Nest selected nodes in Group
Ctrl+I	Open new Compositing Viewer
Ctrl+K	Copy as clone
Ctrl+L	Collapse selected nodes to a LiveGroup or, if no nodes are selected, create an empty LiveGroup
Ctrl+LMB	Reset slider to default
Ctrl+LMB on close (x)	Close all properties panels except the one clicked
Ctrl+P	Toggle Proxy
Ctrl+V	Paste node(s) from the clipboard
Ctrl+X	Cut selected node(s)
Shift+A	Insert AddMix node
Shift+# (1, 2, 3,	Connect selected node to Viewer B buffer

Keystroke(s)	Action
etc.)	
Shift+create node	Create node in new branch, such as Shift+B to create a Blur node in a new branch
Shift+F7..F10	Restore locations 1 through 4
Shift+U	With two or more nodes selected, splay first selected node to input B of all subsequent selections
Shift+X	Swap A/B inputs on node
Shift+Y	With two or more nodes selected, splay the outputs of the selected nodes to the inputs of the last node selected downstream **Note:** If more than two nodes are selected that don't have multiple inputs, Shift+Y ignores all nodes except the last two selections.
Alt+LMB drag	Adjust control values using virtual slider (magnitude dependent on cursor position)
Alt+MMB and drag	Adjust control values using virtual slider (fine)
Ctrl+LMB drag	Expression link controls
Shift+LMB drag	Copy animation from one control to another
Shift+MMB and drag	Adjust control values using virtual slider (coarse)
Shift+Tab	Move to previous control in the properties
Alt+Shift+K	Declone nodes
Alt+Shift+U	With two or more nodes selected, splay last selected node to input A of all previous selections
Ctrl+Alt+A (on a node)	Select all nodes connected to the target node's tree
Ctrl+Alt+G	Replace Group node with nested nodes
Ctrl+Alt+V	Paste knob values to a node of the same class as the copy operation

Keystroke(s)	Action
Ctrl+Shift+/	Search and Replace strings in selected Read and Write nodes
Ctrl+Shift+A	Close all open Properties panels
Ctrl+Shift+B	Toggle selected node(s) bookmark on or off
Ctrl+Shift+C	Change node color
Ctrl+Shift+G	Copy gizmo to group
Ctrl+Shift+P	Create Precomp from selected nodes
Ctrl+Shift+V	Paste a copied node into a new branch from an existing node
Ctrl+Shift+X	Extract selected nodes from tree
Ctrl+Alt+Shift+K	Force clone

Curve Editor/Dope Sheet

Keystroke(s)	Action
A	Frame all keyframes
C	Interpolation (Cubic)
F	Frame all selected keyframes
H	Interpolation (Horizontal)
K	Interpolation (Constant)
L	Interpolation (Linear)
LMB	Select single point
LMB drag on blank space	Select region of points
LMB drag on point	Move all selected points
LMB drag on	Scale points inside selection region

Keystroke(s)	Action
selection box	
LMB drag on transform handle	Move all points in selection region
MMB drag	Draw box in area and zoom to fit area to curve editor panel
MMB or F	Fit selection to window
R	Interpolation (Catmull-Rom)
X	Break selected control points' handles
Z	Interpolation (Smooth [bezier])
Alt+` (backtick)	Display Curve Editor
Alt+LMB drag	Pan
Alt+MMB drag	Variable zoom
Ctrl+A	Select all curves
Ctrl+C	Copy selected keys
Ctrl+E	Copy Expressions
Ctrl+L	Copy Links
Ctrl+LMB drag	Move keyframes freely on the x and y axes
Ctrl+Shift (hold down)	Hide points to click on selection box/transform handle
Ctrl+V	Paste curve
Ctrl+X	Cut selected keys
Shift+LMB	Add or Remove points from selection
Shift+LMB drag	Draw box to add/remove points from selection
Alt+Shift+LMB drag	Move single point

Keystroke(s)	Action
Ctrl+Alt+LMB	Add point to current curve
Ctrl+Shift+C	Copy selected curves
Ctrl+Alt+Shift+LMB	Sketch points freely on current curve

Script Editor

Keystroke(s)	Action
Tab	Increase indentation
Ctrl+[	Load previous script
Ctrl+]	Load next script
Ctrl+Backspace	Clear output window
Ctrl+Return/Enter	Run script in editor
Shift+Tab	Decrease indentation
Ctrl+Shift+[	Decrease indentation of selected text
Ctrl+Shift+]	Increase indentation of selected text

Roto/RotoPaint

Keystroke(s)	Action
Backspace	Delete an item from curve list or Delete points/shapes
C	Toggle Clone tool
D	Toggle Dodge/Burn
Delete	Remove selected point(s)

Keystroke(s)	Action
E	Increase feather on selected point(s)
Esc	Switch back to the current Select tool
I	Pick color
N	Toggle Brush/Eraser
Return (Bezier/Spline tools)	Close shape
S (with Viewer mouse-over focus)	Cycle between the selected tool's modes
T (Clone tool)	Toggle source as onion skin with transform jack
T (Select tool)	Display transform box (points) or jack (shapes)
V	Toggle Bezier/B-Spline/Ellipse/Rectangle tools
X	Toggle Blur/Sharpen/Smear tools
Z	Smooth selected points
Ctrl+A	Select all points
Ctrl+LMB (Bezier/Spline tool)	Sketch a Bezier or B-spline
Ctrl+LMB (on a point)	Break tangent handle for selected point
Ctrl+drag (Clone/Reveal tools)	Set offset between source and destination
Ctrl+Shift (transform box)	Drag transform box points to move them
Shift+LMB (Bezier tool)	Create a sharp point on previous point
Shift+LMB (editing points)	Bring up transform box for selected points
Shift+drag (Brush/Eraser/Clone/Reveal tools)	Change brush size
Shift+drag (editing Bezier/Spline points)	Move both tangent handles at the same time

Keystroke(s)	Action
Shift+E	Remove feather from selected points
Shift+Z	Cusp selected points
Ctrl+Alt+LMB (on Splines)	Add point to curve
Ctrl+Shift+drag (B-Spline)	Increase/decrease tension of B-Spline shape

Appendix C: Supported File and Camera Formats

Notes

When importing and exporting files, remember the following:

- When you import images with a Read node (**Image > Read**), Nuke analyzes the contents of the file to determine the format. The file name extension is not used to determine file format, which allows flexibility with naming conventions in a production environment.
- Regardless of format, Nuke converts all imported image sequences to its native 32-bit linear RGB colorspace.
- When you render new images from Nuke (**Image > Write**), you can use a file name extension to specify format.
- To import and export geometry objects from Alembic, FBX, or OBJ files, use the ReadGeo node (**3D** > **Geometry** > **ReadGeo**). To write them out again, use the WriteGeo node (**3D** > **Geometry** > **WriteGeo**). To import cameras or transforms from Alembic or FBX files, use the Camera node (**3D** > **Camera**). To import lights from an FBX file, use the Light node (**3D** > **Lights** > **Light**).
- To import deep images (either in DTEX or scanline OpenEXR format), use a DeepRead node (**Deep** > **DeepRead**). To export deep images (in scanline OpenEXR format), use a DeepWrite node (**Deep** > **DeepWrite**).

Supported File Formats

The following table lists the supported file formats. The extensions listed under **Extension** let you specify the image format; use these as the actual file name extensions or the prefix to indicate output format for the image sequences.

Format	Bit Depths	Read/Write	Extension	Notes
AAF	n/a	read	aaf	Nuke Studio only

Format	Bit Depths	Read/Write	Extension	Notes
Alembic	n/a	read and write	abc	You can read meshes, point clouds, cameras, and transforms from Alembic files into a Nuke scene using the ReadGeo, Camera, and Axis nodes. To write meshes and point clouds out again, use the WriteGeo node.
Apple ProRes	10, 12	read and write	mov	Support for the following: • Apple ProRes 4444 XQ 12-bit • Apple ProRes 4444 12-bit • Apple ProRes 422 HQ 10-bit • Apple ProRes 422 10-bit • Apple ProRes 422 LT 10-bit • Apple ProRes 422 Proxy 10-bit
ARRIRAW Codex	12	read only	ari arx	Alexa Mini LF Codex HDE

Format	Bit Depths	Read/Write	Extension	Notes

Camera/Sensor	Nuke 15.0	Nuke 15.1	Nuke 15.2	Nuke 16.0	Nuke 16.1	Nuke 17.0
ARRIRAW SDK (**.ari**, **.arx**, and **.mxf**)	v8.0.0	v.8.3.1	v.8.3.1	v.8.3.1	v.8.3.1	v.8.3.1
CodexHDE SDK	4.0.3	5.2.1	5.2.1	5.2.1	5.2.1	5.2.1
CUDA GPU Support	●	●	●	●	●	●
ALEXA 35	●	●	●	●	●	●
ALEXA Mini LF	●	●	●	●	●	●
ALEXA LF	●	●	●	●	●	●
ALEXA SXT(W)	●	●	●	●	●	●
ALEXA Mini	●	●	●	●	●	●
ALEXA 65	●	●	●	●	●	●
ALEXA XT	●	●	●	●	●	●
ALEXA Studio	●	●	●	●	●	●
ALEXA Classic	●	●	●	●	●	●
AMIRA	●	●	●	●	●	●

Note: Nuke, Nuke Studio, and Hiero do not support **.mxf** from the AMIRA camera.

AVI	n/a	read and write	avi	On Windows, the AVI reader uses DirectShow multimedia architecture. When decoding **.avi** files,

Format	Bit Depths	Read/Write	Extension	Notes
				DirectShow tries to find the appropriate codec on the system. If the codec is not available, **.avi** files cannot be opened.
BRAW	12	read	braw	
CIN	10 (log)	read and write	cin	
DNG	8, 12	read	dng	Includes RAW 2.5K CinemaDNG
DPX	8, 10, 12, and 16	read and write	dpx	YCbCr encoded DPX files are not supported on the timeline.
DTEX	32	read only	dtex	To use DTEX files, you need Pixar's PhotoRealistic RenderMan® Pro Server 20, or earlier, on your computer. To read a DTEX file, use the DeepRead node.
EDL	n/a	read and write	edl	Nuke Studio only
FBX		read and write	fbx	You can read meshes, point clouds, cameras, lights, and transforms from FBX files into a Nuke scene using

Format	Bit Depths	Read/Write	Extension	Notes
				the ReadGeo, Camera, Light, and Axis nodes. To write geometry out again, use the WriteGeo node.
GIF	8	read only	gif	
Radiance	16	read and write	hdr, hdri	This format stores an 8-bit mantissa for each of r, g, and b and an additional 8-bit exponent that is shared by all three, which packs the floating point RGB triplet into 32 bits per pixel.
JPEG	8	read and write	jpg, jpeg	Adjust compression levels using the quality slider in the Write node's properties panel.
Maya IFF	8, 16	read only	iff	
MOV	n/a	read and write	mov	The writer supports the following codecs: • Animation • Apple ProRes 4444 XQ 12-bit • Apple ProRes 4444 12-bit • Apple ProRes 422

Format	Bit Depths	Read/Write	Extension	Notes
				HQ 10-bit • Apple ProRes 422 10-bit • Apple ProRes 422 LT 10-bit • Apple ProRes 422 Proxy 10-bit • Avid DNxHD 444 10-bit 440Mbit • Avid DNxHD 422 10-bit 220Mbit • Avid DNxHD 422 8-bit 220Mbit • Avid DNxHD 422 8-bit 145Mbit • Avid DNxHD 422 8-bit 36Mbit **Note:** Interlaced writing is not supported. See Avid DNxHD Notes for more information. • Avid DNxHR: • 8-bit 4:2:2

Format	Bit Depths	Read/Write	Extension	Notes
				DNxHR LB, DNxHR SQ, and DNxHR HQ • 12-bit DNxHR HQX 422 and DNxHR 444 • H.264 Main 4:2:0 8-bit • H.264 High 4:2:0 8-bit • Motion JPEG A • Motion JPEG B • MPEG-4 • NotchLC  **Note:** NotchLC is supported for Windows and Linux currently. • Photo - JPEG • PNG • Uncompressed
		write only	mov	DNxHR Pattern: OP-1A, OP-Atom Profiles:

Format	Bit Depths	Read/Write	Extension	Notes
				• 4:4:4 12-bit • HQX 4:2:2 12-bit • HQ 4:2:2 8-bit • SQ 4:2:2 8-bit • LB 4:2:2 8-bit
MXF	8, 10, 12	read only	mxf	Supported codecs include: • Uncompressed 4:2:2 YCbCr 8-/10-bit • Uncompressed 4:4:4:4 RGBA 8-/10-bit • Uncompressed Avid 4:2:2 YCbCr 8-/10-bit • Uncompressed Avid 4:4:4:4 RGBA 8-/10-bit • JPEG2000 • Avid DNxHD (1080p and 720p 1920x1080 and 1280x720, 4:4:4:4 and 4:2:2) 36, 115, 120, 145, 175, 185, 220, 220x See Avid DNxHD Notes for more information. • Avid DNxHR: • 8-bit 4:2:2 DNxHR LB, DNxHR SQ,

Format	Bit Depths	Read/Write	Extension	Notes
				and DNxHR HQ • 12-bit DNxHR HQX 422 and DNxHR 444 • Sony Raw from the F65, F55, F5 and FS700 cameras. All formats that these cameras provide: 4K, 2K, 1K, 0.5K and 0.25K • Sony X-OCN from the VENICE, VENICE 2, F55, and F5 cameras. ICVFX Mode on X-OCN of MPC-3628 or MPC-3626 V3.0. • ARRIRAW from the Alexa Mini.
		write only	mxf	DNxHD Pattern: OP-1A, OP-Atom Profiles: • HQX 4:4:4 10-bit • HQX 4:2:2 10-bit • HQ 4:2:2 8-bit • SQ 4:2:2 8-bit • LB 4:2:2 8-bit DNxHR Pattern:

Format	Bit Depths	Read/Write	Extension	Notes
				OP-1A, OP-Atom Profiles: • 4:4:4 12-bit • HQX 4:2:2 12-bit • HQ 4:2:2 8-bit • SQ 4:2:2 8-bit • LB 4:2:2 8-bit

Camera/Sensor	Nuke 15.0	Nuke 15.1	Nuke 15.2	Nuke 16.0	Nuke 16.1	Nuke 17.0
SONY SDK (**.mxf**)	v4.0.0	v5.0.0	v5.0.0	v5.0.0	v5.1.0	v5.1.0
SONY SMDK	v4.21.0	v4.24.0	v4.24.0	v4.24.0	v4.26.x	v4.26.x
BURANO	-	●	●	●	●	●
GPU Support	●	●	●	●	●	●
VENICE 2	●	●	●	●	●	●
VENICE	●	●	●	●	●	●
F55	●	●	●	●	●	●
F5	●	●	●	●	●	●
F65	●	●	●	●	●	●
FS700	●	●	●	●	●	●

Avid DNxHD Notes

The bit rates listed in the **Codec Profile** dropdown are the bit rates for 1080p at 29.97 fps EXCEPT for 36 fps (which is actually 45 Mbps @ 29.97fps). You should look at the codec format (422/444, 8/10-bit).

Note: Nuke only supports 1080p and 720p. Non-HD resolutions are scaled to 1080p before writing.

Format	Bit Depths	Read/Write	Extension	Notes
This leads to a set of 1080p bit rates: • 1080p/29.97 440x, 220x, 220, 145, 45 • 1080p/60 N/A, N/A, 440, 290, 90 (same at 59.94) • 1080p/50 N/A, N/A, 367, 242, 75 • 1080p/25 365x, 185x, 185, 120, 36 • 1080p/24 350x, 175x, 175, 115, 36 (same at 23.976) At 720p, the **Codec Profile** dropdown has a different interpretation. The bit rate is taken as the bit rate at 720p at 59.94fps. This leads to another set of bit rates: • 720p/59.94 N/A, 220x, 220, 145, N/A • 720p/50 N/A, 175x, 175, 115, N/A • 720p/29.97 N/A, 110x, 110, 75, N/A • 720p/25 N/A, 90x, 90, 60, N/A • 720p/23.976 N/A, 90x, 90, 60, N/A  **Note:** Since the bit rates are for 1080p at 29.97 fps AND 720p at 59.94 fps (except for 36 Mbit which should read 45 Mbit). It is possible to calculate the bandwidth for all the other frame rates by: BandWidth@1080p = fps/29.97 * NominalBandWidth, or BandWidth@720p = fps/59094 * NominalBandWidth where **NominalBandWidth** is the bandwidth listed in the codec profile knob OR 45 if the bandwidth listed is 36 Mbit. (Avid labels the codec profile names by the approximate bandwidth.)				
OBJ	n/a	read and write	obj	
OpenEXR and OpenEXR 3.3.6	16, 32	read and write	exr	OpenEXR handles 16- and 32-bit float. This 16 is also called "half float" and is different from the 16-bit integer that all the other formats that support 16-bit use.

Format	Bit Depths	Read/Write	Extension	Notes
				Nuke supports multi-part OpenEXR files. For more information, see Notes on Importing OpenEXR Files and Notes on Rendering OpenEXR Files. When working with deep data, Nuke supports scanline OpenEXR files. For more information, see Importing Scanline OpenEXR Files and Writing Deep Data.

EXR Compression

EXR file metadata contains a **compression** key/value pair detailing the compression type used to write the **.exr** file. The value is expressed as the name of the compression type or an integer referencing the compression used:

0 - no compression

1 - RLE compression, run length encoding

2 - Zip compression, one scan line at a time

3 - Zip compression, in blocks of 16 scan lines

4 - PIZ-based wavelet compression, in blocks of 32 scan lines

5 - PXR24 compression, lossy 24-bit float

6 - B44 compression, lossy 4-by-4 pixel block, fixed rate

Format	Bit Depths	Read/Write	Extension	Notes
7 - B44A compression, lossy 4-by-4 pixel block, flat fields are compressed more 8 - DWAA compression, lossy DCT based compression, in blocks of 32 scan lines 9 - DWAB compression, lossy DCT based compression, in blocks of 256 scan lines				
OTIO	n/a	read and write	otio	Nuke Studio only
PIC SoftImage®	8	read and write	pic	
PNG	8, 16	read and write	png (8-bit) png16 (16-bit)	
PSD	8, 16	read only	psd	While Nuke reads standard Photoshop® blend modes, it doesn't read Photoshop layer comps or recognize group blend modes. Photoshop layers are read into separate Nuke layers and anything that doesn't map into that is ignored.
R3D	16	read only	r3d	

Camera/Sensor	Nuke 15.0	Nuke 15.1	Nuke 15.2	Nuke 16.0	Nuke 16.1	Nuke 17.0
R3D SDK (**.r3d**)	v8.3.0	v8.4.0	v8.4.0	v8.4.0	v8.4.0	v8.4.0
CUDA GPU Support	●	●	●	●	●	●

Format	Bit Depths	Read/Write	Extension	Notes
Camera/Sensor	**Nuke 15.0**	**Nuke 15.1**	**Nuke 15.2**	**Nuke 16.0**	**Nuke 16.1**	**Nuke 17.0**
V_RAPTOR XL	●	●	●	●	●	●
V-RAPTOR	●	●	●	●	●	●
V-RAPTOR (X)	●	●	●	●	●	●
KOMODO	●	●	●	●	●	●
MONSTRO 8K VV	●	●	●	●	●	●
HELIUM 8K S35	●	●	●	●	●	●
SCARLET-W 5K	●	●	●	●	●	●
RED RAVEN 4.5K	●	●	●	●	●	●
WEAPON DRAGON 6K	●	●	●	●	●	●
EPIC/SCARLET DRAGON	●	●	●	●	●	●
RED ONE MYSTERIUM 4K	●	●	●	●	●	●
MYSTERIUM-X	●	●	●	●	●	●
MYSTERIUM-X MONOCHROME	●	●	●	●	●	●
RAW	n/a	read only	n/a	DSLR raw data files, such as Canon .CR2 files. These are only supported via the dcraw command line program, which you can download from the dcraw

Format	Bit Depths	Read/Write	Extension	Notes
				website. Bit depth and other specifications depend on the device. Some devices may not be supported.
RLA Wavefront®	8	read only	rla	
SGI	8, 16	read and write	sgi, rgb, rgba (8-bit sequences) sgi16 (for 16-bit sequences)	
TIFF	8, 16, and 32	read and write	tif, tiff (8-bit sequences) tif16, tiff16 (16-bit sequences) ftif, ftiff (32-bit sequences)	If utilized, the compression schema on imported TIFF sequences must be LZW®.
TARGA Truevision®	8	read and write	tga, targa	
USD	n/a	read	usd, usda	
XML	n/a	read and write	xml	Nuke Studio only
XPM	8	read and write	xpm	This is the text-based format in which Nuke's interface elements are stored.

Format	Bit Depths	Read/Write	Extension	Notes
YUV	8	read and write	yuv	This format does not specify resolution, so Nuke assumes a width of 720 pixels.

Supported Audio Formats

The following table lists supported audio formats.

Format Name	Extension
Audio only formats	
All platforms	
Wave	wav
Audio interchange format	aiff
Audio Codec 3	ac3
MPEG 1 Audio Stream	mp2
Mac and Windows	
MPEG 4 Audio	m4a
AAC (Advanced Audio Coding)	aac
Integrated audio formats	
RED Audio	r3d
QuickTime Audio	mov

End User License Agreement (EULA)

PLEASE READ THIS EULA CAREFULLY BEFORE ORDERING OR DOWNLOADING OR USING ANY SOFTWARE PRODUCTS OF FOUNDRY. YOUR ATTENTION IS PARTICULARLY DRAWN TO: (A) CLAUSE 8 IN WHICH SUBSCRIPTION CUSTOMERS (LICENSEES) AGREE TO THE AUTO-RENEWAL OF THEIR LICENSE(S) ON AN ANNUAL BASIS; (B) CLAUSE 15 WHERE WE LIMIT OUR LIABILITY TO USERS OF OUR SOFTWARE PRODUCTS; (C) CLAUSE 19.2 REGARDING THE DATA WE MAY COLLECT AND HOW WE MAY USE IT; AND (D) CLAUSE 19.3 WHERE YOU AUTHORISE FOUNDRY TO USE THE SOFTWARE TO ACCESS AND COLLECT CERTAIN INFORMATION FROM YOUR COMPUTER NETWORKS AND TO TRANSMIT THIS INFORMATION TO FOUNDRY.

IMPORTANT NOTICE TO ALL USERS: BY DOWNLOADING AND/OR INSTALLING AND/OR USING THIS SOFTWARE YOU ACKNOWLEDGE THAT YOU HAVE READ THIS EULA, UNDERSTAND IT AND AGREE TO BE BOUND BY ITS TERMS AND CONDITIONS. IF YOU DO NOT AGREE TO THE TERMS OF THIS EULA DO NOT DOWNLOAD, INSTALL, COPY OR USE THE SOFTWARE.

IMPORTANT NOTICE TO CONSUMERS WHO PURCHASE SOFTWARE PRODUCTS DIRECT FROM FOUNDRY: YOU HAVE THE RIGHT TO CANCEL YOUR CONTRACT AND OBTAIN A FULL REFUND IN ACCORDANCE WITH CLAUSE 9. HOWEVER YOU WILL LOSE THIS RIGHT ONCE YOU INSTALL THE SOFTWARE OR LOGIN TO AN INDIVIDUAL LOGIN OR TEAM LOGIN LICENSE. THIS DOES NOT AFFECT YOUR CONSUMER RIGHTS IN RELATION TO DEFECTIVE PRODUCTS OR SERVICES.

This END USER LICENSE AGREEMENT ("**EULA**") is, in cases where you purchase our product(s) direct from Foundry, incorporated into the agreement between The Foundry Visionmongers Ltd a company registered in England and Wales with company number 4642027 and whose registered office is at Squire Patton Boggs Secretarial Services Limited, Rutland House, 148 Edmund Street, Birmingham, United Kingdom, B3 2JR and whose address for correspondence is 5 Golden Square, London W1F 9HT, ("**Foundry**"), and you, as either an individual or a single company or other legal entity ("**Licensee**") on the terms of which you will purchase the products and services of Foundry (the "**Agreement**"). In cases where you purchase our product(s) from one of our resellers, the use of the term "Agreement" in this EULA refers to the arrangements between Foundry and Licensee on which Licensee is permitted to use Foundry's product(s), including this EULA.

Foundry reserves the right to refuse to grant a License (as defined in clause 1.1) to any Licensee who has failed to pay any sum due either to Foundry or to a reseller of Foundry, in connection with the Agreement, in connection with any other software license, in connection with any other agreed terms between Foundry and the Licensee, to use any Software product(s) of Foundry.

1. GRANT OF LICENSE

1.1 Subject to terms and the scope of the applicable licence model as set out in clause 2, the limitations of clause 3 and all the other terms of the Agreement, Foundry grants to Licensee a limited, non-transferable (subject to clause 2.1(b) below) and non-exclusive license to download, install and use a machine readable, object code version (subject to clauses 3 and 4 below) of the software program(s) purchased by Licensee (the "**Software**") together with, and in accordance with, any accompanying user guide and other documentation (the "**Documentation**"), solely for Licensee's, where the Licensee is a business, own internal purposes or, where the Licensee is a consumer, domestic and private purposes (the "**License**"); provided, however, that Licensee's right to download, install and use the Software and the Documentation is limited to those rights expressly set out in this EULA.

1.2 Some types of license models set out in clause 2.1 limit the installation and use of the Software to the country in which Licensee is based at the date of purchase (the "**Home Country**"), unless otherwise agreed by Foundry in writing. For the avoidance of doubt, 'use' in a Home Country requires the server, network and user to be based in the Home Country. Notwithstanding such limits, Licensee may still use the Software outside the Home Country if traveling or working outside the Home Country on a temporary basis provided that: a) such use does not exceed 70 days in aggregate in any rolling twelve month period or, in the case of any license which lasts for less than twelve months, does not exceed the number of days representing 20% of the term of the license. Notwithstanding any other terms, our Software may not be accessed or used in countries which are subject to import/export controls or sanctions, as further detailed in clause 23.

1.3 Only to the extent that is proportionate to, and reasonably necessary to support, Licensee's licensed use of the Software in accordance with the Agreement, Licensee may (provided valid license keys or license entitlements have been obtained) install the Software on more than one computer, provided always that Licensee's concurrent use of different installations of the Software does not exceed the number of valid Licenses that Licensee has paid for or licensed (as applicable). Where Licensee has purchased an Enterprise License, Licensee may install the Software on more than one computer, pursuant to the terms of the specific Enterprise License Agreement for that Licensee.

2. LICENSE MODELS

2.1 For each Software product that you purchase from Foundry, the product will be licensed (and not sold) to you on a subscription basis (or a rental basis, if available) in accordance with one or more of the license models set out in this clause 2.1 and clause 2.2, as specified in Foundry's invoice or order confirmation (as applicable), and subject to the other terms and conditions of this EULA. Please note that some licensing models set out below do not apply to certain Software products of Foundry, and certain Software products may be sold on an Enterprise License basis (as set out in the definitions below). Whichever licensing model applies, Licensee shall not, at any one time, use more copies of the Software than the total number of valid licenses purchased by Licensee (other than in relation to Enterprise Licenses, in which case the total number

of users shall be in accordance with the terms of the specific Enterprise License Agreement for that Licensee). In addition to the terms expressly stated in this EULA, by using any tool, plug-in, SDK or other functionality in conjunction with the Software which is owned or licensed by any third party (each, a "**Third Party Licensor**"), Licensee agrees to review, comply with, and be legally bound by the licence terms and conditions of that Third Party Licensor.

(a) "**Educational License**" means a License that has been purchased by an educational institution in accordance with Foundry's eligibility criteria as published on its website and amended from time to time (at https://www.foundry.com/education/students).

If Licensee has purchased the Software on the discounted terms of Foundry's educational program and eligibility criteria as published on its website and amended from time to time, Licensee warrants and represents to Foundry as a condition of the Educational License that: (i) it is an educational institution that will use the Software only for the purpose of training and instruction, and for no other purpose, (ii) Licensee will at all times comply with any eligibility criteria and guidelines in respect of the Educational License, as may be communicated to a Licensee and/or published on Foundry's website and amended from time to time, and (iii) use of the Software shall be limited to any site in the Home Country. Unless the Educational License is a Floating License, Licensee shall use the Software on only one computer at a time.

(b) "**Enterprise License**" means a License that is available for certain software only, relating to large-scale commercial deployments to enable a specific number of users to access the Software within an enterprise environment, with specific additional terms and support arrangements, as agreed separately in an Enterprise License Agreement in writing between Foundry and the Licensee. Use of the Software shall be limited to any site in the Home Country, unless agreed otherwise in the Enterprise License Agreement. Any Enterprise License may also be subject to an additional Enterprise License agreement, as agreed between the Licensee and Foundry (the "**Enterprise License Agreement**").

(c) "**Graduate License**" means a License that has been purchased by a graduate in accordance with Foundry's eligibility criteria as published on its website and amended from time to time (at https://www.foundry.com/education/students).

If Licensee has purchased the Software on the discounted terms of Foundry's Graduate Licence, Licensee warrants and represents to Foundry as a condition of the Graduate Licence that the Licensee has graduated from a university or tertiary education institute no more than six (6) months prior to the grant of the Graduate Licence. The Licensee may use the Software in a personal capacity only (namely, as a sole trader or freelancer). For the avoidance of doubt, Licensee may not use the Graduate License in connection with any work or employment, whether paid or unpaid, or other such collaborations with studios or similar organisations. Foundry reserves the right to require evidence of graduation by the Licensee. Use of the Software shall be limited to any site in the Home Country.

(d) "**Individual Login License**"

If Licensee purchases an Individual Login License, Licensee warrants and represents that Licensee is a natural person and that only Licensee shall use the Software. Licensee will be issued with log in details and may use

the Software on any number of computers (but not simultaneously). Use of the Software shall be limited to any site in the Home Country.

(e) "**Mari indie**"

Variants of Mari with limited functionality as described in the Documentation are available to purchase. If Licensee has purchased such a variant, Licensee warrants and represents to Foundry as a condition of the Agreement that: (i) Licensee is a natural person; or (ii) Licensee is an entity in the direct ownership of a single natural person; (iii) Licensee will only access and/or use one copy of the Software; and (iv) only Licensee will use the Software.

(f) "**Mari Individual Subscription License**"

If Licensee purchases a Mari Individual Subscription License then: (a) Licensee warrants and represents that Licensee is a natural person and that only Licensee will use the Software; (b) Licensee shall not share its login details for the Software with any third party; (c) Licensee may use the Software on different computers which may be located anywhere and use is not restricted to the Home Country; (d) Licensee may use the License on different computers, subject to (i) a maximum of two computer authorisations at any one time and (ii) no more than one concurrent user at any one time; (e) Licensee shall not purchase or use more than one Mari Individual Subscription License; and (f) the provisions of clause 8 shall apply.

(g) "**Non-Commercial License**"

If the License is a Non-Commercial License, Licensee warrants and represents that Licensee is a natural person, that they will only access and/or use one copy of a Non-Commercial License for personal, recreational and non-commercial purposes and that only Licensee will use the Software. Under a Non-Commercial License, Licensee will not use the Software: (a) in conjunction with any other copies or versions of the Software, under any type of License model; (b) for any commercial, professional, for-profit and/or on-sale purpose or otherwise to provide any commercial service(s) to a third party (whether or not for financial or other reward and including for education, instruction of or demonstration to any third party for commercial purposes); (c) in the course of any employment or business undertaking of Licensee; (d) on any commercial premises during business hours (except where use of the Software is solely for a personal, recreational, educational or other non-commercial purpose); and/or (e) to create any commercial tools or plug ins.

(h) "**Nuke Indie License**"

If Licensee purchases a License for Nuke Indie, then: (a) Licensee warrants and represents that Licensee (i) is a natural person and that only Licensee will use the Software; (ii) is working independently and shall not use the Nuke Indie License in a pipeline with other Nuke commercial or Nuke Indie licenses, whether those licenses are held by the Licensee, other individuals or other businesses or organisations; and (iii) earns less than $100,000 USD (or local equivalent) a year; and (iv) that Licensee satisfies all criteria set out in Foundry's Nuke Indie Eligibility Requirements as published on its website and which may be amended from time to time (the "**Nuke Indie Eligibility Requirements**"); (b) Licensee shall not share its login details for the

Software with any third party; (c) Licensee shall not purchase or use more than one Nuke Indie License; (d) Licensee may use the License on different computers, subject to (i) a maximum of two computer authorisations at any one time and (ii) no more than one concurrent user at any one time; and (e) Licensee shall use the Software in accordance with terms of the Nuke Indie Eligibility Requirements, including abiding by any functional restrictions; and (f) the provisions of clause 8 shall apply.

(i) "**Nuke Stage License**"

If Licensee purchases a Nuke Stage License, then: (a) Licensee may use the Software on any number of computers, provided that the number of concurrent Nuke Stage Output Device(s) shall never exceed the total number of valid Nuke Stage Floating Licenses purchased by Licensee; and (b) use of the Software shall be limited to any site in the Home Country. "Nuke Stage Output Device(s)" shall mean one region designated for rendered output, which may be mapped to one endpoint such as display windows, SDI outputs, files, or other render targets. If Licensee chooses to use the OptiTrack functionality as part of a Nuke Stage Licence, Licensee acknowledges and agrees that its use of this functionality is subject to the terms of OptiTrack's Plugins EULA available at https://optitrack.com/about/legal/eula.html ("**OptiTrack Plugins EULA**"). Licensee agrees to abide by the terms of OptiTrack Plugins EULA. The OptiTrack Plugin EULA applies in addition to any other provisions in this EULA relating to Third Party Licensors. If there is any conflict or inconsistency between this EULA and the OptiTrack Plugin EULA, the OptiTrack Plugin EULA shall take precedence if the conflict or inconsistency relates to the Optitrack Plugin itself.

(j) "**Offline Floating License**"

If Licensee purchases an Offline Floating License, then: (a) Licensee may use the Software on any number of computers, provided that the number of concurrent users shall never exceed the total number of valid Offline Floating Licenses purchased by Licensee; and (b) use of the Software shall be limited to any site in the Home Country.

(k) "**Offline Node Locked License**"

If Licensee purchases an Offline Node Locked License, Licensee will install and use only a single copy of the Software on only one computer at a time in the Home Country.

(l) "**Rental License**" means a License that has been purchased by a Licensee on a rental basis.

If Licensee has purchased the Software on a rental basis, the License shall be limited to the term of the rental as agreed in writing with Foundry after which it shall automatically expire.

(m) "**Student License**" means a License (whether purchased or provided free of charge) issued to a student in accordance with Foundry's eligibility criteria as published on its website and amended from time to time (at https://www.foundry.com/education/students).

If Licensee has purchased the Software on the discounted terms of Foundry's student program and eligibility criteria as published on its website and amended from time to time, Licensee warrants and represents to Foundry as a condition of the Student License that: (i) the Licensee is a part-time or full-time student at the

time of purchase and will not use the Software for any commercial, professional or for-profit purposes; (ii) Licensee will at all times comply with any eligibility criteria and guidelines in respect of the Student License, as may be communicated to a Licensee and/or published on Foundry's website and amended from time to time, and (iii) use of the Software shall be limited to any site in the Home Country. Unless the Student License is a Floating License, Licensee shall use the Software on only one computer at a time.

(n) "**Subscription License**" means a License that has been purchased on a subscription basis, as set out below.

If Licensee has purchased the Software on a subscription basis then: (a) the License shall be limited to the Subscription Period (as defined in clause 8.1) after which it shall automatically expire and; (b) the provisions of clause 8, including but not limited to auto-renewal, shall apply.

(o) "**Team Login License**"

If Licensee purchases a Team Login License, then: (a) Licensee may use the Software on any number of computers, provided that the number of concurrent users shall never exceed the total number of valid Team Login Licenses purchased by Licensee; (b) use of the Software shall be limited to any site in the Home Country; and (c) use of the Software, and changes to the user authorizations within Licensee's team organization(s), shall be in accordance with any Team Login License guidance and instructions, as published on Foundry's website and which may be amended from time to time.

(p) "**Trial License**"

Licensee may register for a "Trial License" of the Software (not available for all products or in all regions or markets). A Trial License is for product evaluation and learning purposes only and may only be used by a natural person. A Trial License lasts a limited specified period, on the expiry of which the Software will automatically cease to function. Foundry may terminate any Trial License for convenience immediately on notice to Licensee. Licensee will use the Software for product evaluation and learning purposes only.

2.2 If Licensee has purchased a License that permits "non-interactive" use of the Software ("**Headless Rendering**"), Licensee is authorized to use a non-interactive version of the Software for rendering purposes only (i.e. without a user, in a non-interactive capacity) and shall not use such Software on workstations or otherwise in a user-interactive capacity. Headless Rendering is not available on all products. In all cases, Headless Rendering licenses may be used on any number of computers, provided that the number of concurrent computers shall never exceed the total number of valid Headless Rendering licenses purchased by Licensee.

3. RESTRICTIONS ON USE

Please note that in order to guard against unlicensed use of the Software, a license key is required to access and enable the Software. Licensee is authorised to use the Software in machine readable, object code form only (subject to clause 4), and Licensee shall not: (a) assign, sublicense, sell, distribute, transfer,

pledge, lease, rent, lend, share or export the Software (including any tool, plug-in, SDK or other functionality), the Documentation or Licensee's rights under this EULA; (b) alter or circumvent the license keys or other copy protection mechanisms in the Software (including any tool, plug-in, SDK or other functionality) or reverse engineer, decompile, disassemble or otherwise attempt to discover the source code of the Software (including any tool, plug-in, SDK or other functionality) in each case except as and to the extent that applicable law requires such reverse engineering, decompilation, disassembly or discovery to be permitted and it is not lawful to contract out of such requirement; (c) implement or use any method or mechanism designed to enable product functionality not available in the Software but available in (i) other Foundry products; or (ii) other Foundry releases of the same product; (d) (subject to clause 4) modify, adapt, translate or create derivative works based on the Software (including any tool, plug-in, SDK or other functionality) or Documentation; (e) use, or allow the use of, the Software (including any tool, plug-in, SDK or other functionality) or Documentation on any project other than a project produced by Licensee (an "**Authorized Project**") or to provide a service (whether or not any charge is made) to any third party; (f) allow or permit anyone (other than Licensee and Licensee's authorized employees to the extent they are working on an Authorized Project) to use or have access to the Software (including any tool, plug-in, SDK or other functionality) or Documentation; (g) copy or install the Software (including any tool, plug-in, SDK or other functionality) or Documentation other than as expressly provided for in this EULA; or (h) take any action, or fail to take action, that could adversely affect the trademarks, service marks, patents, trade secrets, copyrights or other intellectual property rights of Foundry or any a Third Party Licensor; (i) use the Software, or permit any third party to use the Software, for any illegal purpose. The Licensee shall maintain appropriate internal security, safeguards and controls which shall be at least as great as the measures the Licensee uses to keep Licensee's own information secure (but in any case, using no less than a reasonable degree of care) to ensure that the Software is not used in any way which is in violation of this clause 3 or of this Agreement. For the avoidance of doubt, Licensee remains fully liable for any unauthorized use of the Software by its staff. For purposes of this clause 3, the term "Software" shall include any derivatives of the Software.

Notwithstanding clause 3(b) above, where the reduction of the Software to human readable form is necessary for the purposes of integrating the operation of the Software with the operation of other software or systems used by the Licensee in accordance with section 50B of the Copyright Designs and Patents Act 1988 (or any analogous legislation in other jurisdictions), prior to reducing the Software to human readable form (whether by reverse engineering, decompilation or disassembly), the Licensee shall notify Foundry in advance and allow Foundry a reasonable period to either carry out such action as is required to fulfil the integration or provide the information necessary to achieve such integration to the Licensee, and in either case the Licensee shall meet Foundry's reasonable costs in doing so.

Unless Licensee has purchased an Individual License, a Team Login License or an Individual Login License, for other license types, if the Software is moved from one computer to another (or, in the case of an Offline Floating License, from one license server to another), the issuing of replacement or substituted license keys or approval of the new server location, is in Foundry's sole discretion and is subject to and strictly in accordance with Foundry's License Transfer Policy, which is available on Foundry's website and which requires a fee to be paid in certain circumstances. Foundry may from time to time and at its sole discretion vary the terms and conditions of the License Transfer Policy.

The Software and/or any components thereof are not intended to be used for and Licensee shall not use the Software for (or modify the Software such that is suitable for): (i) prohibited practices under the European Union Artificial Intelligence Act (the "**EU AI Act**") or similar legislation, (ii) any use that would result in the Software being considered a high-risk AI system or that otherwise qualifies as "high-risk" under the EU AI Act. Licensee may not remove, alter or obscure any metadata fields which are automatically applied to content generated by the Software which indicate that such content has been artificially generated or manipulated.

PROHIBITED AND HIGH-RISK USES. Licensee shall not use the Software for: (i) any application that would classify the Software as 'high-risk' under EU AI Act or a similar concept in other applicable regulations; (ii) life-critical systems, medical devices, or safety-critical infrastructure; (iii) autonomous weapons systems or mass surveillance applications; (iv) any illegal, immoral or improper use (including inputting illegal, immoral, or inappropriate content or prompts into any AI models in connection with the Software), or (v) any use that would subject Foundry to additional regulatory obligations. Licensee indemnifies Foundry against all costs and liabilities arising from such prohibited and high-risk uses.

Licensee indemnifies Foundry against regulatory fines, costs or expenses related to Licensee's non-compliant use of the Software.

4. SOURCE CODE

Notwithstanding that clause 1 defines "Software" as an object code version and that clause 3 provides that Licensee may use the Software in object code form only:

4.1 if Foundry has agreed to license to Licensee (including by way of providing SDKs, upgrades, updates or enhancements/customization) source code or elements of the source code of the Software, the intellectual property rights in which belong either to Foundry or to a Third Party Licensor ("**Source Code**"), Licensee shall be licensed to use the Source Code as Software on the terms of this EULA and: (a) notwithstanding clause 3 (c), Licensee may use the Source Code at its own risk in any reasonable way for the limited purpose of enhancing its use of the Software solely for, where the Licensee is a business, its own internal business purposes or, where the Licensee is a consumer, domestic and private purposes and in all respects in accordance with this EULA; (b) Licensee shall in respect of the Source Code comply strictly with all other restrictions applying to its use of the Software under this EULA as well as any other restriction or instruction that is communicated to it by Foundry at any time during the Agreement (whether imposed or requested by Foundry or by any Third Party Licensor);

4.2 to the extent that the Software links to or itself incorporates any open source software and/or software libraries ("**OSS Components**") that are provided to Licensee with or as part of the Software, then where such OSS Components are licensed on the terms of an open source software licence that requires Foundry to make the OSS Components available to the Licensee on specific terms (the "**OSS Licence Terms**"), those OSS Components are licensed to Licensee on, and subject to, the terms of the relevant OSS Licence Terms;

4.3 where Foundry is required by any OSS Licence Terms to make the source code of the relevant OSS Component available to the Licensee, Foundry will at any time during the three year period starting on the date of the Agreement, at the request of Licensee and subject to Licensee paying to Foundry a charge that does not exceed Foundry's costs of doing so, provide Licensee with the source code of the relevant OSS Component (the "**OSS Source Code**") in order that Licensee may modify the OSS Component in accordance with the relevant OSS Licence Terms, together (where appropriate) with certain object code of the Software necessary to enable Licensee to re-link any modified OSS Components to the Software (the "**Object**"); and

4.4 notwithstanding any other term of the Agreement, Foundry gives no express or implied warranty, undertaking or indemnity whatsoever in respect of the Source Code, the OSS Components, the OSS Source Code or the Object, all of which are licensed on an "as is" basis, or in respect of any modification of the Source Code, the OSS Components or the OSS Source Code made by Licensee ("**Modification**"). Licensee may not use the Object for any purpose other than its use of the Software in accordance with this EULA. Notwithstanding any other term of the Agreement, Foundry shall have no obligation to provide support, maintenance, upgrades or updates of or in respect of any of the Source Code, the OSS Components (save for any obligations Foundry may have in respect of any elements that form part of the Software as a whole), the OSS Source Code, the Object or any Modification. Licensee shall indemnify Foundry against all liabilities and expenses (including reasonable legal costs) incurred by Foundry in relation to any claim asserting that any Modification infringes the intellectual property rights of any third party.

5. BACK-UP COPY

Licensee may store one copy of the Software and Documentation off-line and off-site in a secured location within the Home Country that is owned or leased by Licensee in order to provide a back-up in the event of destruction by fire, flood, acts of war, acts of nature, vandalism or other incident. In no event may Licensee use the back-up copy of the Software or Documentation to circumvent the usage or other limitations set forth in this EULA.

6. OWNERSHIP

Licensee acknowledges that the Software (including, for the avoidance of doubt, any Source Code that is licensed to Licensee) and Documentation and all related intellectual property rights and other proprietary rights are and shall remain the sole property of Foundry and the Third Party Licensors. Licensee shall not remove, or allow the removal of, any copyright or other proprietary rights notice included in and on the Software or Documentation or take any other action that could adversely affect the property rights of Foundry or any Third Party Licensor. To the extent that Licensee is authorized to make copies of the Software or Documentation under this EULA, Licensee shall reproduce in and on all such copies any copyright and/or other proprietary rights notices provided in and on the materials supplied by Foundry hereunder. Nothing in the Agreement shall be deemed to give Licensee any rights in the trademarks, service marks, patents, trade secrets, confidential information, copyrights or other intellectual property rights of Foundry or any Third

Party Licensor, and Licensee shall be strictly prohibited from using the name, trademarks or service marks of Foundry or any Third Party Licensor in Licensee's promotion or publicity without Foundry's prior express written approval.

Subject to clause 4.3, Foundry undertakes (the "**Undertaking**") to defend Licensee or at Foundry's option settle any claim brought against Licensee alleging that Licensee's possession or use of the Software or Documentation in accordance with the Agreement infringes the intellectual property rights of a third party in the same country as Licensee ("**Claim**") and shall reimburse all reasonable losses, damages, costs (including reasonable legal fees) and expenses incurred by or awarded against Licensee in connection with any such Claim, provided that the Undertaking shall not apply where the Claim in question is attributable to possession or use of the Software or Documentation other than in accordance with the Agreement, or in combination with any hardware, software or service not supplied or specified by Foundry. The Undertaking is conditional on Licensee giving written notice of the Claim to Foundry as soon as reasonably possible, cooperating in the defence of the Claim and not making any admission of liability or taking any step prejudicial to the defence of the Claim. If any Claim is made, or in Foundry's reasonable opinion is likely to be made, against Licensee, Foundry may at its sole option and expense (a) procure for Licensee the right to continue using the Software, (b) modify the Software so that it ceases to be infringing, (c) replace the Software with non-infringing software, or (d) terminate the Agreement immediately by notice in writing to Licensee and refund the License Fee (less a reasonable sum in respect of Licensee's use of the Software to the date of termination) on return of the Software and all copies by Licensee. The Undertaking constitutes Licensee's exclusive remedy and Foundry's only liability in respect of any Claim.

Where the Software integrates with third-party AI models, tools, libraries, or services provided by Third Party Licensor(s), Foundry has not reviewed, approved, confirmed, pre-vetted any such third-party components (including their fitness for a particular purpose), nor does Foundry host, control, or endorse such third-party components. Licensee acknowledges and agrees that use of such components is at the Licensee's own risk and is subject to the Third Party Licensor's terms and conditions, as applicable.

Where the Software enables generation of AI-powered content or workflows ("**AI Outputs**"), ownership of such AI Outputs vests in the Licensee (in so far as applicable law allows), subject to any third-party rights (including any rights of any Third Party Licensors) in the underlying AI models, training data, or other components. Licensee is solely responsible for ensuring that all inputs are legally compliant and appropriate, and for reviewing, validating, and ensuring the accuracy, quality, legality, and appropriateness of all AI Outputs before use or distribution, and Foundry disclaims all warranties and liabilities in relation to AI Outputs (as further set out in clause 13). Licensee shall not input illegal, immoral or inappropriate content to, or use the Software to generate AI Outputs that infringe third-party rights or violate applicable laws.

7. LICENSE FEE

7.1 Licensee acknowledges that the rights granted to Licensee under this EULA are conditional on Licensee's timely payment of the license fee payable to Foundry in connection with the Agreement or, as the case may

be, payable to Foundry's reseller (the "**License Fee**"). Except where the applicable invoice expressly sets out payment in instalments, the License Fee shall be payable in full as one single payment.

7.2 Licensee will be charged and agrees to pay to Foundry or Foundry's authorized reseller (as applicable): (a) the License Fee as notified by Foundry (or its reseller) at the time of the initial purchase of the License; and (b) in respect of any Subscription Auto-renewal Period for a Subscription License, the License Fee as notified by Foundry (or its reseller) on or about the applicable Renewal Date.

7.3 Unless stated otherwise, any Licence Fee notified to the Licensee by Foundry (or its reseller) is exclusive of sales tax, VAT and any other similar taxes, duties or levies, which shall be payable by the Licensee (and the Licensee agrees to pay) in addition to the Licence Fee.

7.4 In the cases of Non-Commercial Licenses or Trial Licenses or specific End of Life Software, the fact that no License Fee may be payable shall not be construed as a waiver by Foundry of any right or remedy available to it in relation to any breach by the Licensee of this EULA or the Agreement, or of any other right or remedy arising under applicable law, all of which are expressly reserved.

8. SUBSCRIPTION LICENSES AND AUTO-RENEWAL

8.1 If Licensee has purchased a Subscription Licence, the License shall be limited to the Initial Subscription Period and any/all Auto-renewal Periods (each as defined below) (together the "**Subscription Period**") after which it shall automatically expire.

8.2 The Subscription Licence shall begin as soon as Foundry accepts Licensee's order by issuing Licensee with a license key (the "**Subscription Start Date**") and shall continue for an initial period of twelve (12) months (the "**Initial Subscription Period**") unless earlier terminated in accordance the terms of this EULA, or unless otherwise communicated by Foundry.

8.3 Unless Licensee opts out of auto-renewal in accordance with clause 8.5 then upon the first anniversary of the Subscription Start Date and each subsequent anniversary (each a "**Subscription Renewal Date**"), Licensee's Subscription Licence shall renew automatically for a further twelve (12) months (each an "**Auto-renewal Period**"). Licensee's Subscription License will continue to auto-renew in this manner until Licensee opts out of auto-renewal or unless earlier terminated in accordance with the terms of this EULA.

8.4 Prior to each Subscription Renewal Date, Foundry shall send one (1) email to advise you that your Subscription License is approaching auto-renewal to the contact email address as provided by Licensee in accordance with clause 24. The reminder email will be sent not less than thirty (30) days prior to the relevant Subscription Renewal Date.

8.5 **Opting Out of Auto-renewal and License Expiry.** If Licensee wishes to opt out of auto-renewal then you must email licenses@foundry.com providing details of the Subscription Licence(s) which you wish to opt out not less than seventy-two (72) hours prior to the relevant Subscription Renewal Date. Provided that

Licensee notifies Foundry in accordance with the provisions of this clause 8.5 then your Subscription License (s) will not auto-renew and shall expire at the end of the then-current Subscription Period.

8.6 **Increases to the License Fee for Subscription Licenses.** Foundry reserves the right to increase the License Fee for Subscription Licenses from time to time provided that it shall provide Licensee with not less than thirty (30) days' notice of any increase prior to the relevant Subscription Renewal Date.

9. CANCELLATIONS

9.1 Licensee may cancel a License within 14 days of the original purchase date to obtain a full refund and Licensee will no longer be able to use the Software from the cancellation date. Licensee's right to obtain a refund will be lost once the Software has been installed.

9.2 Refunds are not payable for cancellations made after such date. This includes Subscription licenses which are subject to the fixed twelve (12) month terms and for which the Licensee may opt out of auto-renewal in accordance with clause 8.

9.3 Cancellations and requests for refunds can be made by contacting Foundry's Sales Support team at licenses@foundry.com.

10. END OF LIFE

10.1 From time to time, Foundry will retire Software product(s) (and any/ all versions) by withdrawing them from the market. Such Software product(s) will be designated as end of life ("**EoL**") and a corresponding announcement (the "**Announcement**") will be published on Foundry's website from time to time (the date of such announcement being the "**EoL Notification Date**"). The announcement will include an EoL date ("**EoL Date**") and other relevant information.

10.2 No new License(s) (including renewals, if applicable) for the EoL Software product(s) will be sold after the EoL Notification Date. Unless otherwise provided for in the Announcement, the EoL Date will be the last day that the EoL Software product(s) is sold, improved, maintained, updated, or supported (subject to the below).

10.3 Unless otherwise provided for in the Announcement, a Licensee with an active Subscription License(s) at the time of the EoL Date will receive certain limited support until the relevant Subscription Period expires.

10.4 In the event of EoL, Foundry advises Licensee to install the latest product releases, security updates, fixes and patches available until the EoL Date, to remain as secure as possible. Older products may not meet security or performance requirements, and Foundry cannot confirm compatibility with future operating system updates after the EoL Date. To the maximum extent permitted by applicable law, Foundry shall not be liable for any form of loss, damage or disruption in respect of the EoL Software product(s) beyond the EoL Date.

10.5 Foundry retains all rights, including intellectual property rights, associated with the EoL Software product(s) beyond the EoL Date. Any use by Licensee of the EoL Software product(s) beyond the EoL Date will continue to be governed by the terms of this Agreement.

11. MAINTENANCE AND SUPPORT

Subject to Licensee's timely payment of the applicable License Fee, each Subscription Licence (and, if sold by Foundry and applicable, each Rental License) shall include access to certain maintenance and support services in accordance with the Maintenance and Support Terms which are available on Foundry's website (the "**Maintenance and Support Terms**"). The specific maintenance and support services offered or made available by Foundry shall be those set out in the table at clause 1.2 of the Maintenance and Support Terms (the "**Maintenance and Support**").

12. TAXES AND DUTIES

Licensee agrees to pay, and indemnify Foundry from claims for, any local, state or national tax (exclusive of taxes based on net income), duty, tariff or other impost related to or arising from the transaction contemplated by the Agreement.

13. LIMITED WARRANTY

13.1 Subject to clause 13.3, Foundry warrants that, for a period of ninety (90) days after Licensee first downloads the Software ("**Warranty Period**"): (a) the Software will, when properly used on an operating system for which it was designed, perform substantially in accordance with the functions described in the Documentation; and (b) that the Documentation correctly describes the operation of the Software in all material respects. If, within the Warranty Period, Licensee notifies Foundry in writing of any defect or fault in the Software as a result of which it fails to perform substantially in accordance with the Documentation, Foundry will, at its sole option, either repair or replace the Software, provided that Licensee makes available all the information that may be necessary to identify, recreate and remedy the defect or fault. This warranty will not apply to, and Foundry shall have no liability for, any defect or fault caused by: (a) unauthorised use of or any amendment made to the Software by any person other than Foundry; and/or (b) use of the Software in conjunction with third party technology. If Licensee is a consumer, the warranty given in this clause is in addition to Licensee's legal rights in relation to any Software or Documentation that is faulty or not as described.

13.2 Foundry does not warrant that the Software or Documentation will meet Licensee's requirements or that Licensee's use of the Software will be uninterrupted or error free. Where the Software includes AI-enabled features, Foundry makes no warranties, express or implied, regarding the accuracy, quality, originality, merchantability, non-infringement or fitness for purpose of any AI Outputs, and Licensee uses such AI Outputs entirely at its own risk.

13.3 If Licensee purchases a license of the Software that is of a fixed term duration, the Warranty Period in clause 13.1 shall apply only to Licensee's first purchase of such license and not to any subsequent renewal(s) even if a renewal involves another download.

13.4 Where consumers based in the European Union use the Software, the safety of the Software is dependent on Licensee's use in accordance with this Agreement and the Documentation.

14. INDEMNIFICATION

14.1 Licensee agrees to indemnify, hold harmless and defend Foundry, the Third Party Licensors and Foundry's and each Third Party Licensor's respective affiliates, officers, directors, shareholders, employees, authorized resellers, agents and other representatives from all claims, defence costs (including, but not limited to, legal fees), judgments, settlements and other expenses arising from or connected with any claim that any authorised or unauthorised use including but not limited to modification of the Software or Documentation by Licensee or any person connected with Licensee or any person provided with the Software by Licensee, infringes the intellectual property rights or other proprietary rights of any third party, including without limitation any claims arising from Licensee's input into, or outputs from any AI-enabled features (including Licensee's creation, use or distribution of any AI Outputs) in connection with the Software.

14.2 Licensee agrees to indemnify, hold harmless and defend Foundry, the Third Party Licensors and Foundry's and each Third Party Licensor's respective affiliates, officers, directors, shareholders, employees, authorized resellers, agents and other representatives from all claims, defence costs (including, but not limited to, legal fees), judgments, settlements and other expenses arising from or connected with any claim that any authorised or unauthorised use (including but not limited to modification) of the Software or Documentation, by Licensee, any person connected with Licensee or any person provided with the Software by Licensee, causes any of them or any third party (including but not limited to those viewing a product of the Software) to suffer harm or loss of any kind.

15. LIMITATION OF LIABILITY

15.1 **To Business Users:** This sub-clause applies where Licensee is a business user (and for these purposes any Licensee that is not a consumer shall be treated as a business user). Licensee acknowledges that the Software has not been developed to meet its individual requirements, and that it is therefore Licensee's responsibility to ensure that the facilities and functions of the Software as described in the Documentation meet such Licensee requirements. The Software and Documentation is supplied only for Licensee's internal use for its business, and not for any re-sale purposes or for the provision of the Software (whether directly or indirectly) to third parties. Foundry shall not under any circumstances whatever be liable to Licensee, its affiliates, officers, directors, shareholders, employees, agents or other representatives, whether in contract, tort (including negligence), breach of statutory duty, or otherwise, arising under or in connection with the Agreement for loss of profits, sales, business, or revenue, business interruption, loss of anticipated savings,

loss or corruption of data or information, loss of business opportunity, goodwill or reputation (in each case whether the loss is direct or indirect) or any indirect or consequential loss or damage. In respect of any other losses, Foundry's maximum aggregate liability under or in connection with the Agreement whether in contract, tort (including negligence) or otherwise, shall in all circumstances be limited to the greater of US$5,000 (five thousand USD) and a sum equal to the License Fee. Nothing in the Agreement shall limit or exclude Foundry's liability for death or personal injury resulting from our negligence, for fraud or fraudulent misrepresentation or for any other liability that cannot be excluded or limited by applicable law. This EULA sets out the full extent of our obligations and liabilities in respect of the supply of the Software and Documentation. Except as expressly stated in writing in this EULA, there are no conditions, warranties, representations or other terms, express or implied, that are binding on Foundry. Any condition, warranty, representation or other term concerning the supply of the Software and Documentation which might otherwise be implied into, or incorporated in, the Agreement, whether by statute, common law or otherwise, is excluded to the fullest extent permitted by law.

15.2 **To Consumers:** This sub-clause applies where Licensee is a consumer. Licensee acknowledges that the Software has not been developed to meet Licensee's individual requirements, and that it is therefore Licensee's responsibility to ensure that the facilities and functions of the Software as described in the Documentation meet such Licensee requirements. The Software and Documentation are only supplied for Licensee's domestic and private use. Licensee agrees not to use the Software and Documentation for any commercial, business or re-sale purposes, and Foundry has no liability to Licensee for any loss of profit, loss of business, business interruption, loss of business opportunity or loss or harm caused to property or data used for professional purposes. Foundry is only responsible for loss or damage suffered by Licensee that is a foreseeable result of Foundry's breach of the Agreement or its negligence but Foundry is not responsible for any loss or damage that is not foreseeable. Our maximum aggregate liability under or in connection with the Agreement, whether in contract, tort (including negligence) or otherwise, shall in all circumstances be limited to a sum equal to the greater of US$5,000 (five thousand USD) and a sum equal to the latest annual License Fee. Nothing in the Agreement shall limit or exclude Foundry's liability for death or personal injury resulting from our negligence, for fraud or fraudulent misrepresentation or for any other liability that cannot be excluded or limited by applicable law.

15.3 Where applicable, Foundry's liability under the EU Product Liability Directive is subject to the following conditions: (a) Licensee must prove that a defect in the Software directly caused the claimed damage; (b) Licensee must demonstrate that the Software was used strictly in accordance with this EULA and Documentation; (c) Liability is excluded where damage results from Licensee's failure to install available updates within 90 days of notification; (d) Liability is excluded where damage results from use of the Software in combination with third-party software, hardware, or services not approved by Foundry; (e) Foundry's maximum aggregate liability under the Directive shall not exceed the greater of €100,000 or three times the total License Fees paid by Licensee in the 12 months preceding the claim; and (f) Claims must be brought within 3 years of discovery of the defect or damage, whichever is earlier.

16. TERM; TERMINATION

16.1 The Agreement is effective upon Licensee's download of the Software, and the Agreement will remain in effect until termination or expiry. Licensee may terminate the Agreement on written notice to Foundry if Foundry is in material breach of this Agreement and fails to cure the breach within 10 (ten) working days of receiving notice of such breach. If Licensee breaches the Agreement, Foundry may terminate the License immediately by notice to Licensee.

16.2 If the Agreement expires or is terminated, the License will cease immediately and Licensee will immediately cease use of any Software and Documentation and either return to Foundry all copies of the Software and Documentation in Licensee's possession, custody or power or, if Foundry directs in writing, destroy all such copies. In the latter case, if requested by Foundry, Licensee shall provide Foundry with a certificate confirming that such destruction has been completed within 10 days.

16.3 Foundry reserves the right to terminate and/or suspend the License as it deems reasonable in its sole discretion by notice to Licensee if: (a) it becomes aware that Licensee has failed to pay any sum due either to Foundry or to a reseller of Foundry, either in connection with the Agreement, or in connection with any other Software license to use any product(s) of Foundry, and/or in connection with any other agreed terms between Foundry and the Licensee in connection with the Software, and/or in connection with any Maintenance and Support; or (b) the Licensee is otherwise in breach of or fails to comply with any term of the Agreement and/or with any other agreed terms between Foundry and the Licensee.

16.4 Foundry may also terminate this Agreement if Licensee becomes subject to bankruptcy proceedings, becomes insolvent, or makes an arrangement with Licensee's creditors. This Agreement, and for the avoidance of doubt the Licenses to which it relates, will terminate automatically without further notice or action by Foundry if Licensee goes into liquidation. In the event of termination under this sub-clause, Foundry reserves its rights in full to any amounts due and not paid by the Licensee, including any interest payments, and to any other remedies whether under law, contract, statute or otherwise.

17. CONFIDENTIALITY

17.1 Licensee agrees that the Software (including, for the avoidance of doubt, any Source Code that is licensed to Licensee) and Documentation are proprietary to and the confidential information of Foundry or, as the case may be, the Third Party Licensors, and that all such information and any related communications (collectively, "**Confidential Information**") are confidential and a fundamental and important trade secret of Foundry and/or the Third Party Licensors. If Licensee is a business user, Licensee shall disclose Confidential Information only to Licensee's employees who are working on an Authorized Project and have a "need-to-know" such Confidential Information for the purposes of that Authorized Project, and shall advise any recipients of Confidential Information that it is to be used only as expressly authorized in the Agreement. Licensee shall not disclose Confidential Information or otherwise make any Confidential Information available to any other of Licensee's employees or to any third parties without the express written consent of

Foundry. Licensee agrees to segregate, to the extent it can be reasonably done, the Confidential Information from the confidential information and materials of others in order to prevent commingling. Licensee shall take reasonable security measures, which measures shall be at least as great as the measures Licensee uses to keep Licensee's own confidential information secure (but in any case using no less than a reasonable degree of care), to hold the Software, Documentation and any other Confidential Information in strict confidence and safe custody. Foundry may request, in which case Licensee agrees to comply with, certain reasonable security measures as part of the use of the Software and Documentation. This clause shall not apply to any information that is in or comes into the public domain (other than as a result of the Licensee's breach of its obligations under the Agreement), or was in Licensee's lawful possession before receipt or which Licensee develops independently and without breach of this clause. Licensee acknowledges that monetary damages may not be a sufficient remedy for unauthorized disclosure of Confidential Information, and that Foundry shall be entitled, without waiving any other rights or remedies, to such injunctive or other equitable relief as may be deemed proper by a court of competent jurisdiction.

17.2 Licensee acknowledges that Foundry may provide Licensee's identity to any Third Party Licensor for legitimate purposes, including, but not limited to, that Third Party Licensor's monitoring of Licensee's use of its intellectual property rights and enforcement of its license terms.

18. FEEDBACK

Licensee may, but is not required to, provide suggestions, comments, ideas, or know-how, in any form, to Foundry's products, services or technology ("**Feedback**"). Any such Feedback shall be owned in full by Foundry and Foundry may use and incorporate such Feedback for any purpose, without payment of royalties or other consideration to Licensee. There shall also be no obligation to provide attribution for use of Feedback.

19. INSPECTION AND INFORMATION

19.1 Licensee shall advise Foundry on demand of all locations where the Software or Documentation is used or stored. Licensee shall permit Foundry and/or its authorized agents to audit all such locations during normal business hours and on reasonable advance notice, and in the case of Enterprise Licenses, Licensee shall provide Foundry with accurate records of the total number of users who have accessed or used the Software, at any time upon Foundry or its authorized agent's request.

19.2 The Software may include mechanisms to access and collect limited information from computer(s) on which it is installed and from any IT systems to which those computer(s) may be connected (including any system registry files) and transmit it to Foundry and/or its resellers, including the ability to locally cache such information on such computers. Such information (the "**Information**") may include: details of relevant license(s) to Foundry products, details of computer and network equipment, details of the operating system (s) in use on such computer equipment, email domain relating to owners of such computer and network equipment, the location of the computer(s) on which the Software is installed and the profile and extent of

use of the different elements of the Software and other Foundry software. Foundry may use the Information to (a) model the profiles of usage, hardware and operating systems in use collectively across its customer base in order to focus development and support, (b) to provide targeted support to individual customers, (c) to ensure that the usage of the Software by Licensee is in accordance with the Agreement and does not exceed any user number or other limits on its use, (d) to confirm the identity of Licensee, to identify unlicensed use of the Software (including use of pirated or other unlicensed copies of the Software) and to assist Foundry (and its resellers and any enforcement bodies) in contacting any unlicensed users of the Software and seeking to terminate unlicensed use of the Software, and (e) to advise Licensee about any maintenance and/or service issues such as available upgrades. To the extent that any Information constitutes personal data for the purposes of the General Data Protection Regulation (EU) 2016/679 ("**EU GDPR**") and the version of the EU GDPR retained in UK domestic law as further defined in the Data Protection Act 2018 ("**UK GDPR**"), in each case, as amended, superseded or replaced from time to time (as applicable, "**GDPR**") it shall be processed in accordance with the GDPR and with Foundry's Privacy Notice (seE https://www.foundry.com/privacy-notice), as may be updated by Foundry from time to time. Licensee undertakes to make all of its users of the Software (including its Licensee Representatives, as defined in the Maintenance and Support Terms, as applicable) aware of, and procure they all agree to, the uses which Foundry will make of the Information and of the terms of Foundry's Privacy Policy.

19.3 By downloading or using the Software, you (i) warrant that you are entitled to control access to the computer(s) on which the Software is downloaded and any IT systems to which they may be connected, and (ii) irrevocably authorise Foundry (through the use of the Software) to access such computer(s) and IT systems (including any system registry files) and collect the Information from them and to transmit that Information to Foundry and its resellers (and any enforcement bodies) and use it for the purposes identified at clause 19.2(a) to (e) above.

20. DEFECT NOTIFICATION AND LICENSEE OBLIGATIONS TO CONSUMERS

20.1 Notification Procedures: Foundry will notify its Licensees and other consumer customers, as applicable, of material security vulnerabilities through reasonable means, which may include email, website posting, or in-software notifications. Where the Software is made available to consumers in the European Union, Foundry will comply with notification obligations under the EU General Product Safety Regulation (GPSR) 2023/988, including notifying relevant authorities where required by law. Notification timing is at Foundry's discretion based on patch availability and risk assessment. Foundry has no obligation to provide individual notifications for non-material issues, to be determined in Foundry's sole discretion.

20.2 Licensee Response Obligations: Licensee must implement provided security updates (as described further in clause 1.6.2 of the Maintenance and Support Terms) within 90 days of notification to maintain warranty and support coverage. Licensee must maintain current contact information with Foundry for security notifications. Failure to respond to critical security or safety notifications may result in license suspension. Licensee shall promptly report to Foundry via our Support Portal any safety concerns, incidents, or accidents involving the Software.

20.3 Liability Limitations: Foundry's liability to the Licensee for notification delays is limited to direct damages not exceeding the Licensee's annual (one-year) License Fee. Foundry has no liability for damages resulting from Licensee's failure to act on security notifications.

21. U.S. GOVERNMENT LICENSE RIGHTS

All Software, including all components thereof, and Documentation qualify as "commercial items," as that term is defined at Federal Acquisition Regulation ("**FAR**") (48 C.F.R.) 2.101, consisting of "commercial computer software" and "commercial computer software documentation" as such terms are used in FAR 12.212. Consistent with FAR 12.212 and DoD FAR Supp. 227.7202-1 through 227.7202-4, and notwithstanding any other FAR or other contractual clause to the contrary in any agreement into which this Agreement may be incorporated, a government end user will acquire the Software and Documentation with only those rights set forth in this Agreement. Use of either the Software or Documentation or both constitutes agreement by the government that all Software and Documentation are "commercial computer software" and "commercial computer software documentation," and constitutes acceptance of the rights and restrictions herein. The Software is the subject of the following notices:

* Copyright © 2001 - 2026 The Foundry Visionmongers Ltd. All Rights Reserved.

* Unpublished- rights reserved under the Copyright Laws of the United Kingdom.

22. SURVIVAL

Clause 6, clause 7 and clauses 12 to 27 inclusive shall survive any termination or expiration of the Agreement.

23. IMPORT/EXPORT CONTROLS AND SANCTIONS

To the extent that any Software made available under the Agreement is subject to restrictions upon export and/or re-export from any applicable jurisdiction (including the United States), Licensee agrees to comply with, and not act or fail to act in any way that would violate, applicable international, national, state, regional or local laws and regulations, including, without limitation, the U.S. Export Administration Act and the Export Administration Regulations, the regulations of the U.S. Department of Treasury Office of Foreign Assets Control, the International Traffic in Arms regulations, the United States Foreign Corrupt Practices Act, the UK Export Control Act 2002 and the UK Export Control Order 2008 (collectively, "**Export Laws**") as those laws may be amended or otherwise modified from time to time, and neither Foundry nor Licensee shall be required under the Agreement to act or fail to act in any way which it believes in good faith will violate any such laws or regulations. Without limiting the foregoing, Licensee agrees that it will not sell, supply, transfer, export or re-export, directly or indirectly, Foundry's Software or related products and services, or any commodity, technology, technical data, software or service that incorporates, contains or is a direct product of Foundry's Software, products and/or services, (i) in violation of the Export Laws; (ii) to any country for

which an export license or other governmental approval is required at the time of export, without first obtaining all necessary export licenses or other approvals; (iii) to any country, or national or resident of a country, to which trade is embargoed by the United States and/or sanctioned by the United Kingdom or the European Union; (iv) to any person or firm on any government agency's list of blocked, denied or barred persons or entities, including but not limited to the U.S. Department of Commerce's Denied Persons List and Entities List, and the U.S Treasury Department's Specially Designated Nationals List; or (v) to the Russian Federation and/or Belarus or for use in the Russian Federation and/or Belarus, or (vi) for use in any nuclear, chemical or biological weapons, or missile technology end-use. Licensee warrants and represents that it is not subject to any Export Law, sanction and/or other form of trading compliance restriction or regulation which may limit or prohibit its right to download, install and/or use the Software in accordance with this EULA.

24. MISCELLANEOUS

Unless Licensee is a consumer, the Agreement is the exclusive agreement between the parties concerning its subject matter and supersedes any and all prior oral or written agreements, negotiations, or other dealings between the parties concerning such subject matter. Licensee acknowledges that Licensee has not relied upon any representation or collateral warranty not recorded in the Agreement inducing it to enter into the Agreement.

The Agreement may be modified only in writing, by Foundry, at any time. Any such modified terms will become effective upon posting a revised version on Foundry's website. It is your responsibility to check the Foundry website regularly for modifications to this Agreement.

Foundry shall not be in breach of this Agreement nor liable for delay in performing, or failure to perform, any of its obligations under this Agreement if such delay or failure result from events, circumstances or causes beyond its reasonable control.

The failure of either party to enforce any rights granted under the Agreement or to take action against the other party in the event of any such breach shall not be deemed a waiver by that party as to subsequent enforcement of rights or subsequent actions in the event of future breaches.

The Agreement and any dispute or claim arising out of or in connection with it or its subject matter or formation (including, unless Licensee is a consumer, non-contractual disputes or claims) shall be governed by, and construed in accordance with English Law and the parties irrevocably submit to the non-exclusive jurisdiction of the English Courts, subject to any right that a consumer may have to bring proceedings or to have proceedings brought against them in a different jurisdiction.

If Foundry fails to insist that Licensee performs any obligation under the Agreement, or delays in doing so, that will not mean that Foundry has waived its rights or any other right or remedy, nor shall it prevent or restrict the further exercise of that or any other right or remedy. No single or partial exercise of such right or remedy shall prevent or restrict the further exercise of that or any other right or remedy by Foundry.

If any provision or part-provision of the Agreement is or becomes invalid, illegal or unenforceable, it shall be deemed modified to the minimum extent necessary to make it valid, legal and enforceable. If such modification is not possible, the relevant provision or part-provision shall be deemed deleted. Any modification to or deletion of a provision or part-provision under this clause shall not affect the validity and enforceability of the rest of the Agreement.

Unless Licensee is a consumer, Licensee agrees that Foundry may refer to Licensee as a client or a user of the Software, may display its logo(s) for this purpose and may publish quotations and testimonials from Licensee, its directors, partners, officers or employees. Foundry agrees to promptly cease any such use on Licensee's written request.

Foundry and Licensee intend that each Third Party Licensor may enforce against Licensee under the Contracts (Rights of Third Parties) Act 1999 (the "Act") any obligation owed by Licensee to Foundry under this EULA (or any associated Third Party Licensor terms) that is capable of application to any proprietary or other right of that Third Party Licensor in or in relation to the Software. Foundry and Licensee reserve the right under section 2(3)(a) of the Act to rescind, terminate or vary this EULA without the consent of any Third Party Licensor.

Email Address for General Correspondence: Licensee shall notify Foundry of an email address for the provision of any correspondence in connection with the Agreement and shall notify Foundry via licenses@foundry.com of any change(s) to that email address. Please note, the email address you provide is important for the provision of information and notices to you, including in relation to the auto-renewal of any Subscription License (if applicable). It is your responsibility to provide and maintain an up-to-date email address. Foundry shall store details of and may use the email address to notify you in accordance with the terms of this Agreement.

25. NOTICES

Any legal notices from the Licensee to Foundry should be sent via postal mail or international delivery to The Foundry Visionmongers Ltd, 5 Golden Square, London, W1F 9HT, addressed to The General Counsel. These notices will be effective upon receipt by Foundry. Additionally, a copy of the notice should be emailed to Foundry at legal@foundry.com. Unless otherwise agreed in writing, any notices from Foundry to the Licensee will be provided either (a) by email to the registered email address associated with the Licensee's account, or (b) in any other reasonable manner that involves specific notification to the Licensee. Notices from Foundry to the Licensee will be effective (i) one day after being sent by email, and (ii) five days after being posted or sent by international delivery service. The Licensee agrees that service of process may be effected on the Licensee by registered mail sent to the last known address on record with Foundry, if permitted by applicable law.

26. COMPLAINTS & ONLINE DISPUTE RESOLUTION PLATFORM

We hope that you are satisfied with any Software purchase made or service received from Foundry, but if you have a complaint, in the first instance, please contact us on licenses@foundry.com or through our Support Portal: https://support.foundry.com/hc/en-us (for technical support and bug reports), or you can request a call back from the Sales team here: https://www.foundry.com/contact-us.

27. COMPLAINTS & ONLINE DISPUTE RESOLUTION PLATFORM

The Agreement and any dispute or claim (including non-contractual disputes or claims) arising out of or in connection with it or its subject matter or formation shall be governed by and construed in accordance with the law of England and Wales and the courts of England and Wales shall have non-exclusive jurisdiction to settle any dispute or claim (including non-contractual disputes or claims) arising out of or in connection with the Agreement or its subject matter or formation.

Last updated March 2026.

Copyright © March 2026 The Foundry Visionmongers Ltd.

All Rights Reserved. Do not duplicate.