



REFERENCE GUIDE

VERSION 2.1V5

Katana™ Reference Guide. Copyright © 2016 The Foundry Visionmongers Ltd. All Rights Reserved. Use of this document and the Katana software is subject to an End User License Agreement (the "EULA"), the terms of which are incorporated herein by reference. This document and the Katana software may be used or copied only in accordance with the terms of the EULA. This document, the Katana software and all intellectual property rights relating thereto are and shall remain the sole property of The Foundry Visionmongers Ltd. ("The Foundry") and/or The Foundry's licensors.

The EULA can be read in the Katana User Guide.

The 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 The Foundry. To the extent that the EULA authorizes the making of copies of this Reference 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 The Foundry and/or The 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):

Katana™ software © 2016 The Foundry Visionmongers Ltd. All Rights Reserved. Katana™ is a trademark of The Foundry Visionmongers Ltd.

Sony Pictures Imageworks is a trademark of Sony Pictures Imageworks.

Mudbox™ is a trademark of Autodesk, Inc.

RenderMan ® is a registered trademark of Pixar.

In addition to those names set forth on this page, the names of other actual companies and products mentioned in this Reference 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 document.

Linux ® is a registered trademark of Linus Torvalds.

The Foundry

5 Golden Square,

London,

W1F 7HT

Rev: Tuesday, August 23, 2016

CONTENTS

Preface

About this Manual	11
Getting Help	11
Viewing Additional Help	11
Contacting Customer Support	12

Common Parameter Widgets

Asset (assetIdInput) and File Path (fileInput) Widget Types	13
CEL Statement (cel) Widget Type	14
Color (color) Widget Type	16
Common 2D Node (node2d) Widget Type	17
New Scene Graph Location (newScenegraphLocation) Widget Type	18
Locations (locations) Widget Type	19
Look File Pass Name (lookfilePassname) Widget Type	20
Rectangle (rectangle) Widget Type	21
Scene Graph Location (scenegraphLocation) Widget Type	21
Scene Graph Locations (scenegraphLocationArray)Widget Type	22
Transform Controls Widget Type	23
Transform Tools Widget Type	25

2D Nodes

ImageAddMix	26
ImageAtop	27
ImageAverage	31
ImageBackgroundColor	34
ImageBlack	34
ImageBlur	35
ImageBrightness	37

ImageChannels	37
ImageCheckerboard	39
ImageClamp	40
ImageColor	42
ImageContrast	43
ImageCoordinate	44
ImageCrop	44
ImageDifference	45
ImageDistort	49
ImageDivide	51
ImageExclusion	55
ImageExposure	58
ImageFade	59
ImageFrom	59
ImageGain	63
ImageGamma	64
ImageGeometric	65
ImageHypot	69
ImageIn	72
ImageInvert	75
ImageLevels	75
ImageMatte	77
ImageMatteMix	80
ImageMax	81
ImageMerge	84
ImageMin	87
ImageMinus	90
ImageMix	93
ImageMultiply	93

ImageOrient	97
ImageOut	98
ImageOver	101
ImagePlus	104
ImagePosition	107
ImagePremultiply	108
ImageRamp	108
ImageRead	112
ImageReformat	116
ImageSaturation	119
ImageScreen	120
ImageText	123
ImageThreshold	126
ImageTransform2D	127
ImageUnder	132
ImageUnpremultiply	136
ImageWhite	136
ImageWrite	137
ImageZMerge	146
OCIOCDLTransform	147
OCIOColorSpace	147
OCIODisplay	148
OCIOFileTransform	149
OCIOLogConvert	150
OCIOLookTransform	150

3D Nodes

AimConstraint	152
Alembic_In	153
ArnoldGlobalSettings	156

ArnoldObjectSettings	156
ArnoldOutputChannelDefine	157
ArnoldShadingNode	157
AttributeCopy	158
AttributeEditor	160
AttributeFile_In	161
AttributeModifierDefine	163
AttributeScript	165
AttributeSet	168
AttributeUpgrade	170
Backdrop	172
BillboardConstraint	172
BoundsAdjust	173
CameraClippingPlaneEdit	175
CameraCreate	175
CameraImagePlaneCreate	177
CameraScreenWindowConstraint	180
ClippingConstraint	181
CollectionCreate	182
ConstraintCache	183
ConstraintListEdit	184
ConstraintResolve	184
CoordinateSystemDefine	185
DollyConstraint	186
FaceSetCreate	187
FOVConstraint	188
GenericOp	189
GroupMerge	195
HierarchyCopy	196

Importomatic	197
Right-Click Menu	198
InfoCreate	198
Isolate	199
LightCreate	201
LightLink	203
LightLinkEdit	206
LightLinkResolve	208
LightLinkSetup	208
LightListEdit	211
LocationCreate	212
LocationGenerate	212
LodGroupCreate	213
LodSelect	214
LodValuesAssign	215
LookFileBake	217
LookFileLightAndConstraintActivator	219
LookFileManager	219
LookFileMaterialsIn	220
LookFileMaterialsOut	221
LookFileMultiBake	221
LookFileOverrideEnable	223
LookFileResolve	225
Material	225
MaterialResolve	228
MaterialStack	228
Merge	229
NetworkMaterial	231
NetworkMaterialInterfaceControls	232
NetworkMaterialParameterEdit	236

NetworkMaterialSplice	236
OpResolve	238
OpScript	239
OrientConstraint	243
ParentChildConstraint	245
PointConstraint	245
PrimitiveCreate	247
PrmanGlobalSettings	248
PrmanObjectSettings	248
PrmanOutputChannelDefine	249
PrmanShadingNode	250
Prune	251
ReflectionConstraint	252
Rename	252
Render	253
RendererProceduralArgs	256
RenderOutputDefine	265
ReverseNormals	270
ScaleConstraint	270
ScenegraphGeneratorResolve	271
ScenegraphGeneratorSetup	271
ScenegraphXml_In	273
ScreenCoordinateConstraint	274
ShadingNodeArrayConnector	276
ShadingNodeSubnet	276
ShadowBranch	276
TeapotCreate	278
Transform3D	279
TransformEdit	279

VelocityApply	280
ZoomToRect	281

Miscellaneous Nodes

ArnoldLiveRenderSettings	283
DependencyMerge	283
Dot	284
Gaffer	284
GafferThree	287
Group	299
GroupStack	299
InteractiveRenderFilters	300
LiveGroup	301
LiveGroupStack	302
LookFileAssign	303
LookFileGlobalsAssign	304
MaterialAssign	305
NonpersistentSwitch	306
PonyFarm	306
PonyStack	309
PrmanLiveRenderSettings	309
RendererProceduralAssign	309
RenderScript	310
RenderSettings	313
ScenegraphObjectSettings	315
ShadowManager	315
Switch	316
Teleport	317
TimeOffset	317
VariableDelete	317

VariableEnabledGroup	318
VariableSet	318
VariableSwitch	318
ViewerMaterialEdit	319
ViewerObjectSettings	320
VisibilityAssign	322

Preface

Katana is a 3D application specifically designed for the needs of look development and lighting in an asset based pipeline. Originally developed at Sony Pictures Imageworks, Katana has been their core tool for look development and lighting for all their productions since "Spider-Man 3", "Beowulf", and "Surf's Up!".

Katana provides a very general framework for efficient look development and lighting, with the goals of scalability, flexibility, and supporting an asset based pipeline.

About this Manual

This manual aims to provide a complete reference for all the controls within each node in Katana. It does not give you any instructions on using Katana. For details on installing and using Katana, read the *Katana User Guide*.



NOTE: For the most up-to-date information, please see the Katana product page and the latest *Katana Reference Guide* on our web site at www.thefoundry.co.uk.

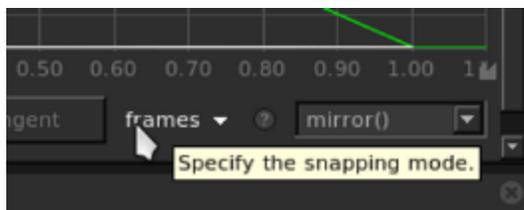
Getting Help

If you can't find what you need in this document, there are other sources of help available to you for all aspects of Katana and its operation.

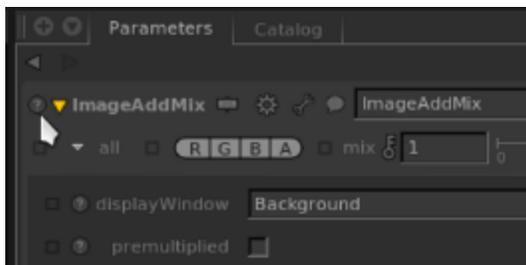
Viewing Additional Help

Katana features several forms of help:

- Some controls offer concise instructions in the form of tooltips. To display the tooltips, move your mouse pointer over a control or node parameter.



- Many **Parameters** tabs include contextual descriptions of the node's parameters. To display these descriptions, click the ? icon.



- Finally, you can click the **Help** menu to access the following:
 - **User Guide** - the Katana user guide, which is aimed at users of all levels and covers most operations inside Katana.
 - **Technical Guide** - a more technical overview of Katana, aimed at those with a more technical role such as pipeline engineers.
 - **Reference Guide** - a reference list of the nodes, their parameters, and how to use them.
 - **Node Reference** - a reference list of the nodes, their parameters, and how to use them in HTML format.
 - **Documentation** - a full list of all the accompanying documents and examples.
 - **API Reference** - information on Katana APIs.
 - **Examples** - a list of accompanying example files.

Contacting Customer Support

Should questions arise that the guides or in-application help system fail to address, you can contact Customer Support directly by e-mail at support@thefoundry.co.uk.

Please note that technical support is only provided during UK hours.

Common Parameter Widgets

These widget groups are common to many nodes in Katana and are outlined here to save undue repetition throughout the *Katana Reference Guide*. For more information regarding the addition of user parameters and specific widget types, refer to *User Parameters and Widget Types* in the *Katana User Guide*, or *Widget Types* and *Widget Options* within the *Args Files in Shaders* chapter, in the *Katana Technical Guide*.

Asset (assetIdInput) and File Path (fileInput) Widget Types

The **Asset** and **File Path** widget types allow you to navigate to assets and files on your file system. Several node types that ship with Katana use the **Asset** and **File Path** widget types for parameters of various names, for example: **abcAsset**, **file**, **filePath**, **lookfile**, **procedural**, **saveTo**, and **source**.

Parameters that use the **Asset** and **File Path** widget types can be found on the following types of nodes:

- [Alembic_In](#)
- [AttributeFile_In](#)
- [CameraImagePlaneCreate](#)
- [GafferThree](#)
- [ImageCoordinate](#)
- [ImageRead](#)
- [LiveGroup](#)
- [LiveGroupStack](#)
- [LookFileAssign](#)
- [LookFileBake](#)
- [LookFileGlobalsAssign](#)
- [LookFileMaterialsIn](#)
- [LookFileMaterialsOut](#)
- [LookFileMultiBake](#)
- [LookFileOverrideEnable](#)
- [Material](#)
- [RendererProceduralArgs](#)
- [RenderOutputDefine](#)
- [ScenegraphGeneratorSetup](#)
- [ScenegraphXml_In](#)

Menu Command	Description
▼	
Browse...	Brings up the file browser or your studio's asset management browser and enables you to select the asset to use.
Set Node Name From Path	Changes the name of the node to match the filename but without the path or extension.

CEL Statement (cel) Widget Type

The **CEL Statement** widget type allows you to build and edit CEL statements that are stored in string parameters on nodes, allowing you to select scene graph locations using paths, collections, or through custom CEL expressions. Several node types that ship with Katana use the **CEL Statement** widget type for parameters of various names, for example: **CEL**, **cel**, **toCel**, **fromCel**, **celSelection**, **disableAt**, and **exclusivity**.

Parameters that use the **CEL Statement** widget type can be found on the following types of nodes:

- [AttributeCopy](#)
- [AttributeEditor](#)
- [AttributeFile_In](#)
- [AttributeModifierDefine](#)
- [AttributeScript](#)
- [AttributeSet](#)
- [AttributeUpgrade](#)
- [CollectionCreate](#)
- [GafferThree](#)
- [GenericOp](#)
- [LightLink](#)
- [LightLinkEdit](#)
- [LightLinkSetup](#)
- [LocationGenerate](#)
- [LodSelect](#)
- [LodValuesAssign](#)
- [LookFileAssign](#)
- [Material](#)
- [OpScript](#)
- [Prune](#)
- [RendererProceduralArgs](#)
- [RendererProceduralAssign](#)
- [ReverseNormals](#)
- [ScenegraphObjectSettings](#)
- [VelocityApply](#)
- [ViewerObjectSettings](#)
- [VisibilityAssign](#)

Menu Command	Description
Add Statements	<ul style="list-style-type: none"> • Paths - Adds a Paths list to this CEL parameter. • Collections - Adds a Collections list to this CEL parameter. • Custom - Adds a Custom parameter to this CEL parameter. • Append Scene Graph Selection - Adds a Paths list to this CEL parameter and places selected scene graph locations in the new list. • Replace With Scene Graph Selection - Removes any parameters within this CEL parameter and creates a new Paths list and populates it with any selected scene graph locations. • Copy CEL Statement As Text - Copies this CEL statement to the clipboard. • Paste CEL Statement - Removes any parameters within this CEL parameter and pastes the CEL statement in the clipboard to this parameter. • Replace With Parameter Expression - Converts the current CEL parameter into an expression.
Paths > Action	

Menu Command	Description
Add Scenegraph Selection	Adds the currently selected scene graph location to this list.
Remove Scenegraph Selection	Removes the currently selected scene graph location from this list.
Remove Selected Paths	Removes the path(s), selected in this Paths list, from this list.
Select All	Selects all the paths in this list.
Select Selected Paths In Scenegraph	Selects the scene graph locations of the selected paths in this list.
Copy Selected Paths to Clipboard	Copies the selected paths from this list to the clipboard.
Show Extended View...	Brings up a dialog with the contents of this Paths list.
Collections > Action	
Add Collections From Scenegraph Selection...	Brings up a dialog box with a list of the collections from the currently selected scene graph locations. You can then select from these collections to add them to this list.
Add Scene Root Collections...	Brings up a dialog box populated with the collections currently on /root . You can then select from these collections to add them to this list.
Remove Selected Paths	Removes the selected collection(s) from this list.
Select All	Selects all the collections in this list.
Copy Selected Paths to Clipboard	Copies all the selected collections and their paths to the clipboard.
"Find And Select" Selected Items...	
Union dropdown	<ul style="list-style-type: none"> • Union • Difference • Intersect <p>Does not occur in all nodes with CEL widgets. Only occurs on additional statements added to the widget after an initial statement.</p>

Color (color) Widget Type

The **Color** widget type allows you to pick a color by specifying RGB or RGBA component values directly in the **Parameters** tab, or through a color picker dialog. Several node types that ship with Katana use the **Color** widget type for parameters of various names, for example: **bottomLeft**, **bottomRight**, **color**, **constantColor**, **fadeToColor**, **gamma**, and **previewColor**.

Parameters that use the **Color** widget type can be found on the following types of nodes:

- [Gaffer](#)
- [GafferThree](#)
- [ImageAddMix](#)
- [ImageBackgroundColor](#)
- [ImageBlack](#)
- [ImageChannels](#)
- [ImageCheckerboard](#)
- [ImageColor](#)
- [ImageClamp](#)
- [ImageContrast](#)
- [ImageFade](#)
- [ImageGamma](#)
- [ImageInvert](#)
- [ImageLevels](#)
- [ImageRamp](#)
- [ImageText](#)
- [ImageThreshold](#)
- [ImageWhite](#)
- [LightCreate](#)
- [ShadowBranch](#)
- [ViewerObjectSettings](#)

Menu Command	Description
color	The color (RGBA values) for the given parameter.
	Picks the color (RGBA) values of the selection.
	
Average	Sets the color picker to use the average values.
Min	Sets the color picker to use the minimum values.
Max	Sets the color picker to use the maximum values.
Front	Sets the color picker to use the front values.
Back	Sets the color picker to use the back values.
Auto-Disable Upon Release	Toggle the ability to automatically disable the picker on mouse-button release.
color > RGB	
red	Sets the red value of the pixels.

green	Sets the green value of the pixels.
blue	Sets the blue value of the pixels.
alpha	Sets the alpha value of the pixels.
color > HSL	
hue	Sets the hue of the pixels.
saturation	Sets the saturation of the pixels.
lightness	Sets the lightness of the pixels.
alpha	Sets the alpha value of the pixels.
color > HSV	
hue	Sets the hue of the pixels.
saturation	Sets the saturation of the pixels.
value	Sets the value of the pixels.
alpha	Sets the alpha value of the pixels.
color options continued	
Enable Display Transform	Toggles gamma correction in the color picker, which is especially useful when working with OCIO.
Restrict RGBA Components	Restricts the alpha to 0,1 and the color channels to 0,a.

Common 2D Node (node2d) Widget Type

The **Common 2D Node** widget type allows you to pick channels that are affected by a particular 2D node and specify masking parameters. This widget type is not exposed for use as a custom user parameter.

The **Common 2D Node** widget type can be found in each Image node, except for Image BackgroundColor, ImageBlack, ImageChannels, ImageCheckerboard, ImageColor, ImageCoordinate, ImageCrop, ImageRamp, ImageRead, ImageText, ImageWhite, and ImageWrite.



NOTE: Not all of the Image nodes have the **Mask** parameter.

Menu Command	Description
▼ [view]	Specify whether the controls are set for the main , left , or right views, or set the controls to Enable All .
	When a specific component (R, G, B, or A) is enabled, the controls affect only that component.
mix	Dissolves between the bg image at 0 and the full merge effect at 1.
Mask ▼	
channel	<p>The channel from the out_mask input to use as a mask:</p> <ul style="list-style-type: none"> • R - use the red channel as the mask. • G - use the green channel as the mask. • B - use the blue channel as the mask. • A - use the alpha channel as the mask. <p>By default, the merge is limited to the non-black areas of the mask.</p>
invert	Inverts the use of the mask channel so that the merge is limited to the non-white areas of the mask.
fringe	<p>When enabled, the mask is modified so that, by default, the merge is limited to the fringe (semi-transparent areas).</p> <p>This is common alpha treatment, which modifies a normal mask such that it only affects the fringe (semi-transparent) areas.</p>

New Scene Graph Location (newScenegraphLocation) Widget Type

The **newScenegraphLocation** widget type allows you to specify the path of a scene graph location that is to be created by the respective node. It is used, for example, for the **name** parameter of CameraCreate nodes.

This widget type is not exposed for use as a custom user parameter, but can be accessed by setting the widget type hint to **newScenegraphLocation** (refer to *the Katana Technical Guide* in the *UI Hints for Plug-ins Using Argument Templates* chapter). Several node types that ship with Katana use the **newScenegraphLocation** widget type for parameters of various names, for example: **location**, **locations**, and **name**.

Parameters that use the **newScenegraphLocation** widget type can be found on the following types of nodes:

- [Alembic_In](#)
- [CameraCreate](#)
- [ImageCoordinate](#)
- [LightCreate](#)
- [PonyStack](#)
- [PrimitiveCreate](#)
- [ScenegraphGeneratorSetup](#)
- [ScenegraphXml_In](#)
- [TeapotCreate](#)

Menu Command	Description
Parent to Scenegraph Selection	Sets the parent location of the object created to be the current scene graph selection.
Parent to /root/world/geo/...	Sets the parent location of the object created to be /root/world/geo/ .
Parent to /root/world/lgt/...	Sets the parent location of the object created to be /root/world/lgt/ .
Parent to /root/world/cam/...	Sets the parent location of the object created to be /root/world/cam/ .
 Select In Scene Graph	<p>Selects the location specified by this parameter in the Scene Graph tab.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  NOTE: This option may be included in both the ▼ dropdown menu and as an icon to the right of the parameter name, or may only be present in one of these locations. </div>

Locations (locations) Widget Type

The **Locations** widget type allows you to select scene graph locations by paths or expressions. It is not an exposed widget type for use as a custom user parameter. Out of all node types that ship with Katana, only [Isolate](#) nodes use the **Locations** widget type, namely for their **isolateLocation** parameter.

Menu Command	Description
Path	Adds another path to this parameter's list of paths.
Expressions	

Append Scene Graph Selection	For each selected scene graph location, a new path is added to this parameter's list of paths and populated with the location.
Replace with Scene Graph Selection	Removes all existing paths and replaces them with paths populated with the currently selected Scene Graph tab locations.
Append Node Graph Node Locations	For each selected Node Graph node, a new path is added to this parameter's list of paths and an expression that links the scene graph location created by that node to the path.
Replace with Node Graph Node Locations	Removes all existing paths and replaces them with a path for each selected Node Graph node and links the scene graph location created by that node to the path.
Append Node Graph Locations as Parameter Expressions	
Replace Node Graph Locations as Parameter Expressions	
Clear All	Removes all paths from this parameter.

Look File Pass Name (lookfilePassname) Widget Type

The **Look File Pass Name** widget type allows you to set the pass name to use from a chosen Look File. This widget type is not exposed for use as a custom user parameter, but can be accessed by setting the widget type hint to **lookfilePassname**. Node types that ship with Katana use the **Look File Pass Name** widget type for the **passName** parameter.

Parameters that use the **Look File Pass Name** widget type can be found on the following types of nodes:

- [LookFileMaterialsIn](#)
- [LookFileOverrideEnable](#)
- [LookFileResolve](#)

Menu Command	Description
Choose Look File Pass from Selection...	

Rectangle (rectangle) Widget Type

The **Rectangle** widget type allows you to specify rectangular bounds to use by a node. This widget type is not exposed for use as a custom user parameter. Several node types that ship with Katana use the **Rectangle** widget type for parameters of various names, for example: **bounds**, **resolution**, or **rect**.

Parameters that use the **Rectangle** widget type can be found on the following types of nodes:

Menu Command	Description
bounds or rect > ▾	
Copy from Monitor ROI	
Copy to Monitor ROI	

Scene Graph Location (scenegraphLocation) Widget Type

The **Scene Graph Location** widget type allows you to specify the path of an existing scene graph location that a node is meant to work with. Several node types that ship with Katana use the **Scene Graph Location** widget type for parameters of various names, for example: **baseLocation**, **cameraLocation**, **location**, **locations**, **path**, **paths**, **sourceLocation**, and **targetPath**.

Parameters that use the **Scene Graph Location** widget type can be found on the following types of nodes:

- [AimConstraint](#)
- [AttributeCopy](#)
- [AttributeSet](#)
- [BillboardConstraint](#)
- [BoundsAdjust](#)
- [CameraClippingPlaneEdit](#)
- [CameraImagePlaneCreate](#)
- [CameraScreenWindowConstraint](#)
- [ClippingConstraint](#)
- [GafferThree](#)
- [HierarchyCopy](#)
- [InfoCreate](#)
- [Isolate](#)
- [LightLink](#)
- [LightLinkEdit](#)
- [LightLinkSetup](#)
- [LightLinkEdit](#)
- [LocationCreate](#)
- [OpScript](#)
- [OrientConstraint](#)
- [ParentChildConstraint](#)
- [PointConstraint](#)
- [PonyFarm](#)
- [ReflectionConstraint](#)
- [Rename](#)
- [RendererProceduralArgs](#)
- [RendererProceduralAssign](#)

- [CollectionCreate](#)
- [ConstraintCache](#)
- [CoordinateSystemDefine](#)
- [DollyConstraint](#)
- [FaceSetCreate](#)
- [FOVConstraint](#)
- [LookFileBake](#)
- [LookFileMultiBake](#)
- [Material](#)
- [NetworkMaterialInterfaceControls](#)
- [NetworkMaterialParameterEdit](#)
- [NetworkMaterialSplice](#)
- [RenderOutputDefine](#)
- [RenderSettings](#)
- [ScreenCoordinateConstraint](#)
- [Transform3D](#)
- [TransformEdit](#)
- [ViewerMaterialEdit](#)

Menu Command	Description
 Adopt Scenegraph Selection	The currently selected Scene Graph tab location is used to populate the parameter.
 Adopt Selected Nodegraph Node	Creates an expression from the currently selected Node Graph node linking the scene graph location created by that node to this parameter.
 Select In Scenegraph	Selects the location specified by this parameter in the Scene Graph tab.  NOTE: This option may be included in both the ▼ dropdown menu and as an icon to the right of the parameter name, or may only be present in one of these locations.
Adjust Path Relative To 'basePath'	Converts the current targetPath to a path relative to the basePath . If the targetPath is an expression, it is converted to a constant.  NOTE: This option does not appear in all instances and may only be available when both the basePath and targetPath parameters exist for a node.

Scene Graph Locations (scenegraphLocationArray) Widget Type

The **Scene Graph Locations** widget type allows you to specify a list of paths of existing scene graph locations that a node is meant to work with. Several node types that ship with Katana use the **Scene Graph Locations** widget type for parameters of various names, for example: **destinationLocations**, **lightPaths**, **locations**, **paths**, and **rootLocations**.

Parameters that use the **Scene Graph Locations** widget type can be found on the following types of nodes:

- [AimConstraint](#)
- [AttributeSet](#)
- [BillboardConstraint](#)
- [ClippingConstraint](#)
- [ConstraintCache](#)
- [ConstraintListEdit](#)
- [DollyConstraint](#)
- [FOVConstraint](#)
- [HierarchyCopy](#)
- [InfoCreate](#)
- [LightLink](#)
- [LightListEdit](#)
- [LocationCreate](#)
- [LookFileBake](#)
- [LookFileMultiBake](#)
- [PointConstraint](#)
- [ScreenCoordinateConstraint](#)

Menu Command	Description
Path	Adds another path to this parameter's list of paths.
Append Scenegraph Selection	For each selected scene graph location, a new path is added to this parameter's list of paths and populated with the location.
Replace with Scenegraph Selection	Removes all existing paths and replaces them with paths populated with the currently selected Scene Graph tab locations.
Append Nodegraph Node Locations	For each selected Node Graph node, a new path is added to this parameter's list of paths and an expression that links the scene graph location created by that node to the path.
Replace with Nodegraph Node Locations	Removes all existing paths and replaces them with a path for each selected Node Graph node and links the scene graph location created by that node to the path.
Clear All	Removes all paths from this parameter.
Find Instances beneath Scene Graph Selection...	 NOTE: This option is only found on the rootLocations parameter.

Transform Controls Widget Type

The **Transform Controls** widget type allows you to manipulate the transformation matrix. This widget type is not exposed for use as a custom user parameter. Node types that ship with Katana use the **Transform Controls** widget type for the transform parameter.

Parameters that use the **Transform Controls** widget type can be found on the following types of nodes:

- [CameraCreate](#)
- [Gaffer](#)
- [GafferThree](#)
- [LightCreate](#)
- [PonyStack](#)
- [PrimitiveCreate](#)
- [TeapotCreate](#)

Menu Command	Description
transform	
interface	<p>Sets the transform control layout:</p> <ul style="list-style-type: none"> • SRT Values - exposes the scale, rotation, and translation controls. • Transform Matrix - exposes a matrix to control transformations. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  <p>NOTE: If you select Transform Matrix, the translate, rotation, and scale fields are replaced by a matrix field instead.</p> </div>
transformOrder	Sets the order in which transforms are applied: Scale Rotate Translate, Scale Translate Rotate, Rotate Scale Translate, Rotate Translate Scale, Translate Scale Rotate, Translate Rotate Scale.
rotationOrder	Sets the order in which rotation is applied: XYZ, XZY, YXZ, YZX, ZXY, ZYX.
transform > interface: SRT Values	
translate	Controls camera translation on the xyz axes.
rotate	Controls camera rotation on the xyz axes.
scale	Controls camera scale on the xyz axes.
transform > interface: Transform Matrix	
matrix	<p>Controls transformations using a matrix in place of individual SRT controls.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  <p>NOTE: This field is only available if you have selected Transform Matrix in the interface field.</p> </div>

Transform Tools Widget Type

The **Transform Tools** widget type allows you to manipulate transformation data for scene graph locations. This widget type is not exposed for use as a custom user parameter. Node types that ship with Katana use the **Transform Tools** widget type for the transform parameter.

Parameters that use the **Transform Tools** widget type can be found on the following types of nodes:

- [CameraCreate](#)
- [Gaffer](#)
- [GafferThree](#)
- [LightCreate](#)
- [PrimitiveCreate](#)
- [TeapotCreate](#)

Menu Command	Description
transform > Tools ▼	
Snap to Position of Scene Graph Selection	Moves the position of the light to the position of the item selected in the Scene Graph tab.
Copy Scene Graph Selection World Transform	Copies the world SRT values of the item selected in the Scene Graph tab into the translate, rotate, and scale parameters under Object tab > transform .
Copy Scene Graph Selection Local Transform	Copies the local SRT values of the item selected in the Scene Graph tab into the translate, rotate, and scale parameters under Object tab > transform .
Fit to Bounds of Scene Graph Selection	Fits the light to the bounds set by the item selected in the Scene Graph tab.
Register to Scene Graph Camera	Places an object at a specified distance from a camera that is selected in the Scene Graph tab, oriented to face the camera, and scaled to fit the camera's screen window exactly. This is designed for use with primitive planes and may set unexpected transforms on other object types. This option is only available if a camera or light is selected in the Scene Graph tab.
Reset Transform	Resets any previous transforms, bringing the light back to the origin (0,0,0).

2D Nodes

The nodes in this chapter are 2D nodes that you can use within Katana. These are listed alphabetically, and each node includes a short description followed by a list of the node's parameters and their functions.

ImageAddMix

This node performs a typical **over** operation using a curve LUT on the foreground input alpha, then computes the result to pass down the node tree.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Sets the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.
displayWindow	Background	The frame size to output in the event that the fg and bg inputs are different sizes: <ul style="list-style-type: none">• Background - output the frame size of the bg input.• Foreground - output the frame size of the fg input.• Union - output a combination of the bg and fg inputs' frame sizes.• Intersection - output an intersection of the bg and fg inputs' frame sizes. This restricts the output to the area where the two frame sizes overlap.
premultiplied	disabled	When enabled, treat the fg colors as being premultiplied by the original alpha, before applying the new alpha adjustment.
curves		
	N/A	Picks the color (RGBA) values of the selection. For more information, refer to the Color Widget Type in Common Parameter Widgets on page 13.

Control (UI)	Default Value	Function
[bg and fg curves]	N/A	Adjusts the background and foreground values.
X	0	Displays the currently selected X point position on the active curve. You can also adjust the selected X point by entering positional
Y	1	Displays the currently selected Y point position on the active curve.
Reset Tangent	N/A	Click to reset any tangential changes applied to the selected curve.
[snapping mode]	off	Sets the point snapping mode: <ul style="list-style-type: none"> • off - no snapping is active. • frames - point movement snaps to the frames within a sequence. • grid - point movement snaps to the grid in the curves display.
[segment expression]	bezier()	Sets the segment expression which governs how the curve is interpolated between the two control points.

ImageAtop

This node layers images together using the Atop compositing algorithm: **Fb+B(1-f)**. It shows the shape of the background, with the foreground covering the background where the images overlap.

You can also specify a different compositing algorithm using the **operation** control.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.
ImageAtop parameters continued		

Control (UI)	Default Value	Function
operation	Atop	<p>If you don't want to layer the images together using the Atop compositing operation, select the operation to use instead.</p> <p>The following conventions apply to the below operation descriptions:</p> <ul style="list-style-type: none">• F refers to the fg input.• f refers to the fg input's alpha channel.• B refers to the bg input.• b refers to the bg input's alpha channel.

Control (UI)	Default Value	Function
operation (continued)		<p>The available operations (based on the Porter & Duff paper) are:</p> <ul style="list-style-type: none"> • Atop - $F + B(1 - f)$. This shows the shape of the background, with the foreground covering the background where the images overlap. • Average - $(F + B) / 2$. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • Difference - $\text{abs}(F - B)$. This shows how much the pixels differ and is useful for comparing two very similar images. • Divide - B / F. This divides the background values by the foreground values. • Exclusion - $F + B - 2FB$. This is a more photographic form of Difference. • From - $F - B$. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see Minus. • Geometric - $2FB / (F + B)$. This is another way of averaging two images. Visually, it's close to Min. • Hypot - $\text{sqrt}(F^2 + B^2)$. This resembles the Add and Screen operations. The result is not as bright as Add, but brighter than Screen. Hypot works with values above 1. It can be useful for adding reflections, as an alternative to Screen. • In - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • Matte - $Ff * B(1 - f)$. This is a premultiplied Over. Use unpremultiplied images with this operation. • Max - $\text{max}(F, B)$. This takes the maximum values of both images. This is a good way to combine mattes and useful for bringing aspects like bright hair detail through.

Control (UI)	Default Value	Function
operation (continued)		<ul style="list-style-type: none"> • Min - $\min(F,B)$. This takes the minimum values of both images. • Minus - $B-F$. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see From. • Multiply - FB. This multiplies the values of the foreground by the values of the background. It can be used to composite darker values from the foreground with the background image - dark gray smoke shot against a white background, for example. • Out - $B(1-f)$. This only shows the areas of the background that do not overlap with the alpha of the foreground. This can be useful for combining mattes. • Over - $F+B(1-f)$. This layers the foreground over the background according to the alpha of the foreground. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • Plus - $F+B$. This produces the sum of the foreground and background. Note that the add algorithm may result in pixel values higher than 1.0. • Screen - $F+B-FB$. This is similar to Hypot, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • Under - $F(1-b)+B$. This is the reverse of the Over operation. It layers the background over the foreground according to the alpha of the background.
amount	1	Dissolves between the bg image at 0 and the full merge effect at 1.
displayWindow	Background	<p>The frame size to output in the event that the fg and bg inputs are different sizes:</p> <ul style="list-style-type: none"> • Background - output the frame size of the bg input. • Foreground - output the frame size of the fg input. • Union - output a combination of the bg and fg inputs' frame sizes. • Intersection - output an intersection of the bg and fg inputs' frame sizes. This restricts the output to the area where the two frame sizes overlap.

Control (UI)	Default Value	Function
clampAlpha	enabled	When enabled, the output alpha channel is clamped to the 0-1 range. Color channels (RGB) are not affected.

ImageAverage

This node layers images together using the **Average** compositing algorithm: $(F+B)/2$. It produces the average of the two images. The result is darker than the original images, but accentuates highlights.

You can also specify a different compositing algorithm using the **operation** control.

Also see [ImageGeometric](#) on page 65 and [ImageHypot](#) on page 69.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.

ImageAverage parameters continued

operation	ImageAverage	<p>If you don't want layer the images together using the ImageAverage compositing operation, select the operation to use instead.</p> <p>The following conventions apply to the below operation descriptions:</p> <ul style="list-style-type: none"> • F refers to the fg input. • f refers to the fg input's alpha channel. • B refers to the bg input. • b refers to the bg input's alpha channel.
-----------	--------------	---

Control (UI)	Default Value	Function
operation (continued)		<p>The available operations (based on the Porter & Duff paper) are:</p> <ul style="list-style-type: none"> • Atop - $F+B(1-f)$. This shows the shape of the background, with the foreground covering the background where the images overlap. • Average - $(F+B)/2$. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • Difference - $\text{abs}(F-B)$. This shows how much the pixels differ and is useful for comparing two very similar images. • Divide - B/F. This divides the background values by the foreground values. • Exclusion - $F+B-2FB$. This is a more photographic form of Difference. • From - $F-B$. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see Minus. • Geometric - $2FB/(F+B)$. This is another way of averaging two images. Visually, it's close to Min. • Hypot - $\text{sqrt}(F^2+B^2)$. This resembles the Add and Screen operations. The result is not as bright as Add, but brighter than Screen. Hypot works with values above 1. It can be useful for adding reflections, as an alternative to Screen. • In - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • Matte - $Ff*B(1-f)$. This is a premultiplied Over. Use unpremultiplied images with this operation. • Max - $\text{max}(F,B)$. This takes the maximum values of both images. This is a good way to combine mattes and useful for bringing aspects like bright hair detail through.

Control (UI)	Default Value	Function
operation (continued)		<ul style="list-style-type: none"> • Min - $\min(F,B)$. This takes the minimum values of both images. • Minus - $B-F$. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see ImageFrom. • Multiply - FB. This multiplies the values of the foreground by the values of the background. It can be used to composite darker values from the foreground with the background image - dark gray smoke shot against a white background, for example. • Out - $B(1-f)$. This only shows the areas of the background that do not overlap with the alpha of the foreground. This can be useful for combining mattes. • Over - $F+B(1-f)$. This layers the foreground over the background according to the alpha of the foreground. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • Plus - $F+B$. This produces the sum of the foreground and background. Note that the add algorithm may result in pixel values higher than 1.0. • Screen - $F+B-FB$. This is similar to Hypot, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • Under - $F(1-b)+B$. This is the reverse of the Over operation. It layers the background over the foreground according to the alpha of the background.
amount	1	Dissolves between the bg image at 0 and the full merge effect at 1.
displayWindow	Background	<p>The frame size to output in the event that the fg and bg inputs are different sizes:</p> <ul style="list-style-type: none"> • Background - output the frame size of the bg input. • Foreground - output the frame size of the fg input. • Union - output a combination of the bg and fg inputs' frame sizes. • Intersection - output an intersection of the bg and fg inputs' frame sizes. This restricts the output to the area where the two frame sizes overlap.

Control (UI)	Default Value	Function
clampAlpha	enabled	When enabled, the output alpha channel is clamped to the 0-1 range. Color channels (RGB) are not affected.

ImageBackgroundColor

Controls the background color of the input image using RGB, HSL, and/or HSV parameters.

Control (UI)	Default Value	Function
color		
color	0.0, 0.0, 0.0, 1.0	The color (RGBA values) of every pixel in the image. You can also use the below RGB, HSL, or HSV controls to set the color. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.

ImageBlack

Generates an image where every pixel is black.

Control (UI)	Default Value	Function
bounds		
[resolution]	Dependent on Project Settings	Select the size of the image.
left	0	Lets you offset the image by adding this number of pixels to the left side of the image.
bottom	0	Lets you offset the image by adding this number of pixels below the image.

Control (UI)	Default Value	Function
width	globals.width	The width of the image in pixels. The default setting, globals.width , resizes the image to the width of the resolution indicated on the Project Settings tab.
height	globals.height	The height of the image in pixels. The default setting, globals.height , resizes the image to the height of the resolution indicated on the Project Settings tab.
infiniteExtent	Disabled	When enabled, the color extends beyond the bounds.
color		
color	0.0, 0.0, 0.0, 1.0	The color (RGBA values) of every pixel in the image. You can also use the below RGB, HSL, or HSV controls to set the color. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.

ImageBlur

Adds blur to an image or matte using Box, Triangle, Gaussian, Bell, BSpline, or Mitchell filter algorithms. The blur value is calculated for image pixels by examining their neighbors within the constraints of the **xAmount** and **yAmount** controls, and applying the selected algorithms.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.
ImageBlur parameters continued		

Control (UI)	Default Value	Function
xAmount	0	Sets the horizontal radius (in pixels) within which pixels are compared to calculate the blur. Higher values widen the compare area, producing more blur.
yAmount	xAmount	<p>Sets the vertical radius (in pixels) within which pixels are compared to calculate the blur. Higher values widen the compare area, producing more blur.</p> <p>By default, this value is the same as xAmount.</p>
filter	Gaussian	<p>Select the filtering algorithm to use:</p> <ul style="list-style-type: none"> • Box • Triangle • Gaussian • Bell • BSpline • Mitchell
borderExtend	Clamp	<p>Select the border extend method for pixels required beyond the image borders:</p> <ul style="list-style-type: none"> • Mirror • Clamp • Background
useOverscan	globals.compDefaults.useOverscan	<p>Sets whether to use upstream overscan (if available) during the border extension process. If overscan is available (and of usable quality), this typically yields superior results around frame edges.</p> <p>However, if you are unsure of this procedure or the integrity of overscanned areas is unknown, it's safer to leave this disabled.</p> <p>Overscan refers to image pixel data outside of the displayWindow and can be inspected using options in the Monitor.</p> <p>For information on explicitly manipulating these regions, see the ImageCrop node.</p>

Control (UI)	Default Value	Function
channelAmounts		
red	1	Applies a multiplier to the blur amount for the red channel.
green	1	Applies a multiplier to the blur amount for the green channel.
blue	1	Applies a multiplier to the blur amount for the blue channel.
alpha	1	Applies a multiplier to the blur amount for the alpha channel.

ImageBrightness

This node multiplies the image's channels to increase or decrease brightness.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.
ImageBrightness parameters continued		
brightness	1	Adjusts the brightness of the rgb channels in the image.
alpha	1	Adjusts the brightness of the alpha channel in the image.

ImageChannels

This node lets you:

- rearrange up to 4 channels from a single image (one input)

- combine channels from several inputs into one output. For example, you can use it to combine two separate passes (such as the beauty pass and the reflection pass) into the same data stream.
- replace a channel with luminance, black (removing the alpha channel, for example), white (making the alpha solid, for example), or any other constant color.

Control (UI)	Default Value	Function
redSource	i0	Select the input from which to take the red channel.
redChannel	R	Select what to use as the red channel: <ul style="list-style-type: none"> • R - use the red channel from redSource. • G - use the green channel from redSource. • B - use the blue channel from redSource. • A - use the alpha channel from redSource. • Lum - use the luminance from redSource. • 1 - set the red channel to white. • 0 - set the red channel to black. • Const - set the red channel to any constant color. You can select the color using the constantColor controls.
greenSource	i0	Select the input from which to take the green channel.
greenChannel	G	Select what to use as the green channel: <ul style="list-style-type: none"> • R - use the red channel from greenSource. • G - use the green channel from greenSource. • B - use the blue channel from greenSource. • A - use the alpha channel from greenSource. • Lum - use the luminance from greenSource. • 1 - set the green channel to white. • 0 - set the green channel to black. • Const - set the green channel to any constant color. You can select the color using the constantColor controls.
blueSource	i0	Select the input from which to take the blue channel.

Control (UI)	Default Value	Function
blueChannel	B	<p>Select what to use as the blue channel:</p> <ul style="list-style-type: none"> • R - use the red channel from blueSource. • G - use the green channel from blueSource. • B - use the blue channel from blueSource. • A - use the alpha channel from blueSource. • Lum - use the luminance from blueSource. • 1 - set the blue channel to white. • 0 - set the blue channel to black. • Const - set the blue channel to any constant color. You can select the color using the constantColor controls.
alphaSource	i0	Select the input from which to take the alpha channel.
alphaChannel	A	<p>Select what to use as the alpha channel:</p> <ul style="list-style-type: none"> • R - use the red channel from alphaSource. • G - use the green channel from alphaSource. • B - use the blue channel from alphaSource. • A - use the alpha channel from alphaSource. • Lum - use the luminance from alphaSource. • 1 - set the alpha channel to white. • 0 - set the alpha channel to black. • Const - set the alpha channel to any constant color. You can select the color using the constantColor controls.
constantColor		
color	0.0000, 0.0000, 0.0000, 1.0000	<p>The color (RGBA values) of the pixels in any channels that you have set to Const. You can also use the below RGB, HSL, or HSV controls to set the color.</p> <p>For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.</p>

ImageCheckerboard

The ImageCheckerboard allows you to create a checkerboard pattern. You can specify the checkers' size and colors and the checkerboard's bounds.

Control (UI)	Default Value	Function
bounds		
[resolution]	Dependent on Project Settings	Select the size of the image.
▼	N/A	For more information, refer to the Rectangle Widget Type in the Common Parameter Widgets on page 13.
left	0	Set the left position of the ROI in the Monitor tab.
bottom	0	Set the bottom position of the ROI in the Monitor tab.
width	globals.width	Set the width of the ROI in the Monitor tab.
height	global.height	Set the height of the ROI in the Monitor tab.
color1		
color1	0.1000, 0.1000, 0.1000, 1.0000	The color (RGBA values) of the pixels for the first color in the checkerboard. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.
color2		
color2	0.5000, 0.5000, 0.5000, 1.000	The color (RGBA values) of the pixels for the second color in the checkerboard. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.
ImageCheckerboard parameters (Cont.)		
checkerSize	64.0, 64.0	Sets the size of the checkers under width and height .

ImageClamp

This node constrains, or clamps, values in the selected channels to a specified minimum and/or maximum range.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.
ImageClamp parameters continued		
clamp	Both	Sets which values are use to clamp the input: <ul style="list-style-type: none"> • Both - min and max rgba values are clamped. • Max - only the max rgba clamps are used. • Min - only the min rgba clamps are used.
min		
red	0	Sets the minimum values at which the rgba channels are clamped. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.
green	0	
blue	0	
alpha	0	
max		
red	16	Sets the maximum values at which the rgba channels are clamped. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.
green	16	
blue	16	
alpha	1	
inputs	Unpremultiplied	Select whether you are using a premultiplied or unpremultiplied input image: <ul style="list-style-type: none"> • Premultiplied - the ImageClamp node unpremultiplies the input, applies the clamp effect, and premultiplies the input again. This simulates applying the clamp before the premultiplication was done, as color corrections are typically applied on unpremultiplied images. • Unpremultiplied - the ImageClamp node simply applies the contrast change.

ImageColor

Generates an image where every pixel is the same color. By default, the image is white.

Control (UI)	Default Value	Function
bounds		
[resolution]	Dependent on Project Settings	Select the size of the image.
bounds > ▾	N/A	For more information, refer to the Rectangle Widget Type in the Common Parameter Widgets on page 13.
left	0	Lets you offset the image by adding this number of pixels to the left side of the image.
bottom	0	Lets you offset the image by adding this number of pixels below the image.
width	globals.width	The width of the image in pixels. The default setting, globals.width , resizes the image to the width of the resolution indicated on the Project Settings tab.
height	globals.height	The height of the image in pixels. The default setting, globals.height , resizes the image to the height of the resolution indicated on the Project Settings tab.
ImageColor parameters continued		
infiniteExtent	Disabled	When enabled, the color extends beyond the bounds.
color		
color	1.0, 1.0, 1.0, 1.0	The color (RGBA values) of every pixel in the image. You can also use the RGB, HSL, or HSV controls to set the color. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.

ImageContrast

This adjusts the input image's contrast around a fixed color point.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.
contrast		
rgb	1	Adjusts the image contrast in the r, g, and b channels.
r	1	Adjusts the image contrast in the red channel only.
g	1	Adjusts the image contrast in the green channel only.
b	1	Adjusts the image contrast in the blue channel only.
a	1	Adjusts the image contrast in the alpha channel only.
fixedPoint		
fixedPoint	0.1800, 0.1800, 0.1800, 0.500	The point from which to influence the contrast. When contrast is greater than one, colors are moved away from this value, when the contrast is below one, colors are moved towards this value. You can also use the below RGB, HSL, or HSV controls to set the color. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.
contrastFunction	Power	Select the contrast function to use: <ul style="list-style-type: none"> • Power • Linear

Control (UI)	Default Value	Function
inputs	Unpremultiplied	<p>Select whether you are using a premultiplied or unpremultiplied input image:</p> <ul style="list-style-type: none"> • Premultiplied - the ImageContrast node unpremultiplies the input, applies the contrast change, and premultiplies the input again. This simulates applying the contrast change before the premultiplication was done, as color corrections are typically applied on unpremultiplied images. • Unpremultiplied - the ImageContrast node simply applies the contrast change.

ImageCoordinate

ImageCoordinate allows you to load an image into the interface and specify a 2D-point (x, y coordinates) that is then stored as attribute data on a scene graph location.

Control (UI)	Default Value	Function
location	/root/world/2d_image/points	<p>The scene graph location to where (x, y) coordinates are written. The location parameter options are available in either the scene graph widget or ▼ dropdown menu to the right of the parameter.</p> <p>For more information, refer to the Create Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.</p>
filePath	N/A	<p>Specifies the file path to the image to load here.</p> <p>For more information, refer to the Asset and File Path Widget Types in the Common Parameter Widgets on page 13.</p>
numberValue	1x2	Sets the override value.

ImageCrop

This node removes, or crops, image information outside a defined area, though Katana has both a data window and a display window (to use the EXR terminology).

- The display window is the image frame.
- The data window is the area that actually contains pixels.



NOTE: The data window may be larger or smaller than the display window. If it is larger, image data exists that can be pulled into the frame by downstream operations. If smaller, savings in processing time and memory are achieved by not explicitly storing pixel values for all the constant color outside the useful image area.

Control (UI)	Default Value	Function
bounds		
[resolution]	Dependent on Project Settings	Sets the size of the display window using the dropdown menu.
bounds > ▾	N/A	For more information, refer to the Rectangle Widget Type in the Common Parameter Widgets on page 13.
left	0	Offset the display window by this number of pixels from the left side of the data window.
bottom	0	Offset the display window by this number of pixels from the bottom side of the data window.
width	globals.width	Adjusts the width of the display window in pixels.
height	globals.height	Adjusts the height of the display window in pixels.
ImageCrop parameters continued		
reformat	disabled	When enabled, reposition the cropped area to the origin and changes the display window.
reformat: enabled		
allowOverscan	disabled	This allows the node to generate overscan (if possible). Overscan refers to image pixel data outside of the display window and can be inspected using options in the Monitor.

ImageDifference

This node layers images together using the **Difference** compositing algorithm: **abs(F-B)**. It shows how much the pixels differ and is useful for comparing two very similar images.

You can also specify a different compositing algorithm using the **operation** control.



NOTE: ImageDifference outputs the absolute value of F-B, whereas [ImageMinus](#) outputs exactly F-B. For the standard "difference matte" situation, you want ImageDifference.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.
ImageDifference parameters continued		
operation	Difference	<p>If you don't want layer the images together using the Difference compositing operation, select the operation to use instead.</p> <p>The following conventions apply to the below operation descriptions:</p> <ul style="list-style-type: none"> • F refers to the fg input. • f refers to the fg input's alpha channel. • B refers to the bg input. • b refers to the bg input's alpha channel.

Control (UI)	Default Value	Function
operation (continued)		<p>The available operations (based on the Porter & Duff paper) are:</p> <ul style="list-style-type: none"> • Atop - $F + B(1 - f)$. This shows the shape of the background, with the foreground covering the background where the images overlap. • Average - $(F + B) / 2$. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • Difference - $\text{abs}(F - B)$. This shows how much the pixels differ and is useful for comparing two very similar images. • Divide - B / F. This divides the background values by the foreground values. • Exclusion - $F + B - 2FB$. This is a more photographic form of Difference. • From - $F - B$. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see Minus. • Geometric - $2FB / (F + B)$. This is another way of averaging two images. Visually, it's close to Min. • Hypot - $\text{sqrt}(F^2 + B^2)$. This resembles the Add and Screen operations. The result is not as bright as Add, but brighter than Screen. Hypot works with values above 1. It can be useful for adding reflections, as an alternative to Screen. • In - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • Matte - $Ff * B(1 - f)$. This is a premultiplied Over. Use unpremultiplied images with this operation. • Max - $\text{max}(F, B)$. This takes the maximum values of both images. This is a good way to combine mattes and useful for bringing aspects like bright hair detail through.

Control (UI)	Default Value	Function
operation (continued)		<ul style="list-style-type: none"> • Min - $\min(F,B)$. This takes the minimum values of both images. • Minus - $B-F$. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see From. • Multiply - FB. This multiplies the values of the foreground by the values of the background. It can be used to composite darker values from the foreground with the background image - dark gray smoke shot against a white background, for example. • Out - $B(1-f)$. This only shows the areas of the background that do not overlap with the alpha of the foreground. This can be useful for combining mattes. • Over - $F+B(1-f)$. This layers the foreground over the background according to the alpha of the foreground. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • Plus - $F+B$. This produces the sum of the foreground and background. Note that the add algorithm may result in pixel values higher than 1.0. • Screen - $F+B-FB$. This is similar to Hypot, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • Under - $F(1-b)+B$. This is the reverse of the Over operation. It layers the background over the foreground according to the alpha of the background.
amount	1	Dissolves between the input image at 0 and the full merge effect at 1.
displayWindow	Background	<p>The frame size to output in the event that the fg and bg inputs are different sizes:</p> <ul style="list-style-type: none"> • Background - output the frame size of the bg input. • Foreground - output the frame size of the fg input. • Union - output a combination of the bg and fg inputs' frame sizes. • Intersection - output an intersection of the bg and fg inputs' frame sizes. This restricts the output to the area where the two frame sizes overlap.

Control (UI)	Default Value	Function
clampAlpha	enabled	When enabled, the output alpha channel is clamped to the 0-1 range. Color channels (RGB) are not affected.

ImageDistort

ImageDistort warps the **input** image based on the values in the **control** image. The pixel values in the **control** image are offsets for where a pixel comes from. For example, if pixel 51, 23 has a value of -1, 5 in the **control** image, the pixel's value comes from pixel 50, 28 in the **input** image.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.
padding		
x	128	Sets the padding required to create the distorted image. These values should be larger than the largest x and y displacements you are expecting in the control image.
y	128	
borderExtend	Clamp	Select the border extend method for pixels required beyond the image borders: <ul style="list-style-type: none"> • Mirror • Clamp • Background

Control (UI)	Default Value	Function
useOverscan	globals.compDefaults.useOverscan	<p>When enabled, use upstream overscan (if available) during the border extension process. If overscan is available (and of usable quality), this typically yields superior results around frame edges.</p> <p>However, if you are unsure of this procedure or the integrity of overscanned areas is unknown, it's safer to leave this disabled.</p> <p>Overscan refers to image pixel data outside of the displayWindow and can be inspected using options in the Monitor.</p> <p>For information on explicitly manipulating these regions, see the ImageCrop node.</p>
processMode	Full Frame	<p>Sets how to render this node:</p> <ul style="list-style-type: none"> • Tiled - The node is rendered incrementally in tiles. This is preferable when interactive feedback is desired or the kernel sizes are small. • Full Frame - The node is rendered in a single pass as a full frame. This option can reduce render times for very large kernel sizes. <p> NOTE: Output in either mode is identical.</p>
useBlueChannelAs Mask	disabled	<p>When enabled, only pixels that have non-zero blue values are sampled from the input image. Pixels with zero blue values (that is, where the blue channel is black) are output as black.</p>
filtering		
xFilter	Lanczos3	The filter kernel to use for sampling on the x axis.
yFilter	Lanczos3	The filter kernel to use for sampling on the y axis.

Control (UI)	Default Value	Function
highlightCompensation	enabled	When enabled, Katana adaptively compresses pixel values prior to transform filtering and re-expands them afterwards. This helps to reduce the ringing in high-contrast areas that can be a problem in linear floating point images (as we have in Katana).
clampOutput	enabled	<p>Filtering can introduce negative values and send values above 1.0.</p> <p>When clampOutput is enabled, Katana clamps the rgb channels low at 0 and the alpha channel between 1 and 0 after the image is filtered. This is recommended for transforms on color/alpha images.</p> <p>When clampOutput is disabled, no clamping is done and values below 0 and above 1 are allowed. This is recommended for transforms applied to images that contain data which may (correctly) range more widely.</p>

ImageDivide

This node layers images together using the **Divide** compositing algorithm: **B/F**. It divides the background values by the foreground values.

You can also specify a different compositing algorithm using the **operation** control.

Control (UI)	Default Value	Function
[2D node controls]	N/A	<p>Set the controls for the stereo view.</p> <p>For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.</p>

Control (UI)	Default Value	Function
ImageDivide parameters continued		
operation	Divide	<p>If you don't want layer the images together using the Divide compositing operation, select the operation to use instead.</p> <p>The following conventions apply to the below operation descriptions:</p> <ul style="list-style-type: none"> • F refers to the fg input. • f refers to the fg input's alpha channel. • B refers to the bg input. • b refers to the bg input's alpha channel.

Control (UI)	Default Value	Function
operation (continued)		<p>The available operations (based on the Porter & Duff paper) are:</p> <ul style="list-style-type: none"> • Atop - $F + B(1-f)$. This shows the shape of the background, with the foreground covering the background where the images overlap. • Average - $(F+B)/2$. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • Difference - $\text{abs}(F-B)$. This shows how much the pixels differ and is useful for comparing two very similar images. • Divide - B/F. This divides the background values by the foreground values. • Exclusion - $F+B-2FB$. This is a more photographic form of Difference. • From - $F-B$. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see ImageMinus. • Geometric - $2FB/(F+B)$. This is another way of averaging two images. Visually, it's close to Min. • Hypot - $\text{sqrt}(F^2+B^2)$. This resembles the Add and Screen operations. The result is not as bright as Add, but brighter than Screen. Hypot works with values above 1. It can be useful for adding reflections, as an alternative to Screen. • In - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • Matte - $Ff*B(1-f)$. This is a premultiplied Over. Use unpremultiplied images with this operation. • Max - $\text{max}(F,B)$. This takes the maximum values of both images. This is a good way to combine mattes and useful for bringing aspects like bright hair detail through.

Control (UI)	Default Value	Function
operation (continued)		<ul style="list-style-type: none"> • Min - $\min(F,B)$. This takes the minimum values of both images. • Minus - $B-F$. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see From. • Multiply - FB. This multiplies the values of the foreground by the values of the background. It can be used to composite darker values from the foreground with the background image - dark gray smoke shot against a white background, for example. • Out - $B(1-f)$. This only shows the areas of the background that do not overlap with the alpha of the foreground. This can be useful for combining mattes. • Over - $F+B(1-f)$. This layers the foreground over the background according to the alpha of the foreground. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • Plus - $F+B$. This produces the sum of the foreground and background. Note that the add algorithm may result in pixel values higher than 1.0. • Screen - $F+B-FB$. This is similar to Hypot, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • Under - $F(1-b)+B$. This is the reverse of the Over operation. It layers the background over the foreground according to the alpha of the background.
amount	1	Dissolves between the bg image at 0 and the full merge effect at 1.
displayWindow	Background	<p>The frame size to output in the event that the fg and bg inputs are different sizes:</p> <ul style="list-style-type: none"> • Background - output the frame size of the bg input. • Foreground - output the frame size of the fg input. • Union - output a combination of the bg and fg inputs' frame sizes. • Intersection - output an intersection of the bg and fg inputs' frame sizes. This restricts the output to the area where the two frame sizes overlap.

Control (UI)	Default Value	Function
clampAlpha	enabled	When enabled, the output alpha channel is clamped to the 0-1 range. Color channels (RGB) are not affected.

ImageExclusion

This node layers images together using the **Exclusion** compositing algorithm: **F+B-2FB**. It's a more photographic form of [ImageDifference](#).

You can also specify a different compositing algorithm using the **operation** control.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.

ImageExclusion parameters continued

operation	Exclusion	<p>If you don't want layer the images together using the Exclusion compositing operation, select the operation to use instead.</p> <p>The following conventions apply to the below operation descriptions:</p> <ul style="list-style-type: none"> • F refers to the fg input. • f refers to the fg input's alpha channel. • B refers to the bg input. • b refers to the bg input's alpha channel.
-----------	-----------	--

Control (UI)	Default Value	Function
operation (continued)		<p>The available operations (based on the Porter & Duff paper) are:</p> <ul style="list-style-type: none"> • Atop - $F+B(1-f)$. This shows the shape of the background, with the foreground covering the background where the images overlap. • Average - $(F+B)/2$. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • Difference - $\text{abs}(F-B)$. This shows how much the pixels differ and is useful for comparing two very similar images. • Divide - B/F. This divides the background values by the foreground values. • Exclusion - $F+B-2FB$. This is a more photographic form of Difference. • From - $F-B$. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see Minus. • Geometric - $2FB/(F+B)$. This is another way of averaging two images. Visually, it's close to Min. • Hypot - $\text{sqrt}(F^2+B^2)$. This resembles the Add and Screen operations. The result is not as bright as Add, but brighter than Screen. Hypot works with values above 1. It can be useful for adding reflections, as an alternative to Screen. • In - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • Matte - $Ff*B(1-f)$. This is a premultiplied Over. Use unpremultiplied images with this operation. • Max - $\text{max}(F,B)$. This takes the maximum values of both images. This is a good way to combine mattes and useful for bringing aspects like bright hair detail through.

Control (UI)	Default Value	Function
operation (continued)		<ul style="list-style-type: none"> • Min - $\min(F,B)$. This takes the minimum values of both images. • Minus - $B-F$. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see From. • Multiply - FB. This multiplies the values of the foreground by the values of the background. It can be used to composite darker values from the foreground with the background image - dark gray smoke shot against a white background, for example. • Out - $B(1-f)$. This only shows the areas of the background that do not overlap with the alpha of the foreground. This can be useful for combining mattes. • Over - $F+B(1-f)$. This layers the foreground over the background according to the alpha of the foreground. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • Plus - $F+B$. This produces the sum of the foreground and background. Note that the add algorithm may result in pixel values higher than 1.0. • Screen - $F+B-FB$. This is similar to Hypot, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • Under - $F(1-b)+B$. This is the reverse of the Over operation. It layers the background over the foreground according to the alpha of the background.
amount	1	Dissolves between the bg image at 0 and the full merge effect at 1.
displayWindow	Background	<p>The frame size to output in the event that the fg and bg inputs are different sizes:</p> <ul style="list-style-type: none"> • Background - output the frame size of the bg input. • Foreground - output the frame size of the fg input. • Union - output a combination of the bg and fg inputs' frame sizes. • Intersection - output an intersection of the bg and fg inputs' frame sizes. This restricts the output to the area where the two frame sizes overlap.

Control (UI)	Default Value	Function
clampAlpha	enabled	When enabled, the output alpha channel is clamped to the 0-1 range. Color channels (RGB) are not affected.

ImageExposure

Allows you to adjust the exposure of the input sequence using f-stops or gain.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.
units: F-Stops		
fsIntensity	0	Adjusts f-stop intensity.
units: Gain		
gIntensity	1	Adjusts gain intensity.
units: F-Stops > fsColor		
red	0	Adjusts f-stop exposure in the red channel.
green	0	Adjusts f-stop exposure in the green channel.
blue	0	Adjusts f-stop exposure in the blue channel.
fsAlpha	0	Adjusts f-stop exposure in the alpha channel.
units: Gain > gColor		
red	1	Adjusts gain exposure in the red channel.
green	1	Adjusts gain exposure in the green channel.
blue	1	Adjusts gain exposure in the blue channel.

Control (UI)	Default Value	Function
gAlpha	1	Adjusts gain exposure in the alpha channel.
units	F-Stops	Select the units in which the exposure is altered: <ul style="list-style-type: none"> • F-Stops - use the fsColor controls to adjust exposure. • Gain - use the gColor controls to adjust exposure.

ImageFade

This node fades the input image to a color of your choosing. By default, the image is faded to black.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.
ImageFade parameters continued		
amount	1	Dissolves between the bg image at 0 and the full merge effect at 1.
fadeToColor		
fadeToColor	0.0000, 0.0000, 0.0000, 0.0000	The color (RGBA values) of the fade color. You can also use the below RGB, HSL, or HSV controls to set the color. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.

ImageFrom

This node layers images together using the **From** compositing algorithm: **F-B**. It subtracts the background from the foreground. You can also specify a different compositing algorithm using the **operation** control.

For subtracting the foreground from the background instead, see [ImageUnder](#) on page 132.

Control (UI)	Default Value	Function
[2D node controls]	N/A	<p>Set the controls for the stereo view.</p> <p>For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.</p>
ImageFrom parameters continued		
operation	From	<p>If you don't want layer the images together using the From compositing operation, select the operation to use instead.</p> <p>The following conventions apply to the below operation descriptions:</p> <ul style="list-style-type: none"> • F refers to the fg input. • f refers to the fg input's alpha channel. • B refers to the bg input. • b refers to the bg input's alpha channel.

Control (UI)	Default Value	Function
operation (continued)		<p>The available operations (based on the Porter & Duff paper) are:</p> <ul style="list-style-type: none"> • Atop - $F + B(1 - f)$. This shows the shape of the background, with the foreground covering the background where the images overlap. • Average - $(F + B) / 2$. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • Difference - $\text{abs}(F - B)$. This shows how much the pixels differ and is useful for comparing two very similar images. • Divide - B / F. This divides the background values by the foreground values. • Exclusion - $F + B - 2FB$. This is a more photographic form of Difference. • From - $F - B$. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see Minus. • Geometric - $2FB / (F + B)$. This is another way of averaging two images. Visually, it's close to Min. • Hypot - $\sqrt{F^2 + B^2}$. This resembles the Add and Screen operations. The result is not as bright as Add, but brighter than Screen. Hypot works with values above 1. It can be useful for adding reflections, as an alternative to Screen. • In - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • Matte - $Ff * B(1 - f)$. This is a premultiplied Over. Use unpremultiplied images with this operation. • Max - $\max(F, B)$. This takes the maximum values of both images. This is a good way to combine mattes and useful for bringing aspects like bright hair detail through.

Control (UI)	Default Value	Function
operation (continued)		<ul style="list-style-type: none"> • Min - $\min(F,B)$. This takes the minimum values of both images. • Minus - $B-F$. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see From. • Multiply - FB. This multiplies the values of the foreground by the values of the background. It can be used to composite darker values from the foreground with the background image - dark gray smoke shot against a white background, for example. • Out - $B(1-f)$. This only shows the areas of the background that do not overlap with the alpha of the foreground. This can be useful for combining mattes. • Over - $F+B(1-f)$. This layers the foreground over the background according to the alpha of the foreground. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • Plus - $F+B$. This produces the sum of the foreground and background. Note that the add algorithm may result in pixel values higher than 1.0. • Screen - $F+B-FB$. This is similar to Hypot, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • Under - $F(1-b)+B$. This is the reverse of the Over operation. It layers the background over the foreground according to the alpha of the background.
amount	1	Dissolves between the bg image at 0 and the full merge effect at 1.
displayWindow	Background	<p>The frame size to output in the event that the fg and bg inputs are different sizes:</p> <ul style="list-style-type: none"> • Background - output the frame size of the bg input. • Foreground - output the frame size of the fg input. • Union - output a combination of the bg and fg inputs' frame sizes. • Intersection - output an intersection of the bg and fg inputs' frame sizes. This restricts the output to the area where the two frame sizes overlap.

Control (UI)	Default Value	Function
clampAlpha	enabled	When enabled, the output alpha channel is clamped to the 0-1 range. Color channels (RGB) are not affected.

ImageGain

This node lets you adjust the gain in your input image. In other words, it multiplies a channel's values by a given factor, which has the effect of lightening the channel while preserving the blackpoint.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.
units: F-Stops		
fsIntensity	0	Adjusts f-stop intensity.
units: Gain		
gIntensity	1	Adjusts gain intensity.
units: F-Stops > fsColor		
red	0	Adjusts f-stop in the red channel.
green	0	Adjusts f-stop in the green channel.
blue	0	Adjusts f-stop in the blue channel.
fsAlpha	0	Adjusts f-stop in the alpha channel.
units: Gain > gColor		
red	1	Adjusts gain in the red channel.
green	1	Adjusts gain in the green channel.
blue	1	Adjusts gain in the blue channel.

Control (UI)	Default Value	Function
gAlpha	1	Adjusts gain in the alpha channel.
units	F-Stops	Select the units in which the gain is altered: <ul style="list-style-type: none"> • F-Stops - use the fsColor controls to adjust gain. • Gain - use the gColor controls to adjust gain.

ImageGamma

Applies a constant gamma value to the selected channels. This lightens or darkens the mid-tones.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.
gamma		
rgb	1	Adjusts gamma in the red, green, and blue channels.
r	1	Adjusts gamma in the red channel.
g	1	Adjusts gamma in the green channel.
b	1	Adjusts gamma in the blue channel.
a	1	Adjusts gamma in the alpha channel.
fixedPoint		
fixedPoint	1.0000, 1.0000, 1.0000, 1.0000	The color (RGBA values). You can also use the below RGB, HSL, or HSV controls to set the color. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.

Control (UI)	Default Value	Function
inputs	Unpremultiplied	<p>Select whether you are using a premultiplied or unpremultiplied input image:</p> <ul style="list-style-type: none"> • Premultiplied - the ImageGamma node unpremultiplies the input, applies the gamma change, and premultiplies the input again. This simulates applying the gamma change before the premultiplication was done, as color corrections are typically applied on unpremultiplied images. • Unpremultiplied - the ImageGamma node simply applies the gamma change.

ImageGeometric

This node layers images together using the **Geometric** compositing algorithm: $2FB/(F+B)$. This is another way of averaging two images. Visually, it's close to ImageMin.

You can also specify a different compositing algorithm using the **operation** control.

Also see [ImageAverage](#) on page 31 and [ImageHypot](#) on page 69.

Control (UI)	Default Value	Function
[2D node controls]	N/A	<p>Set the controls for the stereo view.</p> <p>For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.</p>

ImageGeometric parameters continued

Control (UI)	Default Value	Function
operation	Geometric	<p>If you don't want layer the images together using the Geometric compositing operation, select the operation to use instead.</p> <p>The following conventions apply to the below operation descriptions:</p> <ul style="list-style-type: none">• F refers to the fg input.• f refers to the fg input's alpha channel.• B refers to the bg input.• b refers to the bg input's alpha channel.

Control (UI)	Default Value	Function
operation (continued)		<p>The available operations (based on the Porter & Duff paper) are:</p> <ul style="list-style-type: none"> • Atop - $F + B(1 - f)$. This shows the shape of the background, with the foreground covering the background where the images overlap. • Average - $(F + B) / 2$. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • Difference - $\text{abs}(F - B)$. This shows how much the pixels differ and is useful for comparing two very similar images. • Divide - B / F. This divides the background values by the foreground values. • Exclusion - $F + B - 2FB$. This is a more photographic form of Difference. • From - $F - B$. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see Minus. • Geometric - $2FB / (F + B)$. This is another way of averaging two images. Visually, it's close to Min. • Hypot - $\text{sqrt}(F^2 + B^2)$. This resembles the Add and Screen operations. The result is not as bright as Add, but brighter than Screen. Hypot works with values above 1. It can be useful for adding reflections, as an alternative to Screen. • In - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • Matte - $Ff * B(1 - f)$. This is a premultiplied Over. Use unpremultiplied images with this operation. • Max - $\text{max}(F, B)$. This takes the maximum values of both images. This is a good way to combine mattes and useful for bringing aspects like bright hair detail through.

Control (UI)	Default Value	Function
operation (continued)		<ul style="list-style-type: none"> • Min - $\min(F,B)$. This takes the minimum values of both images. • Minus - $B-F$. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see From. • Multiply - FB. This multiplies the values of the foreground by the values of the background. It can be used to composite darker values from the foreground with the background image - dark gray smoke shot against a white background, for example. • Out - $B(1-f)$. This only shows the areas of the background that do not overlap with the alpha of the foreground. This can be useful for combining mattes. • Over - $F+B(1-f)$. This layers the foreground over the background according to the alpha of the foreground. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • Plus - $F+B$. This produces the sum of the foreground and background. Note that the add algorithm may result in pixel values higher than 1.0. • Screen - $F+B-FB$. This is similar to Hypot, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • Under - $F(1-b)+B$. This is the reverse of the Over operation. It layers the background over the foreground according to the alpha of the background.
amount	1	Dissolves between the bg image at 0 and the full merge effect at 1.
displayWindow	Background	<p>The frame size to output in the event that the fg and bg inputs are different sizes:</p> <ul style="list-style-type: none"> • Background - output the frame size of the bg input. • Foreground - output the frame size of the fg input. • Union - output a combination of the bg and fg inputs' frame sizes. • Intersection - output an intersection of the bg and fg inputs' frame sizes. This restricts the output to the area where the two frame sizes overlap.

Control (UI)	Default Value	Function
clampAlpha	enabled	When enabled, the output alpha channel is clamped to the 0-1 range. Color channels (RGB) are not affected.

ImageHypot

This node layers images together using the **Hypot** compositing algorithm: $\sqrt{F^2+B^2}$. This resembles the ImageAdd and ImageScreen operations. The result is not as bright as ImageAdd, but brighter than ImageScreen.

ImageHypot works with values above 1. It can be useful for adding reflections, as an alternative to ImageScreen.

You can also specify a different compositing algorithm using the **operation** control.

Also see [ImagePlus](#) on page 104 and [ImageScreen](#) on page 120.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.

ImageHypot parameters continued

operation	Hypot	<p>If you don't want layer the images together using the Hypot compositing operation, select the operation to use instead.</p> <p>The following conventions apply to the below operation descriptions:</p> <ul style="list-style-type: none"> • F refers to the fg input. • f refers to the fg input's alpha channel. • B refers to the bg input. • b refers to the bg input's alpha channel.
-----------	-------	--

Control (UI)	Default Value	Function
operation (continued)		<p>The available operations (based on the Porter & Duff paper) are:</p> <ul style="list-style-type: none"> • Atop - $F + B(1 - f)$. This shows the shape of the background, with the foreground covering the background where the images overlap. • Average - $(F + B) / 2$. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • Difference - $\text{abs}(F - B)$. This shows how much the pixels differ and is useful for comparing two very similar images. • Divide - B / F. This divides the background values by the foreground values. • Exclusion - $F + B - 2FB$. This is a more photographic form of Difference. • From - $F - B$. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see ImageMinus. • Geometric - $2FB / (F + B)$. This is another way of averaging two images. Visually, it's close to Min. • Hypot - $\sqrt{F^2 + B^2}$. This resembles the Add and Screen operations. The result is not as bright as Add, but brighter than Screen. Hypot works with values above 1. It can be useful for adding reflections, as an alternative to Screen. • In - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • Matte - $Ff * B(1 - f)$. This is a premultiplied Over. Use unpremultiplied images with this operation. • Max - $\max(F, B)$. This takes the maximum values of both images. This is a good way to combine mattes and useful for bringing aspects like bright hair detail through.

Control (UI)	Default Value	Function
operation (continued)		<ul style="list-style-type: none"> • Min - $\min(F,B)$. This takes the minimum values of both images. • Minus - $B-F$. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see From. • Multiply - FB. This multiplies the values of the foreground by the values of the background. It can be used to composite darker values from the foreground with the background image - dark gray smoke shot against a white background, for example. • Out - $B(1-f)$. This only shows the areas of the background that do not overlap with the alpha of the foreground. This can be useful for combining mattes. • Over - $F+B(1-f)$. This layers the foreground over the background according to the alpha of the foreground. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • Plus - $F+B$. This produces the sum of the foreground and background. Note that the add algorithm may result in pixel values higher than 1.0. • Screen - $F+B-FB$. This is similar to Hypot, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • Under - $F(1-b)+B$. This is the reverse of the Over operation. It layers the background over the foreground according to the alpha of the background.
amount	1	Dissolves between the bg image at 0 and the full merge effect at 1.
displayWindow	Background	<p>The frame size to output in the event that the fg and bg inputs are different sizes:</p> <ul style="list-style-type: none"> • Background - output the frame size of the bg input. • Foreground - output the frame size of the fg input. • Union - output a combination of the bg and fg inputs' frame sizes. • Intersection - output an intersection of the bg and fg inputs' frame sizes. This restricts the output to the area where the two frame sizes overlap.

Control (UI)	Default Value	Function
clampAlpha	enabled	When enabled, the output alpha channel is clamped to the 0-1 range. Color channels (RGB) are not affected.

ImageIn

This node layers images together using the **In** compositing algorithm: **Bf**. It only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes.

You can also specify a different compositing algorithm using the **operation** control.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.

ImageIn parameters continued

operation	In	<p>If you don't want layer the images together using the In compositing operation, select the operation to use instead.</p> <p>The following conventions apply to the below operation descriptions:</p> <ul style="list-style-type: none"> • F refers to the fg input. • f refers to the fg input's alpha channel. • B refers to the bg input. • b refers to the bg input's alpha channel.
-----------	----	---

Control (UI)	Default Value	Function
operation (continued)		<p>The available operations (based on the Porter & Duff paper) are:</p> <ul style="list-style-type: none"> • Atop - $F + B(1 - f)$. This shows the shape of the background, with the foreground covering the background where the images overlap. • Average - $(F + B) / 2$. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • Difference - $\text{abs}(F - B)$. This shows how much the pixels differ and is useful for comparing two very similar images. • Divide - B / F. This divides the background values by the foreground values. • Exclusion - $F + B - 2FB$. This is a more photographic form of Difference. • From - $F - B$. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see Minus. • Geometric - $2FB / (F + B)$. This is another way of averaging two images. Visually, it's close to Min. • Hypot - $\text{sqrt}(F^2 + B^2)$. This resembles the Add and Screen operations. The result is not as bright as Add, but brighter than Screen. Hypot works with values above 1. It can be useful for adding reflections, as an alternative to Screen. • In - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • Matte - $Ff * B(1 - f)$. This is a premultiplied Over. Use unpremultiplied images with this operation. • Max - $\text{max}(F, B)$. This takes the maximum values of both images. This is a good way to combine mattes and useful for bringing aspects like bright hair detail through.

Control (UI)	Default Value	Function
operation (continued)		<ul style="list-style-type: none"> • Min - $\min(F,B)$. This takes the minimum values of both images. • Minus - $B-F$. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see From. • Multiply - FB. This multiplies the values of the foreground by the values of the background. It can be used to composite darker values from the foreground with the background image - dark gray smoke shot against a white background, for example. • Out - $B(1-f)$. This only shows the areas of the background that do not overlap with the alpha of the foreground. This can be useful for combining mattes. • Over - $F+B(1-f)$. This layers the foreground over the background according to the alpha of the foreground. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • Plus - $F+B$. This produces the sum of the foreground and background. Note that the add algorithm may result in pixel values higher than 1.0. • Screen - $F+B-FB$. This is similar to Hypot, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • Under - $F(1-b)+B$. This is the reverse of the Over operation. It layers the background over the foreground according to the alpha of the background.
amount	1	Dissolves between the bg image at 0 and the full merge effect at 1.
displayWindow	Background	<p>The frame size to output in the event that the fg and bg inputs are different sizes:</p> <ul style="list-style-type: none"> • Background - output the frame size of the bg input. • Foreground - output the frame size of the fg input. • Union - output a combination of the bg and fg inputs' frame sizes. • Intersection - output an intersection of the bg and fg inputs' frame sizes. This restricts the output to the area where the two frame sizes overlap.

Control (UI)	Default Value	Function
clampAlpha	enabled	When enabled, the output alpha channel is clamped to the 0-1 range. Color channels (RGB) are not affected.

ImageInvert

Inverts a channel's values. To invert a channel is to subtract its values from 1, which causes its blacks to become white and its whites to become black. You may find this particularly useful to invert mattes.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.

ImageInvert parameters continued

mode	Additive	Sets the invert calculation mode: <ul style="list-style-type: none"> • Additive • Multiplicative
------	----------	--

mode: Additive

max	1.0000, 1.0000, 1.0000, 1.0000	The max color (RGBA values). You can also use the below RGB, HSL, or HSV controls to set the color. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.
-----	-----------------------------------	---

ImageLevels

This node controls the input, gamma, and output levels of the input image.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.
inputMin		
inputMin	0.0000, 0.0000, 0.0000, 0.0000	Sets the minimum input level for the RGBA values. You can also use the HSL or HSV controls to set the color. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.
inputMax		
inputMax	1.0000, 1.0000, 1.0000, 1.0000	Sets the maximum input level for the RGBA values. You can also use the HSL or HSV controls to set the color. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.
gamma		
gamma	1.0000, 1.0000, 1.0000, 1.0000	Sets the gamma levels for the RGBA values. You can also use the HSL or HSV controls to set the color. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.
outputMin		
outputMin	0.0000, 0.0000, 0.0000, 0.0000	Sets the minimum output level for the RGBA values. You can also use the HSL or HSV controls to set the color. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.
outputMax		
outputMax	1.0000, 1.0000, 1.0000, 1.0000	Sets the maximum output level for the RGBA values. You can also use the HSL or HSV controls to set the color. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.

Control (UI)	Default Value	Function
direction	Forward	
clampMin	No	When set to Yes , levels are clamped to the specified inputMin and outputMin values.
clampMax	No	When set to Yes , levels are clamped to the specified inputMax and outputMax values.

ImageMatte

This node layers images together using the **Matte** compositing algorithm: **Ff*B(1-f)**. It's a premultiplied ImageOver. Use unpremultiplied images with this operation.

You can also specify a different compositing algorithm using the **operation** control.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.

ImageMatte parameters continued

operation	Matte	<p>If you don't want layer the images together using the Matte compositing operation, select the operation to use instead.</p> <p>The following conventions apply to the below operation descriptions:</p> <ul style="list-style-type: none"> • F refers to the fg input. • f refers to the fg input's alpha channel. • B refers to the bg input. • b refers to the bg input's alpha channel.
-----------	-------	--

Control (UI)	Default Value	Function
operation (continued)		<p>The available operations (based on the Porter & Duff paper) are:</p> <ul style="list-style-type: none"> • Atop - $F + B(1 - f)$. This shows the shape of the background, with the foreground covering the background where the images overlap. • Average - $(F + B) / 2$. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • Difference - $\text{abs}(F - B)$. This shows how much the pixels differ and is useful for comparing two very similar images. • Divide - B / F. This divides the background values by the foreground values. • Exclusion - $F + B - 2FB$. This is a more photographic form of Difference. • From - $F - B$. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see ImageMinus. • Geometric - $2FB / (F + B)$. This is another way of averaging two images. Visually, it's close to Min. • Hypot - $\text{sqrt}(F^2 + B^2)$. This resembles the Add and Screen operations. The result is not as bright as Add, but brighter than Screen. Hypot works with values above 1. It can be useful for adding reflections, as an alternative to Screen. • In - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • Matte - $Ff * B(1 - f)$. This is a premultiplied Over. Use unpremultiplied images with this operation. • Max - $\text{max}(F, B)$. This takes the maximum values of both images. This is a good way to combine mattes and useful for bringing aspects like bright hair detail through.

Control (UI)	Default Value	Function
operation (continued)		<ul style="list-style-type: none"> • Min - $\min(F,B)$. This takes the minimum values of both images. • Minus - $B-F$. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see From. • Multiply - FB. This multiplies the values of the foreground by the values of the background. It can be used to composite darker values from the foreground with the background image - dark gray smoke shot against a white background, for example. • Out - $B(1-f)$. This only shows the areas of the background that do not overlap with the alpha of the foreground. This can be useful for combining mattes. • Over - $F+B(1-f)$. This layers the foreground over the background according to the alpha of the foreground. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • Plus - $F+B$. This produces the sum of the foreground and background. Note that the add algorithm may result in pixel values higher than 1.0. • Screen - $F+B-FB$. This is similar to Hypot, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • Under - $F(1-b)+B$. This is the reverse of the Over operation. It layers the background over the foreground according to the alpha of the background.
amount	1	Dissolves between the bg image at 0 and the full merge effect at 1.
displayWindow	Background	<p>The frame size to output in the event that the fg and bg inputs are different sizes:</p> <ul style="list-style-type: none"> • Background - output the frame size of the bg input. • Foreground - output the frame size of the fg input. • Union - output a combination of the bg and fg inputs' frame sizes. • Intersection - output an intersection of the bg and fg inputs' frame sizes. This restricts the output to the area where the two frame sizes overlap.

Control (UI)	Default Value	Function
clampAlpha	enabled	When enabled, the output alpha channel is clamped to the 0-1 range. Color channels (RGB) are not affected.

ImageMatteMix

This node uses a matte to control the mixing of two images: background and foreground.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.

ImageMatteMix parameters continued

amount	1	An optional bias for the mix operation. See amountMath for details.
amountMath	Normal	Select the mode to use: <ul style="list-style-type: none"> • Normal - In this mode, an amount of 1 means to obey the matte fully. An amount of less than 1 means to mix in more of the original bg image. This corresponds to artists' expectations. • Compatibility - In this mode, an amount of 0 favors the bg image. An amount towards 1 favors the fg image. 0.5 is the non-biased setting.
matteChannel	A	Specifies which channel in the matte input contains the matte.

Control (UI)	Default Value	Function
displayWindow	Background	<p>The frame size to output in the event that the fg and bg inputs are different sizes:</p> <ul style="list-style-type: none"> • Background - output the frame size of the bg input. • Foreground - output the frame size of the fg input. • Union - output a combination of the bg and fg inputs' frame sizes. • Intersection - output an intersection of the bg and fg inputs' frame sizes. This restricts the output to the area where the two frame sizes overlap.

ImageMax

This node layers images together using the **Max** compositing algorithm: **max(F,B)**. It takes the maximum values of both images. This is a good way to combine mattes and useful for bringing aspects like bright hair detail through.

You can also specify a different compositing algorithm using the **operation** control.

Control (UI)	Default Value	Function
[2D node controls]	N/A	<p>Set the controls for the stereo view.</p> <p>For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.</p>
ImageMax parameters continued		
operation	Max	<p>If you don't want layer the images together using the Max compositing operation, select the operation to use instead.</p> <p>The following conventions apply to the below operation descriptions:</p> <ul style="list-style-type: none"> • F refers to the fg input. • f refers to the fg input's alpha channel. • B refers to the bg input. • b refers to the bg input's alpha channel.

Control (UI)	Default Value	Function
operation (continued)		<p>The available operations (based on the Porter & Duff paper) are:</p> <ul style="list-style-type: none"> • Atop - $F+B(1-f)$. This shows the shape of the background, with the foreground covering the background where the images overlap. • Average - $(F+B)/2$. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • Difference - $\text{abs}(F-B)$. This shows how much the pixels differ and is useful for comparing two very similar images. • Divide - B/F. This divides the background values by the foreground values. • Exclusion - $F+B-2FB$. This is a more photographic form of Difference. • From - $F-B$. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see Minus. • Geometric - $2FB/(F+B)$. This is another way of averaging two images. Visually, it's close to Min. • Hypot - $\text{sqrt}(F^2+B^2)$. This resembles the Add and Screen operations. The result is not as bright as Add, but brighter than Screen. Hypot works with values above 1. It can be useful for adding reflections, as an alternative to Screen. • In - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • Matte - $Ff*B(1-f)$. This is a premultiplied Over. Use unpremultiplied images with this operation. • Max - $\text{max}(F,B)$. This takes the maximum values of both images. This is a good way to combine mattes and useful for bringing aspects like bright hair detail through.

Control (UI)	Default Value	Function
operation (continued)		<ul style="list-style-type: none"> • Min - $\min(F,B)$. This takes the minimum values of both images. • Minus - $B-F$. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see From. • Multiply - FB. This multiplies the values of the foreground by the values of the background. It can be used to composite darker values from the foreground with the background image - dark gray smoke shot against a white background, for example. • Out - $B(1-f)$. This only shows the areas of the background that do not overlap with the alpha of the foreground. This can be useful for combining mattes. • Over - $F+B(1-f)$. This layers the foreground over the background according to the alpha of the foreground. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • Plus - $F+B$. This produces the sum of the foreground and background. Note that the add algorithm may result in pixel values higher than 1.0. • Screen - $F+B-FB$. This is similar to Hypot, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • Under - $F(1-b)+B$. This is the reverse of the Over operation. It layers the background over the foreground according to the alpha of the background.
amount	1	Dissolves between the bg image at 0 and the full merge effect at 1.
displayWindow	Background	<p>The frame size to output in the event that the fg and bg inputs are different sizes:</p> <ul style="list-style-type: none"> • Background - output the frame size of the bg input. • Foreground - output the frame size of the fg input. • Union - output a combination of the bg and fg inputs' frame sizes. • Intersection - output an intersection of the bg and fg inputs' frame sizes. This restricts the output to the area where the two frame sizes overlap.

Control (UI)	Default Value	Function
ignoreZero	disabled	
clampAlpha	enabled	When enabled, the output alpha channel is clamped to the 0-1 range. Color channels (RGB) are not affected.

ImageMerge

This node is a generic merge node that is able to perform all the other merge operations supported by Katana.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.

ImageMerge parameters continued

operation	Merge	<p>If you don't want layer the images together using the Merge compositing operation, select the operation to use instead.</p> <p>The following conventions apply to the below operation descriptions:</p> <ul style="list-style-type: none"> • F refers to the fg input. • f refers to the fg input's alpha channel. • B refers to the bg input. • b refers to the bg input's alpha channel.
-----------	-------	--

Control (UI)	Default Value	Function
operation (continued)		<p>The available operations (based on the Porter & Duff paper) are:</p> <ul style="list-style-type: none"> • Atop - $F + B(1 - f)$. This shows the shape of the background, with the foreground covering the background where the images overlap. • Average - $(F + B) / 2$. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • Difference - $\text{abs}(F - B)$. This shows how much the pixels differ and is useful for comparing two very similar images. • Divide - B / F. This divides the background values by the foreground values. • Exclusion - $F + B - 2FB$. This is a more photographic form of Difference. • From - $F - B$. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see Minus. • Geometric - $2FB / (F + B)$. This is another way of averaging two images. Visually, it's close to Min. • Hypot - $\text{sqrt}(F^2 + B^2)$. This resembles the Add and Screen operations. The result is not as bright as Add, but brighter than Screen. Hypot works with values above 1. It can be useful for adding reflections, as an alternative to Screen. • In - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • Matte - $Ff * B(1 - f)$. This is a premultiplied Over. Use unpremultiplied images with this operation. • Max - $\text{max}(F, B)$. This takes the maximum values of both images. This is a good way to combine mattes and useful for bringing aspects like bright hair detail through.

Control (UI)	Default Value	Function
operation (continued)		<ul style="list-style-type: none"> • Min - $\min(F,B)$. This takes the minimum values of both images. • Minus - $B-F$. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see From. • Multiply - FB. This multiplies the values of the foreground by the values of the background. It can be used to composite darker values from the foreground with the background image - dark gray smoke shot against a white background, for example. • Out - $B(1-f)$. This only shows the areas of the background that do not overlap with the alpha of the foreground. This can be useful for combining mattes. • Over - $F+B(1-f)$. This layers the foreground over the background according to the alpha of the foreground. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • Plus - $F+B$. This produces the sum of the foreground and background. Note that the add algorithm may result in pixel values higher than 1.0. • Screen - $F+B-FB$. This is similar to Hypot, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • Under - $F(1-b)+B$. This is the reverse of the Over operation. It layers the background over the foreground according to the alpha of the background.
amount	1	Dissolves between the bg image at 0 and the full merge effect at 1.
displayWindow	Background	<p>The frame size to output in the event that the fg and bg inputs are different sizes:</p> <ul style="list-style-type: none"> • Background - output the frame size of the bg input. • Foreground - output the frame size of the fg input. • Union - output a combination of the bg and fg inputs' frame sizes. • Intersection - output an intersection of the bg and fg inputs' frame sizes. This restricts the output to the area where the two frame sizes overlap.

Control (UI)	Default Value	Function
clampAlpha	enabled	When enabled, the output alpha channel is clamped to the 0-1 range. Color channels (RGB) are not affected.

ImageMin

This node layers images together using the **Min** compositing algorithm: **min(F,B)**. It takes the minimum values of both images.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.

ImageMin parameters continued

operation	Min	<p>If you don't want layer the images together using the Min compositing operation, select the operation to use instead.</p> <p>The following conventions apply to the below operation descriptions:</p> <ul style="list-style-type: none"> • F refers to the fg input. • f refers to the fg input's alpha channel. • B refers to the bg input. • b refers to the bg input's alpha channel.
-----------	-----	--

Control (UI)	Default Value	Function
operation (continued)		<p>The available operations (based on the Porter & Duff paper) are:</p> <ul style="list-style-type: none"> • Atop - $F + B(1-f)$. This shows the shape of the background, with the foreground covering the background where the images overlap. • Average - $(F+B)/2$. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • Difference - $\text{abs}(F-B)$. This shows how much the pixels differ and is useful for comparing two very similar images. • Divide - B/F. This divides the background values by the foreground values. • Exclusion - $F+B-2FB$. This is a more photographic form of Difference. • From - $F-B$. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see Minus. • Geometric - $2FB/(F+B)$. This is another way of averaging two images. Visually, it's close to Min. • Hypot - $\text{sqrt}(F^2+B^2)$. This resembles the Add and Screen operations. The result is not as bright as Add, but brighter than Screen. Hypot works with values above 1. It can be useful for adding reflections, as an alternative to Screen. • In - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • Matte - $Ff*B(1-f)$. This is a premultiplied Over. Use unpremultiplied images with this operation. • Max - $\text{max}(F,B)$. This takes the maximum values of both images. This is a good way to combine mattes and useful for bringing aspects like bright hair detail through.

Control (UI)	Default Value	Function
operation (continued)		<ul style="list-style-type: none"> • Min - $\min(F,B)$. This takes the minimum values of both images. • Minus - $B-F$. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see From. • Multiply - FB. This multiplies the values of the foreground by the values of the background. It can be used to composite darker values from the foreground with the background image - dark gray smoke shot against a white background, for example. • Out - $B(1-f)$. This only shows the areas of the background that do not overlap with the alpha of the foreground. This can be useful for combining mattes. • Over - $F+B(1-f)$. This layers the foreground over the background according to the alpha of the foreground. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • Plus - $F+B$. This produces the sum of the foreground and background. Note that the add algorithm may result in pixel values higher than 1.0. • Screen - $F+B-FB$. This is similar to Hypot, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • Under - $F(1-b)+B$. This is the reverse of the Over operation. It layers the background over the foreground according to the alpha of the background.
amount	1	Dissolves between the bg image at 0 and the full merge effect at 1.
displayWindow	Background	<p>The frame size to output in the event that the fg and bg inputs are different sizes:</p> <ul style="list-style-type: none"> • Background - output the frame size of the bg input. • Foreground - output the frame size of the fg input. • Union - output a combination of the bg and fg inputs' frame sizes. • Intersection - output an intersection of the bg and fg inputs' frame sizes. This restricts the output to the area where the two frame sizes overlap.

Control (UI)	Default Value	Function
ignoreZero	disabled	
clampAlpha	enabled	When enabled, the output alpha channel is clamped to the 0-1 range. Color channels (RGB) are not affected.

ImageMinus

This node layers images together using the **Minus** compositing algorithm: **B-F**. It subtracts the foreground from the background.

Also see [ImageFrom](#) on page 59.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.

ImageMinus parameters continued

operation	Minus	<p>If you don't want layer the images together using the Minus compositing operation, select the operation to use instead.</p> <p>The following conventions apply to the below operation descriptions:</p> <ul style="list-style-type: none"> • F refers to the fg input. • f refers to the fg input's alpha channel. • B refers to the bg input. • b refers to the bg input's alpha channel.
-----------	-------	--

Control (UI)	Default Value	Function
operation (continued)		<p>The available operations (based on the Porter & Duff paper) are:</p> <ul style="list-style-type: none"> • Atop - $F + B(1 - f)$. This shows the shape of the background, with the foreground covering the background where the images overlap. • Average - $(F + B) / 2$. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • Difference - $\text{abs}(F - B)$. This shows how much the pixels differ and is useful for comparing two very similar images. • Divide - B / F. This divides the background values by the foreground values. • Exclusion - $F + B - 2FB$. This is a more photographic form of Difference. • From - $F - B$. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see Minus. • Geometric - $2FB / (F + B)$. This is another way of averaging two images. Visually, it's close to Min. • Hypot - $\text{sqrt}(F^2 + B^2)$. This resembles the Add and Screen operations. The result is not as bright as Add, but brighter than Screen. Hypot works with values above 1. It can be useful for adding reflections, as an alternative to Screen. • In - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • Matte - $Ff * B(1 - f)$. This is a premultiplied Over. Use unpremultiplied images with this operation. • Max - $\text{max}(F, B)$. This takes the maximum values of both images. This is a good way to combine mattes and useful for bringing aspects like bright hair detail through.

Control (UI)	Default Value	Function
operation (continued)		<ul style="list-style-type: none"> • Min - $\min(F,B)$. This takes the minimum values of both images. • Minus - $B-F$. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see From. • Multiply - FB. This multiplies the values of the foreground by the values of the background. It can be used to composite darker values from the foreground with the background image - dark gray smoke shot against a white background, for example. • Out - $B(1-f)$. This only shows the areas of the background that do not overlap with the alpha of the foreground. This can be useful for combining mattes. • Over - $F+B(1-f)$. This layers the foreground over the background according to the alpha of the foreground. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • Plus - $F+B$. This produces the sum of the foreground and background. Note that the add algorithm may result in pixel values higher than 1.0. • Screen - $F+B-FB$. This is similar to Hypot, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • Under - $F(1-b)+B$. This is the reverse of the Over operation. It layers the background over the foreground according to the alpha of the background.
amount	1	Dissolves between the bg image at 0 and the full merge effect at 1.
displayWindow	Background	<p>The frame size to output in the event that the fg and bg inputs are different sizes:</p> <ul style="list-style-type: none"> • Background - output the frame size of the bg input. • Foreground - output the frame size of the fg input. • Union - output a combination of the bg and fg inputs' frame sizes. • Intersection - output an intersection of the bg and fg inputs' frame sizes. This restricts the output to the area where the two frame sizes overlap.

Control (UI)	Default Value	Function
clampAlpha	enabled	When enabled, the output alpha channel is clamped to the 0-1 range. Color channels (RGB) are not affected.

ImageMix

This node performs a simple foreground over background mix of two inputs.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.
ImageMix parameters continued		
amount	0.5	Sets the amount of mix between the background image at 0 and both images at 1.
displayWindow	Background	The frame size to output in the event that the fg and bg inputs are different sizes: <ul style="list-style-type: none"> • Background - output the frame size of the bg input. • Foreground - output the frame size of the fg input. • Union - output a combination of the bg and fg inputs' frame sizes. • Intersection - output an intersection of the bg and fg inputs' frame sizes. This restricts the output to the area where the two frame sizes overlap.

ImageMultiply

This node layers images together using the **Multiply** compositing algorithm: **FB**. It multiplies the values of the foreground by the values of the background. It can be used to composite darker values from the foreground with the background image - dark gray smoke shot against a white background, for example.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.
ImageMultiply parameters continued		

Control (UI)	Default Value	Function
operation	Multiply	<p>If you don't want layer the images together using the Multiply compositing operation, select the operation to use instead.</p> <p>The following conventions apply to the below operation descriptions:</p> <ul style="list-style-type: none"> • F refers to the fg input. • f refers to the fg input's alpha channel. • B refers to the bg input. • b refers to the bg input's alpha channel. <p>The available operations (based on the Porter & Duff paper) are:</p> <ul style="list-style-type: none"> • Atop - $Fb+B(1-f)$. This shows the shape of the background, with the foreground covering the background where the images overlap. • Average - $(F+B)/2$. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • Difference - $\text{abs}(F-B)$. This shows how much the pixels differ and is useful for comparing two very similar images. • Divide - B/F. This divides the background values by the foreground values. • Exclusion - $F+B-2FB$. This is a more photographic form of Difference. • From - $F-B$. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see Minus. • Geometric - $2FB/(F+B)$. This is another way of averaging two images. Visually, it's close to Min. • Hypot - $\text{sqrt}(F^2+B^2)$. This resembles the Add and Screen operations. The result is not as bright as Add, but brighter than Screen. Hypot works with values above 1. It can be useful for adding reflections, as an alternative to Screen.

Control (UI)	Default Value	Function
operation (continued)		<ul style="list-style-type: none"> • In - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • Matte - $Ff*B(1-f)$. This is a premultiplied Over. Use unpremultiplied images with this operation. • Max - $\max(F,B)$. This takes the maximum values of both images. This is a good way to combine mattes and useful for bringing aspects like bright hair detail through. • Min - $\min(F,B)$. This takes the minimum values of both images. • Minus - $B-F$. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see From. • Multiply - FB. This multiplies the values of the foreground by the values of the background. It can be used to composite darker values from the foreground with the background image - dark gray smoke shot against a white background, for example. • Out - $B(1-f)$. This only shows the areas of the background that do not overlap with the alpha of the foreground. This can be useful for combining mattes. • Over - $F+B(1-f)$. This layers the foreground over the background according to the alpha of the foreground. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • Plus - $F+B$. This produces the sum of the foreground and background. Note that the add algorithm may result in pixel values higher than 1.0. • Screen - $F+B-FB$. This is similar to Hypot, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • Under - $F(1-b)+B$. This is the reverse of the Over operation. It layers the background over the foreground according to the alpha of the background.
amount	1	Dissolves between the bg image at 0 and the full merge effect at 1.

Control (UI)	Default Value	Function
displayWindow	Background	<p>The frame size to output in the event that the fg and bg inputs are different sizes:</p> <ul style="list-style-type: none"> • Background - output the frame size of the bg input. • Foreground - output the frame size of the fg input. • Union - output a combination of the bg and fg inputs' frame sizes. • Intersection - output an intersection of the bg and fg inputs' frame sizes. This restricts the output to the area where the two frame sizes overlap.
clampAlpha	enabled	When enabled, the output alpha channel is clamped to the 0-1 range. Color channels (RGB) are not affected.

ImageOrient

This node allows you to rotate, flip, and flop the input image around its center. A flip on the x axis mirrors the image vertically. A flop on the on the y axis mirrors the image horizontally.

Control (UI)	Default Value	Function
[2D node controls]	N/A	<p>Set the controls for the stereo view.</p> <p>For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.</p>

ImageOrient parameters continued

orientation	No Change	<p>Select how to rotate the input image:</p> <ul style="list-style-type: none"> • No Change - Do not rotate the image. • Rotate 90 - Rotate the image 90 degrees clockwise. • Rotate 180 - Rotate the image 180 degrees clockwise. • Rotate 270 - Rotate the image 270 degrees clockwise. • Flip - Mirror the image vertically (turning the image upside down). • Flop - Mirror the image horizontally. • FlipFlop - Mirror the image vertically and horizontally. This is the same as Rotate 180.
-------------	-----------	---

ImageOut

This node layers images together using the **Out** compositing algorithm: **B(1-f)**. Only shows the areas of the background that do not overlap with the alpha of the foreground. This can be useful for combining mattes.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.
ImageOut parameters continued		
operation	Out	If you don't want layer the images together using the Out compositing operation, select the operation to use instead. The following conventions apply to the below operation descriptions: <ul style="list-style-type: none"> • F refers to the fg input. • f refers to the fg input's alpha channel. • B refers to the bg input. • b refers to the bg input's alpha channel.

Control (UI)	Default Value	Function
operation (continued)		<p>The available operations (based on the Porter & Duff paper) are:</p> <ul style="list-style-type: none"> • Atop - $F+B(1-f)$. This shows the shape of the background, with the foreground covering the background where the images overlap. • Average - $(F+B)/2$. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • Difference - $\text{abs}(F-B)$. This shows how much the pixels differ and is useful for comparing two very similar images. • Divide - B/F. This divides the background values by the foreground values. • Exclusion - $F+B-2FB$. This is a more photographic form of Difference. • From - $F-B$. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see Minus. • Geometric - $2FB/(F+B)$. This is another way of averaging two images. Visually, it's close to Min. • Hypot - $\text{sqrt}(F^2+B^2)$. This resembles the Add and Screen operations. The result is not as bright as Add, but brighter than Screen. Hypot works with values above 1. It can be useful for adding reflections, as an alternative to Screen. • In - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • Matte - $Ff*B(1-f)$. This is a premultiplied Over. Use unpremultiplied images with this operation. • Max - $\text{max}(F,B)$. This takes the maximum values of both images. This is a good way to combine mattes and useful for bringing aspects like bright hair detail through.

Control (UI)	Default Value	Function
operation (continued)		<ul style="list-style-type: none"> • Min - $\min(F,B)$. This takes the minimum values of both images. • Minus - $B-F$. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see From. • Multiply - FB. This multiplies the values of the foreground by the values of the background. It can be used to composite darker values from the foreground with the background image - dark gray smoke shot against a white background, for example. • Out - $B(1-f)$. This only shows the areas of the background that do not overlap with the alpha of the foreground. This can be useful for combining mattes. • Over - $F+B(1-f)$. This layers the foreground over the background according to the alpha of the foreground. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • Plus - $F+B$. This produces the sum of the foreground and background. Note that the add algorithm may result in pixel values higher than 1.0. • Screen - $F+B-FB$. This is similar to Hypot, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • Under - $F(1-b)+B$. This is the reverse of the Over operation. It layers the background over the foreground according to the alpha of the background.
amount	1	Dissolves between the bg image at 0 and the full merge effect at 1.
displayWindow	Background	<p>The frame size to output in the event that the fg and bg inputs are different sizes:</p> <ul style="list-style-type: none"> • Background - output the frame size of the bg input. • Foreground - output the frame size of the fg input. • Union - output a combination of the bg and fg inputs' frame sizes. • Intersection - output an intersection of the bg and fg inputs' frame sizes. This restricts the output to the area where the two frame sizes overlap.

Control (UI)	Default Value	Function
clampAlpha	enabled	When enabled, the output alpha channel is clamped to the 0-1 range. Color channels (RGB) are not affected.

ImageOver

This node layers images together using the **Over** compositing algorithm: $F+B(1-f)$. This layers the foreground over the background according to the alpha of the foreground. This is the most commonly used operation. It's used when layering a foreground element over a background plate.

Also see [ImageUnder](#) on page 132.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.

ImageOver parameters continued

operation	Over	<p>If you don't want layer the images together using the Over compositing operation, select the operation to use instead.</p> <p>The following conventions apply to the below operation descriptions:</p> <ul style="list-style-type: none"> • F refers to the fg input. • f refers to the fg input's alpha channel. • B refers to the bg input. • b refers to the bg input's alpha channel.
-----------	------	---

Control (UI)	Default Value	Function
operation (continued)		<p>The available operations (based on the Porter & Duff paper) are:</p> <ul style="list-style-type: none"> • Atop - $F + B(1 - f)$. This shows the shape of the background, with the foreground covering the background where the images overlap. • Average - $(F + B) / 2$. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • Difference - $\text{abs}(F - B)$. This shows how much the pixels differ and is useful for comparing two very similar images. • Divide - B / F. This divides the background values by the foreground values. • Exclusion - $F + B - 2FB$. This is a more photographic form of Difference. • From - $F - B$. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see Minus. • Geometric - $2FB / (F + B)$. This is another way of averaging two images. Visually, it's close to Min. • Hypot - $\text{sqrt}(F^2 + B^2)$. This resembles the Add and Screen operations. The result is not as bright as Add, but brighter than Screen. Hypot works with values above 1. It can be useful for adding reflections, as an alternative to Screen. • In - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • Matte - $Ff * B(1 - f)$. This is a premultiplied Over. Use unpremultiplied images with this operation. • Max - $\text{max}(F, B)$. This takes the maximum values of both images. This is a good way to combine mattes and useful for bringing aspects like bright hair detail through.

Control (UI)	Default Value	Function
operation (continued)		<ul style="list-style-type: none"> • Min - $\min(F,B)$. This takes the minimum values of both images. • Minus - $B-F$. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see From. • Multiply - FB. This multiplies the values of the foreground by the values of the background. It can be used to composite darker values from the foreground with the background image - dark gray smoke shot against a white background, for example. • Out - $B(1-f)$. This only shows the areas of the background that do not overlap with the alpha of the foreground. This can be useful for combining mattes. • Over - $F+B(1-f)$. This layers the foreground over the background according to the alpha of the foreground. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • Plus - $F+B$. This produces the sum of the foreground and background. Note that the add algorithm may result in pixel values higher than 1.0. • Screen - $F+B-FB$. This is similar to Hypot, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • Under - $F(1-b)+B$. This is the reverse of the Over operation. It layers the background over the foreground according to the alpha of the background.
amount	1	Dissolves between the bg image at 0 and the full merge effect at 1.
displayWindow	Background	<p>The frame size to output in the event that the fg and bg inputs are different sizes:</p> <ul style="list-style-type: none"> • Background - output the frame size of the bg input. • Foreground - output the frame size of the fg input. • Union - output a combination of the bg and fg inputs' frame sizes. • Intersection - output an intersection of the bg and fg inputs' frame sizes. This restricts the output to the area where the two frame sizes overlap.

Control (UI)	Default Value	Function
clampAlpha	enabled	When enabled, the output alpha channel is clamped to the 0-1 range. Color channels (RGB) are not affected.

ImagePlus

This node layers images together using the **Plus** compositing algorithm: **F+B**. It produces the sum of the foreground and background. Note that the add algorithm may result in pixel values higher than 1.0.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.

ImagePlus parameters continued

operation	Plus	<p>If you don't want layer the images together using the Plus compositing operation, select the operation to use instead.</p> <p>The following conventions apply to the below operation descriptions:</p> <ul style="list-style-type: none"> • F refers to the fg input. • f refers to the fg input's alpha channel. • B refers to the bg input. • b refers to the bg input's alpha channel.
-----------	------	---

Control (UI)	Default Value	Function
operation (continued)		<p>The available operations (based on the Porter & Duff paper) are:</p> <ul style="list-style-type: none"> • Atop - $F + B(1 - f)$. This shows the shape of the background, with the foreground covering the background where the images overlap. • Average - $(F + B) / 2$. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • Difference - $\text{abs}(F - B)$. This shows how much the pixels differ and is useful for comparing two very similar images. • Divide - B / F. This divides the background values by the foreground values. • Exclusion - $F + B - 2FB$. This is a more photographic form of Difference. • From - $F - B$. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see Minus. • Geometric - $2FB / (F + B)$. This is another way of averaging two images. Visually, it's close to Min. • Hypot - $\sqrt{F^2 + B^2}$. This resembles the Add and Screen operations. The result is not as bright as Add, but brighter than Screen. Hypot works with values above 1. It can be useful for adding reflections, as an alternative to Screen. • In - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • Matte - $Ff * B(1 - f)$. This is a premultiplied Over. Use unpremultiplied images with this operation. • Max - $\max(F, B)$. This takes the maximum values of both images. This is a good way to combine mattes and useful for bringing aspects like bright hair detail through.

Control (UI)	Default Value	Function
operation (continued)		<ul style="list-style-type: none"> • Min - $\min(F,B)$. This takes the minimum values of both images. • Minus - $B-F$. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see From. • Multiply - FB. This multiplies the values of the foreground by the values of the background. It can be used to composite darker values from the foreground with the background image - dark gray smoke shot against a white background, for example. • Out - $B(1-f)$. This only shows the areas of the background that do not overlap with the alpha of the foreground. This can be useful for combining mattes. • Over - $F+B(1-f)$. This layers the foreground over the background according to the alpha of the foreground. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • Plus - $F+B$. This produces the sum of the foreground and background. Note that the add algorithm may result in pixel values higher than 1.0. • Screen - $F+B-FB$. This is similar to Hypot, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • Under - $F(1-b)+B$. This is the reverse of the Over operation. It layers the background over the foreground according to the alpha of the background.
amount	1	Dissolves between the bg image at 0 and the full merge effect at 1.
displayWindow	Background	<p>The frame size to output in the event that the fg and bg inputs are different sizes:</p> <ul style="list-style-type: none"> • Background - output the frame size of the bg input. • Foreground - output the frame size of the fg input. • Union - output a combination of the bg and fg inputs' frame sizes. • Intersection - output an intersection of the bg and fg inputs' frame sizes. This restricts the output to the area where the two frame sizes overlap.

Control (UI)	Default Value	Function
clampAlpha	enabled	When enabled, the output alpha channel is clamped to the 0-1 range. Color channels (RGB) are not affected.

ImagePosition

This node applies an integer, non-resampled offset to the input image.

If you are looking to do a transform with sub-pixel re-sampling, see [ImageTransform2D](#) on page 127 instead.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.
offset		
x	0	The number of pixels by which you want to offset the input image along the x axis. For example, if you enter 2 in this field, 2 is added to the x values.
y	0	The number of pixels by which you want to offset the input image along the y axis. For example, if you enter 2 in this field, 2 is added to the y values.
adjustDisplayWindow	disabled	When enabled, the displayWindow is repositioned along with the image content.  NOTE: This is very rarely desired, as convention dictates that the displayWindow should always have the lower left corner pinned to 0, 0.

ImagePremultiply

This node premultiplies (mult) the rgb channels by the **alphaChannel** when an image is connected to the **alpha** input. Otherwise, **a** is read from the **input** leaving the alpha channel unchanged.

Also see [ImageUnpremultiply](#) on page 136.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.
ImagePremultiply parameters continued		
alphaChannel	A	If a separate alpha input is provided, choose which of its channels to use to premultiply. If no separate alpha input is provided, the alpha from input is used and this control is disabled.

ImageRamp

Generates a gradation with various parameters.

Control (UI)	Default Value	Function
bounds		
[resolution]	Dependent on Project Settings	Sets the size of the display window using the dropdown menu.
bounds > ▾	N/A	For more information, refer to the Rectangle Widget Type in the Common Parameter Widgets on page 13.
left	0	Set the left position of the ROI in the Monitor tab.

Control (UI)	Default Value	Function
bottom	0	Set the bottom position of the ROI in the Monitor tab.
width	globals.width	Set the width of the ROI in the Monitor tab.
height	global.height	Set the height of the ROI in the Monitor tab.
ImageRamp parameters continued		
type	Horizontal	Set the pattern for the gradation from the options: <ul style="list-style-type: none"> • Horizontal • Vertical • Diagonal • Linear • Corner • Radial
interpolationColorspaces	linear	Specify the colorspace used for the color fields.
type: Horizontal or Vertical		
start > color	1.0000, 1.0000, 1.0000, 1.0000	The color (RGBA values) at the start of the ImageRamp. Alternatively, you can also use the color options below to set the color. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.
end > color	0.0000, 0.0000, 0.0000, 0.0000	The color (RGBA values) at the end of the ImageRamp. Alternatively, you can also use the color options below to set the color. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.
type: Diagonal		

Control (UI)	Default Value	Function
start > color	1.0000, 1.0000, 1.0000, 1.0000	The color (RGBA values) at the start of the ImageRamp. Alternatively, you can also use the color options below to set the color. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.
end > color	0.0000, 0.0000, 0.0000, 0.0000	The color (RGBA values) at the end of the ImageRamp. Alternatively, you can also use the color options below to set the color. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.
sense	Left to Right	Set whether the start color begins on the left and transitions to the end color on the right, or the other way around.
type: Linear		
start > color	1.0000, 1.0000, 1.0000, 1.0000	The color (RGBA values) at the start of the ImageRamp. Alternatively, you can also use the color options below to set the color. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.
start > Start	globals.width, globals.height	Set the position in the Monitor tab, or with global manipulators turned on, reposition the start point in the Monitor tab to adjust the values in the Start field.
inner > add point	N/A	Add a point to the gradation, where a specific color can be set to a specific position . By default, when you add a point, the name is filled in the blank box with p_0, ascending in number as you add more points. This can be changed to a more meaningful name.
end > color	0.0000, 0.0000, 0.0000, 0.0000	The color (RGBA values) at the end of the ImageRamp. Alternatively, you can also use the color options below to set the color. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.

Control (UI)	Default Value	Function
end > End	globals.width, globals.height	Set the position in the Monitor tab, or with global manipulators turned on, reposition the end point in the Monitor tab to adjust the values in the End field.
type: Corner		
bottomLeft	1.0000, 0.0000, 0.0000, 1.0000	The color (RGBA values) at the bottom-left corner of the ImageRamp. Alternatively, you can also use the color options below to set the color. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.
bottomRight	0.0000, 1.0000, 0.0000, 1.0000	The color (RGBA values) at the bottom-right corner of the ImageRamp. Alternatively, you can also use the color options below to set the color. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.
topRight	0.0000, 0.0000, 1.0000, 1.0000	The color (RGBA values) at the top-right corner of the ImageRamp. Alternatively, you can also use the color options below to set the color. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.
topLeft	0.0000, 0.0000, 0.0000, 0.0000	The color (RGBA values) at the top-left corner of the ImageRamp. Alternatively, you can also use the color options below to set the color. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.
type: Radial		
start > color	1.0000, 1.0000, 1.0000, 1.0000	The color (RGBA values) at the start of the ImageRamp. Alternatively, you can also use the color options below to set the color. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.

Control (UI)	Default Value	Function
start > Start	globals.width, globals.height	Set the position in the Monitor tab, or with global manipulators turned on, reposition the start point in the Monitor tab to adjust the values in the Start field.
end > color	0.0000, 0.0000, 0.0000, 0.0000	The color (RGBA values) at the end of the ImageRamp. Alternatively, you can also use the color options below to set the color. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.
radial > confineTo Circle	disabled	Set whether the start and end colors in the radial pattern are confined to a circle.
radial > radius	778	Specify the radius of the circle.
radial > aspectRatio	1	Specify the aspect ratio of the circle.
radial > innerRadius	0	Specify the radius of the inner circle, as set by the start color.
radial > fallOff	0.5	Controls the profile of the fall off between the start and end radii.

ImageRead

This node loads images from disk, using the native resolution and the frame range for the sequence. It converts all imported sequences to Katana's linear colorspace automatically, but there are options to control this. Note that Katana's image processing operations are written assuming they are working on linear images, so be careful if you change the default input colorspace conversion. All of Katana's image processing is implemented in floating point, so files are converted to float at input.

Control (UI)	Default Value	Function
file	none	The image sequence to load. For more information, refer to the Asset and File Path Widget Types in the Common Parameter Widgets on page 13.

Control (UI)	Default Value	Function
image		
rawData	disabled	When enabled, Katana skips the automatic colorspace conversion. Note that Katana is inherently a floating-point system. Thus, if integer data is loaded (at any bit-depth), the pixels are mapped to the range of [0, 1].
colorspace	auto	<p>Select the colorspace for the file on disk. Upon load, the image sequence is converted from this colorspace to Katana's native floating-point linear colorspace. The default value, auto, means Katana tries to determine the bit depth from the file header and the colorspace from the file name. If Katana gets this wrong or the file is not named in a standard way, you can use this control to force Katana to assume the image data is in the selected colorspace and bit depth.</p> <p>You can also use this control to avoid the colorspace conversion entirely by specifying lnzf or ncf, which indicate that the file is already linear. Bear in mind, however, that most image processing operations in Katana presume linear input data. The results of image processing operations in Katana are not defined, tested, or supported for non-linear image data. All operators have been implemented assuming input images are linear.</p> <p>Note: This option only appears when rawData is disabled.</p>
isProxy	disabled	When enabled, Katana assumes the loaded image sequence is a proxy rather than a full-resolution image. This is preferable to manually resizing the image, as it is more efficient when proxy-rendering is enabled.
image > isProxy: enabled > fullResFrame		

Control (UI)	Default Value	Function
[resolution]	Dependent on Project Settings	When isProxy is enabled, you can use this control to select the resolution for the full-resolution image.
left	timing.missingFrameBounds.left	Sets the left position of the rectangle.
bottom	timing.missingFrameBounds.bottom	Sets the bottom position of the rectangle.
width	timing.missingFrameBounds.width	Sets the width of the rectangle.
height	timing.missingFrameBounds.height	Sets the height of the rectangle.
timing		
frame	frame	<p>Sets the frame number actually read from disk prior to applying the inMode, outMode, firstFrame, and lastFrame settings. When a downstream node requests an image from an ImageRead node, ImageRead evaluates this control to determine the frame number to read from disk (by default this is the current time). The result is compared against the firstFrame and lastFrame values and, if necessary, any remapping of the actual frame number is done based on the inMode and outMode settings.</p> <p>You can retime or offset your input by using an expression or a curve here, but note that currently Katana only reads the nearest frame and doesn't generate in-between frames (no optical flow interpolation). The value is forced to an integer at the time it's used, so you don't need to worry about this if you don't want to.</p>
inMode	Black	<p>Sets what to do when a frame is required at a time value prior to firstFrame:</p> <ul style="list-style-type: none"> • Black firstFrame to black. • Freeze • Repeat • Mirror

Control (UI)	Default Value	Function
outMode	Black	<p>Sets what to do when a frame is required at a time value after lastFrame:</p> <ul style="list-style-type: none"> • Black lastFrame to black. • Freeze • Repeat • Mirror
firstFrame	globals.inTime	<p>Sets the first valid frame of the sequence of images on disk. If a frame prior to firstFrame is required, its contents are determined based on inMode.</p> <p>If the file control has frame range values in it and this control is left at its default value, the value in the file control is obeyed.</p>
lastFrame	globals.outTime	<p>Sets the last valid frame of the sequence of images on disk. If a frame beyond lastFrame is required, its contents are determined based on outMode.</p> <p>If the file control has frame range values in it and this control is left at its default value, the value in the file control is obeyed.</p>
lockSettings	disabled	<p>When enabled, the firstFrame, lastFrame, inMode, and outMode values aren't automatically updated when a new file sequence is chosen.</p>

Control (UI)	Default Value	Function
missingFrames	Error	<p>Specifies what to do if a frame is not found:</p> <ul style="list-style-type: none"> • Error - have the render fail with an error. • Black - replace any missing frames with black. • Nearest - replace any missing frames with the nearest frame. • Checkerboard - replace any missing frames with a checkerboard image. <p> NOTE: If no frames in the image sequence are present, the render fails regardless of this control's setting.</p>
advanced		
includeInErrorChecking	enabled	<p>When enabled, Katana includes this node when it automatically checks ImageRead nodes for errors.</p> <p>When disabled, Katana excludes this node when it automatically checks ImageRead nodes for errors.</p>

ImageReformat

Reformat lets you resize your image sequence width and height using the incoming displayWindow to determine the scale factor. This also allows you to use plates of varying image resolution on a single recipe without running into issues when combining them.



NOTE: If no resize is needed, filtering is NOT applied (unlike in the [ImageTransform2D](#) node, which always applies filtering).

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.
resolution		
[resolution]	Dependent on Project Settings	The format to which you want to output the image sequence. The default setting resizes the image to the format indicated in the Project Settings.
width	globals.width	Sets a custom reformat width.  NOTE: You can only edit this field when the initial resolution is modified.
height	globals.height	Sets a custom reformat height.  NOTE: You can only edit this field when the initial resolution is modified.
preserveAspect	enabled	When enabled, Katana preserves the input image's aspect ratio.
allowOverscan	disabled	This allows the node to generate overscan (if possible). Overscan refers to image pixel data outside of the displayWindow and can be inspected using options in the Monitor.
preserveAspect: enabled		
center	enabled	When enabled, Katana pads the output image if any gaps remain after reformatting while preserving the original aspect ratio.
filtering		

Control (UI)	Default Value	Function
downFilter	Lanczos3	<p>The filter kernel to use for downsampling:</p> <ul style="list-style-type: none"> • Gaussian • Triangle • Box • Bell • BSpline • Sinc • Lanczos2 • Lanczos3 • Lanczos5 • Mitchell • Bilinear • Bicubic • Nearest
upFilter	Mitchell	<p>The filter kernel to use for upsampling:</p> <ul style="list-style-type: none"> • Gaussian • Triangle • Box • Bell • BSpline • Sinc • Lanczos2 • Lanczos3 • Lanczos5 • Mitchell • Bilinear • Bicubic • Nearest
highlightCompensation	enabled	<p>When enabled, Katana adaptively compresses pixel values prior to transform filtering and re-expands them afterwards. This helps to reduce the ringing in high-contrast areas that can be a problem in linear floating point images.</p>

Control (UI)	Default Value	Function
clampOutput	enabled	<p>Filtering can introduce negative values and send values above 1.0.</p> <p>When clampOutput is enabled, Katana clamps the RGB channels low at 0 and the alpha channel between 1 and 0 after the image is filtered. This is recommended for transforms on color/alpha images.</p> <p>When clampOutput is disabled, no clamping is done and values below 0 and above 1 are allowed. This is recommended for transforms applied to images that contain data which may (correctly) range more widely.</p>

ImageSaturation

This node is used to correct the input image's saturation (color intensity).

Control (UI)	Default Value	Function
[2D node controls]	N/A	<p>Set the controls for the stereo view.</p> <p>For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.</p>
ImageSaturation parameters continued		
saturation	1	Controls overall image saturation. Values less than 1 reduce saturation, and the other way around.
coefficients		
red	0.2126	Adjusts the image in conjunction with the saturation control, but only affects the red channel.
green	0.7152	Adjusts the image in conjunction with the saturation control, but only affects the green channel.
blue	0.0722	Adjusts the image in conjunction with the saturation control, but only affects the blue channel.
normalize	enabled	When enabled, saturation calculations are normalized.

ImageScreen

This node layers images together using the **Screen** compositing algorithm: **F+B-FB**. It's similar to ImageHypot, but clamps pixel values to 1.0. This is mostly useful for combining mattes.



NOTE: ImageScreen math assumes image values are between 0 and 1. If your highlights go above 1.0, ImageScreen may produce unexpected results.

Also see [ImageHypot](#).

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.
ImageScreen parameters continued		
operation	Screen	If you don't want layer the images together using the Screen compositing operation, select the operation to use instead. The following conventions apply to the below operation descriptions: <ul style="list-style-type: none"> • F refers to the fg input. • f refers to the fg input's alpha channel. • B refers to the bg input. • b refers to the bg input's alpha channel.

Control (UI)	Default Value	Function
operation (continued)		<p>The available operations (based on the Porter & Duff paper) are:</p> <ul style="list-style-type: none"> • Atop - $F + B(1-f)$. This shows the shape of the background, with the foreground covering the background where the images overlap. • Average - $(F+B)/2$. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • Difference - $\text{abs}(F-B)$. This shows how much the pixels differ and is useful for comparing two very similar images. • Divide - B/F. This divides the background values by the foreground values. • Exclusion - $F+B-2FB$. This is a more photographic form of Difference. • From - $F-B$. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see Minus. • Geometric - $2FB/(F+B)$. This is another way of averaging two images. Visually, it's close to Min. • Hypot - $\text{sqrt}(F^2+B^2)$. This resembles the Add and Screen operations. The result is not as bright as Add, but brighter than Screen. Hypot works with values above 1. It can be useful for adding reflections, as an alternative to Screen. • In - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • Matte - $Ff*B(1-f)$. This is a premultiplied Over. Use unpremultiplied images with this operation. • Max - $\text{max}(F,B)$. This takes the maximum values of both images. This is a good way to combine mattes and useful for bringing aspects like bright hair detail through.

Control (UI)	Default Value	Function
operation (continued)		<ul style="list-style-type: none"> • Min - $\min(F,B)$. This takes the minimum values of both images. • Minus - $B-F$. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see From. • Multiply - FB. This multiplies the values of the foreground by the values of the background. It can be used to composite darker values from the foreground with the background image - dark gray smoke shot against a white background, for example. • Out - $B(1-f)$. This only shows the areas of the background that do not overlap with the alpha of the foreground. This can be useful for combining mattes. • Over - $F+B(1-f)$. This layers the foreground over the background according to the alpha of the foreground. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • Plus - $F+B$. This produces the sum of the foreground and background. Note that the add algorithm may result in pixel values higher than 1.0. • Screen - $F+B-FB$. This is similar to Hypot, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • Under - $F(1-b)+B$. This is the reverse of the Over operation. It layers the background over the foreground according to the alpha of the background.
amount	1	Dissolves between the bg image at 0 and the full merge effect at 1.
displayWindow	Background	<p>The frame size to output in the event that the fg and bg inputs are different sizes:</p> <ul style="list-style-type: none"> • Background - output the frame size of the bg input. • Foreground - output the frame size of the fg input. • Union - output a combination of the bg and fg inputs' frame sizes. • Intersection - output an intersection of the bg and fg inputs' frame sizes. This restricts the output to the area where the two frame sizes overlap.

Control (UI)	Default Value	Function
clampAlpha	enabled	When enabled, the output alpha channel is clamped to the 0-1 range. Color channels (RGB) are not affected.

ImageText

This node generates text.

Control (UI)	Default Value	Function
bounds		
[resolution]	Dependent on Project Settings	Select the size of the text frame.
left	0	Lets you offset the text frame by this number of pixels from the left.
bottom	0	Lets you offset the text frame by this number of pixels from the bottom.
width	globals.width	<p>The width of the text frame in pixels.</p> <p>The default setting, globals.width, resizes the text frame to the width of the resolution indicated on the Project Settings tab.</p> <p> NOTE: You can only edit this field when the initial resolution is modified.</p>
height	globals.height	<p>The height of the text frame in pixels.</p> <p>The default setting, globals.height, resizes the text frame to the height of the resolution indicated on the Project Settings tab.</p> <p> NOTE: You can only edit this field when the initial resolution is modified.</p>

Control (UI)	Default Value	Function
text	none	<p>Enter the text you want to display here.</p> <p>You can optionally query scene graph values from an incoming 3D scene by:</p> <ul style="list-style-type: none"> • Connecting a 3D scene as input • Creating a text GroupAttribute at /root, containing the attributes you are interested in using as children. The AttributeScript node is usually most convenient for this manipulation. GroupAttribute structure is preserved, so for example you can place a whole materialAttr as a child of text, and then query the individual values below. • Reference attrs within the text node using the {attr:ATTRNAME} syntax.
fontSource	Builtin	<p>Select:</p> <ul style="list-style-type: none"> • Builtin - to use a built-in font (either Arial or Courier) for the text. • File - to use a font from an external font file for the text. Enter the file path to the font or use the file browser to browse to it. Fonts are loaded using FreeType2, which supports TrueType and OpenType fonts among others.
fontSource: Builtin		
font	Arial	Lets you select a font for the text when fontSource is set to Builtin : either Arial or Courier .
fontSource: File		
fontFile	none	Lets you select a font for the text when fontSource is set to File .
parameters continued		
size	18.0, size[0]	<p>Sets the pixel size of the font.</p> <p>Note that because of the way fonts are generated from control splines that vary in size, you rarely get a character that is exactly this size.</p> <p>No character ever renders larger than this size.</p>

Control (UI)	Default Value	Function
position	getDisplayWindow().width/2, getDisplayWindow().height/2	<p>The pixel position at which the justified text is placed.</p> <p>For example, if you set hjustify to Left and vjustify to Top, the left side of the baseline of the first line of text is placed at this location.</p> <p>If you set hjustify to Center and vjustify to Bottom, the baseline of the last line of text is centered on this position horizontally.</p>
hjustify	Center	<p>Sets how to align the text horizontally:</p> <ul style="list-style-type: none"> • Left - align the text along the left edge of the text frame, placing the left side of the text block at the location defined by position. This leaves the right edge of the text ragged. • Center - align the text from the center of the text frame, placing the center of the text block at the location defined by position. This leaves both edges of the text ragged. • Right - align the text along the right edge of the text frame, placing the right side of the text block at the location defined by position. This leaves the left edge of the text ragged.
vjustify	Center	<p>Sets how to align the text vertically:</p> <ul style="list-style-type: none"> • Top - align the text along the top edge of the text frame, placing the top baseline of the text block at the location defined by position. • Center - align the text from the center of the text frame, placing the center baseline of the text block at the location defined by position. • Bottom - align the text along the bottom of the text frame, placing the bottom baseline of the text block at the location defined by position. <p>The baseline is the imaginary line upon which most letters rest.</p>
lineSpace	0	<p>If you have several lines of text, this adjusts the spacing between each line. By using negative values, you can make the letters overlap.</p>

Control (UI)	Default Value	Function
wrapMode	None	Sets how to wrap long lines of text to fit inside the text frame: <ul style="list-style-type: none"> • None - long lines are not wrapped to fit inside the text frame. Some parts of the text may fall outside the frame and not be visible. • Word - long lines are split into several lines at word boundaries. • Exact - long lines are split into several lines at the closest point in the text that fits the text frame width, regardless of word boundaries.
wrapMode: Word or Exact		
wrapWidth	bounds.width	The width to use when calculating when to wrap the text.
color		
color	1.0, 1.0, 1.0, 1.0	The color (RGBA values) of the rendered text. You can also use the below RGB, HSL, or HSV controls to set the color of the text. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.
parameters continued		
antiAliasingGamma	2.2	Gamma applied after text rasterization but before applying color. This affects anti-aliasing appearance.
missingAttributes	Ignore	Specifies behavior when missing 3D scene graph attributes are encountered: <ul style="list-style-type: none"> • Ignore - the missing attributes are ignored. • Error - the missing attributes cause a render error.

ImageThreshold

The ImageThreshold node sets the value of the output pixels of an image, based on whether the value is above or below the value in the **level** parameter.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.
level		
level	0.5000, 0.5000, 0.5000, 0.5000	For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.
enableLow	Enabled	If enabled, any channel values below their corresponding level are set to the corresponding low value.
low		
low	0.0000, 0.0000, 0.0000, 0.0000	The output low value. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.
enableHigh	Enabled	If enabled, any channel values greater than or equal to their corresponding level are set to the corresponding high value.
high		
high	1.0000, 1.0000, 1.0000, 1.0000	The output high value. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.

ImageTransform2D

ImageTransform2D lets you not only translate elements, but also rotate, scale, and shear them.

Control (UI)	Default Value	Function
--------------	---------------	----------

[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.
transform		
order	trsx	Sets the operation order for translate (t), rotate (r), scale (s), and shear (x).
translate x, y	0, 0	Translates the image along the x and y axes.
rotate	0	Rotates the image around the pivot x y coordinates.
aspectRatio	1	Sets the pixel aspect ratio. This allows you to maintain aspect ratio when rotating anamorphic images.
scale x, y	1, 1	Scales the image width and height around the pivot x y coordinates.
shear x, y	0, 0	Shears the image around the pivot x y coordinates.
pivot x, y	(getDisplayWindow().x1 + getDisplayWindow().x0)/2, (getDisplayWindow().y1 + getDisplayWindow().y0)/2	Sets the center of rotation, scale, and shear on the x and y axes.
invert	disabled	When enabled, any transform you applied using the translate, rotate, scale, shear, or pivot controls is inverted.
filtering		

downFilter	Lanczos3	<p>The filter kernel to use for downsampling:</p> <ul style="list-style-type: none">• Gaussian• Triangle• Box• Bell• BSpline• Sinc• Lanczos2• Lanczos3• Lanczos5• Mitchell - remapped pixels receive some smoothing, plus blurring to hide pixelation.• Bilinear - gives good results, but can produce square artifacts at extreme zoom.• Bicubic - provides more rounded results, slightly blurrier but without the square artifacts.• Nearest - preserves edge detail, but gives quite "blocky" textures.
------------	----------	--

upFilter	Lanczos3	<p>The filter kernel to use for upsampling:</p> <ul style="list-style-type: none"> • Gaussian • Triangle • Box • Bell • BSpline • Sinc • Lanczos2 • Lanczos3 • Lanczos5 • Mitchell - remapped pixels receive some smoothing, plus blurring to hide pixelation. • Bilinear - gives good results, but can produce square artifacts at extreme zoom. • Bicubic - provides more rounded results, slightly blurrier but without the square artifacts. • Nearest - preserves edge detail, but gives quite "blocky" textures.
highlightCompensation	enabled	<p>When enabled, Katana adaptively compresses pixel values prior to transform filtering and re-expands them afterwards. This helps to reduce the ringing in high-contrast areas that can be a problem in linear floating point images (as we have in Katana).</p>

clampOutput	enabled	<p>Filtering can introduce negative values and send values above 1.0.</p> <p>When clampOutput is enabled, Katana clamps the RGB channels low at 0 and the alpha channel between 1 and 0 after the image is filtered. This is recommended for transforms on color/alpha images.</p> <p>When clampOutput is disabled, no clamping is done and values below 0 and above 1 are allowed. This is recommended for transforms applied to images that contain data which may (correctly) range more widely.</p>
onlyApplyMotion	disabled	<p>When enabled, Katana does not apply the node's full transform. Instead, it only applies the motion-vector component of the transform to the incoming image.</p> <p>If you apply onlyApplyMotion to the incoming image, and then transform the result by the node (with motion blur disabled), the results are similar (except for sampling differences).</p>
motionBlur		
enable	globals.compDefaults. motionBlur.enable	When enabled, you can add motion blur to the transform.
linearParamSubframeInterp	enabled	When enabled, use a fast sampling of the parameters using slerped end points for each sub-frame of motion blur. This is preferable in all cases except where lengthy blur strokes undergo subframe acceleration .

shutter	globals.compDefaults. motionBlur.shutter.i0, globals.compDefaults. motionBlur.shutter.i1	Sets the open and close time of the shutter when motion blurring, relative to the current frame. Changing the second number is the primary way to control the amount of motion blur applied. For example, a value of 0.5 corresponds to half a frame. Increasing the value produces more blur, and decreasing the value less.
numSamples	globals.compDefaults. motionBlur.numSamples	Sets the number of motion blur samples to compute and merge. Increase the value to produce more samples for higher quality, or decrease it to shorten the processing time. The higher the value, the smoother the result.

ImageUnder

This node layers images together using the **Under** compositing algorithm: $F(1-b)+B$. This is the reverse of the ImageOver operation. It layers the background over the foreground according to the matte of the background.

Also see [ImageOver](#) on page 101.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.
ImageUnder parameters continued		

Control (UI)	Default Value	Function
operation	Under	<p>If you don't want layer the images together using the Under compositing operation, select the operation to use instead.</p> <p>The following conventions apply to the below operation descriptions:</p> <ul style="list-style-type: none">• F refers to the fg input.• f refers to the fg input's alpha channel.• B refers to the bg input.• b refers to the bg input's alpha channel.

Control (UI)	Default Value	Function
operation (continued)		<p>The available operations (based on the Porter & Duff paper) are:</p> <ul style="list-style-type: none"> • Atop - $F + B(1 - f)$. This shows the shape of the background, with the foreground covering the background where the images overlap. • Average - $(F + B) / 2$. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • Difference - $\text{abs}(F - B)$. This shows how much the pixels differ and is useful for comparing two very similar images. • Divide - B / F. This divides the background values by the foreground values. • Exclusion - $F + B - 2FB$. This is a more photographic form of Difference. • IFrom - $F - B$. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see Minus. • Geometric - $2FB / (F + B)$. This is another way of averaging two images. Visually, it's close to Min. • Hypot - $\text{sqrt}(F^2 + B^2)$. This resembles the Add and Screen operations. The result is not as bright as Add, but brighter than Screen. Hypot works with values above 1. It can be useful for adding reflections, as an alternative to Screen. • In - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • Matte - $Ff * B(1 - f)$. This is a premultiplied Over. Use unpremultiplied images with this operation. • Max - $\text{max}(F, B)$. This takes the maximum values of both images. This is a good way to combine mattes and useful for bringing aspects like bright hair detail through.

Control (UI)	Default Value	Function
operation (continued)		<ul style="list-style-type: none"> • Min - $\min(F,B)$. This takes the minimum values of both images. • Minus - $B-F$. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see From. • Multiply - FB. This multiplies the values of the foreground by the values of the background. It can be used to composite darker values from the foreground with the background image - dark gray smoke shot against a white background, for example. • Out - $B(1-f)$. This only shows the areas of the background that do not overlap with the alpha of the foreground. This can be useful for combining mattes. • Over - $F+B(1-f)$. This layers the foreground over the background according to the alpha of the foreground. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • Plus - $F+B$. This produces the sum of the foreground and background. Note that the add algorithm may result in pixel values higher than 1.0. • Screen - $F+B-FB$. This is similar to Hypot, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • Under - $F(1-b)+B$. This is the reverse of the Over operation. It layers the background over the foreground according to the alpha of the background.
amount	1	Dissolves between the bg image at 0 and the full merge effect at 1.
displayWindow	Background	<p>The frame size to output in the event that the fg and bg inputs are different sizes:</p> <ul style="list-style-type: none"> • Background - output the frame size of the bg input. • Foreground - output the frame size of the fg input. • Union - output a combination of the bg and fg inputs' frame sizes. • Intersection - output an intersection of the bg and fg inputs' frame sizes. This restricts the output to the area where the two frame sizes overlap.

Control (UI)	Default Value	Function
clampAlpha	enabled	When enabled, the output alpha channel is clamped to the 0-1 range. Color channels (RGB) are not affected.

ImageUnpremultiply

This node divides the RGB channels by the **alphaChannel** when an image is connected to the **alpha** input. Otherwise, **a** is read from the **input** leaving the alpha channel unchanged.

Also see [ImagePremultiply](#) on page 108.

Control (UI)	Default Value	Function
[2D node controls]	N/A	Set the controls for the stereo view. For more information, refer to the Common 2D Nodes Widget Type in Common Parameter Widgets on page 13.
ImageUnpremultiply parameters continued		
alphaChannel	A	If a separate alpha input is provided, choose which of its channels to use to unpremultiply. If no separate alpha input is provided, the alpha from input is used and this control is disabled.

ImageWhite

Generates an image where every pixel is white, by default.

Control (UI)	Default Value	Function
bounds		

Control (UI)	Default Value	Function
[resolution]	Dependent on Project Settings	Select the size of the image.
left	0	Lets you offset the text frame by this number of pixels from the left.
bottom	0	Lets you offset the text frame by this number of pixels from the bottom.
width	globals.width	<p>The width of the text frame in pixels.</p> <p>The default setting, globals.width, resizes the text frame to the width of the resolution indicated on the Project Settings tab.</p> <p> NOTE: You can only edit this field when the initial resolution is modified.</p>
height	globals.height	<p>The height of the text frame in pixels.</p> <p>The default setting, globals.height, resizes the text frame to the height of the resolution indicated on the Project Settings tab.</p> <p> NOTE: You can only edit this field when the initial resolution is modified.</p>
infiniteExtent	Disabled	When enabled, the color extends beyond the bounds.
color		
color	1.0, 1.0, 1.0, 1.0	<p>The color (RGBA values) of every pixel in the image. You can also use the below RGB, HSL, or HSV controls to set the color.</p> <p>For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.</p>

ImageWrite

ImageWrite writes its incoming image to a file on disk. The ImageWrite node, unless defaults are overridden, converts images from Katana's linear colorspace to the colorspace named in the filename. Katana image processing nodes

work entirely in floating point, so images are also converted from floating point to the bit depth specified in the options for the format.

ImageWrite contains controls (**channels** and **outputFrame**) to force the output regardless of what may be coming into the node. However, if the output format cannot support the settings (for example, **.jpeg** doesn't support an alpha channel), the extra information is discarded.

The ImageWrite node supports the following file formats: **.exr**, **.rla**, **.cin**, **.png**, **.tif**, **.tiff**, **.jpg**, **.jpeg**, **.dpx**, and **.hist**.

Control (UI)	Default Value	Function
passName	comp	Sets the name used in the directories generated for this ImageWrite node. The passName should be unique for each ImageWrite node in the scene.
activeViews	main	Determines which views generate output images when hot-rendered or batch-rendered <ul style="list-style-type: none"> • Enable All - All views generate output images. • main - Only the main view generates output images. • left - Only the left view generates output images. • right - Only the right view generates output images.
singleFrame	disabled	When enabled, Katana only renders a single frame (for example, image_res.0001.exr) rather than an image sequence (for example, image_res.#.exr). You can specify the frame number using the frame control below. This also produces a render error when rendering on any frame other than the specified frame.
singleFrame: enabled		
frame	globals.inTime	Sets the frame to render when singleFrame is enabled.
inputs		

Control (UI)	Default Value	Function
[identifier]	none	<p>Defines short input identifiers. The identifier is included in the input/output input names and is used as a prefix for the output asset rep.</p> <p>%V is replaced with the view name.</p> <p>%v is replaced with the appropriate asset token.</p> <p> NOTE: Input identifiers have no effect on file names, only assets.</p>
mode	file	<p>Sets whether to write a file or define a dependency:</p> <ul style="list-style-type: none"> • file • dependency
file	none	<p>Sets the file path and name for the rendered image(s).</p> <p> NOTE: If mode is set to dependency, this control is hidden.</p>
inputs > mode: file > image		
proxyOnCue	enabled	
channels	Input	<p>Selects the channels to render:</p> <ul style="list-style-type: none"> • RGBA - Render the red, green, blue, and alpha channels. If any of the color channels are missing from the input, they are filled with 0 (pure black). If the alpha channel is missing, it is filled with 1 (pure white or fully opaque). • RGB - Render the red, green, and blue channels. If any of these channels are missing from the input, they are filled with 0 (pure black). • A - Only render the alpha channel. If this channel is missing from the input, it is filled with 1 (pure white or fully opaque). • Input - Render all channels that exist in the input. If the file format does not support the input channel configuration, required but missing channels are filled with 0 (color channels) or 1 (alpha).

Control (UI)	Default Value	Function
rawData	disabled	When enabled, Katana skips the automatic colorspace conversion (that is, the conversion from its native linear floating-point format to the output colorspace).
colorspace	linear	Katana converts from linear to this colorspace when writing the file to disk. The default value, auto , means Katana tries to determine the output colorspace from the file name.
colorConvert	enabled	<p>When enabled, Katana converts rendered image data from its native linear colorspace to the output colorspace specified in the file name. This is desirable in nearly every situation.</p> <p>A case where you would want to set this to disabled is if you know the data being rendered is in a colorspace other than linear (such as the re-projection of a log plate) and you want to name the output file log without a linear to log conversion.</p>
fileFormat	exr	<p>Sets the file format to output:</p> <ul style="list-style-type: none"> • auto - Katana tries to determine the output format from the file name. • exr • rla • cin • png • tif • tiff • jpg • jpeg • dpx • hist

inputs > mode: file > image > fileFormat: exr

Control (UI)	Default Value	Function
exrCompression	Wavelet	Defines the exr compression method to use. All methods are lossless (with the exception of Pixar 24 , which is lossless but restricts the pixels to 24-bit float). Wavelet is generally preferable as it offers ~2:1 compression even on grainy data.
exrBitDepth	16	Sets the floating point precision of the rendered exr file: <ul style="list-style-type: none"> • 16 - half float. This is recommended for all color passes. • 32 - full float. This is recommended for all ncf data arbitrary output variables (AOVs).
exrType	Tiled	Sets whether the exr file is written to support: <ul style="list-style-type: none"> • Tiled - random tile access. • Scanline - random scanline access.
comments	none	Optional field for any comments you want to store in the output file's comment metadata field. Currently, this is only supported on the exr file format.

inputs > mode: file > image > fileFormat: exr > exrType: Tiled

exrTileWidth	256	Sets the tile width to use when writing to tiled exr files.
exrTileHeight	256	Sets the tile height to use when writing to tiled exr files.
exrTileWorldAlign	disabled	When enabled (in conjunction with shrinkwrapping), the data rectangle is adjusted (top+left) so that the internal tile boundaries are aligned with world coordinates. This improves memory usage / performance for programs that process image tiles (such as Katana). <div data-bbox="805 1409 878 1470" data-label="Image"> </div> NOTE: This does not guarantee that tiles are aligned - it merely attempts to meet this condition.

inputs > mode: file > image > fileFormat: rla

Control (UI)	Default Value	Function
rlaBitDepth	auto	<p>Sets the bit depth of the rendered file. The default value, auto, means Katana tries to determine the bit depth from the colorspace. The other options are:</p> <ul style="list-style-type: none"> • 8-bit • 10-bit • 16-bit • 32-bit
inputs > mode: file > image > fileFormat: png		
pngBitDepth	auto	<p>Sets the bit depth of the rendered file. The default value, auto, means Katana tries to determine the bit depth from the colorspace. The other options are:</p> <ul style="list-style-type: none"> • 8-bit • 16-bit
inputs > mode: file > image > fileFormat: tif or tiff		
tifCompression	LZW	<p>The tiff compression method to use:</p> <ul style="list-style-type: none"> • None - No compression method is used. • LZW - The LZW compression method is used. This is lossless, so it is usually preferable to use it unless there is an issue with compatibility in the target reader.
tifBitDepth	auto	<p>The bit depth of the rendered file. The default value, auto, means Katana tries to determine the bit depth from the colorspace. The other options are:</p> <ul style="list-style-type: none"> • 8-bit • 16-bit • 32-bit
tifPredictor	None	<p>The predictor type to use when tifCompression is enabled:</p> <ul style="list-style-type: none"> • None - No prediction is used. • Horizontal - Horizontal prediction is used. This can result in smaller file sizes, but may present compatibility issues for some programs, such as MAXON's Cinema4D.

Control (UI)	Default Value	Function
inputs > mode: file > image > fileFormat: jpg or jpeg		
jpgQuality	100	The quality to use when generating the jpg file. Higher values generate larger file sizes, with 100 representing the best quality image and 0 representing the lowest.
inputs > mode: file > bounds		
displayWindow	input	<p>The frame size to write to the file:</p> <ul style="list-style-type: none"> • input - Use the frame size from the input. This crops off image data outside the frame or pads the frame with black if the image bounds do not fill the frame already. • manual - Crop the output to the specified frame size, padding with black if necessary.
dataWindow	shrinkwrap	<p>The image area to write to the file:</p> <ul style="list-style-type: none"> • shrinkwrap - Make sure the area is no larger than the frame size. This is the typical choice. If the format supports separate data and display windows (for example, the exr format does), the data window is clipped to the frame. • displayWindow - Write whatever area the input image data window covers (even if it exceeds the frame size). This only works with formats like exr that support a data window different from the display window. This is useful for writing out overscan images where the data extends beyond the frame. • manual - Crop the image area to the specified size, padding with black if necessary.



NOTE: Make sure other applications you are using support the selection you make. For example, if you select **displayWindow**, any other applications that read the output need to be able to handle separate data and display windows. You also need to use a format (like exr) that supports the concept, otherwise the data window is still clipped to the frame.

Control (UI)	Default Value	Function
inputs > mode: file > bounds > overscan		
left	0	Overscan specifies the number of pixels to pad the render request in each direction during a disk render (including batch renders). The display window is unchanged, but this expands the data window to include any extra input data that has been made available by the expanded render request.
bottom	0	Note that if dataWindow is set to shrinkwrap , the data window is still shrunk inward to encompass only the non-zero pixels in the image. Overscan simply enlarges the area that is initially rendered and under consideration for shrinkwrapping.
right	0	
top	0	Overscan has no effect when dataWindow is set to manual . You must include the desired overscan amount directly in the manual data window that you set.
inputs > mode: file > bounds > displayWindow: manual		
displayWindowResolution	512sq	Sets the resolution of the display window using the dropdown menu. This is a useful override if there exists different resolution names with the same resolution width and height.
width	512	Defines the display window resolution manually.
height	512	
postScripts > Add		
Add Post Script	N/A	Allows you to add post script commands.
farmSettings		

Control (UI)	Default Value	Function
setActiveFrameRange	disabled	<p>Sets how the active frame range for rendering is defined:</p> <p>When enabled, the activeFrameRange controls are displayed which define the active frame range for rendering.</p> <p>When disabled, Katana assumes that the active frame range is the same as the range between <code>globals.inTime</code> and <code>globals.outTime</code>.</p> <p>These settings affect outline file generation and guarantee that even if the node is called to render, it only writes files for frames in the active range.</p>
farmFileName	none	Defines the farm file name and path.
versionUp	Auto	<p>Sets whether the outputs of this node are versioned up when rendered on the queue:</p> <ul style="list-style-type: none"> • Auto - use the global setting specified in the outline file. • Yes - outputs version up. • No - outputs don't version up.
threadable	enabled	<p>Determines whether the queue is allowed to assign multiple cores to a frame of this render.</p> <p>When enabled, the queue may optionally thread the render.</p> <p>When disabled, the queue must use only one core.</p>
memory	none	Sets the memory requirement for the farm layer. Memory can be defined as m for megabyte or g for gigabyte. For example, 512m or 2g.
excludeFromFarmOutput Generation	disabled	<p>When enabled, this node does not appear in any generated farm file (however, the node is still renderable if called directly).</p> <p>Enabling this control hides the forceFarmOutputGeneration control.</p>

Control (UI)	Default Value	Function
forceFarmOutputGeneration	disabled	<p>When enabled, this node always appears in generated farm files (regardless of whether it has any valid outputs).</p> <p> NOTE: If excludeFromFarmOutputGeneration is also enabled, the node does not appear in the generated farm file (excludeFromFarmOutputGeneration overrides forceFarmOutputGeneration).</p>
farmSettings > setActiveFrameRange: enabled > activeFrameRange		
start	1	Sets the first frame in the active frame range when setActiveFrameRange is enabled.
end	1	Sets the last frame in the active frame range when setActiveFrameRange is enabled.

ImageZMerge

The ImageZMerge node applies a simple A over B composite. A and B are determined by examining the corresponding depth image and making A the image that has the lower (closer) depth value. A depth of **0** is assumed to be infinitely far away. An accumulated depth for the result is also available as the second output. Inputs are ordered as follows: Image1, Depth1, Image2, Depth2.

Control (UI)	Default Value	Function
depthChannel 	N/A	Specifies the channel (R, G, B, or A) that contains the depth values in the depth images.

OCIOCDLTransform

This node applies an ASC CDL grade. The calculation uses **output = (i * s + o)^p** where **i** is the input value, **s** is **slope**, **o** is **offset** and **p** is **power**.

Control (UI)	Default Value	Function
slope		
r	1	Adjusts the slope value in the red channel.
g	1	Adjusts the slope value in the green channel.
b	1	Adjusts the slope value in the blue channel.
offset		
r	0	Adjusts the offset value in the red channel.
g	0	Adjusts the offset value in the green channel.
b	0	Adjusts the offset value in the blue channel.
power		
r	1	Adjusts the power value in the red channel.
g	1	Adjusts the power value in the green channel.
b	1	Adjusts the power value in the blue channel.
saturation	1	Scales the image saturation using the 709 ASC primaries.
direction	forward	

OCIOColorSpace

This node covers the input colorspace to another specified colorspace.

Control (UI)	Default Value	Function
inColorSpace	Inf	Sets the input colorspace to convert from.
outColorSpace	Inf	Sets the output colorspace to convert to.
context		
key1	none	
value1	none	
key2	none	
value2	none	
key3	none	
value3	none	
key4	none	
value4	none	

OCIODisplay

This node is used to convert the input colorspace to display device suitable values.

Control (UI)	Default Value	Function
inputColorSpace	Inf	Sets the input colorspace to convert from.
display	sRGB	Sets the output display colorspace to convert to.
view	Film	<ul style="list-style-type: none"> • Film • Log • Raw
exposure		
rgb	0	Sets the exposure level for the r, g, and b channels together.
r	0	Sets the exposure level for the red channel.

Control (UI)	Default Value	Function
g	0	Sets the exposure level for the green channel.
b	0	Sets the exposure level for the blue channel.
context		
key1	none	
value1	none	
key2	none	
value2	none	
key3	none	
value3	none	
key4	none	
value4	none	

OCIOFileTransform

This node applies a LUT transform using a specified file.

Control (UI)	Default Value	Function
src	none	Specifies the src file path and name to use for the transform. This can be any file format that OpenColorIO supports: .3dl, .cc, .ccc, .csp, .cub, .cube, .lut (Houdini), .mga, .m3d, .spi1d, .spi3d, .spimtx, .vf
cccid	none	When src points to a .ccc file, specify the id to lookup. OpenColorIO::Contexts (envvars) are obeyed.
direction	forward	
interpolation	linear	
context		

Control (UI)	Default Value	Function
key1	none	
value1	none	
key2	none	
value2	none	
key3	none	
value3	none	
key4	none	
value4	none	

OCIOLogConvert

This node can be used to override the Kodak-recommended settings when making Cineon conversions in either direction (lin to log or log to lin). It's rare that you would want to override these settings, but if it becomes necessary you can use the OCIOLogConvert node. If you do, you should also check **rawData** in the ImageRead and ImageWrite node controls to skip the automatic conversion.

Control (UI)	Default Value	Function
operation	Lin To Log	Select the operation to perform: <ul style="list-style-type: none"> • Log To Lin - convert from a logarithmic (Cineon) format to Katana's linear colorspace. • Lin To Log - convert from Katana's linear colorspace to a logarithmic (Cineon) format.

OCIOLookTransform

This node provides a way to apply per-shot color correction as specified using the OpenColorIO look mechanism.

Control (UI)	Default Value	Function
look	none	Sets which looks to apply, referencing the OCIO configuration. You can chain looks together using a list delimited by commas or colons. To indicate direction, you can also use the + and - modifiers.
direction	forward	
inColorSpace	Inf	Sets the input colorspace to convert from.
outColorSpace	Inf	Sets the output colorspace to convert to.
ignoreErrors	disabled	When enabled, a missing OpenColorIO look forces this fail. When disabled, a missing OpenColorIO look is treated as a normal colorspace conversion.
context		
key1	none	
value1	none	
key2	none	
value2	none	
key3	none	
value3	none	
key4	none	
value4	none	

3D Nodes

The nodes in this chapter are 3D nodes that you can use within Katana. These are listed alphabetically, and each node includes a short description followed by a list of the node's parameters and their functions.

AimConstraint

Applies an aim constraint to an object in the scene graph.

Control (UI)	Default Value	Function
basePath	None	<p>Describes the scene graph location of the object to constrain. The basePath parameter options are available in either the scene graph widget or ▼ dropdown menu to the right of the parameter.</p> <p>For more information, refer to the Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.</p>
targetPath	None	<p>Describes the object(s) location to which the basePath object is constrained. The targetPath parameter options are available by clicking Add Locations or ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location and Locations Widget Types in Common Parameter Widgets on page 13.</p>
targetOrigin	Object	<p>Sets how the center of the target object is calculated:</p> <ul style="list-style-type: none">• Object - uses the local origin of the object as the target.• Bounding Box - uses the center of the object's bounding box as the target.• Face Center Average - uses the face center average of the object as the target.• Face Bounding Box - uses the face center average of the object's bounding box as the target.

Control (UI)	Default Value	Function
baseAimAxis	0.0, 0.0, -1.0	The axis of the base object that is pointed at the target. Adjusting these values changes the axis of the object that is aimed at the target.
baseUpAxis	0.0, 1.0, 0.0	The axis of the base object that is pointed upwards relative to the target. Adjusting these values changes the rotation of the base object, while keeping the aim constant.
targetUpAxis	0.0, 1.0, 0.0	The world space axis from the target object's position that defines the up direction for the base object Adjusting these values changes the axis of the base object's up axis.
allowMissingTargets	No	When set to Yes , silently ignore the constraint if its target is not in the scene graph. When set to No , produce an error on constraint resolution if the target is missing.
addToConstraintList	No	Adds base path to globals.constraintList at /root/world . This is only needed for cases in which one constraint depends on another constraint already being evaluated. The globals.constraintList is used to specify the order of evaluation of constraints.
setRelativeTargets	No	Stores target paths in the scene graph constraint definition as paths relative to the base path. Targets should still be specified as absolute paths in this node's parameters.

Alembic_In

The Alembic_In node enables you to import Alembic assets. Alembic is an open source scene information interchange framework which distills complex, animated scenes into non-procedural, application-independent, baked geometric results. It stores only the baked information and not how that information was obtained. You can export to Alembic from most popular 3D applications.

Alembic caches are retrieved with reference to time, not a particular frame; because of this, Katana needs to know what frame rate to use when querying the alembic file.

Control (UI)	Default Value	Function
name	/root/world/geo/asset	<p>Specifies the scene graph location where the Alembic asset is to be placed. The name parameter options are available in either the scene graph widget or  dropdown menu to the right of the parameter.</p> <p>For more information, refer to the Create Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.</p>
abcAsset	none	<p>Specifies where to retrieve the asset, an Alembic (.abc) file.</p> <p>For more information, refer to the Asset and File Path Widget Types in the Common Parameter Widgets on page 13.</p>
addForceExpand	Yes	<p>When set to a particular location in the scene graph, it forces expansion of the hierarchy under that location rather than stopping when a bounding box is reached.</p>
addBounds	Root	<p>Specifies where the overall bounds should be placed: the root location, its direct children, both, or none of these. Adding the bounds to the root has the disadvantage of producing the wrong bounds when the same root location is used by several Alembic_In nodes (the last one loaded overwrites the bounds added by the other ones).</p> <p>Adding the bounds to the direct children of the root location has the disadvantage of repeating the same overall bounds on each child, which means that these bounds can be potentially bigger than the real bounds of that child.</p>
fps	24	<p>Sets how many frames constitute a second inside the Alembic file.</p>

Control (UI)	Default Value	Function
addToCameraList	No	<p>For archives expected to contain cameras, this enables a light-weight traversal of the archive at /root/world so that the camera paths may be included in <code>globals.cameraList</code>.</p> <p>This parameter is disabled by default as it does work at a shallower point in the scene, independent of any downstream actions.</p> <p>In typical cases, the initial traversal is trivial and is then cached. Even so it's good practice to enable this only when necessary.</p>
timing		
mode	Current Frame	<p>Sets the timing mode to apply to the asset:</p> <ul style="list-style-type: none"> • Current Frame - uses the current frame to access the Alembic asset. • Hold Frame - uses the frame specified by holdTime to access the Alembic asset. • Clamp Range - forces Katana to only retrieve geometry from between the inTime and outTime frames. The frames specified by inTime and outTime are used for frames before and after the clamp range respectively.
timing > mode: Hold Frame		
holdTime	<code>globals.inTime</code> (an expression)	The frame to retrieve from the Alembic asset.
timing > mode: Clamp Range		
inTime	<code>globals.inTime</code> (an expression)	The start frame for retrieving geometry from the Alembic asset.
outTime	<code>globals.outTime</code> (an expression)	The end frame for retrieving geometry from the Alembic asset.
advanced		

Control (UI)	Default Value	Function
useOnlyShutterOpenCloseTimes	No	When set to Yes , it forces the Alembic cache to only use the time samples corresponding to shutter open and close times when ' maxTimeSamples ' is set to 2 in a RenderSettings node.

ArnoldGlobalSettings

The ArnoldGlobalSettings node is populated by the XML file located at `${KATANA_ROOT}/plugins/Resources/Arnold4.1/GenericAssign`.



NOTE: The parameters that are available for this node are dependent on which version of Arnold you are using. As such, only renderer-agnostic parameters are listed. For more information on some of the other parameters you may encounter, please refer to the documentation that ships with Arnold.

ArnoldObjectSettings

The ArnoldObjectSettings node is populated by the XML files located at `${KATANA_ROOT}/plugins/Resources/Arnold4.1/GenericAssign`. These parameter names and defaults can change between Arnold versions and, as such, they are provided for you to change.



NOTE: The parameters that are available for this node are dependent on which version of Arnold you are using. As such, only renderer-agnostic parameters are listed. For more information on some of the other parameters you may encounter, please refer to the documentation that ships with Arnold.

Control (UI)	Default Value	Function
CEL	none	<p>The scene graph locations are specified using the Collection Expression Language (CEL). The CEL parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>

ArnoldOutputChannelDefine

Builds the parameters used by Arnold during render.



NOTE: driverParameters change depending on the selected driver.



NOTE: The parameters that are available for this node are dependent on which version of Arnold you are using. As such, only renderer-agnostic parameters are listed. For more information on some of the other parameters you may encounter, please refer to the documentation that ships with Arnold.

Control (UI)	Default Value	Function
name	none	<p>The name used by the RenderOutputDefine node for these output channel settings.</p> <p>This parameter usually matches the channel name. By allowing a different name to be sent to the RenderOutputDefine node, different filter and file types can be used for the same output variable.</p>
type	RGBA	<p>Sets the output channel type:</p> <p>BYTE, INT, LONG, BOOL, FLOAT, DOUBLE, RGB, RGBA, ABSRGB, VECTOR, POINT, POINT2, STRING, POINTER, ARRAY, MATRIX, and ENUM</p>

ArnoldShadingNode

The ArnoldShadingNode allows you to select an Arnold-specific shader to build complex shading networks. The last shading node in the shading network needs to be connected to a [NetworkMaterial](#) node in order to be connected to the 3D node graph and assigned to objects in the scene.



NOTE: The parameters that are available for this node are dependent on which version of Arnold you are using. As such, only renderer-agnostic parameters are listed. For more information on some of the other parameters you may encounter, please refer to the documentation that ships with Arnold.

Control (UI)	Default Value	Function
name	ArnoldShadingNode	Determines the attribute identifier for this shader node beneath the 'material' attribute. This must be unique among all upstream nodes connected into a single NetworkMaterial node.
nodeType	none	Selects the available shader from the dropdown list. The parameters for each shader in the dropdown list are not included, as they are renderer-specific. Use the file browser or your studio's asset management browser to select the shader to use.
parameters	N/A	Once you've added a shader, the shader's parameters are populated under the Parameter group.
publicInterface		
namePrefix	none	Specifies the name's prefix for the exposed parameter.
pagePrefix	none	Allows you to organize the shading node's exposed parameters in groups in the NetworkMaterial node's Material Interface.
nameRegExFind	none	Finds and deletes the name specified in namePrefix field.
nameRegExReplace	none	When used with nameRegExFind , finds and replaces the name with the name specified by nameRegExReplace .
pageRegExFind	none	Finds and deletes the name specified in pagePrefix field.
pageRegExReplace	none	When used with pageRegExFind , finds and replaces the name with the name specified by pageRegExReplace .
ArnoldShadingNode parameters continued		
Force Refresh	N/A	Reloads the shader file's information.

AttributeCopy

Copies an attribute from location(s) in the copyFrom scene to location(s) in the input scene. Attribute data is shared between copies, so it's cheap to copy large attributes like geometry.point.P.

This node traverses the copyFrom scene at location fromRoot, and the input scene at location toRoot. From these locations on, it expects to find identical hierarchy and location names. For each location, if the copyFrom location has the attribute specified by fromAttr, for example, geometry.point.P, the attribute is copied to the location specified by toAttr, for example, geometry.point.Pref on the input location.

The optional **toCEL** parameter allows you to filter the evaluation of this node. Only locations in the destination scene that match **toCEL** are evaluated. If **toCEL** is empty, all locations in the destination scene are evaluated.

Control (UI)	Default Value	Function
fromRoot	/root/world	<p>Defines the copyFrom location. The fromRoot parameter options are available in either the scene graph widget or ▼ dropdown menu to the right of the parameter.</p> <p>For more information, refer to the Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.</p>
fromAttr	none	Defines the attribute that is copied.
toRoot	/root/world	<p>Defines the copyTo location. The toRoot parameter options are available in either the scene graph widget or ▼ dropdown menu to the right of the parameter.</p> <p>For more information, refer to the Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.</p>
toAttr	none	Defines the location where the attribute is copied to.
toCEL	none	<p>Allows you to filter the evaluation of this node. Only locations in the destination scene that match toCEL are evaluated. If toCEL is empty, all locations in the destination scene are evaluated.</p> <p>The scene graph locations are specified using the Collection Expression Language (CEL). The toCEL parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>

AttributeEditor

The AttributeEditor node is used to edit specific attributes of objects in the scene graph.

Control (UI)	Default Value	Function
exclusivity	none	<p>Exclusivity locks the interactive Viewer tab edits of a location to this node.</p> <p>The scene graph locations are specified using the Collection Expression Language (CEL). The exclusivity parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>
Existing Edits Table	N/A	Attributes dragged onto the Drop Attributes Here hotbox, located below the table, are placed here for editing.
Drop Attributes Here	N/A	Drag and drop attributes to this location to add them to the Existing Edits Table.
Existing Edits Table		
Existing Edits	none	The name of the attribute, grouped under its scene graph location, is displayed in this column.
Index	none	When the edited attribute is a number or string array, the Index column controls which value is displayed in the Value column. Click in the column to popup a slider which changes the index.
Value	none	The value of the edit is displayed in the Value column. Clicking on the value pops up a simple string or number field allowing you to change the value.

Control (UI)	Function
Existing Edits Table > [right-click menu]	

Control (UI)	Function
 Go To Location	Selects the scene graph location these edits affect.
 Disable Overrides	Disables the edit
 Enable Overrides	Enables the edit
Move Overrides To Selected Scenegraph Location...	Moves the override to the scene graph location currently selected.
Copy Overrides To Selected Scenegraph Location...	Copies the override to the scene graph location currently selected.
Delete Overrides	Deletes the override.
Drop Attributes Here	Middle-click and drag attributes from the Attributes tab to this hotspot to use that attribute.

AttributeFile_In

This node reads in an attribute file from a specified location and applies the attribute changes to the scene graph locations specified by the **CEL** statement.

Control (UI)	Default Value	Function
CEL	none	<p>The scene graph locations are specified using the Collection Expression Language (CEL). The CEL parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>
File Path	none	<p>Describes the filepath to an Attributes File.</p> <p>For more information, refer to the Asset and File Path Widget Types in the Common Parameter Widgets on page 13.</p>

Control (UI)	Default Value	Function
Custom File Parser	none	<p>Specifies the .so file with the Attributes File parser. Leave it empty to use the default one.</p> <p>For more information, refer to the Asset and File Path Widget Types in the Common Parameter Widgets on page 13.</p>
Attribute Group Name	attributeFile	<p>Specifies the name of the group attribute where the attributes from the file are stored. When empty, the attributes are stored directly under the location (without a group attribute).</p>
Apply When	immediate	<p>Determines when the script runs:</p> <ul style="list-style-type: none"> • immediate - the filter runs at the locations specified by the CEL statement as they are evaluated at this node's point in the graph. • deferred or during katana standard resolve - the script and its arguments are added as attributes under the scenegraphLocationModifiers group attribute. When deferred, they are run later by the implicit ScenegraphLocationModifierResolve filter added at render time. When during katana standard resolve, they are evaluated by a LookFileResolve node or by the first implicit resolver if no LookFileResolve node is present. (They may be tested either by enabling implicit scene graph resolvers in the Scene Graph tab or with a ScenegraphLocationModifierResolve node.) • during material resolve - the script and its arguments are added as attributes under the material. Scene GraphLocationModifiers group attribute. This is primarily intended for material scene graph locations. The material resolve process evaluates the script at the locations at which the material is assign or applied. This can be useful for building randomization or procedural control over shader parameters at the material level without having to apply materialOverride attributes at the geometry level.
Recursive Enable	No	<p>When applying in a non-immediate state, enabling this results in the script running at every location beneath the assigned locations. In general this is more efficient than using an equivalent recursive CEL statement.</p>

Control (UI)	Default Value	Function
timing		
mode	Current Frame	Sets the timing mode to apply to the asset: <ul style="list-style-type: none"> • Current Frame - uses the current frame to access the attribute file. • Hold Frame - uses the frame specified by holdTime to access the attribute file. • Clamp Range - forces Katana to only apply the attributes stored in the attributes file between the inTime and outTime frames. For before and after the clamp range, the frames specified by inTime and outTime respectively are used.
timing > mode: Hold Frame		
holdTime	globals.inTime (an expression)	Specifies which frame to use.
timing > mode: Clamp Range		
inTime	globals.inTime (an expression)	Specifies the start frame for the clamp range. It is also used for all frames before the inTime .
outTime	globals.outTime (an expression)	Specifies the end frame for the clamp range. It is also used for all frames after the outTime .

AttributeModifierDefine

This node makes use of the Attribute Modifier Plug-in API (AMP) to modify attributes in a scene graph. For more details on writing an AMP, please consult the documentation which is located through the **Help > Documentation** menu.

Control (UI)	Default Value	Function
CEL	none	<p>The scene graph locations are specified using the Collection Expression Language (CEL). The CEL parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>
applyWhen	immediate	<p>Determines when the script runs:</p> <ul style="list-style-type: none"> • immediate - the filter runs at the locations specified by the CEL statement as they are evaluated at this node's point in the graph. • deferred - the script details and its arguments are added as attributes under the attributeModifiers group attribute. They are run later by either an AttributeModifierResolve node or the implicit AttributeModifierResolve filter added at render time. They may be tested by enabling Scenegraph Implicit Resolvers located in the menu bar or in the Scene Graph tab.
applyWhen: deferred		
modifierNameMode	node name	Set whether to use the node name or a specified modifier name.
recursiveEnable	No	When applying in a non-immediate state, enabling this results in the Op running at every location beneath the assigned locations. In general this is more efficient than using an equivalent recursive CEL statement.
resolvelds	none	<p>When applyWhen is set to deferred, you may specify a space-delimited list of strings to indicate that this attribute script should only be resolved by attribute resolvers that contain at least one matching "resolveld." This is an advanced feature for greater control over order of evaluation.</p> <p>A useful resolvelds is <code>implicit_preprocess</code>, which runs at the first implicit resolver, before other implicit resolvers, such as <code>MaterialResolve</code> and <code>ConstraintResolve</code> are run.</p>
applyWhen: deferred: modifierNameMode: specified		

Control (UI)	Default Value	Function
modifierName	none	The name of the attribute group.
modifierType	none	Sets the attribute modifier plug-in to apply. The modifierType determines which arguments are available in the args dropdown.
args: AttributeFile		
filepath	none	Describes the attribute file asset location.
sofilepath	none	Describes the .so asset location.
groupAttr	none	Describes the group name where the attribute file modifications are placed.
args: GeoScaler		
scale	0.5	Sets the scaling factor for the GeoScaler AMP plug-in.
args: Messer		
displacement	2	Sets the maximum amount the geometry is displaced using the Messer AMP plug-in.

AttributeScript

This node runs Python scripts with the ability to query and modify attributes at locations specified by the CEL parameter. These functions are available:

- **DelAttr**(attributePath) - deletes the attribute at the given attribute path.
- **GetArguments**([asOpArgs=False, asFnAttribute=False]) - returns a list of arguments to the attribute script. If the **asOpArgs** parameter is set to **True** returns the new style Op arguments to the AttributeScript Op. If **False**, it returns the old style AMP arguments. The arguments include CEL expressions, the name of the node, and similar. If **asFnAttribute** is **True**, return the results as an **asFnAttribute** object, otherwise use the older **ScenegraphAttr** class.
- **GetAttr**(attributePath, [inherit=False, asAttr=False, atLocation=None, asNumpy=False, asFnAttribute=False]) - given an attribute (for instance, **visible** or **material.surfaceParams.Surf_Col**), **GetAttr** returns a list of values, with the following exceptions:
 - If an attribute does not exist at the given attribute, **None** is returned.
 - If **inherit** is set to **True**, the attribute is queried globally.

- If **asAttr** is **True**, a successful query returns an object of type `ScenegraphAttr`. `ScenegraphAttr` is the internal format of attributes in Katana. They are less immediate and convenient than raw lists of values but are useful in two ways:
 - They store multiple time samples of data.
 - They are more efficient than converting to a list if the individual values of the queried attribute do not need to be read or changed.

If the **GetAttr** option **atLocation** is given, the attribute is read from the named scene graph location, rather than the script's current location. If the **GetAttr** option **atLocation** isn't being used, the results from the `AttributeScript` running at the parent location is affected when using `inherit=True`.



NOTE: The **asNumpy** parameter is included for legacy reasons and should not be changed from **False**. If the option is set to **True**, Katana raises an exception.

- **GetChildNames**([atLocation=None]) - returns the child names from the current or specified scene graph location.
- **GetConfig**(name) - returns a named configuration value.
- **GetFrameTime**() - returns the current frame time as a float.
- **GetFullName**() - returns the full path of the current scene graph location. This is useful as a random seed.
- **GetName**() - returns the base name of the current scene graph location.
- **GetNumSamples**() - returns the scene-wide recommendation for the number of samples to generate.
- **GetResolution**([name=None, asDict=False]) - returns the dimensions of a named resolution, or the current resolution if the name is not specified. If the **asDict** parameter is **False**, the value returned is a 2-tuple of the width and height values. If the **asDict** parameter is set to **True**, then it returns a dictionary containing more information about the resolution than simply the dimensions.
- **GetShutterOpen**() - returns the shutter open value as defined (and only as defined) by `renderSettings.shutterOpen`.
- **GetShutterClose**() - returns the shutter close value as defined (and only as defined) by `renderSettings.shutterClose`.
- **GetType**() - returns the type of the current scene graph location.
- **GetXform**([worldSpace=False, atLocation=None, asAttr=False, atSamples=None, asFnAttribute=False]) - returns the local or worldspace transformation as a 16-element list of float values representing a 4x4 transform matrix. If the **atSamples** parameter is specified it should be a sequence of time samples at which to return the xform.
- **Util.AssetResolve**(assetID) - given an asset ID, **Util.AssetResolve** returns a file path.
- **SetAttr**(attributePath, [attrValue]) - given an attribute path, **SetAttr** overrides an attribute with the value provided. If **attrValue** is a list of values, the length and type must match any pre-existing attribute at that location or an error is raised. This is to prevent accidental changes to the definition of an attribute. If **attrValue** is a `ScenegraphAttr`, it is used without regard to any existing attribute. If **attrValue** is not provided, the existing attribute at that location is deleted.
- **User Parameters** - in addition, user parameters of the `AttributeScript` node itself are available in a module called **user**. Each parameter (array or otherwise) is converted to a list of values. For example, a number parameter is available as a list containing one float value.

Control (UI)	Default Value	Function
CEL	none	<p>The scene graph locations are specified using the Collection Expression Language (CEL). The toCEL parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>
applyWhen	immediate	<p>Determines when the script is run:</p> <ul style="list-style-type: none"> • immediate - the script runs at the locations specified by the CEL statement as they are evaluated at this node's point in the graph. • during attribute modifier plugin resolve - the script and its arguments are added as attributes under the Scene GraphLocationModifiers group attribute and run later by the implicit ScenegraphLocationModifierResolve filter added at render time. • during katana look file resolve - the script and its arguments are added as attributes under the Scene GraphLocationModifiers group attribute and are evaluated by a LookFileResolve node or by the first implicit resolver if no LookFileResolve node is present. <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <p> NOTE: Plug-in resolve and look file resolve may be tested either by enabling Scenegraph Implicit Resolvers in the Scene Graph tab or with an AttributeModifierResolve or LookFileResolve node.</p> </div> <ul style="list-style-type: none"> • during material resolve - the script and its arguments are added as attributes under the material.ops group attribute. This is primarily intended for material scene graph locations. The material resolve process evaluates the script at the locations at which the material is assign or applied. This can be useful for building randomization or procedural control over shader parameters at the material level without having to apply materialOverride attributes at the geometry level.

applyWhen: during attribute modifier plugin resolve, during katana look file resolve, or during material resolve.

Control (UI)	Default Value	Function
recursiveEnable	No	When applying in a non-immediate state, enabling this results in the attribute running at every location beneath the assigned locations. In general this is more efficient than using an equivalent recursive CEL statement.
initializationScript	No	Enables an optional setup script to be run before the first location at which the attribute script evaluates. The user module is available to store persistent data and retrieve parameter values
initializationScript: Yes		
setup	none	Contains a Python script to run once before the script parameter is run for each location. It is used to run scripts that are computationally or resource heavy and have the same result for each location, for instance file reads. To store persistent data from this script to the script parameter script, use the user module.
script	none	Contains the Python scripts to run.

AttributeSet

This node is used for creating, modifying, or deleting scene graph attribute locations.

Control (UI)	Default Value	Function
mode	paths	Specifies the location to be overridden: <ul style="list-style-type: none"> • paths • CEL
mode: paths		

Control (UI)	Default Value	Function
paths	none	<p>Sets the paths of the attribute. For example, /root/world/geo. The paths parameter options are available by clicking Add Locations or ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location and Locations Widget Types in Common Parameter Widgets on page 13.</p>
mode: CEL		
celSelection	none	<p>Sets the attribute location to be overridden.</p> <p>The scene graph locations are specified using the Collection Expression Language (CEL). The celSelection parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>
AttributeSet parameters continued		
action	Create/Override	<p>Describes the action to apply to the scene graph attribute:</p> <ul style="list-style-type: none"> • Create/Override • Delete • Force Default
attributeName	none	The attribute name to override.
Drop Attributes Here	N/A	Drag and drop attributes to this location to add them to the attributeName field.
action: Create/Override		

Control (UI)	Default Value	Function
attributeType	double	The type of the attribute that is overridden: <ul style="list-style-type: none"> • integer • double • float • string • group
groupInherit	Yes	Decides whether or not implicitly-created groups are inherited lower in the scene graph hierarchy. For instance, creating foo.bar implicitly creates the group foo. This group is either inherited or not, depending on this parameter.
action: Create/Override: attributeType: integer, double, or float		
numberValue	0.0	Sets the override value.
action: Create/Override: attributeType: string		
stringValue	none	Sets the override value.
action: Create/Override: attributeType: group		
applyAtLeaves	No	
groupValue	N/A	Collects a number of attributes into one group.
Drop Attributes Here	N/A	Middle-click and drag attributes from the Attributes tab to this hotspot to use that attribute.
action: Force Default		
groupInherit	Yes	Decides whether or not implicitly-created groups are inherited lower in the scene graph hierarchy. For instance, creating foo.bar implicitly creates the group foo. This group is either inherited or not, depending on this parameter.

AttributeUpgrade

Upgrades scene data with Katana 1.x attribute conventions for use with Katana 2.0. The conventions that are supported for conversion are attributeModifiers and scenegraphLocationModifiers (including AttributeScripts).

For example, if - in a 1.x version of Katana - you created an AttributeScript node in **during attribute modifier resolve** mode instead of **immediate**, it created attributes at your matching CEL locations, like so:

```
.
|-- name = 'root'
|-- scenegraphLocationModifiers
|   |-- AttributeScript
|       |-- type = 'OP_Python'
|       |-- args
|           |-- expr = 'print "hi"'
|           |-- historyName = 'AttributeScript'
|           |-- resolveIds = 'test'
|-- type = 'group'
```

AttributeUpgrade is useful when loading saved or serialized attributes that are in this form into Katana post-2.0 ops. Simply loading a 1.x file that contains an AttributeScript node in 2.0 prompts Katana to run through standard upgrade mechanisms, which means this AttributeUpgrade node isn't necessary in that case. This node is only needed when you're loading old attributes directly. Using an AttributeUpgrade node would, therefore, transform the previous example into:

```
.
|-- ops
|   |-- AttributeScript
|       |-- opType = 'AttributeScript'
|       |-- opArgs
|           |-- script = 'print "hi"'
|           |-- nodeName = 'AttributeScript'
|           |-- resolveIds = 'test'
|-- type = 'group'
```



NOTE: The AttributeUpgrade Op is run automatically during a LookFileResolve, which means that most of the time, you shouldn't need to use this node explicitly.

Control (UI)	Default Value	Function
celSelection	N/A	<p>Sets the attribute location to be upgraded.</p> <p>The scene graph locations are specified using the Collection Expression Language (CEL). The celSelection parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>

Backdrop

The Backdrop node allows you to improve readability and navigation of your recipes. For example, you can set a colored background around a group of nodes, or use the Backdrop node to add comments in a scene. For more details on the Backdrop node, please consult the *Working with Nodes* chapter in the *Katana User Guide*.

BillboardConstraint

Applies an aim constraint to an object in a scene. To get the best possible aim, the constraint only rotates around the axis defined by **baseRotateAxis**.

Control (UI)	Default Value	Function
basePath	None	<p>Describes the scene graph location of the object to constrain. The basePath parameter options are available in either the scene graph widget or ▼ dropdown menu to the right of the parameter.</p> <p>For more information, refer to the Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.</p>
targetPath	none	<p>Describes the object(s) location to which the basePath object is constrained. The targetPath parameter options are available by clicking Add Locations or ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location and Locations Widget Types in Common Parameter Widgets on page 13.</p>
targetOrigin	object	<p>Sets how the center of the target object is calculated:</p> <ul style="list-style-type: none"> • object - uses the local origin of the object as the target. • boundingBox - uses the center of the object's bounding box as the target.
baseAimAxis	0.0, 0.0, -1.0	<p>The axis of the base object that is pointed at the target.</p> <p>Adjusting these values changes the axis of the object that is aimed at the target.</p>

Control (UI)	Default Value	Function
baseRotateAxis	0.0, 1.0, 0.0	<p>The axis of the base object that is rotated to maintain orientation to the target.</p> <p>Adjusting these values changes the rotation of the base object, while keeping the aim constant.</p>
allowMissingTargets	No	<p>When set to Yes, silently ignore the constraint if its target is not in the scene graph.</p> <p>When set to No, produce an error on constraint resolution if the target is missing.</p>
addToConstraintList	No	<p>Adds base path to globals.constraintList at /root/world.</p> <p>This is only needed for cases in which one constraint depends on another constraint already being evaluated. The globals.constraintList is used to specify the order of evaluation of constraints.</p>
setRelativeTargets	No	<p>Stores target paths in the scene graph constraint definition as paths relative to the base path.</p> <p>Targets should still be specified as absolute paths in this node's parameters.</p>

BoundsAdjust

Allows you to adjust the bounding box of a geometry location.

Control (UI)	Default Value	Function
targetPath	none	<p>Describes the location of the geometry who's bounding box is being adjusted. The targetPath parameter options are available in either a scene graph widget or ▼ dropdown menu to the right of the parameter.</p> <p>For more information, refer to the Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.</p>

Control (UI)	Default Value	Function
action	Strip Bounds	Sets the type of bounding adjustment to apply: <ul style="list-style-type: none"> • Strip Bounds - removes the bounding box from the geometry entirely. • Pad Bounds (%) - allows you to pad the bounding box by a user-defined percentage. • Pad Bounds (local) - allows you to pad the bounding box by an amount specified in units.
action: Strip Bounds		
stripAncestors	No	Specify whether or not to strip the bound attributes from the ancestor locations.
action: Pad Bounds (%) > padAmount		
percentage	0	Specify percentage to add to the original bounding box size. This is a keyable attribute.
adjustAncestors	No	Specify whether or not to expand the bound attributes of any ancestor location to account for the newly adjusted child location bounds.
when	immediate	Sets when the specified adjustment is applied: <ul style="list-style-type: none"> • immediate - pad the bounds immediately. • deferred - pad the bounds only at render time (more efficient). Padding is calculated in the PRMan plug-in, so the result is not visible in Katana even with implicit resolvers on.
action: Pad Bounds (Local) > padAmount		
localSpace	0	Specifies the number of local space units to pad the bounding box.
adjustAncestors	No	Specify whether or not to expand the bound attributes of any ancestor location to account for the newly adjusted child location bounds.
when	immediate	Sets when the specified adjustment is applied: <ul style="list-style-type: none"> • immediate - pad the bounds immediately. • deferred - pad the bounds only at render time (more efficient). Padding is calculated in the PRMan plug-in, so the result is not visible in Katana even with implicit resolvers on.

CameraClippingPlaneEdit

Edits the camera near and far clipping attributes for a single camera.



NOTE: The default values change when initially connected to a camera.

Control (UI)	Default Value	Function
cameraLocation	/root/world/cam/camera	Describes the location of the camera. For more information, refer to the Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.
with location specified		
near	1 (see note)	Sets the near clipping plane for the specified camera.
far	10000 (see note)	Sets the far clipping plane for the specified camera.
claimExclusivity	No	When set to No , the camera is not controlled by another node in the scene graph and is effectively locked.

CameraCreate

The CameraCreate node is used to create a scene graph containing a camera. CameraCreate does not load the camera from any file or asset but instead builds an entirely new camera from the parameters you specify on this node.



NOTE: LightCreate and CameraCreate are identical, except for the type of scene graph locations they create, and the population of the lightList vs. cameraList.



TIP: To lock a camera's position after it's created, set **claimExclusivity** on the CameraCreate to **No**.

Control (UI)	Default Value	Function
name	/root/world/cam/camera	<p>This is the scene graph location where the camera is created. For example, the default value of /root/world/cam/camera creates a camera at the location /root/world/cam/camera. The name parameter options are available in either the scene graph widget or ▼ dropdown menu to the right of the parameter.</p> <p>For more information, refer to the Create Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.</p>
projection	perspective	<p>Toggles the projection type of the camera:</p> <ul style="list-style-type: none"> • perspective • orthographic
fov	70	Controls the field of view angle in degrees.
near	0.1	Sets the near clipping plane distance.
far	100000	Sets the far clipping plane distance.
screenWindow		
left	-1	This set of four number parameters controls the screen window placement on the imaging plane. They are, in order, left, right, bottom and top bounds of the screen window.
right	1	
bottom	-1	
top	1	
centerOfInterest	20	Offsets the center of interest of the camera.
orthographicWidth	30	Sets the orthographic projection width.
includeInCameraList	Yes	When enabled, the camera is visible in the camera list on the /root/world location, under the Scene Graph tab.
transform		

Control (UI)	Default Value	Function
transform	N/A	Transforms the camera according to the SRT or matrix controls. For more information, refer to the Transform Controls Widget Type in the Common Parameter Widgets on page 13.
transform > Tools ▾	N/A	Adjusts the camera to match selected scene graph selection options in the dropdown menu. For more information, refer to the Transform Tools Widget Type in the Common Parameter Widgets on page 13.
CameraCreate parameters continued		
makeInteractive	Yes	When set to Yes , you can drag objects in the Viewer and Katana retains the information from the Viewer.

CameraImagePlaneCreate

Creates attributes on a camera that describe an image plane. In the Viewer, the camera displays the image plane.



NOTE: If you are not seeing image planes in the Viewer, make sure to turn on the image plane button in the Viewer tab. This button globally turns on/off all image planes in the Viewer.

Control (UI)	Default Value	Function
cameraLocation	none	This is the scene graph location where the target camera resides. For example, the default value of /root/world/cam/camera references a camera at the location /root/world/cam/camera . For more information, refer to the Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.
name	plane	Sets the name of the image plane created.

Control (UI)	Default Value	Function
imagePath	none	<p>Sets the filepath of the image or sequence to display in the image plane. Supported file formats include .cin, .dpx, .rla, .iff, .tif, .jpg, .tga, .rgb, and .tga. Floating point data (.exr, .tif, .zfile) is not currently supported.</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;">  <p>NOTE: Image sequences must contain a padded frame number.</p> </div> <p>For more information, refer to the Asset and File Path Widget Types in the Common Parameter Widgets on page 13.</p>
frame	1	Sets the frame of the image sequence to use.
depth	10000	Sets the distance from the camera to image plane.
alpha	1	Sets the image plane's alpha value.
displayOnlyIfCurrent	No	<p>When set to Yes, this image plane is only displayed when looking through the camera it is attached to.</p> <p>When set to No, you can see the image plane in all views.</p>
displayMode	RGBA	<p>Sets the image plane display mode:</p> <ul style="list-style-type: none"> • None • Outline • RGB • RGBA

Control (UI)	Default Value	Function
fit	Best	<p>Controls how the image file fits into the image plane if there is a mismatch between aspect ratios:</p> <ul style="list-style-type: none"> • Fill - the image is scaled as required to fill the plane, without being squashed or stretched. Any excess is cropped. • Best - the image is scaled as required to display it entirely within the plane, without being squashed, stretched or cropped. • Horizontal - the image is scaled as required so that its aspect ratio is maintained within the horizontal bounds of the plane. Any excess at the top or bottom is cropped. • Vertical - the image is scaled as required so that its aspect ratio is maintained within the vertical bounds of the plane. Any excess at the left or right is cropped. • To Size - the image is stretched or squashed to fit with the plane both horizontally and vertically. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  <p>NOTE: To control the image's SRT values directly, enable the manualPlacementSettings checkbox.</p> </div>
crop		
left	0	Sets the amount of manual crop to apply to the edges of the image plane.
bottom	0	
right	1	
top	1	
CameraImagePlaneCreate parameters continued		
manualPlacementSettings	No	When set to Yes , you can adjust the image manually using its SRT values.
manualPlacementSettings: Yes > size		
x	1	
y	1	

Control (UI)	Default Value	Function
manualPlacementSettings: Yes > offset		
x	0	
y	0	
manualPlacementSettings: Yes		
rotate	0	

CameraScreenWindowConstraint

This node is used to orient, scale, and position the base scene graph location so that it sits at a specified distance from the camera and fits the camera screen window exactly.

Control (UI)	Default Value	Function
basePath	none	<p>Describes the scene graph location of the object to constrain. This should be plane geometry. The basePath parameter options are available in either the scene graph widget or ▼ dropdown menu to the right of the parameter.</p> <p>For more information, refer to the Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.</p>
targetPath	none	<p>Describes the camera location to which the basePath object is constrained. The targetPath parameter options are available in either a scene graph widget or ▼ dropdown menu to the right of the parameter.</p> <p>For more information, refer to the Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.</p>
distance	1	Sets distance from the camera at which the base scene graph location is constrained.
planeType	XY	<p>The type of plane that is constrained:</p> <ul style="list-style-type: none"> • XY • XZ

Control (UI)	Default Value	Function
addToConstraintList	No	<p>Adds base path to globals.constraintList at /root/world.</p> <p>This is only needed for cases in which one constraint depends on another constraint already being evaluated. The globals.constraintList is used to specify the order of evaluation of constraints.</p>
setRelativeTargets	No	<p>Stores target paths in the scene graph constraint definition as paths relative to the base path.</p> <p>Targets should still be specified as absolute paths in this node's parameters.</p>

ClippingConstraint

This node adjusts the camera's near and far clipping planes to fit just in front of and behind the target (along the axis from the camera). You can view the results of the ClippingConstraint node by turning on Scenegraph Implicit Resolvers at the top.

Control (UI)	Default Value	Function
basePath	none	<p>Describes the scene graph location of the object to constrain. The basePath parameter options are available in either the scene graph widget or ▼ dropdown menu to the right of the parameter.</p> <p>For more information, refer to the Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.</p>
targetPath	none	<p>Describes the object(s) location to which the basePath object is constrained. The targetPath parameter options are available by clicking Add Locations or ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location and Locations Widget Types in Common Parameter Widgets on page 13.</p>
pad		

Control (UI)	Default Value	Function
near	0	Sets the amount of space to leave between the nearest extent of the target and the camera's near clipping plane.
far	0	Sets the amount of space to leave between the farthest extent of the target and the camera's far clipping plane.
ClippingConstraint parameters continued		
respectMotionBlur	Yes	When set to Yes , constraints are adjusted to allow for the target's motion within the time the shutter is open.
allowMissingTargets	No	When set to Yes , silently ignore the constraint if its target is not in the scene graph. When set to No , produce an error on constraint resolution if the target is missing.
addToConstraintList	No	Adds base path to globals.constraintList at /root/world . This is only needed for cases in which one constraint depends on another constraint already being evaluated. The globals.constraintList is used to specify the order of evaluation of constraints.
setRelativeTargets	No	Stores target paths in the scene graph constraint definition as paths relative to the base path. Targets should still be specified as absolute paths in this node's parameters.

CollectionCreate

Collections are used to store a CEL statement. They are stored as attributes at **/root**. As they are simply attributes within the scene graph, Collections can be included within Katana Look Files.

Control (UI)	Default Value	Function
location	none	<p>The scene graph location to where (x, y) coordinates are written. The location parameter options are available in either the scene graph widget or ▼ dropdown menu to the right of the parameter.</p> <p>For more information, refer to the Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.</p>
name	none	Sets the name of the collection.
CEL	none	<p>Specifies scene graph locations to store as part of this collection.</p> <p>The scene graph locations are specified using the Collection Expression Language (CEL). The CEL parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>
stripDisabledBlocks	No	

ConstraintCache

The ConstraintCache node caches the transform attributes of specified locations to disk for later use.

Control (UI)	Default Value	Function
Fill Cache	n/a	Fills the cache with the transform matrix for a given location.
startFrame	1	Specifies from which frame to start caching.
endFrame	100	Specifies from which frame to stop caching.

Control (UI)	Default Value	Function
locations	none	<p>The scene graph location to where the cache is written. The locations parameter options are available by clicking Add Locations or ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location and Locations Widget Types in Common Parameter Widgets on page 13.</p>

ConstraintListEdit

Adds locations to the **globals.constraintList** attribute at **/root/world**. This is useful for including constraints loaded from a deferred source, such as a look file. Only constraints on locations listed in the **globals.constraintList** are resolved at render time.

Control (UI)	Default Value	Function
locations	none	<p>Sets the scene graph location(s) to add to the constraint list. The locations parameter options are available by clicking Add Locations.</p> <p>For more information, refer to the Scene Graph Location and Locations Widget Types in Common Parameter Widgets on page 13.</p>
mode	add	<p>Sets the constraint mode:</p> <ul style="list-style-type: none"> • add - adds locations to the globals.constraintList at /root/world. • remove - removes locations from the globals.constraintList at /root/world.

ConstraintResolve

This node resolves all constraints stored on the locations referenced in **globals.constraintList** at **/root/world**.

CoordinateSystemDefine

Creates a named coordinate system accessed by PRMan shaders. The list of all global named coordinate systems can be found in `/root/world, globals.coordinateSystems`.

Control (UI)	Default Value	Function
scope	global	Specifies how the coordinate system is defined. The options are: <ul style="list-style-type: none"> • global • relative scope
coordinateSystemName	none	Specifies the unique name of the coordinate system to create.
scope: global		
referenceLocation	none	Specifies the scene graph location whose global transform defines the coordinate system. For more information, refer to the Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.
scope: relative scoped		
baseLocation	none	Specifies the scene graph location whose scoped transform defines the base coordinate system. For more information, refer to the Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.
relativeLocation	none	Specifies the scene graph location whose scoped transform defines the relative coordinate system. For more information, refer to the Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.

DollyConstraint

This node translates the camera along its look at (or local Z) axis, moving it towards or away from the target. DollyConstraint ensures that the target fits exactly in the camera's screen window and is useful for turntable setup.

See also [FOVConstraint](#) on page 188.

Control (UI)	Default Value	Function
basePath	none	<p>Describes the scene graph location of the camera or light to dolly. The basePath parameter options are available in either the scene graph widget or ▼ dropdown menu to the right of the parameter.</p> <p>For more information, refer to the Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.</p>
targetPath	none	<p>Sets the location of the object(s) to fit within the field of view. The targetPath parameter options are available by clicking Add Locations or ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location and Locations Widget Types in Common Parameter Widgets on page 13.</p>
targetBounds	box	<p>The type of bounds to use for the target object(s):</p> <ul style="list-style-type: none"> • box - the camera is constrained to fit the bounding box of the target object(s). • sphere - the camera is constrained to fit a sphere that encloses the bounding box of the target objects(s).
angleOffset	0	Sets the angle to add to the FOV in the dolly calculation.
allowMissingTargets	No	<p>When set to Yes, silently ignore the constraint if its target is not in the Scene Graph.</p> <p>When set to No, produce an error on constraint resolution if the target is missing.</p>

Control (UI)	Default Value	Function
addToConstraintList	No	<p>Adds base path to globals.constraintList at /root/world.</p> <p>This is only needed for cases in which one constraint depends on another constraint already being evaluated. The globals.constraintList is used to specify the order of evaluation of constraints.</p>
setRelativeTargets	No	<p>Stores target paths in the scene graph constraint definition as paths relative to the base path.</p> <p>Targets should still be specified as absolute paths in this node's parameters.</p>

FaceSetCreate

This node creates a set (or group) of faces in an existing mesh. This is useful in order to more easily re-select them later when making shader, attribute, and visibility assignments to a sub-set of faces on a single mesh.

Control (UI)	Default Value	Function
meshLocation	none	<p>Describes the location of the mesh for which the set is created.</p> <p>For more information, refer to the Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.</p>
faceSetName	faceset	Defines the name of the new set.
invertSelection	Disabled	When enabled, the faces that are not mentioned in the selection are used.
selection	none	Stores the list of faces as an array. Alternatively, stores the list of faces that are currently selected in the Viewer in this parameter by selecting ▼ > Adopt Faces From Viewer .

FOVConstraint

This node constrains the field of view of a camera to fit the target geometry. FOVConstraint closes or opens the field of view of a camera from all sides while the center of the frame remains the same. If an object is located at the edge of a light's view, the FOVConstraint should be combined with an AimConstraint to tighten the view right on the object.

See also [DollyConstraint](#) on page 186.

Control (UI)	Default Value	Function
basePath	none	<p>Describes the scene graph location of the object to constrain. The basePath parameter options are available in either the scene graph widget or ▼ dropdown menu to the right of the parameter.</p> <p>For more information, refer to the Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.</p>
targetPath	none	<p>Describes the object(s) location to which the basePath object is constrained. The targetPath parameter options are available by clicking Add Locations or ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location and Locations Widget Types in Common Parameter Widgets on page 13.</p>
targetBounds	box	<p>Sets the type of bounds to use for the target object(s).</p> <ul style="list-style-type: none"> • box - the field of view is constrained to fit the bounding box of the target object(s). This can be very useful for shadow maps, as it produces a tight fitting bounding box. • sphere - the field of view is constrained to fit a sphere that encloses the bounding box of the target object(s). This can be very useful for turntables when you don't want the field of view to change as the object rotates.
angleOffset	0	Sets the angle added to the FOV during calculation.

Control (UI)	Default Value	Function
allowMissingTargets	No	<p>When set to Yes, silently ignore the constraint if its target is not in the scene graph.</p> <p>When set to No, produce an error on constraint resolution if the target is missing.</p>
addToConstraintList	No	<p>Adds base path to globals.constraintList at /root/world.</p> <p>This is only needed for cases in which one constraint depends on another constraint already being evaluated. The globals.constraintList is used to specify the order of evaluation of constraints.</p>
setRelativeTargets	No	<p>Stores target paths in the scene graph constraint definition as paths relative to the base path.</p> <p>Targets should still be specified as absolute paths in this node's parameters.</p>

GenericOp

The GenericOp node allows you to run a named specific Op. This is particularly useful to run custom Ops written as plug-ins during development and for use within Super Tools and macros. For more information, refer to the *Op API* chapter in the *Katana Technical Guide*.

Control (UI)	Default Value	Function
opType	N/A	Specifies the type of the Op to run, for example, AttributeSet.

Control (UI)	Default Value	Function
opArgs	N/A	<p>Add a new opArgs parameters from the Add dropdown list. The available options are described in greater detail in the <i>User Parameters and Widget Types</i> chapter in the <i>Katana User Guide</i>:</p> <ul style="list-style-type: none"> • Number • String • Group • Number Array • String Array • Float Vector • Color, RGB • Color, RGBA • Button • Toolbar • TeleParameter • Node Drop Proxy
addSystemOpArgs	No	<p>If enabled, adds a 'system' opArg containing information from the graph state time slice, such as frame and shutter timings. This is only necessary for some Ops.</p>
executionMode	immediate	<p>Determines when the Op is run:</p> <ul style="list-style-type: none"> • immediate - the script is run at the locations specified in the applyWhere parameter as it is evaluated at this node's point in the node graph. • deferred - the script is set up by this node but won't actually be run until a later node in the node graph, as specified by the applyWhen parameter.
executionMode: immediate		

Control (UI)	Default Value	Function
applyWhere	at locations matching CEL	<p>Determines where the script is run:</p> <ul style="list-style-type: none"> • at all locations - at all the locations in the node graph. • at specific location - at only the location specified by the location parameter. If this location doesn't exist, it is created automatically. • at locations matching CEL - at only those locations in the node graph that match the CEL statements.
resolvelds	N/A	<p>When executionMode is set to immediate, specify a space-delimited list of strings to indicate that this script should only be resolved by Op resolvers which contain at least one matching "resolveld". This is an advanced feature for greater control over order of evaluation.</p> <p>A useful resolvelds is <code>implicit_preprocess</code>, which runs at the first implicit resolver, before other implicit resolvers, such as <code>MaterialResolve</code> and <code>ConstraintResolve</code> are run.</p>
inputBehavior	only valid	<p>Controls how input ports on the node are mapped onto the inputs of the underlying Op. This parameter is only meaningful when the node has one or more invalid input ports - a port that is not connected to an output port or is connected to an output port that doesn't provide data.</p> <p>When set to only valid, any valid input ports of then ode are skipped when determining which inputs to pass to the underlying Op.</p> <p>When set to by index, all input ports of the node are represented in the list of inputs the Op sees; invalid inputs are represented as an Op of type no-op.</p>
applyWhere: at specific locations		
location	/root/world/location	<p>The location to create, if it doesn't already exist. Otherwise, the scene graph location at which the script is run.</p> <p>For more information on the location widget parameters, see Common Parameter Widgets on page 13.</p>
applyWhere: at locations matching CEL		

Control (UI)	Default Value	Function
CEL	N/A	<p>The scene graph locations are specified using the Collection Expression Language (CEL). The CEL parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>
executionMode: deferred		
applyWhen	during op resolve	<p>Determines when the script is run:</p> <ul style="list-style-type: none"> • during op resolve - the script and its arguments are added as attributes to be executed later by an OpResolve node. If the Op isn't run by an explicit OpResolve node placed in the node graph, it is automatically run at render time by the implicit resolvers. • during material resolve - the script and its arguments are added as attributes under the material.scenegraphLocationModifiers group attribute. This is primarily intended for material scene graph locations, allowing the material to specify a procedural process that is run at every location that the material is assigned to. The script is run as part of the material resolve process, and are executed just after the initial values for the material shader are created at the location. Examples of its use include randomizing or building procedural control over shader parameters. • during katana look file resolve - the script and its arguments are added as attributes under the Scene GraphLocationModifiers group attribute and are evaluated by a LookFileResolve node or by the first implicit resolver if no LookFileResolve node is present.

Control (UI)	Default Value	Function
CEL	N/A	<p>The scene graph locations are specified using the Collection Expression Language (CEL). The CEL parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>
name	GenericOp	The name of the attribute group under which to store the Op type and args.
inputBehavior	only valid	<p>Controls how input ports on the node are mapped onto the inputs of the underlying Op. This parameter is only meaningful when the node has one or more invalid input ports - a port that is not connected to an output port or is connected to an output port that doesn't provide data.</p> <p>When set to only valid, any valid input ports of then ode are skipped when determining which inputs to pass to the underlying Op.</p> <p>When set to by index, all input ports of the node are represented in the list of inputs the Op sees; invalid inputs are represented as an Op of type no-op.</p>
applyWhen: during op resolve		
recursiveEnable	No	<p>When applying in a non-immediate state, enabling this results in the Op running at every location beneath the assigned locations. In general this is more efficient than using an equivalent recursive CEL statement.</p> <p>You can also override the ops.*.recursiveEnable attribute anywhere deeper in the tree to exclude evaluation at those locations. This is similar to the behavior of the visible or light linking attributes.</p>
recursiveEnable: yes		

Control (UI)	Default Value	Function
disableAt	N/A	<p>Execution is disabled for locations at or below this CEL statement. For large scene hierarchies, this is often less expensive than enabling evaluation at a larger number of leaf locations to avoid applying it to a smaller subset.</p> <p>The scene graph locations are specified for the disableAt parameter options by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>
resolvelds	N/A	<p>When applyWhen is set to during op resolve, you may specify a space-delimited list of strings to indicate that this Op should only be resolved by Op resolvers that contain at least one matching "resolveld." This is an advanced feature for greater control over order of evaluation.</p> <p>A useful resolvelds is <code>implicit_preprocess</code>, which runs at the first implicit resolver, before other implicit resolvers, such as <code>MaterialResolve</code> and <code>ConstraintResolve</code> are run.</p>
applyWhen: during katana look file resolve		
recursiveEnable	No	<p>When applying in a non-immediate state, enabling this results in the Op running at every location beneath the assigned locations. In general this is more efficient than using an equivalent recursive CEL statement.</p> <p>You can also override the ops.*.recursiveEnable attribute anywhere deeper in the tree to exclude evaluation at those locations. This is similar to the behavior of the <code>visible</code> or <code>light linking</code> attributes.</p>
recursiveEnable: yes		

Control (UI)	Default Value	Function
disableAt	N/A	<p>Execution is disabled for locations at or below this CEL statement. For large scene hierarchies, this is often less expensive than enabling evaluation at a larger number of leaf locations to avoid applying it to a smaller subset.</p> <p>The scene graph locations are specified for the disableAt parameter options by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>

GroupMerge

The GroupMerge node is a Supertool that creates a convenient interface for managing multiple nodes of the same type.

Within the GroupMerge interface, you can create any number of **nodes of the same type**, and these nodes are combined into a single output by merging them. The nodes are merged in the order they appear in the list.

This node is most often used to group nodes that have no input, but provide a scene graph location as an output. For example, the GroupMerge node could be used to manage multiple PrimitiveCreate nodes, and the output scene graph is all of the primitives merged together.



NOTE: When the GroupMerge node is first created, its type is not defined. You can create a node and then add it to the stacklist by **Shift**+middle-mouse and dragging from the **Node Graph** tab to the node's list in the **Parameters** tab. At that point, the GroupMerge is permanently typed as a group of the type of node that was dragged in.

Control (UI)	Function
	Creates a new node of the type associated with this node and adds it to the node list.
	Brings up a searchable list to aid in selection.
/	Locks all nodes against editing. Unlocks all nodes for editing.

Control (UI)	Function
[Right-click menu]	
Ignore Selected Entries	Disables the selected nodes.
 View At Location	Sets the current view node to the selected node
Delete Selected Entries	Deletes the selected node.
Duplicate Selected Entries	Duplicates the selected node, creating a new copy of both the node and matching its parameters.
Cut Selected Entries	Deletes the selected node and copies it to the clipboard.
Copy Selected Entries	Copies the selected node to the clipboard.
Paste	Paste the current clipboard node into this list.
Tearoff Parameters Of Selected Entries...	Create a new floating window with the parameters of this node on a tab inside.

HierarchyCopy

The HierarchyCopy node takes a scene graph location - or locations - and copies to a given destination location or locations.



NOTE: The HierarchyCopy node copies not just the source locations but all attributes pertaining to those source hierarchies, resulting in a potentially slow operation.

Control (UI)	Default Value	Function
pruneSource	No	When set to Yes, the source location, or locations, are pruned from the recipe.
copies		

Control (UI)	Default Value	Function
sourceLocation	N/A	The scene graph location of the hierarchy to copy. For more information, refer to the Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.
destinationLocations	N/A	The scene graph location under which the copy - or copies - are created. The destinationLocations parameter options are available by clicking Add Locations or ▼ dropdown menu. For more information, refer to the Scene Graph Location and Locations Widget Types in Common Parameter Widgets on page 13.
makeNumberedCopies	No	When set to Yes , the scene graph locations created are sequentially numbered, and the option to make more than one copy is enabled.
makeNumberedCopies:Yes		
numCopies	1	When makeNumberedCopies is set to Yes , sets the number of copies to create.

Importomatic

The Importomatic is a super tool node with a custom interface to load and manage different geometry types. Other geometry or asset types can be added using a custom plug-in.

- Geometry or asset types may be grouped by adding additional outputs, then middle-dragging loaded geometry or assets under the new output.
- Additional outputs may be renamed by selecting the additional output in the Name column, then editing the output name in the parameter field. The default output can not be reordered or renamed.
- The order in which the geometry appears in the GUI determines the merge order, and its listing place in the scene graph.
- Multiple geometry entries may be selected at once, but their parameters are not displayed in the GUI. Multiple entries may be selected, moved, and regrouped at once.
- If a geometry asset has version information, it is displayed in the Version column. A version can be selected by left-clicking on the triangle in the version column for a geometry listing, toggling the Show Explicit Versions button, and selecting the desired version.

Right-Click Menu

The right-click menu options available for each geometry asset allow a user to ignore or delete selected asset entries. Additional outputs can also be deleted from the right-click menu. The default output can not be deleted.

Levels of detail, if available for that asset type can be activated, by selecting Include Levels of Detail from the right-click menu. The Status column indicates that LODs are enabled.

InfoCreate

This node creates a hierarchy of info locations, each tagged with the specified xml block. If **leafName** is specified, locations named with the **leafName** are created as children of the specified locations. If **leafName** is left empty, info locations are created directly at the specified locations.

Images can be embedded using standard syntax, however the node cannot reference web servers (must be links in the file system).

Extra scene graph locations can be baked into Look Files (**.klf**) and are added as new scene graph locations in the scene when a Look File is resolved. A common use of the InfoCreate node is to provide documentation and/or version specific information (either baked in a Look File or as an InfoCreate node in a macro).

Control (UI)	Default Value	Function
leafName	info	<p>If a leafName is populated, the info is created below each specified item in the locations parameter array.</p> <p>Common leaf names are: readme, info, and user.</p>

Control (UI)	Default Value	Function
locations	/root/world	<p>If leafName is not populated, info locations are created directly at the specified locations. If leafName is specified, locations named with the leafName are created as children of the specified locations. The locations parameter options are available by clicking Add Locations or ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location and Locations Widget Types in Common Parameter Widgets on page 13.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  <p>NOTE: An empty or invalid location value generates a location of type error under /root.</p> </div>
text (html editor)	view editor and preview	<p>Sets the mode for the html editor:</p> <ul style="list-style-type: none"> • view editor and preview - the top section of the editor is html source and the bottom section of the editor is a rendered preview. • view only editor - shows only the top section of editor (html source). • view only preview - shows only the bottom section of editor (rendered preview).

Isolate

This node is used to remove objects from a scene. It allows you to select a set of locations to keep and it removes everything else. For example, you could isolate a character or two out of all the geometry in your scene.

The Isolate node cannot take a collection. You can however:

1. Right-click on the collection name in the **Scene Graph** tab and select **Find and Select....**
2. From the **Parameters** tab of the Isolate node, select **isolateLocations > Add Locations > Replace with Scene Graph Selection**.

Control (UI)	Default Value	Function
isolateLocations	none	<p>This is a list of locations to keep while every other location is removed by the Isolate.</p> <p>For more information, refer to the Locations Widget Type in Common Parameter Widgets on page 13.</p>
isolateFrom	/root/world/geo	<p>This is the topmost location to remove from the scene. For example, if you set this to /root/world/geo, then nothing in /root/world/lgt or /root/materials is modified. This parameter allows you to scope the changes. To isolate a single shape from an entire character, set isolateFrom to the character path (for example, /root/world/geo/somecharacter), then set isolateLocations to the shape you'd like to keep.</p> <p>For more information, refer to the Scene Graph Location Widget Type in Common Parameter Widgets on page 13.</p>
Enable secondary (inverse) output	disabled	<p>When enabled, the secondary output provides a scene containing the scene graph locations which have been removed from the primary output. For example, in a scene containing the following locations:</p> <p>/root/world/geo</p> <p>/root/world/geo/box</p> <p>/root/world/geo/circle</p> <p>/root/world/lgts</p> <p>If /root/world/geo/box is isolated using isolateFrom /root/world/geo, the secondary output contains /root/world/geo/circle.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> NOTE: Any scene graph location that is a peer or ancestor of the isolationRoot is present on both outputs.</p> </div>

LightCreate

This node is used to create a scene graph containing a light. LightCreate does not load the light from any file or asset but instead builds an entirely novel light from the parameters you specify on this node. This node is not used generally, the GafferThree node is often used instead.



NOTE: LightCreate requires a light shader to function properly.

LightCreate and CameraCreate are identical, except for the type of scene graph locations they create, and the population of the lightList vs. cameraList.

Control (UI)	Default Value	Function
name	/root/world/lgt/light	<p>Sets the scene graph location where the light is created. For example, the default value of /root/world/lgt/light creates a light at the location /root/world/lgt/light. The name parameter options are available in either the scene graph widget or ▼ dropdown menu to the right of the parameter.</p> <p>For more information, refer to the Create Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.</p>
projection	perspective	<p>Sets the light projection mode:</p> <ul style="list-style-type: none"> • perspective • orthographic
fov	70	Controls the field of view angle in degrees.
near	0.1	Sets the near clipping plane distance.
far	100000	Sets the far clipping plane distance.
screenWindow		
left	-1	This set of four number parameters controls the screen window placement on the imaging plane. They are, in order, left, right, bottom and top bounds of the screen window.
right	1	
bottom	-1	
top	1	

Control (UI)	Default Value	Function
LightCreate parameters continued		
centerOfInterest	20	Offsets the center of interest of the light.
orthographicWidth	30	Sets the orthographic projection width.
radius	1	Sets the light's radius.
previewColor		
previewColor	1, 1, 1	<p>Specifies the color of the light in the Viewer. This value does not affect the color value of the light when rendering, it's used for testing the placement of lights.</p> <p>For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.</p>
manipulators > Add		
Add Entry	none	Adds entries to the manipulators list.
manipulators > manipulator		
name	none	
preset	default	
selected	Handles	<ul style="list-style-type: none"> • Handles - • Outline - • None -
visible	Outline	<ul style="list-style-type: none"> • Handles - • Outline - • None -
lightListDefaults		
enable	1	
transform		

Control (UI)	Default Value	Function
transform	N/A	Transforms the light according to the SRT or matrix controls. For more information, refer to the Transform Controls Widget Type in the Common Parameter Widgets on page 13.
transform > Tools ▾	N/A	Adjusts the light to match selected scene graph selection options in the dropdown menu. For more information, refer to the Transform Tools Widget Type in the Common Parameter Widgets on page 13.
LightCreate parameters continued		
makeInteractive	Yes	When set to Yes , you can drag objects in the Viewer and Katana retains the information from the Viewer.

LightLink

The LightLink node manipulates the lightList attribute on the scene to perform selective lighting of objects. LightLink allows you to control which lights illuminate which objects, using a number of different modes.



NOTE: The GafferThree node uses a LightLink internally to provide light linking. The user interface there is substantially similar to the LightLink node.



NOTE: Light linking information is stored on the objects themselves in the lightList attribute. This stores the enable state of a light for each location in the scene.

Visibility does not have any effect on lights, so a VisibilityAssign does not disable a light. LightLink is the best way to turn a light on or off by hand.

Control (UI)	Default Value	Function
effect	illumination	<p>Determines whether the link is acting upon the light's illumination or shadow visibility of the specified objects:</p> <ul style="list-style-type: none"> • illumination • shadow visibility <p>Shadow visibility is only currently respected by Arnold renders.</p> <ul style="list-style-type: none"> • Custom
action	off	<p>Controls the LightLink node's behavior:</p> <ul style="list-style-type: none"> • on - turn the selected lights on for the selected objects. Does nothing else. • exclusive on - turn the selected lights on for the selected objects. Also turn the selected lights off for all other objects. Use this to force the selected lights to only illuminate the selected objects, and nothing else. • off - turn the selected lights off for the selected objects. Does nothing else. • exclusive off - turn the selected lights off for the selected objects. Also turn the selected lights on for all other objects. Use this to force the selected lights to not illuminate the selected objects, but to illuminate everything else. • clear - remove any local setting for light enable/disable for the selected objects; the inherited settings are used on these objects. • delete - the selected lights are removed from the light list for all objects in the scene. This is more than simply turning the lights off; they're removed from the list, and a LightListEdit is required to turn them on again. • delete inverse - the selected lights are the only lights left in the light list for all objects in the scene.
effect: custom		

Control (UI)	Default Value	Function
customAttrName	custom	<p>Specifies a custom attribute name to set on the lightList for your object scene graph locations. The value of the custom parameter becomes the attribute name, which is set on the object scene graph location for each light.</p> <p>When a LightLink node is used with a LightLinkEdit node, for instance, if the:</p> <ul style="list-style-type: none"> • light parameter is set to /root/world/lgt/spotlight, • custom parameter is set to myAttr, • off CEL parameter is set to /root/world/geo/pony, <p>Then</p> <ul style="list-style-type: none"> • /root/world/geo/pony has an attribute named lightList.root_world_lgt_spotlight.myAttr, whose value is set to 0.
objects	none	<p>Sets the object(s) on which to operate. The scene graph locations are specified for the objects parameter options by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>
lightMode	CEL	<p>Controls how you specify which lights to operate on:</p> <ul style="list-style-type: none"> • CEL • Paths <p>Paths are included for backward compatibility.</p>
lightMode: CEL		

Control (UI)	Default Value	Function
lights	none	<p>When lightMode is set to CEL, this CEL statement is used to select the lights to operate on.</p> <p>The scene graph locations are specified using the Collection Expression Language (CEL). The lights parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>
lightMode: Paths		
lightPaths	none	<p>When lightMode is set to Paths, this list of light path names is used as the set of lights to operate on. The lightPaths parameter options are available by clicking Add Locations or ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location and Locations Widget Types in Common Parameter Widgets on page 13.</p>

LightLinkEdit

The LightLinkEdit node adjusts which objects are illuminated by a light. This node edits **lightList** attributes that were previously set by LightLink or LightLinkResolve nodes.



NOTE: Light linking information is stored on the object scene graph locations themselves in the lightList attribute. This stores the enable state of a light for each location in the scene.

Control (UI)	Default Value	Function
clearUnmatched	disabled	The clearUnmatched parameter determines whether or not existing light linking attributes for this light are removed from locations that do not match the on or off expressions.

Control (UI)	Default Value	Function
effect	illumination	<p>Determines whether the link is acting upon the light's illumination or shadow visibility of the specified objects:</p> <ul style="list-style-type: none"> • illumination • shadow visibility • custom <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  <p>NOTE: Shadow visibility is only currently respected by Arnold renders.</p> </div>
light	N/A	<p>Specifies the scene graph location for the light you want to apply the lighting quality of the effect parameter from.</p> <p>For more information, refer to the Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.</p>
effect: custom		
customAttrName	custom	<p>Specifies a custom attribute name to set on the lightList for your object scene graph locations. The value of the custom parameter becomes the attribute name, which is set on the object scene graph location for each light.</p> <p>When a LightLink node is used with a LightLinkEdit node, for instance, if the:</p> <ul style="list-style-type: none"> • light parameter is set to /root/world/lgt/spotlight, • custom parameter is set to myAttr, • off CEL parameter is set to /root/world/geo/pony, <p>Then</p> <ul style="list-style-type: none"> • /root/world/geo/pony has an attribute named lightList.root_world_lgt_spotlight.myAttr, the value of which is set to 0.
LightLinkEdit parameters continued		

Control (UI)	Default Value	Function
on	N/A	<p>Links matching locations to the effect specified in the effect parameter.</p> <p>The scene graph locations are specified using the Collection Expression Language (CEL). The on parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>
off	N/A	<p>Disables the effect specified in the effect parameter for the matching locations.</p> <p>The scene graph locations are specified using the Collection Expression Language (CEL). The off parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>

LightLinkResolve

The LightLinkResolve node resolves the attributes which the LightLinkSetup node sets on **/root/world.lightList**. The LightLinkSetup node defines an entry on the **lightList** containing CEL expressions defining On and Off locations for a particular light. The LightLinkResolve node resolves these CEL expressions into local attributes on any locations that match the criteria defined by these CEL expressions.

LightLinkSetup

The LightLinkSetup node is an alternative to the [LightLink](#) node and sets the attributes on the lightList at the **/root/world** location instead of the object's scene graph location. It allows the node to set light linking options for locations that don't exist at that point in the node graph. These options are resolved by implicit resolvers or can be resolved manually by a LightLinkResolve node.

Control (UI)	Default Value	Function
clearUnmatched	disabled	<p>When linking is resolved, the clearUnmatched parameter determines whether or not existing light linking attributes for this light are removed from locations that do not match the on or off expressions.</p> <p>The effect of this parameter is only visible in the Attributes tab when linking has been resolved, which means after a LightLinkResolve node or when Implicit Resolvers are active.</p> <p>Examines the lightList attribute on your linked objects to ensure that the attributes have been set correctly. If the attribute has been disabled, the value of the enable child attribute in the lightList attribute for your light is 0; otherwise, the default enabled setting is 1.</p>
action	append linking information	<p>Determines how light linking settings from this node are merged with settings in the incoming scene. If this light doesn't exist in the incoming scene, this option has no effect.</p> <ul style="list-style-type: none"> • append linking information - the new attributes are appended to the incoming options. Where conflicts occur, the attributes that are set from this node are used. • override linking information - the linking options set in this node override information from the incoming scene options.
effect	illumination	<p>Determines whether the link is acting upon the light's illumination or shadow visibility of the specified objects:</p> <ul style="list-style-type: none"> • illumination • shadow visibility • custom <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  <p>NOTE: Shadow visibility is only currently respected by Arnold renders.</p> </div>
effect: custom		

Control (UI)	Default Value	Function
customAttrName	custom	<p>Specifies a custom attribute name to set on the lightList for your object scene graph locations. The value of the custom parameter becomes the attribute name that is set on the object scene graph location for each light.</p> <p>When a LightLinkSetup node is used with a LightLinkResolve node, for instance, if the:</p> <ul style="list-style-type: none"> • light parameter is set to /root/world/lgt/spotlight, • custom parameter is set to myAttr, • off CEL parameter is set to /root/world/geo/pony, <p>Then</p> <ul style="list-style-type: none"> • /root/world/geo/pony has an attribute named lightList.root_world_lgt_spotlight.myAttr, whose value is set to 0. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  <p>NOTE: If you've added a LightLinkSetup node only and have not linked it to a LightLinkResolve node, the attribute on /root/world/geo/pony is not set.</p> </div>
light	N/A	<p>Specifies the scene graph location of the light you want to apply the lighting quality of the effect parameter from.</p> <p>For more information, refer to the Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.</p>
initialState	use existing value	<p>Determines the initial value for the specified effect in the light list entry for this light:</p> <ul style="list-style-type: none"> • use existing value - the attribute uses the value of the attribute in the incoming scene if applicable. • on - sets the initial value to 1. • off - sets the initial value to 0.

LightLinkSetup parameters continued

Control (UI)	Default Value	Function
on	N/A	<p>Links matching locations to the effect specified in the effect parameter.</p> <p>The scene graph locations are specified using the Collection Expression Language (CEL). The on parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>
off	N/A	<p>Disables the effect specified in the effect parameter for the matching locations.</p> <p>The scene graph locations are specified using the Collection Expression Language (CEL). The off parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>

LightListEdit

This node adds locations to the lightList attribute at **/root/world**. This is useful for including lights whose loading is deferred. Only explicit paths are supported because this information is required at the start of rendering. LightListEdit can also be used to extract lights from components and makes them renderable from a Look File.

Control (UI)	Default Value	Function
locations	none	<p>Sets the locations of lights from a path, either in the scene graph or the node graph. The locations parameter options are available by clicking Add Locations or ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location and Locations Widget Types in Common Parameter Widgets on page 13.</p>

Control (UI)	Default Value	Function
mode	add	Sets edit mode, though currently only add is available.
initialState	on	Determines whether the newly-added light locations are initially on or off .

LocationCreate

Allows you to create a scene graph location of any type. Often used in macros to generate one or more scene graph location without the overhead or type-specific attributes created by the other Create nodes.

Control (UI)	Default Value	Function
type	group	Sets the type attribute of the scene graph location(s) to be created (as seen in the 'Type' column of the scene graph).
locations	/root/world	<p>Sets one or more scene graph path(s) to the location(s) to be created. The locations parameter options are available by clicking Add Locations or ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location and Locations Widget Types in Common Parameter Widgets on page 13.</p>
attrs	None	Drag string or number attributes here to have them added to the scene graph location(s) created by this node.

LocationGenerate

The LocationGenerate node allows you to add a child location of a specific type to any locations matching a CEL statement.

Control (UI)	Default Value	Function
CEL	none	<p>The scene graph locations are specified using the Collection Expression Language (CEL). The CEL parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>
name	loc	Specifies the child location name.
locationType	none	Specifies the child location type.
addParentNameAsAttr	none	Optionally adds the basename of the matching CEL location at the specified attribute name. Typically, this parameter is used for locations to run render procedural, if the name of the parent location needs to be passed to the renderer as a parameter.

LodGroupCreate

When pointed at a location, the children are assigned a level of detail (LOD) range for each node input. The parameters of the node directly correspond to the PRMan values and are based on screen area. This means when rendering smaller test images, the resulting output is different to full renders as the calculation of the bounding box mapped to the screen is different. For more on how to get the most out of these parameters, please review the PRMan documentation.

Each node input that requires an LOD range must be added as an additional input using the **inputs > Add > Add Input** menu option.

Control (UI)	Default Value	Function
groupName	lod_group	The name of the level-of-detail group location that is created at the hierarchyTargetLocation .

Control (UI)	Default Value	Function
hierarchyTargetLocation	/root/world/geo	The scene graph location where the level-of-detail group is placed. Each node input creates a level-of-detail location below this location which stores the lodRange attributes for that input.
inputs	N/A	The parameter grouping for the node inputs.
inputs > Add		
Add Input	N/A	Menu option to add a new node input and create an additional level-of-detail location to store its scene.
inputs > input		
minVisible	0	When the bounding box is transformed to screen space, if its pixel count is less than the minVisible parameter, the object is not displayed.
lowerTransition	0	When the bounding box is transformed to screen space, if its pixel count lies between the minVisible and lowerTransition parameters, the object is only part displayed.
upperTransition	99999999999999977 48809823456034029568	When the bounding box is transformed to screen space, if its pixel count is between the upperTransition and maxVisible parameters, the object is only part displayed.
maxVisible	99999999999999977 48809823456034029568	When the bounding box is transformed to screen space, if its pixel count is less than the maxVisible parameter, the object is not displayed.

LodSelect

This node removes all but one LOD (level-of-detail) location beneath the selected level-of-detail groups. The location to keep is selected based on one of three attributes, either:

- **by index** - select the level-of-detail location to keep based on its index in the child list of the level-of-detail group.
- **by tag** - select the **level-of-detail** location to keep based on its **info > componentLodTag** attribute.
- **by weight** - select the **level-of-detail** location to keep based on its **info > componentLodWeight** attribute. The **level-of-detail** location below the **level-of-detail group** location that is closest to the weight specified in the **selectionWeight** parameter is kept.

Control (UI)	Default Value	Function
CEL	none	<p>The scene graph locations are specified using the Collection Expression Language (CEL). The CEL parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p> <p> NOTE: CEL statement should match locations of type level-of-detail group.</p>
mode	by index	<p>Sets the method used to specify levels of detail:</p> <ul style="list-style-type: none"> • by index • by tag • by weight <p> NOTE: In cases of ambiguity, or where no groups match the criteria, the lowest index LOD group is selected (after all possible filtering has taken place) for the by index and by tag modes.</p> <p>The by index mode operates with strict matching, and produces an error if the chosen index does not exist.</p>
mode is: by index		
selectionIndex	0	Sets the index of which LOD child to keep.
mode is: by tag		
selectionTag	hi	Sets the tag of which LOD child to keep.
mode is: by weight		
selectionWeight	1	Sets the weight to use while determining which children to keep.

LodValuesAssign

This node assigns level of detail (LOD) ranges to the child locations for all CEL statement matches.

The **ranges** parameters of the node directly correspond to the PRMan values used for level of detail and are based on screen area. This means when rendering smaller test images, the resulting output is different to full renders as the calculation of the bounding box mapped to the screen is different. For more on how to get the most out of these parameters, please review the PRMan documentation.

Each child location requires an LOD range and must be added using the **ranges > Add > Add Entry** menu option.

Control (UI)	Default Value	Function
CEL	none	<p>The scene graph locations are specified using the Collection Expression Language (CEL). The CEL parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  <p>NOTE: CEL statement should match locations of type level-of-detail group.</p> </div>
ranges	N/A	The parameter grouping that holds the LOD ranges for each child location.
ranges > Add		
Add Entry	N/A	Menu option to add a new ranges parameter with minVisible , lowerTransition , upperTransition , and maxVisible . For every child location of the level-of-detail group location there should be a corresponding level-of-detail location and ranges parameter.
ranges > lod		
minVisible	0	When the bounding box is transformed to screen space, if its pixel count is less than the minVisible parameter, the object is not displayed.
lowerTransition	0	When the bounding box is transformed to screen space, if its pixel count lies between the minVisible and lowerTransition parameters, the object is only part displayed.

upperTransition	9999999999999999 7748809823456034029568	When the bounding box is transformed to screen space, if its pixel count is between the upperTransition and maxVisible parameters, the object is only part displayed.
maxVisible	9999999999999999 7748809823456034029568	When the bounding box is transformed to screen space, if its pixel count is less than the maxVisible parameter, the object is not displayed.

LookFileBake

Bakes a Look File for a scene graph location(s) specified in the **rootLocations** field.

Control (UI)	Default Value	Function
rootLocations		
rootLocations	/root/world	<p>Sets the scene graph location(s) to bake the Look File information for. Any location under /root/world can be used, but it is recommended that components or assembly locations are specified. The rootlocations parameter options are available by clicking Add Locations or ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location and Locations Widget Types in Common Parameter Widgets on page 13.</p>
passes		

Control (UI)	Default Value	Function
passes	None	<p>Passes are typically render passes, but could also be auxiliary baking passes for generating point clouds or brickmaps. A Look File can have one or multiple passes.</p> <p>To add a pass, select Add > Add Pass Input.</p> <p>A new pass input is created on the node, and a pass name field is added to the pass list. To change the pass name, simply change the name text field supplied.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  NOTE: All pass names must be unique. </div>
LookFileBake parameters continued		
saveTo	None	<p>Sets where to store the baked Look File.</p> <p>For more information, refer to the Asset and File Path Widget Types in the Common Parameter Widgets on page 13.</p>
options		
outputFormat	as directory	<p>Specifies whether to bake the Look File to a single archive (currently Zip), or instead to a per-pass directory representation.</p> <p>The directory option is higher performance and recommended for heavy-weight use cases.</p>
includeGlobalAttributes	No	When set to Yes , attributes on /root are stored in the baked Look File.
alwaysIncludeSelectedMaterialTrees	No	When set to Yes , include all material locations at or below the paths specified by selectedMaterialTreeRootLocations without regard to whether they are assigned to geometry within the scope of the rootLocations paths.
LookFileBake parameters continued		
Write Look File	N/A	Click to bake the Look File.

LookFileLightAndConstraintActivator

Katana maintains a list of lights, cameras, and constraints at **/root/world** within the scene graph. When a Look File brings in a light or constraint, the lists at **/root/world** need to be updated. The LookFileLightAndConstraintActivator node activates LookFile lights and constraints by updating the respective lists. It is also used to add constraints from LookFiles to the global constraint list. This list is used to specify the order in which constraints are evaluated, so this only has to be done if the constraints from the LookFile need to be evaluated in a specific order. Because it reads its input from a LookFile-resolved scene, you should place it after either a LookFileManager or LookFileResolve node.

Choose **Search Entire Incoming Scene...** or **Search Incoming Scene From Scene Graph Selection...** from the **Action** menu to find available lights and constraints.

- Entries are organized by asset, location, and then light and constraint paths.
- Gray entries are pending -- found by the searching tools but not yet enabled in the scene.
- Pending entries are not saved from session to session.
- Locations (entries immediately below the asset entries) may be refreshed individually by choosing **Search From Selected Locations** from the right-click menu. This option is only available when one or more location entries are selected.

To enable a pending entry, choose **Enable** from the right-click menu at any point within the hierarchy.

Enable and disable operations executed in this manner always act upon the selected entries and all of their children. Individual light and constraint paths may also be enabled by clicking on the checkbox next their names.

To enable everything at once:

1. First, choose **Search Entire Incoming Scene...** from the **Action** menu.
2. When that has completed, choose **Select All** from the **Action** menu right-click on any entry and choose **Enable**.

Control (UI)	Default Value	Function
Action	none	Searches the scene graph for lights and constraints brought in by Look Files then enables or disables the results as required.

LookFileManager

LookFileManager decodes incoming Look Files that have been set up in another scene. Each Look File piece of imported geometry passed into this node must be assigned through a LookFileAssign node. Once the Look File is assigned, LookFileManager decodes the Look File into the passes set up by the look development artist using a LookFileBake node.

Control (UI)	Default Value	Function
Look Files	none	Lists the Look Files that are being edited by the LookFileManager.
Passes	default	Lists any passes associated with the Look Files.
Add Override	none	Allows you to add overrides to selected Look Files.

LookFileMaterialsIn

This node loads materials from a Look File into the local scene to allow additional edits before they are applied to the scene.

Control (UI)	Default Value	Function
lookfile	none	<p>Sets the Look File path and name.</p> <p>For more information, refer to the Asset and File Path Widget Types in the Common Parameter Widgets on page 13.</p>
passName	none	<p>Sets the pass name to use from the Look File.</p> <p>For more information, refer to the Look File Widget Type in the Common Parameter Widgets on page 13.</p>
asReference	No	<p>When set to Yes, the material is loaded as a reference.</p> <p>Reading the material by reference causes any materials assigned to keep a reference to the Katana Look File from which they got their material.</p>
locationForMaterials	Load at original location	<p>Sets where in the scene graph to import the materials from:</p> <ul style="list-style-type: none"> • Load at original location - the materials maintain the same location. • Load at specified location - provides a parameter, <code>userLocation</code>, that acts as a namespace for the material palette. For instance, a material at <code>/root/materials/geo/chrome</code> with <code>userLocation default_pass</code> is placed at <code>/root/materials/lookfile/default_pass/geo/chrome</code>.

Control (UI)	Default Value	Function
locationForMaterials: Load at specified location		
userLocation	N/A	Specify the location that acts as a namespace for the material palette.

LookFileMaterialsOut

Use this node to write incoming materials into a Look File. This is useful for creating a material library that can be read into other scenes.

Control (UI)	Default Value	Function
saveTo	none	Sets the location of the Look File to contain the material. For more information, refer to the Asset and File Path Widget Types in the Common Parameter Widgets on page 13.
options		
outputFormat	as archive	Specifies whether to write the Look File to a single archive (currently Zip), or instead to a per-pass directory representation. The directory option is higher performance and recommended for heavy-weight use cases.
Write Look File	N/A	Click to write the material to the specified Look File.

LookFileMultiBake

LookFileMultiBake is a convenient SuperTool that wraps multiple LookFileBake nodes into a single node. Passes are shared between nodes, and the Look Files can be baked individually or all at once.

Control (UI)	Default Value	Function
+	N/A	Creates a new LookFileBake and adds it to the LookFileBake list.
When an entry has been added		
rootLocations	/root/world	<p>Sets one or more scene graph path(s) to the location(s) to be created. The locations parameter options are available by clicking Add Locations or ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location and Locations Widget Types in Common Parameter Widgets on page 13.</p>
🔑	N/A	Brings up a searchable list to aid in selection.
saveTo	N/A	<p>Sets where to store the baked Look File.</p> <p>For more information, refer to the Asset and File Path Widget Types in the Common Parameter Widgets on page 13.</p>
Passes		
name	pass	<p>Passes are typically render passes, but could also be auxiliary baking passes for generating point clouds or brickmaps. A LookFile can have one or multiple passes. To add a pass, select Add > Add Pass Input.</p> <p>A new pass input is created on the node, and a pass name field is added to the pass list. To change the pass name, simply change the name text field supplied.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  NOTE: All pass names must be unique. </div>
LookFileMultiBake parameters continued		
includeGlobalAttributes	No	When set to Yes , attributes on /root are stored in the baked Look File.

Control (UI)	Default Value	Function
alwaysIncludeSelectedMaterialTrees	No	If enabled, this includes all material locations at or below the paths specified by selectedMaterialTreeRootLocations without regard to whether they are assigned to geometry within the scope of the rootLocations paths.
When alwaysIncludeSelectedMaterialTrees: yes		
selectedMaterialTreeRootLocations	/root/materials/geo	Sets one or more scene graph path(s) to the location(s) to be created. The locations parameter options are available by clicking Add Locations or ▼ dropdown menu. For more information, refer to the Scene Graph Location and Locations Widget Types in Common Parameter Widgets on page 13.
Write ▼		
Write All Look Files...	N/A	Click to bake all the Look Files.
Writes Selected Look Files...	N/A	Click to bake the selected Look Files.

Control (UI)	Function
LookFileBake list [Right-click menu]	
Delete Selected Entries	Deletes the selected entries.

LookFileOverrideEnable

LookFileOverrideEnable allows you to bring into the scene materials from a specific Look File/pass so you can override them before the Look File is resolved using the LookFileResolve node.

Control (UI)	Default Value	Function
lookfile	N/A	Selects the Look File to import materials from. For more information, refer to the Asset and File Path Widget Types in the Common Parameter Widgets on page 13.
versionsToOverride	Only this Version	Specifies on which version of the Look File the overrides are reflected.
locationForMaterials	Use asset name	Sets where to import the materials in the scene graph. <ul style="list-style-type: none"> • Use asset name - the materials are imported in a unique location based on the lookfile asset fields. • User-specified - provides a parameter, userLocation, for you to specify a namespace for the materials.
When locationForMaterials: User-specified		
userLocation	N/A	Specifies a namespace for the materials. For instance, a material at /root/materials/geo/chrome with userLocation default_pass is placed at /root/materials/default_pass/geo/chrome .
LookFileOverrideEnable parameters continued		
passName	default	Specifies the name of the Look File/pass you want to override. The entry 'default' displays if the field is left empty. For more information, refer to the Look File Widget Type in the Common Parameter Widgets on page 13.
loadAsReference	No	When set to Yes , the material is loaded as a reference. Reading the material by reference causes any materials assigned to keep a reference to the Katana Look File from which they got their material.
enforceVersion	No	When set to Yes , you are enforcing this version to be resolved by making the LookFileResolve node resolve the asset in the LookFileOverrideEnable node instead of the one in the LookFileAssign node.

LookFileResolve

This node applies a specific pass from assigned Look Files to the scene. LookFileResolve is also in the implicit resolver list applied at render time.

Control (UI)	Default Value	Function
passName	none	Sets the name of the Look File pass to use. For more information, refer to the Look File Widget Type in the Common Parameter Widgets on page 13.
Flush Caches	N/A	Click to flush the Look File cache and force a reload.

Material

This node defines a material, which is a set of shader calls and associated parameters. Materials are assigned to geometry using the [MaterialAssign](#) node.

Control (UI)	Default Value	Function
name	Material	Sets the node name. It's a good idea to use a meaningful name such as "mtl_red".

Control (UI)	Default Value	Function
action	create new material	<p>Determines the node's behavior:</p> <ul style="list-style-type: none"> • create new material - creates a new scene graph location of type geometry material or light material beneath /root/materials/(geo lgt) with the name specified by the name parameter. • create from LookFile - creates a new scene graph location of type geometry material or light material from a specified LookFile with the name specified by the name parameter. • create child material - creates a new scene graph location of type geometry material or light material beneath the location specified by inheritsFrom.location parameter with the name specified by the name parameter. • edit material - displays the incoming values of a single scene graph material location specified by the edit.location parameter. This is useful for making changes when the original Material node, which created this location, is not within the current session, or for multiple branches of a graph.
action	(continued)	<ul style="list-style-type: none"> • override materials - accepts drag-and-dropped attributes from material attribute groups. This can be used in two ways: <ul style="list-style-type: none"> • When aimed at locations within the renderable scene, it creates a materialOverride attribute. At resolve time, these values override equivalent values in the material attribute of renderable scene graph locations beneath. This is useful for making global changes to the assigned instances of many different materials at once, regardless of whether they share the same source. • When aimed at locations of type geometry material or lightmaterial, it modifies the material directly. This does not display incoming values because they could differ from location to location. This means that you must specify the shader in order to display adjustable parameters.
When action is: create new material		
namespace	none	Specifies the scene graph location where the material is placed.

Control (UI)	Default Value	Function
makeInteractive	No	When set to Yes , you can drag objects in the Viewer and Katana retains the information from the Viewer.
Add shader	N/A	Click to add a renderer-specific shader to the material. The shaders that are available change depending on the renderers installed.
When action is: create from LookFile		
namespace	none	Specifies the scene graph location where the material is placed.
makeInteractive	No	When set to Yes , you can drag objects in the Viewer and Katana retains the information from the Viewer.
lookfile	N/A	Selects the Look File to import materials from. For more information, refer to the Asset and File Path Widget Types in the Common Parameter Widgets on page 13.
materialPath	N/A	Allows you to select materials from the Look File.
asReference	Yes	When set to Yes , the material is loaded as a reference. Reading the material by reference causes any materials assigned to keep a reference to the Katana Look File from which they got their material.
When action is: create child material		
makeInteractive	No	When set to Yes , you can drag objects in the Viewer and Katana retains the information from the Viewer.
Add shader	N/A	Click to add a renderer-specific shader to the material. The shaders that are available change depending on the renderers installed.
inheritsFrom > location	N/A	Sets the scene graph path to the location to be created. The location parameter options are available by clicking the ▼ dropdown menu. For more information, refer to the Scene Graph Location Widget Type in Common Parameter Widgets on page 13.
When action is: edit material		

Control (UI)	Default Value	Function
makeInteractive	No	When set to Yes , you can drag objects in the Viewer and Katana retains the information from the Viewer.
Add shader	N/A	Click to add a renderer-specific shader to the material. The shaders that are available change depending on the renderers installed.
edit > location	N/A	Sets the scene graph path to the location to be created. The locations parameter options are available by clicking the  dropdown menu. For more information, refer to the Scene Graph Location Widget Type in Common Parameter Widgets on page 13.
When action is: override materials		
CEL	none	The scene graph locations are specified using the Collection Expression Language (CEL). The CEL parameter options are available by clicking Add Statements . For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.
attrs	none	Middle-click and drag attributes from the Attributes tab to this hotspot to use that attribute.

MaterialResolve

Resolves materials in the scene graph. At scene graph locations with **materialAssign** attributes, it finds the material that is referenced and copies its material attributes to the scene graph location. Results of this operation can be viewed in the **Attributes** tab. It can also be used to apply material overrides set by the **Material** node.

MaterialStack

MaterialStack node is a specialized GroupStack for organizing your scene Materials. To move a Material node that is outside the MaterialStack node to inside the stack, hold down the shift key and middle-mouse drag it in.

The Material in the stack are linked together, providing a single output by connecting them one after the other in serial, in the order in which they appear in the stack. Selecting Materials in the stack displays their controls on the right of the stack.

Merge

The merge node allows you to combine multiple scenes into a single output scene. All objects in any of the input scenes are present in the output scene. If a location is present in more than one of the input scenes, then attribute values are taken from the left most input which has the location (however, the **Advanced Options** allow more control over this). Merge is a very versatile node for collecting multiple elements into a scene for rendering.

Tips:

- A merge node with a single input is effectively a no-Op node.
- Right-click on the Merge node in the **Node Graph** tab input ports to delete any unused ports.
- Right-click on a node while connecting a link in the **Node Graph** tab to display a popup menu of ports to connect to; this can be easier than hunting for a specific port on a Merge.
- Hit the tilde key (~) while connecting a link in the **Node Graph** tab to connect to the left most open port on the node, or add a new port if none are free.

Control (UI)	Default Value	Function
showAdvancedOptions	No	When set to Yes , the advanced parameters are available. These are normally only needed when doing something unusual or complex; merging two components together to form a single model is a common case, for example merging cloth and deforming geometry together. Typically, this use of the Merge node is hidden from the user inside a show macro so it's unlikely you'll need the advanced options.
showAdvancedOptions: Yes		
advanced		

Control (UI)	Default Value	Function
sumBounds	No	When enabled, bound attributes are queried for each relevant input location and the total results are used. The output bounding box at each location is expanded to be large enough to contain all the inputs at that location. This is important when merging renderable geometry together inside of components.
preserveWorldSpaceXform	No	When enabled, all inherited xform attributes (preceded by an origin statement) are applied at each location whose source input differs from that of its parent. This is only necessary in exceptional situations where there are conflicting transformations on overlapping locations of the merge inputs. Basically, this forces some locations to ignore their parent transforms so that they appear in the correct location in the scene. This is most commonly used when merging deforming geometry into a component, because the deforming geometry may have different transforms on locations shared with the non deforming geometry. If the result of the merging has objects that seem to be in the wrong position, try this option as a possible solution.
preserveInheritedAttributes	none	Displays a list of attribute names for which inheritance should be preserved when choosing between inputs of the Merge. Whenever a child location's source input differs from that of its parent, these attributes are queried globally and applied locally to the child location.

Control (UI)	Default Value	Function
preferredInputAttributes	none	Displays a list of attribute names and indices of inputs for which the preferred value of an attribute should be read. These are exceptions to the general rule of leftmost input wins. For the listed attribute, a given input is given 'first crack' at providing the attribute in the result before the general rule is used. This is often used when merging two versions of a component to form a single output model; the first input provides most of the attributes, but a second input might provide correctly deformed geometry or other attributes that should be used in preference to the first input. Again, this is typically rolled into a show macro, so it's unlikely you'll need to work with this setting directly.
mergeRoot	/root	
skipLightList	No	
skipGlobals	No	
skipRenderSettingsOutputs	No	
skipRelativeScopedCoordinateSystems	No	
skipCollections	No	
inputs		
inputs	none	Allows you to name the inputs on the Merge node.

NetworkMaterial

The NetworkMaterial node creates a material location for shading nodes that are connected as inputs, in order for the material to be assigned to objects in the scene. You can assign the material locations that are created by a NetworkMaterial node to a location with [MaterialAssign](#) node.

Control (UI)	Default Value	Function
name	NetworkMaterial	Specifies the name of the location.
namespace	N/A	Creates a sub-location for the network material in the Scene Graph tab.
Add Terminal ▼	N/A	Specifies which input to expect from the shading nodes, for instance Arnold renderer-specific shading nodes.
Material Interface ⚙️ or right-click on interface table		
Rebuild with Current State	N/A	Updates the information of the Material Interface.
Remove Selected Local Definition...	N/A	Removes the selected shading node's exposed parameters.
Remove All Local Definitions...	N/A	Removes all selected shading node's exposed parameters.

NetworkMaterialInterfaceControls

The NetworkMaterialInterfaceControls node allows you to apply a visibility or a lock condition on one or several parameters exposed by a [NetworkMaterial](#) node. This node works on a network material location, which can come from a NetworkMaterial node or a Look File.

Control (UI)	Default Value	Function
materialLocation	N/A	Specifies the scene graph location path of the network material to be modified. The materialLocation parameter options are available by clicking the ▼ dropdown menu. For more information, refer to the Scene Graph Location Widget Type in Common Parameter Widgets on page 13.

Control (UI)	Default Value	Function
state	visibility	<p>Selects a condition state to apply:</p> <ul style="list-style-type: none"> • visibility - depending on the condition, displays or hides the parameter or page specified in the targetName parameter. • lock - depending on the condition, locks or unlocks the parameter or page specified in the targetName parameter, making any edits impossible if locked.
targetType	parameter	<p>Selects whether to apply the condition on either:</p> <ul style="list-style-type: none"> • parameter - applies on a single parameter. • page - applies on a set of parameters grouped into a page.
targetName	N/A	<p>Specifies the name of the chosen parameter or page.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  <p>NOTE: The name must be identical to the one that displays in the NetworkMaterial node's Material Interface.</p> </div>
definitionStyle	operator tree	<p>Selects how to set up the condition:</p> <ul style="list-style-type: none"> • operator tree - allows you to set up conditions using an operator tree. • conditional state expression - allows you to set up conditions using one or several expressions.
When definitionStyle: operator tree		
operators		
op	and	<p>Selects which expression operator to use in the operator tree:</p> <ul style="list-style-type: none"> • and - the resulting expression is satisfied only if all of the child expressions are satisfied. • or - the resulting expression is satisfied if at least one of the child expressions is satisfied.

Control (UI)	Default Value	Function
ops > Add ▾		
or		
once an op has been added, [Op]		
contains	N/A	Evaluates if the condition is true by testing if the parameter or page values contain the values set in the expression.
doesNotContain	N/A	Evaluates if the condition is true by testing if the parameter or page values do not contain the values set in the expression.
endsWith	N/A	Evaluates if the condition is true by testing if the parameter or page values end with the values set in the expression.
equalTo	N/A	Evaluates if the condition is true by testing if the parameter or page values are equal to the values set in the expression.
greaterThan	N/A	Evaluates if the condition is true by testing if the parameter or page values are greater than the values set in the expression.
greaterThanOrEqualTo	N/A	Evaluates if the condition is true by testing if the parameter or page values are greater than or equal to the values set in the expression.
in	N/A	Evaluates if the condition is true by testing if the parameter or page values are in the values (separated by a pipe with no spaces) set in the expression.
lessThan	N/A	Evaluates if the condition is true by testing if the parameter or page values are less than the values set in the expression.
lessThanOrEqualTo	N/A	Evaluates if the condition is true by testing if the parameter or page values are less than or equal to the values set in the expression.

Control (UI)	Default Value	Function
notEqualTo	N/A	Evaluates if the condition is true by testing if the parameter or page values are not equal to the values set in the expression.
notIn	N/A	Evaluates if the condition is true by testing if the parameter or page values are not in the values (separated by a pipe with no spaces) set in the expression.
numChildrenEqualTo	N/A	Evaluates if the condition is true by testing if the number of children in the target group parameter is equal to the number of children specified in the parameter or page.
numChildrenGreaterThan OrEqualTo	N/A	Evaluates if the condition is true by testing if the number of children in the target group parameter is greater than or equals to the number of children specified in the parameter or page.
regex	N/A	Evaluates if the condition is true by testing if the parameter or page values match the values set in the regular expression.
and	N/A	Specifies if you want to compare the parameter or page values to another set of values. It uses all the expressions to evaluate the condition.
or	N/A	Specifies if you want to compare the parameter or page values to another set of values. It uses only one of the expressions to evaluate the condition.
Once an Op has been added:		
path	N/A	Specifies the path of the parameter or page to evaluate.
value	N/A	Specifies the values to compare the parameter or page values with, in order to evaluate if the condition is true.
When definitionStyle: conditional state expression		
expression	N/A	Specifies the expression to use to apply a visibility or a lock condition on the exposed parameter(s).

NetworkMaterialParameterEdit

The NetworkMaterialParameterEdit node allows you to edit a shading node's parameters in a non-destructive way.

Control (UI)	Default Value	Function
name	NetworkMaterialParameterEdit	Specifies the name of the location.
action	edit existing location	Determines the node's behavior: <ul style="list-style-type: none"> • edit existing location - specifies the NetworkMaterial scene graph location. • inherit from existing material location - creates a sub-location that inherits the material from the parent NetworkMaterial location.
When action is: edit existing location		
Material to Edit > location	N/A	Specifies the location. The location parameter options are available by clicking the ▼ dropdown menu. For more information, refer to the Scene Graph Location Widget Type in Common Parameter Widgets on page 13.
When action is: inherit from existing material location		
inheritsFrom > location	N/A	Specifies the location. The location parameter options are available by clicking the ▼ dropdown menu. For more information, refer to the Scene Graph Location Widget Type in Common Parameter Widgets on page 13.
nodes		
nodes > Add ▼	N/A	Allows you to add the shading nodes connected to the NetworkMaterial node.

NetworkMaterialSplice

The NetworkMaterialSplice node allows you to make changes to an existing [NetworkMaterial](#) node in a non-destructive way by connecting or inserting new shading nodes within the shading network. You can also disconnect

existing shading nodes within the shading network and rewire the [NetworkMaterial](#) node in different ways.

Control (UI)	Default Value	Function
name	NetworkMaterialSplice	Specifies the name of the node.
action	edit existing location	<p>Determines the node's behavior:</p> <ul style="list-style-type: none"> • edit existing location - makes modifications to an existing network material. • inherit from existing material location - creates a sub-location that inherits the material from the parent NetworkMaterialSplice location.
When action is: edit existing location		
edit > location	N/A	<p>Specifies the location. The location parameter options are available by clicking the ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location Widget Type in Common Parameter Widgets on page 13.</p>
When action is: inherit from existing material location		
inheritsFrom > location	N/A	<p>Specifies the location. The location parameter options are available by clicking the ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location Widget Type in Common Parameter Widgets on page 13.</p>
inputs > append		
action	connect	<p>Allows you to select how the new shading nodes connected to the append input port are to be spliced into the network material:</p> <ul style="list-style-type: none"> • connect - allows you to connect the new nodes to one of the input connections of a shading node in the original network material. • insert - allows you to insert the new nodes between two shading nodes within the shading network.
When action is: connect		
connectToNode	N/A	Specifies which shading node in the original network to connect the nodes to.

Control (UI)	Default Value	Function
connectToPort	N/A	Specifies which port on the shading node we want to connect the new nodes to.
when action is: insert		
connectToNode	N/A	Specifies the point in between two consecutive shading nodes in the original material where you want to insert the new shading nodes.
connectToPort	N/A	Specifies which port to connect to.
reconnectToNode	N/A	Specifies the new shading node that you want the original connection to be re-connected to.
reconnectToPort	N/A	Specifies which port to reconnect to.
extraConnections > Add ▾		
c > connectFromNode	N/A	Specifies the shading node in the original network that you want to create a new connection from.
c > connectFromPort	out	Specifies which output port on the connectFromNode to connect from.
c > connectToNode	N/A	Specifies the node in the original shading network that you want to add a new connection to.
c > connectToPort	N/A	Specifies which port to connect to.
disconnections >Add ▾		
d > node	N/A	Specifies which node you want to disconnect within the shading node.
d > port	N/A	Specifies which port you want to disconnect within the shading node.

OpResolve

This node resolves deferred ops, such as Attribute Modifier Plug-ins, OpScripts, and AttributeScripts.

Control (UI)	Default Value	Function
resolveWithIds	all	When an Op has been deferred to run during op resolve you must specify the resolveID. If the resolveWithIds option is set to all , it processes all the ops set to run during op resolve . If the option is set to selected , it only processes those that have at least one resolveId that matches the values set in the specifiedResolveIds field.
resolveWithIds: specified		
specifiedResolveIds	N/A	A space-separated list of resolveIds to resolve.

OpScript

A Lua-based interface to the Op API, which is both faster and more powerful than AttributeScript. For more information on the OpScript interface, see **Help > Documentation**.

Control (UI)	Default Value	Function
CEL	N/A	<p>The scene graph locations are specified using the Collection Expression Language (CEL). The CEL parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>
script	N/A	<ul style="list-style-type: none"> • Edit in <editor>... - opens an external editor, as set up in the Preferences under externalTools > Editor, for editing the OpScript node's Lua script without blocking Katana's user interface. Katana monitors the text files you are editing, and when it detects that they have changed, it updates the OpScript node accordingly. • lua - contains the lua script to run.

Control (UI)	Default Value	Function
executionMode	immediate	<p>Determines when the script is run:</p> <ul style="list-style-type: none"> • immediate - the script is run at the locations specified in the applyWhere parameter as it is evaluated at this node's point in the node graph. • deferred - the script is set up by this node but won't actually be run until a later node in the node graph, as specified by the applyWhen parameter.
Display as multi-input	Disabled	If enabled, allows multiple inputs to be connected to this OpScript node.
When executionMode is: immediate		
applyWhere	at locations matching CEL	<p>Determines where the script is run:</p> <ul style="list-style-type: none"> • at all locations - at all the locations in the node graph. • at specific location - at only the location specified by the location parameter. If this location doesn't exist, it is created automatically. • at locations matching CEL - at only those locations in the node graph that match the CEL statements.
inputBehavior	only valid	<p>Controls how input ports on the node are mapped onto the inputs of the underlying Op. This parameter is only meaningful when the node has one or more invalid input ports - a port that is not connected to an output port or is connected to an output port that doesn't provide data.</p> <p>When set to only valid, any invalid input ports of the node are skipped when determining which inputs to pass to the underlying Op.</p> <p>When set to by index, all input ports of the node are represented in the list of inputs the Op sees; invalid inputs are represented as an Op of type no-op.</p>
When applyWhere is: at specific locations		

Control (UI)	Default Value	Function
location	/root/world/location	<p>The location to create, if it doesn't already exist. Otherwise, the scene graph location at which the script is run.</p> <p>The location parameter options are available by clicking the ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location Widget Type in Common Parameter Widgets on page 13.</p>
When executionMode is: deferred		
applyWhen	during op resolve	<p>Determines when the script is run:</p> <ul style="list-style-type: none"> • during op resolve - the script and its arguments are added as attributes to be executed later by an OpResolve node. If the Op isn't run by an explicit OpResolve node placed in the node graph, it is automatically run at render time by the implicit resolvers. • during material resolve - the script and its arguments are added as attributes under the material.ops group attribute. This is primarily intended for material scene graph locations, allowing the material to specify a procedural process that is run at every location that the material is assigned to. The script is run as part of the material resolve process, and is executed just after the initial values for the material shader are created at the location. Examples of its use include randomizing or procedural control over shader parameters. • during katana look file resolve - the script and its arguments are added as attributes under the ops group attribute and are evaluated by a LookFileResolve node or by the first implicit resolver if no LookFileResolve node is present.
modifierNameMode	node name	<p>Deferred OpScripts are added as group attributes within the ops group. By default, the name of the node is used for the sub-group. Since node names must be unique in project, the resulting attribute name can change. In nearly all cases, that doesn't matter. For cases in which it does, you can specify a fixed name to use.</p>
When modifierNameMode is: specified		

Control (UI)	Default Value	Function
modifierName	modifier	Sets the name of the attribute group beneath ops to use for describing this deferred script.
When executionMode is: deferred, during op resolve		
resolvelds	N/A	<p>Specify a space-delimited list of strings to indicate that this script should only be resolved by Op resolvers which contain at least one matching "resolveld." This is an advanced feature for greater control over order of evaluation.</p> <p>A useful resolvelds is <code>implicit_preprocess</code>, which runs at the first implicit resolver, before other implicit resolvers, such as <code>MaterialResolve</code> and <code>ConstraintResolve</code> are run.</p>
recursiveEnable	No	<p>When applying in a non-immediate state, enabling this results in the script running at every location beneath the assigned locations. In general this is more efficient than using an equivalent recursive CEL statement.</p> <p>You can also override the ops.*.recursiveEnable attribute anywhere deeper in the tree to exclude evaluation at those locations. This is similar to the behavior of the <code>visible</code> or <code>light linking</code> attributes.</p>
When recursiveEnable is: yes		
disableAt	N/A	<p>Execution is disabled for locations at or below this CEL statement. For large scene hierarchies, this is often less expensive than enabling evaluation at a larger number of leaf locations to avoid applying it to a smaller subset.</p> <p>The scene graph locations are specified for the disableAt parameter options by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>
When executionMode is: deferred, during Katana look file resolve		

Control (UI)	Default Value	Function
recursiveEnable	No	<p>When applying in a non-immediate state, enabling this results in the script running at every location beneath the assigned locations. In general this is more efficient than using an equivalent recursive CEL statement.</p> <p>You can also override the ops.*.recursiveEnable attribute anywhere deeper in the tree to exclude evaluation at those locations. This is similar to the behavior of the visible or light linking attributes.</p>
When recursiveEnable is: Yes		
disableAt	N/A	<p>Execution is disabled for locations at or below this CEL statement. For large scene hierarchies, this is often less expensive than enabling evaluation at a larger number of leaf locations to avoid applying it to a smaller subset.</p> <p>The scene graph locations are specified for the disableAt parameter options by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>

OrientConstraint

OrientConstraint matches the rotation (orientation) of the object in **basePath** to the object in **targetPath**. See also [ParentChildConstraint](#) on page 245 and [PointConstraint](#) on page 245.

Control (UI)	Default Value	Function
basePath	none	<p>Sets the location of the object to constrain. The basePath parameter options are available by clicking the ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location Widget Type in Common Parameter Widgets on page 13.</p>

Control (UI)	Default Value	Function
targetPath	none	<p>Sets the location of the object(s) to constrain the object in basePath to. The targetPath parameter options are available by clicking the ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location Widget Type in Common Parameter Widgets on page 13.</p>
targetOrientation	Object	<p>Sets the type of bounds to use for the target object(s):</p> <ul style="list-style-type: none"> • Object • Face
When targetOrientation: Face		
targetFaceIndex	0	
OrientConstraint parameters continued		
xAxis	Yes	Constrains the x Axis.
yAxis	Yes	Constrains the y Axis.
zAxis	Yes	Constrains the z Axis.
allowMissingTargets	No	<p>When set to Yes, silently ignore the constraint if its target is not in the scene graph.</p> <p>When set to No, produce an error on constraint resolution if the target is missing.</p>
addToConstraintList	No	<p>Adds base path to globals.constraintList at /root/world.</p> <p>This is only needed for cases in which one constraint depends on another constraint already being evaluated. The globals.constraintList is used to specify the order of evaluation of constraints.</p>
setRelativeTargets	No	<p>Stores target paths in the scene graph constraint definition as paths relative to the base path.</p> <p>Targets should still be specified as absolute paths in this node's parameters.</p>

ParentChildConstraint

Constrains the translate, rotate, and scale values of one object (the parent) to another (the child). See also [OrientConstraint](#) on page 243 and [PointConstraint](#) below.

Control (UI)	Default Value	Function
basePath	None	<p>Defines the child object. The location parameter options are available by clicking the ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location Widget Type in Common Parameter Widgets on page 13.</p>
targetPath	None	<p>Defines the parent object. The location parameter options are available by clicking the ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location Widget Type in Common Parameter Widgets on page 13.</p>
addToConstraintList	No	<p>Adds base path to globals.constraintList at /root/world.</p> <p>This is only needed for cases in which one constraint depends on another constraint already being evaluated. The globals.constraintList is used to specify the order of evaluation of constraints.</p>
setRelativeTargets	No	<p>Stores target paths in the scene graph constraint definition as paths relative to the base path.</p> <p>Targets should still be specified as absolute paths in this node's parameters.</p>

PointConstraint

Applies a constraint that translates the base object to a point defined by the target object(s). See also [OrientConstraint](#) on page 243 and [ParentChildConstraint](#) above.

Control (UI)	Default Value	Function
basePath	None	<p>Defines the location of the object to constrain. The location parameter options are available by clicking the ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location Widget Type in Common Parameter Widgets on page 13.</p>
targetPath	None	<p>Defines the location of the object(s) to constrain the object in basePath to. If you set multiple targets, then the constraint moves to the average center of the objects. The targetPath parameter options are available by clicking Add Locations or ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location and Locations Widget Types in Common Parameter Widgets on page 13.</p>
baseOrigin	Object	<p>Sets how the center of the base object is calculated:</p> <ul style="list-style-type: none"> • Object - uses the local origin of the object as the position of the base object. • Bounding Box - uses the center of the object's bounding box as the position of the base object.
targetOrigin	Object	<p>Sets how the center of the target object is calculated:</p> <ul style="list-style-type: none"> • Object - uses the local origin of the object as the target. • Bounding Box - uses the center of the object's bounding box as the target. • Face Center Average - uses the face center average of the object as the target. • Face Bounding Box - uses the face center average of the object's bounding box as the target.
allowMissingTargets	No	<p>When set to Yes, silently ignore the constraint if its target is not in the scene graph.</p> <p>When set to No, produce an error on constraint resolution if the target is missing.</p>
xAxis	disabled	Constrains the x Axis.

Control (UI)	Default Value	Function
yAxis	disabled	Constrains the y Axis.
zAxis	disabled	Constrains the z Axis.
addToConstraintList	No	<p>Adds base path to globals.constraintList at /root/world.</p> <p>This is only needed for cases in which one constraint depends on another constraint already being evaluated. The globals.constraintList is used to specify the order of evaluation of constraints.</p>
setRelativeTargets	No	<p>Stores target paths in the scene graph constraint definition as paths relative to the base path.</p> <p>Targets should still be specified as absolute paths in this node's parameters.</p>

PrimitiveCreate

Adds a primitive geometry element to a scene such as sphere, cube, or cylinder as well as renderer procedural, rib archive, brickmap and clipping plane.

Control (UI)	Default Value	Function
name	/root/world/geo/primitive	<p>Describes the scene graph location where the object is created. The name parameter options are available in either the scene graph widget or ▼ dropdown menu to the right of the parameter.</p> <p>For more information, refer to the Create Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.</p>
type	sphere	Sets the type of primitive created (plane or sphere, for example)
transform		

Control (UI)	Default Value	Function
transform	N/A	Transforms the primitive according to the SRT or matrix controls. For more information, refer to the Transform Controls Widget Type in the Common Parameter Widgets on page 13.
transform > Tools ▾	N/A	Adjusts the primitive to match selected scene graph selection options in the dropdown menu. For more information, refer to the Transform Tools Widget Type in the Common Parameter Widgets on page 13.
PrimitiveCreate parameters continued		
makeInteractive	Yes	When set to Yes , you can drag objects in the Viewer and Katana retains the information from the Viewer.
viewerPickable	Yes	When set to Yes , the object can be selected in the Viewer. When set to No , the object can only be selected through the scene graph.

PrmanGlobalSettings

This is for changing anything that broadly comes under the heading of RenderMan options.



NOTE: The parameters that are available for this node are dependent on which version of PRMan you are using and whether you are using the PRMan plug-in that ships with Katana or RenderMan Studio for Katana. As such, only the renderer-agnostic parameters are listed below. For more information on some of the other parameters you may encounter, please refer to the documentation that ships with RenderMan.

PrmanObjectSettings

The purpose of this node is to set PRMan attributes at levels of the scene graph hierarchy described by the given CEL statement.



NOTE: The parameters that are available for this node are dependent on which version of PRMan you are using and whether you are using the PRMan plug-in that ships with Katana or RenderMan Studio for Katana. As such, only the renderer-agnostic parameters are listed below. For more information on some of the other parameters you may encounter, please refer to the documentation that ships with RenderMan.

Control (UI)	Default Value	Function
CEL	none	<p>The scene graph locations are specified using the Collection Expression Language (CEL). The CEL parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>

PrmanOutputChannelDefine

The PrmanOutputChannelDefine allows you to define custom output channels, so that the final render can be split into separate elements. These custom channels (AOVs) can then be manipulated in a compositing tool.



NOTE: The parameters that are available for this node are dependent on which version of PRMan you are using and whether you are using the PRMan plug-in that ships with Katana or RenderMan Studio for Katana. As such, only the renderer-agnostic parameters are listed below. For more information on some of the other parameters you may encounter, please refer to the documentation that ships with RenderMan.

Control (UI)	Default Value	Function
name	none	The name used by the RenderOutputDefine node for these output channel settings. This parameter usually matches the channel name.
type	varying float	<p>Sets the output channel type:</p> <p>BYTE, INT, LONG, BOOL, FLOAT, DOUBLE, RGB, RGBA, ABSRGB, VECTOR, POINT, POINT2, STRING, POINTER, ARRAY, MATRIX, and ENUM</p>

PrmanShadingNode

The PrmanShadingNode allows you to select a RenderMan-specific shader to build complex shading networks. The last shading node in the shading network needs to be connected to a [NetworkMaterial](#) node in order to be connected to the 3D node graph and assigned to objects in the scene.



NOTE: The parameters that are available for this node are dependent on which version of PRMan you are using and whether you are using the PRMan plug-in that ships with Katana or RenderMan Studio for Katana. As such, only the renderer-agnostic parameters are listed below. For more information on some of the other parameters you may encounter, please refer to the documentation that ships with RenderMan.

Control (UI)	Default Value	Function
name	PrmanShadingNode	Determines the attribute identifier for this shader node beneath the 'material' attribute. This must be unique among all upstream nodes connected into a single NetworkMaterial node.
nodeType	none	Selects the available shader from the dropdown list. The parameters for each shader in the dropdown list are not included, as they are renderer-specific. Use the file browser or your studio's asset management browser to select the shader to use.
parameters	N/A	Once you've added a shader, the shader's parameters are populated under the Parameter group.
publicInterface		
namePrefix	none	Specifies the name's prefix for the exposed parameter.
pagePrefix	none	Allows you to organize the shading node's exposed parameters in groups in the NetworkMaterial node's Material Interface.
nameRegExFind	none	Finds and deletes the name specified in namePrefix field.
nameRegExReplace	none	When used with nameRegExFind , finds and replaces the name with the name specified by nameRegExReplace .

Control (UI)	Default Value	Function
pageRegExFind	none	Finds and deletes the name specified in namePrefix field.
pageRegExReplace	none	When used with pageRegExFind , finds and replaces the name with the name specified by pageRegExReplace .
PrmanShadingNode parameters continued		
Force Refresh	N/A	Reloads the shader file's information.

Prune

The Prune node removes objects from a scene. Any location that matches the given CEL statement is removed from the output. Any parent location that matches the CEL statement also has all children removed from the output, so there's no need to match all the children if you're pruning out an entire tree of locations. See also [Isolate](#) on page 199 and [VisibilityAssign](#) on page 322.

Notes:

- To prune out all polymesh objects, use a 'Custom' type statement that looks like this:
`//*{@type=="polymesh"}` Change polymesh to whatever type you're interested in to remove that type.
- You don't need to prune out an object to prevent it from being used in a render. The [VisibilityAssign](#) node is another way of removing objects from the render without actually removing the object from the scene.
- To see what the Prune is removing, view the node above the prune then click the little arrow on the Prune node's **cel** parameter and select **Find and Select in Scenegraph....**

After processing for a while, all objects that are to be pruned become selected in the **Scene Graph** tab.

If nothing is selected, then nothing matches the CEL statement and nothing is pruned.

Control (UI)	Default Value	Function
cel	none	<p>The CEL statement to use to select locations to remove.</p> <p>The scene graph locations are specified using the Collection Expression Language (CEL). The cel parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>

ReflectionConstraint

ReflectionConstraint transforms the base object to a mirrored position opposite the target plane object.

Control (UI)	Default Value	Function
basePath	none	<p>Sets the object to constrain. The materialAssign parameter options are available by clicking the ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location Widget Type in Common Parameter Widgets on page 13.</p>
targetPath	none	<p>Sets the object(s) to which the object in basePath is constrained. The materialAssign parameter options are available by clicking the ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location Widget Type in Common Parameter Widgets on page 13.</p>
targetFaceIndex	0	The face number that forms the basis for the reflection transform
addToConstraintList	No	<p>Adds base path to globals.constraintList at /root/world.</p> <p>This is only needed for cases in which one constraint depends on another constraint already being evaluated. The globals.constraintList is used to specify the order of evaluation of constraints.</p>
setRelativeTargets	No	<p>Stores target paths in the scene graph constraint definition as paths relative to the base path.</p> <p>Targets should still be specified as absolute paths in this node's parameters.</p>

Rename

This node is useful for renaming scene graph locations according to regular expression matching and substitution. Be aware that many operations are dependent on the names of scene graph locations. Use this with care as it's possible to invalidate subsequent operations by changing scene graph location names.

Control (UI)	Default Value	Function
rootLocation	/root/world/geo	Describes the top-most location on which to perform renaming. The rename parameter options are available by clicking the  dropdown menu. For more information, refer to the Scene Graph Location Widget Type in Common Parameter Widgets on page 13.
locationTypes	none	Accepts a comma-delimited list of scene graph location types on which to act. An empty list acts upon all types.
pattern	none	Defines a POSIX-style regular expression on which to match.
replace	none	Sets the string replacement. \1 through \9 expand to matched groups in the above pattern. \0 expands to the full match string.

Render

The Render node takes a scene as input and renders images. The first input on this node is the scene to render. Additional inputs are dependency connections, which are used to track dependencies between passes. Each Render node is intended to be a single invocation of RenderMan or another renderer.

The Render node is really only used to track render settings, asset names, and which previously defined output passes are to be used. To set up passes, use `RenderOutputDefine`. To change render settings (like the active camera) use `RenderSettings`. To change RenderMan global settings (like `pixelSamples`), use `PrmanGlobalSettings`.

Input Information:

- Don't delete the port 'input' on the render node, or the node becomes unusable.
- Additional inputs to the Render node are dependency inputs and are only used when generating outline files for rendering.

Control (UI)	Default Value	Function
passName	Render	<p>Sets the passName to identify this render node and is used to build the name of assets written from the Render node.</p> <p>When the passName is changed, the name of the Render node is also updated to stay in sync with the pass name. This is a parameter rather than just using the node name itself so you can have more control over this; node names must be unique within a Node Graph, while passName can be duplicated among different Render nodes if you need to for some reason.</p>
lock	disabled	<p>When enabled, the asset information for this Render node is no longer updated.</p> <p>This is useful when you're sharing a Render node between shots and want to use expressions to reference the original output of the Render node. A locked Render node cannot be used to HotRender because the asset it produces is locked. It can be referenced in expressions with 'getRenderLocation'.</p>
outputs		
outputs	none	Manages which available outputs are active.
farmGlobalSettings		
setActiveFrameRange	disabled	<p>When enabled, activeFrameRange parameters are exposed to define the active frame range for rendering.</p> <p>When disabled, the active frame range is assumed to be the same as globals.inTime and globals.outTime.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  NOTE: The active frame range affects farm file generation and ShadowRenderAssign node assignments. </div>

Control (UI)	Default Value	Function
setActiveFrameRange: enabled		
activeFrameRange > start	1	When setActiveFrameRange is enabled, sets the start of the active frame range.
activeFrameRange > end	1	When setActiveFrameRange is enabled, sets the end of the active frame range.
Render parameters continued		
dependAll	disabled	When enabled, farm dependencies wait until all frames of this node are rendered before rendering themselves.
farmFileName	none	
renderInternalDependencies	disabled	When enabled, internal dependencies of this node (input Render nodes that don't have any external (shottree) outputs of their own) are rendered in the same farm process as this node.
excludeFromFarmOutputGeneration	disabled	When enabled, this node does not appear in any generated farm file (however, the node is still renderable if called directly).
forceFarmOutputGeneration	disabled	When enabled, this node always appears in a generated farm file (regardless of whether it has any valid outputs).  NOTE: If excludeFromFarmOutputGeneration is also set, the node does not appear in the generated farm file (excludeFromFarmOutputGeneration overrides forceFarmOutputGeneration).
threadable	enabled	

RendererProceduralArgs

The `RendererProceduralArgs` node allows you to use and declare renderer procedurals, such as native Arnold or RenderMan procedurals. For example you can use procedurals to generate hair or other geometry. Procedurals are assigned to scene graph locations of type 'renderer procedural' with the [RendererProceduralAssign](#) node.

Control (UI)	Default Value	Function
name	RendererProceduralArgs	Sets the name of the procedural.
action	create new location	<p>Determines what action the node takes:</p> <ul style="list-style-type: none"> • create new location - creates a new scene graph location of type "renderer procedural arguments" beneath /root/materials/proc with the name specified by the name parameter. • inherit from existing location - creates a new scene graph location of type "renderer procedural arguments" beneath the location specified by inheritsFrom.location parameter with the name specified by the name parameter. • edit existing location - displays the incoming values of a single scene graph location of type "renderer procedural arguments" or "renderer procedural" specified by the edit.location parameter - applies the specified edits to the values at that location. • edit multiple locations - edits values at the locations specified by the CEL statement from the edit.CEL parameter. This does not display incoming values because they could differ from location to location. This means that you must specify the dso path for the procedural in order to display settable parameters.

Control (UI)	Default Value	Function
action	(continued)	<ul style="list-style-type: none"> • define overrides - accepts drag-and-dropped attributes from <code>rendererProcedural</code> attribute groups. This can be used in two ways: <ol style="list-style-type: none"> 1. When aimed (via CEL) at locations within the renderable scene, it creates a rendererProceduralOverride attribute. At resolve time, these values override equivalent values in the rendererProcedural attribute of scene graph locations of type "RendermanProcedural" beneath. This is useful for making global changes to many different procedurals at once, regardless of whether they share the same source. 2. When aimed (via CEL) at locations of type "renderer procedural arguments", it modifies the <code>rendererProcedural</code> directly in the same manner as the "edit multiple locations" action. • remove overrides - removes or masks inherited attributes in the rendererProceduralOverride at the scene graph locations specified by the CEL statement of the <code>overrides.CEL</code> parameter.
When action is: create new location		
namespace	N/A	Specifies the scene graph location under /root/world/proc where to place the procedural.
procedural	N/A	Brings up the file browser or your studio's asset management browser and enables you to select the procedural to use. For more information, refer to the Asset and File Path Widget Types in the Common Parameter Widgets on page 13.
args	N/A	Specifies the args file attributes.
useInfiniteBounds	Yes	Controls whether the declaration of the procedural includes bounds that the renderer can use to defer evaluation of the procedural until it needs data from inside those bounds. When set to Yes , nearly infinite bounds are used. When set to No , no bounds are used.

Control (UI)	Default Value	Function
includeCameraInfo	None	<p>Specifies the format in which the camera information is passed to the procedural by the render plugin.</p> <p>It includes the following: camera's transform, field of view, and screen window.</p> <ul style="list-style-type: none"> • None - no camera information is passed to the procedural. • As Parameters - the camera information is passed into the procedural as arguments to the procedural. So the procedural you're specifying args for needs to be expecting to receive this camera information, and know what to do with it. • As Attributes - the camera information is put into the RIB stream as a "user" attribute. So the procedural needs to know to look for this attribute if it wants to use it. <p>The Parameters and Attributes formats are the following:</p> <ul style="list-style-type: none"> • string cameraInfo_path - path to the camera. • float cameraInfo_fov - the Field of View. • float cameraInfo_near - distance to the near clipping plane. • float cameraInfo_far - distance to the far clipping plane. • float cameraInfo_left - screen window left. • float cameraInfo_right - screen window right. • float cameraInfo_top - screen window top. • float cameraInfo_bottom - screen window bottom. • float[16] cameraInfo_xform - the camera transform.
When includeCameraInfo is: As Parameters or As Attributes		
cameraInfo > whichCamera	Render Camera	<ul style="list-style-type: none"> • Render Camera - selects the render camera. • Other Camera - lets you set the path for the camera using cameraPath
When action is: inherit from existing location		

Control (UI)	Default Value	Function
InheritsFrom > location	N/A	<p>Specifies the location to use in the Scene Graph tab. The location parameter options are available by clicking the  dropdown menu.</p> <p>For more information, refer to the Scene Graph Location Widget Type in Common Parameter Widgets on page 13.</p>
procedural	N/A	<p>Brings up the file browser or your studio's asset management browser and enables you to select the procedural to use.</p> <p>For more information, refer to the Asset and File Path Widget Types in the Common Parameter Widgets on page 13.</p>
args	N/A	Specifies the args file attributes.
useInfiniteBounds	Yes	<p>Controls whether the declaration of the procedural includes bounds that the renderer can use to defer evaluation of the procedural until it needs data from inside those bounds.</p> <p>When set to Yes, nearly infinite bounds are used. When set to No, no bounds are used.</p>

Control (UI)	Default Value	Function
includeCameraInfo	None	<p>Specifies the format in which the camera information is passed to the procedural by the render plugin.</p> <p>It includes the following: camera's transform, field of view, and screen window.</p> <ul style="list-style-type: none"> • None - no camera information is passed to the procedural. • As Parameters - the camera information is passed into the procedural as arguments to the procedural. So the procedural you're specifying args for needs to be expecting to receive this camera information, and know what to do with it. • As Attributes - the camera information is put into the RIB stream as a "user" attribute. So the procedural needs to know to look for this attribute if it wants to use it. <p>The Parameters and Attributes formats are the following:</p> <ul style="list-style-type: none"> • string cameraInfo_path - path to the camera. • float cameraInfo_fov - the Field of View. • float cameraInfo_near - distance to the near clipping plane. • float cameraInfo_far - distance to the far clipping plane. • float cameraInfo_left - screen window left. • float cameraInfo_right - screen window right. • float cameraInfo_top - screen window top. • float cameraInfo_bottom - screen window bottom. • float[16] cameraInfo_xform - the camera transform.
When includeCameraInfo is: As Parameters or As Attributes		
cameraInfo > whichCamera	Render Camera	<ul style="list-style-type: none"> • Render Camera - selects the render camera. • Other Camera - lets you set the path for the camera using cameraPath
When action is: edit existing location		

Control (UI)	Default Value	Function
edit > location	N/A	<p>Specifies the location to use in the Scene Graph tab. The location parameter options are available by clicking the  dropdown menu.</p> <p>For more information, refer to the Scene Graph Location Widget Type in Common Parameter Widgets on page 13.</p>
procedural	N/A	<p>Brings up the file browser or your studio's asset management browser and enables you to select the procedural to use.</p> <p>For more information, refer to Asset and File Path Widget Types in Common Parameter Widgets on page 13.</p>
args	N/A	Specifies the args file attributes.
useInfiniteBounds	Yes	<p>Controls whether the declaration of the procedural includes bounds that the renderer can use to defer evaluation of the procedural until it needs data from inside those bounds.</p> <p>When set to Yes, nearly infinite bounds are used. When set to No, no bounds are used.</p>

Control (UI)	Default Value	Function
includeCameraInfo	None	<p>Specifies the format in which the camera information is passed to the procedural by the render plugin.</p> <p>It includes the following: camera's transform, field of view, and screen window.</p> <ul style="list-style-type: none"> • None - no camera information is passed to the procedural. • As Parameters - the camera information is passed into the procedural as arguments to the procedural. So the procedural you're specifying args for needs to be expecting to receive this camera information, and know what to do with it. • As Attributes - the camera information is put into the RIB stream as a "user" attribute. So the procedural needs to know to look for this attribute if it wants to use it. <p>The Parameters and Attributes formats are the following:</p> <ul style="list-style-type: none"> • string cameraInfo_path - path to the camera. • float cameraInfo_fov - the Field of View. • float cameraInfo_near - distance to the near clipping plane. • float cameraInfo_far - distance to the far clipping plane. • float cameraInfo_left - screen window left. • float cameraInfo_right - screen window right. • float cameraInfo_top - screen window top. • float cameraInfo_bottom - screen window bottom. • float[16] cameraInfo_xform - the camera transform.
When includeCameraInfo is: As Parameters or As Attributes		
cameraInfo > whichCamera	Render Camera	<ul style="list-style-type: none"> • Render Camera - selects the render camera. • Other Camera - lets you set the path for the camera using cameraPath
When action is: edit multiple locations		

Control (UI)	Default Value	Function
edit > CEL	N/A	<p>Sets the CEL specification of scene graph locations on which the assignment acts.</p> <p>The scene graph locations are specified using the Collection Expression Language (CEL). The CEL parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>
procedural	N/A	<p>Brings up the file browser or your studio's asset management browser and enables you to select the procedural to use.</p> <p>For more information, refer to Asset and File Path Widget Types in Common Parameter Widgets on page 13.</p>
args	N/A	Specifies the args file attributes.
useInfiniteBounds	Yes	<p>Controls whether the declaration of the procedural includes bounds that the renderer can use to defer evaluation of the procedural until it needs data from inside those bounds.</p> <p>When set to Yes, nearly infinite bounds are used.</p> <p>When set to No, no bounds are used.</p>

Control (UI)	Default Value	Function
includeCameraInfo	None	<p>Specifies the format in which the camera information is passed to the procedural by the render plugin.</p> <p>It includes the following: camera's transform, field of view, and screen window.</p> <ul style="list-style-type: none"> • None - no camera information is passed to the procedural. • As Parameters - the camera information is passed into the procedural as arguments to the procedural. So the procedural you're specifying args for needs to be expecting to receive this camera information, and know what to do with it. • As Attributes - the camera information is put into the RIB stream as a "user" attribute. So the procedural needs to know to look for this attribute if it wants to use it. <p>The Parameters and Attributes formats are the following:</p> <ul style="list-style-type: none"> • string cameraInfo_path - path to the camera. • float cameraInfo_fov - the Field of View. • float cameraInfo_near - distance to the near clipping plane. • float cameraInfo_far - distance to the far clipping plane. • float cameraInfo_left - screen window left. • float cameraInfo_right - screen window right. • float cameraInfo_top - screen window top. • float cameraInfo_bottom - screen window bottom. • float[16] cameraInfo_xform - the camera transform.
When includeCameraInfo is: As Parameters or As Attributes		
cameraInfo > whichCamera	Render Camera	<ul style="list-style-type: none"> • Render Camera - selects the render camera. • Other Camera - lets you set the path for the camera using cameraPath
When action is: define overrides		

Control (UI)	Default Value	Function
overrides > CEL	N/A	<p>Sets the CEL specification of scene graph locations on which the assignment acts.</p> <p>The scene graph locations are specified using the Collection Expression Language (CEL). The CEL parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>
attrs	Drop Attributes Here	Middle-click and drag attributes from the Attributes tab to this hotspot to use that attribute.
When action is: remove overrides		
overrides > CEL	N/A	<p>Sets the CEL specification of scene graph locations on which the assignment acts.</p> <p>The scene graph locations are specified using the Collection Expression Language (CEL). The CEL parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>

RenderOutputDefine

Specifies output of an image (color, AOV, shadow map, or similar) to a file. In RIB, this means a Display statement.

Control (UI)	Default Value	Function
outputName	primary	Associates a name with the display. Typically primary by default; often shadow for shadow maps, and similar conventions. This name appears in the Render node, along with (or as) the default primary .

Control (UI)	Default Value	Function
type	color	<p>Specifies the type of output.</p> <ul style="list-style-type: none"> • color - mostly used to render out rgb beauty files, but also can be used for rendering out z, P(point), N(normals), Ci(final shader color) passes. • shadow - renders out PRMan z and deepshad(deep shadow) files. • raw - allows you to directly specify the values for a Display line. Since the output could be anything, Katana doesn't do any colorspace conversion on this output, and can't support tiling. • ptc - prman pointcloud file. The shader is responsible for generating the pointcloud data, but this output sets up a global 'renderOutputs.local.passname' attribute at /root to specify where the pointcloud should go, and also handles stitching of pointcloud files after a tile render. • script - run a script on another RenderOutputDefine, like txmake. • prescript - run a script before the render is started. • none - clears the output. If the output was previously setup by a different RenderOutputDefine node, this removes the entry.
includedByDefault	Yes	When enabled, this Render Definition is sent to the Render node.
rendererSettings		
colorSpace	Inf	Sets the output colorspace used.
fileExtension	exr	Sets the output file format.
channel	rgba	Sets the channels to output. You can also set a user-defined channel from a PrmanDisplayChannelDefine node.
When fileExtension: exr; convertSettings		

Control (UI)	Default Value	Function
exrCompression	Scanline ZIP	Defines the exr compression method to use. All methods are lossless (with the exception of Pixar24 , which is lossless but quantizes the pixels to 24-bit float). Wavelet is generally preferable as it offers ~2:1 compression even on grainy data.
exrBitDepth	16	Sets the floating point precision of the rendered exr file: <ul style="list-style-type: none"> • 16 - half float. This is recommended for all color passes. • 32 - full float. This is recommended for all ncf data arbitrary output variables (AOVs).
exrOptimize	Yes	When enabled, the exr file is written out in an a manner optimized for efficient random tile-access. These optimizations greatly improve memory usage and performance for programs which process images in tiles.
exrType	Tiled	Sets whether the exr file is written to support: <ul style="list-style-type: none"> • Tiled - random tile access. • Scanline - random scanline access.
When fileExtension: exr		
clampOutput	No	When set to Yes , post-render clamp negative rgb values to 0, and clamp alpha values to 0-1. <div data-bbox="732 1150 1495 1270" style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  NOTE: clampOutput has no effect on NaN and inf values. </div>
colorConvert	Yes	When set to Yes , post-render convert rendered image data from linear to the output colorspace specified in the filename. <p>The default value of Yes is suitable for nearly every situation, since the linear output of the render is converted to the colorspace in the filename. A case where you would want to set this to No is if you know the data being rendered is in a colorspace other than linear, such as the reprojection of a log plate, and you want to name the output file log without a linear to log conversion.</p>
When fileExtension: png; convertSettings		

Control (UI)	Default Value	Function
pngBitDepth	16	Sets the bit depth of the rendered file: <ul style="list-style-type: none"> • 8-bit • 16-bit
When fileExtension: rla; convertSettings		
rlaBitDepth	16	Sets the bit depth of the rendered file: <ul style="list-style-type: none"> • 8-bit • 10-bit • 16-bit • 32-bit
When fileExtension: tif; convertSettings		
tifBitDepth	16	The bit depth of the rendered file: <ul style="list-style-type: none"> • 8-bit • 16-bit • 32-bit
tifCompression	LZW	The tif compression method to use: <ul style="list-style-type: none"> • None - No compression method is used. • LZW - The LZW compression method is used. This is lossless, so it is usually preferable to use it unless there is an issue with compatibility in the target reader.
When fileExtension: tif		
clampOutput	No	When set to Yes , post-render clamp negative rgb values to 0, and clamp alpha values to 0-1. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  <p>NOTE: clampOutput has no effect on NaN and inf values.</p> </div>

Control (UI)	Default Value	Function
colorConvert	Yes	<p>When set to Yes, post-render convert rendered image data from linear to the output colorspace specified in the filename.</p> <p>The default value of Yes is suitable for nearly every situation, since the linear output of the render is converted to the colorspace in the filename. A case where you would want to set this to No is if you know the data being rendered is in a colorspace other than linear, such as the reprojection of a log plate, and you want to name the output file log without a linear to log conversion.</p>
When fileExtension: jpg		
jpgQuality	100	The quality to use when generating the jpg file. Higher values generate larger file sizes, with 100 representing the best quality image and 0 representing the lowest.
rendererSettings parameters continued		
computeStats	None	<p>Allows you to compute image statistics as a post process, appending as exr metadata. Select:</p> <ul style="list-style-type: none"> • None • Raw • Depth <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  <p>NOTE: In depth mode, zero values and very large values are ignored. In both modes, only the region within the dataWindow is considered.</p> </div>
tempRenderLocation	none	
cameraName	none	<p>Describes the scene graph location of camera to render from. If empty, render from the camera specified in rendererSettings.cameraName at /root. The cameraName parameter options are available by clicking the ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location Widget Type in Common Parameter Widgets on page 13.</p>

Control (UI)	Default Value	Function
locationType	local	
When locationType: file; locationSettings		
renderLocation		<p>Specify the render location, or bring up the file browser or your studio's asset management browser to select the location to use.</p> <p>For more information, refer to the Asset and File Path Widget Types in the Common Parameter Widgets on page 13.</p>



NOTE: RenderOutputLocation plug-ins that are shipped as source and can be found in `plugins/Src/RenderOutputLocations`.

ReverseNormals

ReverseNormals reverses any point and vertex normals on locations matching its CEL parameter. Point normals are represented by a **geometry.point.N** attribute, and vertex normals are represented by a **geometry.vertex.N** attribute. If neither of these attributes are present, the node has no effect. If they are both present, they are both reversed. Any other normal attributes, such as surface normals, are not recognized or modified by the node.

Control (UI)	Default Value	Function
celSelection	N/A	<p>Sets the normals location to be reversed.</p> <p>The scene graph locations are specified using the Collection Expression Language (CEL). The celSelection parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>

ScaleConstraint

This node constrains the base object to the scale of the target object.

Control (UI)	Default Value	Function
basePath	None	<p>Sets the object to constrain. The basePath parameter options are available by clicking the ▼ dropdown menu.</p> <p>For more information, refer to the Path Selection Widget Types in Common Parameter Widgets on page 13.</p>
targetPath	None	<p>Sets the object(s) to which the object in basePath is constrained. The targetPath parameter options are available by clicking the ▼ dropdown menu.</p> <p>For more information, refer to the Path Selection Widget Types in Common Parameter Widgets on page 13.</p>
addToConstraintList	No	<p>Adds base path to globals.constraintList at /root/world.</p> <p>This is only needed for cases in which one constraint depends on another constraint already being evaluated. The globals.constraintList is used to specify the order of evaluation of constraints.</p>
setRelativeTargets	No	<p>Stores target paths in the scene graph constraint definition as paths relative to the base path.</p> <p>Targets should still be specified as absolute paths in this node's parameters.</p>

ScenegraphGeneratorResolve

This node resolves external procedurals specified by upstream ScenegraphGeneratorSetup nodes.

ScenegraphGeneratorSetup

The ScenegraphGeneratorSetup node is used to put the arguments needed for the external procedure into the scene graph. The procedure itself is not executed until the recipe reaches an associated ScenegraphGeneratorResolve node.

Control (UI)	Default Value	Function
name	/root/world/geo/scenegraphGenerator	Describes the scene graph location where the object is created. The name parameter options are available in either the scene graph widget or ▼ dropdown menu to the right of the parameter. For more information, refer to the Create Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.
generatorType	SphereMaker	
resolvelds	none	
generatorType: Alembic_In; args		
fileName	none	
fps	24	
addForceExpand	1	
pathFromRoot	mone	
addBounds	children	
generatorType: CubeMaker; args		
numberOfCubes	20	
rotateCubes		
generatorType: Obj_In; args		
fileName		
generatorType: PointCloudToLocations; args		
maxChildren	0	
numLevels	0	
locationType	locator	
generatorType: ScenegraphXml; args		

Control (UI)	Default Value	Function
filepath	none	
generatorType: SpeedTree Raw Mesh; args		
filename	none	Describes the scene graph location where the object is created. For more information, refer to the Asset and File Path Widget Types in the Common Parameter Widgets on page 13.
geometries	enabled	
bones	disabled	
spines	disabled	
generatorType: SphereMaker; args		
numSpheres	20	
testPoint	1.0, 1.0, 1.0	
testString	none	
generatorType: SphereMakerMaker; args		
numSphereMakers	2	

ScenegraphXml_In

XML is a simple way to describe a hierarchical structure. Katana leverages this format to provide a rich descriptive asset language. Through XML, assets can be structured so they can be loaded and manipulated in stages. Simpler parts of the asset can load quicker and use less memory, only loading the full asset when absolutely necessary.



NOTE: ScenegraphXML is provided as a reference example only. It is not intended for production use, and is not supported.

Some asset elements that can be described within a ScenegraphXml file are:

- Assembly locations
- Component locations
- Level-of-detail group locations

- Level-of-detail locations
- Other XML locations

Control (UI)	Default Value	Function
name	/root/world/geo	Describes where the scene is imported in the scene graph. The name parameter options are available in either the scene graph widget or ▼ dropdown menu to the right of the parameter. For more information, refer to the Create Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.
asset	none	Describes the file path and name of the XML scene file. For more information, refer to the Asset and File Path Widget Types in the Common Parameter Widgets on page 13.
timing		
mode	Current Frame	Sets the timing mode to apply to the asset: <ul style="list-style-type: none"> • Current Frame • Hold Frame • Clamp Range
When mode: Hold Frame		
holdTime	1	
When mode: Clamp Range		
inTime	1	
outTime	100	

ScreenCoordinateConstraint

ScreenCoordinateConstraint modifies the camera screen window to fit the target object(s).

Control (UI)	Default Value	Function
basePath	None	<p>Sets the object to constrain. The basePath parameter options are available by clicking the ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location Widget Type in Common Parameter Widgets on page 13.</p>
targetPath	None	<p>Sets the object(s) to which the object in basePath is constrained. The targetPath parameter options are available by clicking Add Locations or ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location and Locations Widget Types in Common Parameter Widgets on page 13.</p>
respectMotionBlur	Yes	<p>When set to Yes, constraints are adjusted to allow for the target's motion within the time the shutter is open.</p>
keepAspectRatio	Yes	<p>When set to Yes, maintain the aspect ratio of the screen window.</p> <p>When set to No, modify the aspect ratio of the screen window to fill as much of the frame as possible.</p>
allowMissingTargets	No	<p>When set to Yes, silently ignore the constraint if its target is not in the scene graph.</p> <p>When set to No, produce an error on constraint resolution if the target is missing.</p>
addToConstraintList	No	<p>Adds base path to globals.constraintList at /root/world.</p> <p>This is only needed for cases in which one constraint depends on another constraint already being evaluated. The globals.constraintList is used to specify the order of evaluation of constraints.</p>
setRelativeTargets	No	<p>Stores target paths in the scene graph constraint definition as paths relative to the base path.</p> <p>Targets should still be specified as absolute paths in this node's parameters.</p>

ShadingNodeArrayConnector

The ShadingNodeArrayConnector allows you to collect the connected shading nodes' outputs and build an input connection for shading node array parameters. The ShadingNodeArrayConnector node only works with shading nodes, which define the parameter as an array.

ShadingNodeSubnet

The ShadingNodeSubnet node allows you to group shading nodes together for a better organization of your shading network. You can then connect the ShadingNodeSubnet node to the NetworkMaterial node. To do the inverse (explode the group of shading nodes), select the ShadingNodeSubnet node and press **U**.

Control (UI)	Default Value	Function
Subnet Material Interface  or right-click interface table		
Remove	N/A	Removes the selected shading node parameter from the Subnet Material Interface.
Refresh	N/A	Refreshes the Subnet Material Interface.
Add Exposed Parameters	N/A	Adds the exposed parameters of all of the shading nodes contained within this ShadingNodeSubnet to the Subnet Material Interface.

ShadowBranch

This node generates shadows from a pass (such as a beauty pass) as well as:

1. Pruning all light materials and optionally all surface materials from the scene, even if the materials have already been resolved.
2. Providing quick access to some PRMan options that you generally want to have set in a shadow render.



NOTE: More information on details of settings picked up in the parameters can be found in the PrmanGlobalSettings documentation.

Control (UI)	Default Value	Function
pruneMaterials	lights	<p>Sets the objects to prune:</p> <ul style="list-style-type: none"> • all - prunes all materials. • lights - typically used to prune specific lights. • lights and surfaces - prunes these, but leaves displacement shaders. • none - no pruning occurs.
defineOutputs	primary shadow	<p>Acts like a render output device with override capabilities. Select:</p> <ul style="list-style-type: none"> • no override • primary shadow • primary deepshad • primary shadow, secondary deepshad • primary color, secondary deepshad.
resolution	512x512	Sets the resolution of the image(s).
pixelSamples	1.0, 1.0	Sets the sub-pixel sampling rate (X x Y samples)
zthreshold	0.9960, 0.9960, 0.9960	<p>Sets the minimum opacity for objects to appear in shadow maps.</p> <p>For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.</p>
deepshadowerror	0.01	Sets the allowable error in deep shadow generation, where 0 equals loss-less.
depthfilter	min	<p>Sets the filter to use on depth (Z) values:</p> <ul style="list-style-type: none"> • min • max • average • midpoint - calculates the depth as the midpoint between the object that is closest to the viewpoint and the second closest object.

TeapotCreate

This node creates a specific type of PrimitiveCreate node rendering a teapot instead of a sphere or cube. See also [PrimitiveCreate](#) on page 247.

Control (UI)	Default Value	Function
name	/root/world/geo/primitive	<p>Describes the scene graph location where the object is created. The name parameter options are available in either the scene graph widget or ▼ dropdown menu to the right of the parameter.</p> <p>For more information, refer to the Create Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.</p>
type	teapot	Sets the type of primitive created (plane or sphere, for example)
transform		
transform	N/A	<p>Transforms the teapot according to the SRT or matrix controls.</p> <p>For more information, refer to the Transform Controls Widget Type in the Common Parameter Widgets on page 13.</p>
transform > Tools ▼	N/A	<p>Adjusts the teapot to match selected scene graph selection options in the dropdown menu.</p> <p>For more information, refer to the Transform Tools Widget Type in the Common Parameter Widgets on page 13.</p>
TeapotCreate parameters continued		
makeInteractive	Yes	When set to Yes , you can drag objects in the Viewer and Katana retains the information from the Viewer.
viewerPickable	Yes	<p>When set to Yes, the object can be selected in the Viewer.</p> <p>When set to No, the object can only be selected through the scene graph.</p>

Transform3D

Adds transform attributes to scene graph locations allowing you to control 3D objects in the Viewer.



NOTE: Manipulates the xform attribute and is used by the AttributeEditor node.

Control (UI)	Default Value	Function
path	/root/world/geo	Sets the path to a scene graph location. The path parameter options are available by clicking the ▼ dropdown menu. For more information, refer to the Scene Graph Location Widget Type in Common Parameter Widgets on page 13.
order	Scale Rot Trans	Sets the order to apply the transform.
rotateOrder	Rx Ry Rz	Sets the order of each rotation.
stackOrder	First	Sets whether to apply before or after the transforms.
translate	0.0, 0.0, 0.0	Moves the object up, down, left, right, in or out (of 3D space).
rotate	0.0, 0.0, 0.0	Specifies the pivoting around the pivot (axis).
scale	1.0, 1.0, 1.0	Sets the scale (on individual axis of x, y or z).
pivot	0.0, 0.0, 0.0	Sets the point around which the translate and rotate happens.
uniformScale	1	Scales the translate, rotate and scale uniformly.
makeInteractive	No	When set to Yes , you can drag objects in the Viewer and Katana retains the information from the Viewer.
adjustParentBounds	Yes	Specifies whether or not to adjust the bound attribute of the parent locations affected by the transformations of the child.

TransformEdit

The TransformEdit node allows you to make changes to the transform attributes of a scene graph location.

Control (UI)	Default Value	Function
path	/root/world/geo	<p>Sets the path to a scene graph location. The path parameter options are available by clicking the ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location Widget Type in Common Parameter Widgets on page 13.</p>
action	replace global transform	<p>Determines the node's behavior:</p> <ul style="list-style-type: none"> • override interactive transform - overrides the values set in the scene graph location. • append new transform - adds values to the existing values set in the scene graph location. • replace global transform - replaces the global transform attributes relative to the origin of the world.
rotationOrder	XYZ	Sets the order in which the rotation is applied: XYZ, XZY, YXZ, YZX, ZXY, ZYX.
When action: append new transform		
stackOrder	First	In Katana the local transform at any location is created using a stack of transforms. The stackOrder parameter specifies whether the new transform is appended at the first or last position in this stack.

VelocityApply

Creates extra time samples on the P or Pw attribute of a shape using the V or v attribute describing velocity in units per second.

Control (UI)	Default Value	Function
CEL	none	<p>The scene graph locations are specified using the Collection Expression Language (CEL). The CEL parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>
velocityAttribute	none	<p>The name of the attribute representing the velocity information to be used by the node. If the parameter is not set, the following attributes are checked:</p> <ul style="list-style-type: none"> • geometry.point.V • geometry.point.v • geometry.arbitrary.v
VelocityUnits	units / second	<p>Units to be used to interpret the values stored in the velocity attribute, with the following options:</p> <ul style="list-style-type: none"> • units / second • units / frame
velocityScale	1	<p>Defines a multiplier on the velocity attribute, where 1 = no change.</p>
fps	24	<p>Defines frames per second. Used to determine the amount of the velocity (which is defined in units per second) to apply to the geometry attributes.</p>

ZoomToRect

The ZoomToRect node zooms and/or crops the render by setting crop window and screen window attributes. When this node is edited, a guide rectangle is drawn in the **Monitor** tab. The guide can be resized to adjust what portion of the render is zoomed and/or cropped.

Control (UI)	Default Value	Function
rect		
<resolution>	Dependent on Project Settings	List of preset sizes to zoom to.
rect > ▼	N/A	For more information, refer to the Rectangle Widget Type in the Common Parameter Widgets on page 13.
left	0	Sets the left position of the rectangle.
bottom	0	Sets the bottom position of the rectangle.
width	2048	Sets the width of the rectangle.
height	1556	Sets the height of the rectangle.
ZoomToRect parameters continued		
zoom	Yes	When Yes , sets what portion of the render to zoom in to.
crop	Yes	When Yes , sets what portion of the render to crop out.

Miscellaneous Nodes

The nodes in this chapter are considered Miscellaneous and fall under the **Misc** category in Katana. These are listed alphabetically, and each node includes a short description followed by a list of the node's parameters and their functions.

ArnoldLiveRenderSettings

The ArnoldLiveRenderSettings node provides parameters for use by Arnold when live rendering. The parameters are defined in a GenericAssign **.xml** file, such as the **ArnoldGlobalSettings.xml** and **ArnoldObjectSettings.xml**.



NOTE: The parameters that are available for this node are dependent on which version of Arnold you are using.

For more information on some of the other parameters, please refer to the documentation that ships with Arnold.

DependencyMerge

The DependencyMerge node takes any number of Render dependencies as inputs and consolidates them into a single link that you can wire into your dependent Render node.



NOTE: Add as many ports as necessary by clicking the arrow at the top of the node in the **Node Graph** tab.

Dependencies between Render nodes are represented by links between the nodes. This can rapidly become very complex, since a single Render node may depend on several Render and ShadowRender nodes throughout the node graph.

Control (UI)	Default Value	Function
farmSettings		
farmFileName	none	Sets the location where the farm file is written.

Dot

The Dot node performs no operation on the data passing through it. Its purpose is to improve the appearance and layout of your node graph, but also to disable connections between nodes. In this way, you can use the Dot node as an on/off switch for incoming connections.



TIP: You can insert Dot nodes on-the-fly during link creation by pressing the **.** (**period**) key.

Control (UI)	Default Value	Function
Display As Dot	enabled	When enabled, Dot nodes are displayed as a dot in the node graph instead of the regular rectangle node shape.

Gaffer

The Gaffer node is used to wrap multiple lights into a single node, as well as adding the ability to:

- Use light profiles for different types of light.
- Add light rigs to group lights together.
- Mute and solo lights and groups of lights.
- Link lights to specific geometry.
- Add aim constraints to lights.



NOTE: Some of the controls listed may not be available due to the extensive customizability of Katana. Some of the Gaffer node's menu options are created using profiles, which can result in different light creation menu options.

Control (UI)	Default Value	Function
profile	prman	Sets the basic render profile used by the Gaffer node. The profiles themselves are shipped with the plug-ins and the default can be set through Python using the <code>KATANA_GAFFER_DEFAULT_PROFILE</code> environment variable.

Control (UI)	Default Value	Function
sync selection	Disabled	When enabled, selecting a gaffer light within the Parameters tab selects its location within the Scene Graph tab.
		
 Add Aim Target	N/A	Adds an aim target (a locator) to the gaffer table. The locator is used to provide a target for light aim constraints.
 Add Light	N/A	Adds a light to the gaffer table.
 Add Rig	N/A	Adds a rig to the gaffer table.
 Add Master Material	N/A	Adds a master material to the gaffer table. The master material is assigned to lights, providing the same material for multiple lights. Each light is also capable of overriding the defaults set by this master material.
 Add Sky Dome	N/A	Adds a sky dome light to the gaffer table. As each implementation for a sky dome is different, each studio should implement a version specific to its needs.
Import Rig...	N/A	Adds a previously exported rig to the gaffer table.

Control (UI)	Default Value	Function
[Gaffer object table]	gaffer	<p>Displays a list of all objects controlled by this Gaffer node. The object table contains the following information:</p> <ul style="list-style-type: none"> • Name - the name of the object. • M - click to mute the object so that it is omitted from renders. • S - click to solo the object so that everything not soloed is omitted from interactive renders. • Shader - displays the shader associated with the object. You can also click this column to select a shader. • Color - specifies the color of a light. Click the swatch to activate the color picker. • Int - sets the light intensity. • Exp - sets the light exposure. • Link - indicates whether or not the item is linked. A star in the entry indicates there are exceptions. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  <p>NOTE: The Object, Material, and Linking tabs under the object table are populated depending on the object selected.</p> </div>
[Gaffer object table item] > [right-click]		
Add >  Add Light	N/A	Adds a light to the gaffer table.
Add >  Add Rig	N/A	Adds a rig to the gaffer table.
Add >  Add Aim Target	N/A	Adds an aim target (a locator) to the gaffer table. The locator is used to provide a target for light aim constraints.
Add >  Add Master Material	N/A	Adds a master material to the gaffer table. The master material is assigned to lights, providing the same material for multiple lights. Each light is also capable of overriding the defaults set by this master material.
Add >  Add Sky Dome	N/A	Adds a sky dome light to the gaffer table. As each implementation for a sky dome is different, each studio should implement a version specific to its needs.
Add > Import Rig...	N/A	Adds a previously exported rig to the gaffer table.

Control (UI)	Default Value	Function
Delete	N/A	Deletes the entry in the Gaffer object table selected.
Lock	N/A	Locks the entry to prevent transformation. Materials are still editable.
Duplicate	N/A	Creates a copy of the currently selected entry.
Export Rig...	N/A	Exports the currently selected item as a GPRIG file.

Object, Material and Linking tabs - display parameters dependent on what is selected in the Gaffer object table.

GafferThree

The GafferThree node type allows you to create lights under an arbitrary hierarchy of rigs. Materials, transformations and constraints can be applied to lights from within the GafferThree's **Parameters** interface. This node also supports:

- Creating and applying master materials to lights.
- Muting and soloing lights, or all lights under a rig.
- Linking lights to specific objects.
- Editing lights from the incoming scene, and editing multiple lights at once.
- Adding aim constraints to lights.



NOTE: The GafferThree node is an improvement upon the legacy Gaffer node from Katana 1.x. Notable improvements from the Gaffer node are the following:

- Improved performance when dealing with large numbers of lights in Katana projects.
- Full advantage of Geolib3, the scene graph processing library.
- Allows editing of existing lights from the incoming scene.

The legacy Gaffer node type from Katana 1.x is still present, and current scenes should continue to work, but it is advised to move to using the newer GafferThree node type where possible.

Control (UI)	Function
 > Select In Scenegraph	Sets the scene graph path to the location to be created. For more information, see Common Parameter Widgets on page 13.
 > Show Incoming Scene	Displays all incoming lights, rigs, and master materials from upstream Gaffer-type nodes.
sync selection	When enabled, selecting a Gaffer light within the Parameters tab selects its location within the Scene Graph tab: <ul style="list-style-type: none"> • off - no syncing is performed (the default). • out - selection of a light in the GafferThree node is mirrored in the Scene Graph tab, but not the other way around. • in/out - selecting in either the Scene Graph tab or GafferThree node results in the corresponding entry in the other also being selected.
[Gaffer object table]	Displays a list of all objects controlled by this GafferThree node. The object table contains the following information: <ul style="list-style-type: none"> • Name - the name of the object. Double-click in this column to change the name of the item. • M - click to mute the object so that it is omitted from renders. • S - click to solo the object so that everything not solo-ed is omitted from interactive renders. • Shader - displays the shader associated with the object. You can also right-click in this column to select a shader. Once you've added a shader, double-click in this column to assign or change it at any time. • Color - specifies the color of a light. Double-click the swatch to activate the color picker. If there isn't a swatch, you need to add a color in the Material tab before you can change it in this column. • Int - sets the light intensity. • Exp - sets the light exposure. • Linking - indicates whether or not the item is linked. A star in the entry indicates there are exceptions.
Right-click the [Gaffer object table item]	
Add >  Master Material	Adds a master material to the gaffer table. The master material is assigned to lights, providing the same material for multiple lights. Each light is also capable of overriding the defaults set by this master material.

Control (UI)	Function
Add >  Light	Adds a light to the object table.
Add >  Rig	Adds a rig for multiple lights to the object table.
Add > Import Rig...	Adds a previously exported rig to the object table.
Delete	Deletes the selected entity in the object table.
Duplicate	Creates a copy of the currently selected entity.
Adopt for Editing	Allow you to make edits on a light, rig, or master material that has been shown from an incoming scene, which can be any upstream Gaffer-type node.
Delete Edit Package	If you adopted a light, rig, or master material for editing, you can revert back to a read-only state and reverse the changes that you applied.
Toggle Lock State of Selected Items	Toggles the lock state of the selected entity in the object table.
Group Selected Siblings Under Rig	Groups the selected siblings under a newly created rig.
Export Rig	Exports the currently selected item as a .rig file.
Expand All	<p>Expands the selected branch in the object table to reveal all children. If the selected branch does not have any children, nothing happens when attempting to expand.</p> <p> NOTE: In the menu, Expand All becomes Expand Branch whenever there is more than one item in the Gaffer object table.</p>
Expand All To	<p>Expand the branch to a specific type, either assembly, component, or level-of-detail group. This method of expansion applies specifically to the scene graph, and has limited use for the GafferThree.</p> <p> NOTE: In the menu, Expand All To becomes Expand Branch To whenever there is more than one item in the Gaffer object table.</p>

Control (UI)	Function
Collapse All To	<p>Collapse the branch to a specific type, either assembly, component, or level-of-detail group. This method of collapse applies specifically to the scene graph, and has limited use for the GafferThree.</p> <p> NOTE: In the menu, Collapse All To becomes Collapse Branch To whenever there is more than one item in the Gaffer object table.</p>
Collapse All	<p>Collapses the selected branch in the object table to hide all children. If the selected branch does not have any children, nothing happens when attempting to collapse.</p> <p> NOTE: In the menu, Collapse All becomes Collapse Branch whenever there is more than one item in the Gaffer object table.</p>
Expand Location	Expands the selected location to only the children and leaves directly below it in the hierarchy.
Collapse Location	Collapses the selected location and any children and leaves directly below it, but not any entities higher than the location in the hierarchy.

The display parameters for the **Object**, **Material**, and **Linking** tabs are dependent on what's selected in the Gaffer object table. Where a particular tab isn't listed for an object type, there are no parameters in that tab.

Control (UI)	Default Value	Function
Object list: master material		
Material tab		
useLookFileMaterial	Disabled	When enabled, the material from an associated Look File is used.
Add Shader	N/A	Click to add a renderer-specific shader from the dropdown list. The Material tab is populated with controls appropriate to the shader selected, and are dependent on the renderers installed.

Control (UI)	Default Value	Function
Object list: light		

Control (UI)	Default Value	Function
Object tab > geometry		
projection	perspective	<p>Sets the light projection mode:</p> <ul style="list-style-type: none"> • perspective - a warped projection where distant objects/features appear smaller than those nearer the camera. • orthographic - a two-dimensional representation of a three-dimensional object.
radius	1	Sets the light's radius.
fov	70	Controls the field of view angle in degrees.
orthographicWidth	30	Sets the orthographic projection width.
centerOfInterest	20	Sets the center of interest.
near	0.1	Sets the near clipping plane distance.
far	100000	Sets the far clipping plane distance.
screenWindow	-1.0, 1.0, -1.0, 1.0	Controls the screen window placement on the imaging plane. They are the left , right , bottom , and top bounds of the screen window.
Object tab > transform		
transform	N/A	<p>Transforms the light according to the SRT or matrix controls.</p> <p>For more information, refer to the Transform Controls Widget Type in the Common Parameter Widgets on page 13.</p>
transform > Tools ▾	N/A	<p>Adjusts the light to match selected scene graph selection options in the dropdown menu.</p> <p>For more information, refer to the Transform Tools Widget Type in the Common Parameter Widgets on page 13.</p>
enable aim constraint: enabled; aim constraint options		

Control (UI)	Default Value	Function
targetPath	none	<p>Specifies the object(s) to constrain to. If you want to aim a light to point at a target, this is the target. If you set multiple targets, then the constraint aims at the average center of the objects. The targetPath parameter options are available in either a scene graph widget or ▼ dropdown menu to the right of the parameter.</p> <p>For more information, refer to the Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.</p>
addToConstraintList	Yes	Specifies whether or not to add the base path for the light to the globals.constraintList at /root/world in the Attributes tab.
targetOrigin	Object	<p>Sets how the center of the target object is calculated:</p> <ul style="list-style-type: none"> • Object - uses the local origin of the object as the target. • Bounding Box - uses the center of the object's bounding box as the target. • Face Center Average - uses the face center average of the object as the target. • Face Bounding Box - uses the face center average of the object's bounding box as the target.
baseAimAxis	0.0, 0.0, -1.0	The axis of the base object that is pointed at the target. Adjusting these values changes the side of the object that is aimed at the target.
baseUpAxis	0.0, 1.0, 0.0	The axis of the base object that is pointed upwards relative to the target. Adjusting these values changes the rotation of the base object, while keeping the aim constant.
targetUpAxis	0.0, 1.0, 0.0	The axis of the target object that is pointed upwards relative to the base object. Adjusting these values changes the rotation of the target object, while maintaining the aim constant.
setRelativeTargets	No	<p>Stores target paths in the scene graph constraint definition as paths relative to the base path.</p> <p>Targets should still be specified as absolute paths in this node's parameters.</p>

Control (UI)	Default Value	Function
Object tab > annotation		
text	none	Places a label in the Viewer containing the string entered in the text field.
previewColor	1.0, 1.0, 1.0	Specifies the color of the light annotation in the Viewer. This value does not affect the color value of the light. For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.
Material tab > material		
useLookFileMaterial	Disabled	When enabled, the material from an associated Look File is used.
Add Shader	N/A	Click to add a renderer-specific shader from the dropdown list. The Material tab is populated with controls appropriate to the shader selected, and are dependent on the renderers installed.
useLookFileMaterial: enabled		
reference > asset	N/A	Set the path to the asset you want in your scene. For more information, refer to the Asset and File Path Widget Types in the Common Parameter Widgets on page 13.
reference > materialPath ▼	N/A	Choose the material path from the dropdown list, based on the asset you have listed in the asset field above. If you do not have an asset listed in the asset field, nothing appears in the dropdown list for materialPath .
Linking tab > light linking		
Support for light linking is dependent on your renderer.		
action	append linking information	Determines whether the linking options set in this node override the incoming scene options or if the new settings are appended to the incoming options. If this light doesn't exist in the incoming scene, this option has no effect.

Control (UI)	Default Value	Function
initialState	on (or use existing value for adopted lights)	<p>Determines whether the newly-added light location is initially on, off, or use existing value.</p> <p> NOTE: The use existing value option is only available for adopted lights.</p>
clearUnmatched	disabled	<p>When linking is resolved, the clearUnmatched parameter determines whether or not existing light linking attributes for this light are removed from locations which do not match the on or off expressions.</p> <p>The effect of this parameter is only visible in the Attributes tab when linking has been resolved, which means after a LightLinkResolve node or when Implicit Resolvers are active.</p> <p>Examines the lightList attribute on your linked objects to ensure that the attributes have been set correctly. If the parameter has been disabled, the value of the enable child attribute in the lightList attribute for your light is 0; otherwise, the default enabled setting is 1.</p>
on	N/A	<p>A CEL Statement through which scene graph locations can be linked to the selected light, thereby turning the light on for those locations.</p> <p>The scene graph locations are specified using the Collection Expression Language (CEL). For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation).</p> <p>CEL Statements are edited using CEL Statement Widgets. For more information on the CEL Statement Widget type, refer to CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>

Control (UI)	Default Value	Function
off	N/A	<p>A CEL Statement through which scene graph locations can be linked to the selected light, thereby turning the light off for those locations.</p> <p>The scene graph locations are specified using the Collection Expression Language (CEL). For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation).</p> <p>CEL Statements are edited using the CEL Statement Widgets. For more information on the CEL Statement Widget type, refer to CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>

Linking tab > shadow linking

Support for shadow linking is dependent on your renderer.

action	append linking information	Determines whether the linking options set in this node override the incoming scene options or if the new settings are appended to the incoming options. If this light doesn't exist in the incoming scene, this option has no effect.
initialState	don't set value (or use existing value for adopted lights)	<p>Determines the initial value for shadow visibility in the lightList entry for the newly-added light: on, off, don't set value, or use existing value.</p> <p> NOTE: The use existing value option is only available for adopted lights.</p>

Control (UI)	Default Value	Function
clearUnmatched	disabled	<p>When linking is resolved, the clearUnmatched parameter determines whether or not existing light linking attributes for this light are removed from locations which do not match the on or off expressions.</p> <p>The effect of this parameter is only visible in the Attributes tab when linking has been resolved, which means after a LightLinkResolve node or when Implicit Resolvers are active.</p> <p>Examines the lightList attribute on your linked objects to ensure that the attributes have been set correctly. If the parameter has been disabled, the value of the enable child attribute in the lightList attribute for your light is 0; otherwise, the default enabled setting is 1.</p>
on	N/A	<p>A CEL Statement through which scene graph locations can be linked to the selected light, thereby turning shadow casting from this light on for those locations.</p> <p>The scene graph locations are specified using the Collection Expression Language (CEL). For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation).</p> <p>CEL Statements are edited using CEL Statement Widgets. For more information on the CEL Statement Widget type, refer to the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>
off	N/A	<p>A CEL Statement through which scene graph locations can be linked to the selected light, thereby turning shadow casting from this light off for those locations.</p> <p>The scene graph locations are specified using the Collection Expression Language (CEL). For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation).</p> <p>CEL Statements are edited using CEL Statement Widgets. For more information on the CEL Statement Widget type, refer to the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>

Control (UI)	Default Value	Function
Object list: rig		
Object tab > transform		
transform	N/A	Transforms the rig according to the SRT or matrix controls. For more information, refer to the Transform Controls Widget Type in the Common Parameter Widgets on page 13.
transform > Tools ▼	N/A	Adjusts the rig to match selected scene graph selection options in the dropdown menu. For more information, refer to the Transform Tools Widget Type in the Common Parameter Widgets on page 13.
Object tab parameters continued		
enable point constraint	disabled	When enabled, specifies the point constraint options .
enable orient constraint	disabled	When enabled, specifies the orient constraint options .
when enable point constraint: enabled; point constraint options		
targetPath	N/A	Specifies the object(s) to constrain to. If you want to aim a rig to point at a target, this is the target. If you set multiple targets, then the constraint aims at the average center of the objects. The targetPath parameter options are available in either a scene graph widget or ▼ dropdown menu to the right of the parameter. For more information, refer to the Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.
addToConstraintList	Yes	Specifies whether or not to add the base path for the rig to the globals.constraintList at /root/world in the Attributes tab.
allowMissingTargets	No	When set to Yes , silently ignore the constraint if its target is not in the scene graph. When set to No , produce an error on constraint resolution if the target is missing.

Control (UI)	Default Value	Function
baseOrigin	Object	<p>Sets how the center of the base object is calculated:</p> <ul style="list-style-type: none"> • Object - uses the local origin of the object as the origin. • BoundingBox - uses the center of the object's bounding box as the base origin.
targetOrigin	Object	<p>Sets how the center of the target object is calculated:</p> <ul style="list-style-type: none"> • Object - uses the local origin of the object as the target origin. • BoundingBox - uses the center of the object's bounding box as the target origin. • FaceCenterAverage - uses the face center average of the object as the target origin. • FaceBoundingBox - uses the face center average of the object's bounding box as the target origin.
setRelativeTargets	No	<p>Stores target paths in the scene graph constraint definition as paths relative to the base path.</p> <p>Targets should still be specified as absolute paths in this node's parameters.</p>
when enable orient constraint: enabled; orient constraint options		
targetPath	N/A	<p>Specifies the object(s) to constrain to. If you want to aim a rig to point at a target, this is the target. If you set multiple targets, then the constraint aims at the average center of the objects. The targetPath parameter options are available in either a scene graph widget or ▼ dropdown menu to the right of the parameter.</p> <p>For more information, refer to the Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.</p>
addToConstraintList	yes	<p>Specifies whether or not to add the base path for the rig to the globals.constraintList at /root/world in the Attributes tab.</p>
targetOrientation	Object	<p>Sets how the orientation of the target object is calculated:</p> <ul style="list-style-type: none"> • Object - uses the local origin of the object as the target. • Face - uses the local origin on the face as the target.

Control (UI)	Default Value	Function
allowMissingTargets	No	When set to Yes , silently ignore the constraint if its target is not in the scene graph. When set to No , produce an error on constraint resolution if the target is missing.
xAxis	Enabled	When enabled, orientation is constrained on the x axis.
yAxis	Enabled	When enabled, orientation is constrained on the y axis.
zAxis	Enabled	When enabled, orientation is constrained on the z axis.
setRelativeTargets	No	Stores target paths in the scene graph constraint definition as paths relative to the base path. Targets should still be specified as absolute paths in this node's parameters.

Group

The Group node is a node that contains other nodes. You can create Group nodes by selecting some nodes and pressing **G** to collapse them into a Group. To do the inverse (explode a Group), select a Group node and press **U**.

GroupStack

The GroupStack node is a SuperTool that creates a convenient interface for managing a list of nodes of the same type.

Within the GroupStack interface, you can create any number of **nodes of the same type**, and these nodes are linked together, providing a single output by connecting them one after the other in serial, in the order in which they appear in the stack. GroupStack is similar to GroupMerge, except with GroupMerge the nodes are merged together instead of creating a list of nodes where a 3D input is passed through.

This node is most often used to group nodes that have one input and modify the scene graph in some way. For example, the GroupStack node could be used to manage multiple CollectionCreate nodes, or multiple material edits.



NOTE: When the GroupStack node is first created, its type is not defined. You can create a node and then add it to the stacklist by **Shift**+middle-mouse and dragging from the **Node Graph** tab to the node's list in the **Parameters** tab. At that point, the GroupStack is permanently typed as a group of the type of node that was dragged in.

Control (UI)	Default Value	Function
+	N/A	Creates a new node of the type associated with this node and adds it to the node list.
🔍	N/A	Brings up a searchable list to aid in selection.
/	N/A	Locks all nodes against editing. Unlocks all nodes for editing.
[Right-click menu]		
Ignore Selected Entries	N/A	Disables the selected nodes.
📍 View At Location	N/A	Sets the current view node to the selected node
Delete Selected Entries	N/A	Deletes the selected node.
Duplicate Selected Entries	N/A	Duplicates the selected node, creating a new copy of both the node and matching its parameters.
Cut Selected Entries	N/A	Deletes the selected node and copies it to the clipboard.
Copy Selected Entries	N/A	Copies the selected node to the clipboard.
Paste	N/A	Paste the current clipboard node into this list.
Tearoff Parameters Of Selected Entries...	N/A	Create a new floating window with the parameters of this node on a tab inside.

InteractiveRenderFilters

Interactive render filters enable you to setup common interactive render recipe changes without having to include them within the recipe. These filters are designed to only be included when performing an interactive render and are ignored for hotrenders.

InteractiveRenderFilters nodes don't need to be connected into a recipe to take affect.

An example of a filter is a resolution change. You can set up an interactive render filter to reduce the size of a render, thus making debugging and light tests much quicker. Other examples might be anti-aliasing settings, shading rate changes (if using RenderMan), or the number of light bounces.

LiveGroup

The LiveGroup node is similar to the Group node except the contents are loaded from an external file. The contents of the LiveGroup can be locked (non-editable) or unlocked, and is used mainly during look development. To change the contents of a LiveGroup, you can modify the file it references or change the contents of a LiveGroup in a Katana session, then publish it back to the source file. The LiveGroup context menu is dynamic, and the options that appear in the menu change depending whether the node is in a non-editable or editable state.



NOTE: The source file is automatically reloaded each time a scene is opened in Katana. If the file has been changed, the changes are picked up automatically. If the source file cannot be read or no longer exists, a copy stored in the scene file is used instead, and a warning is printed to the shell.

Control (UI)	Default Value	Function
source	none	Sets the path to load in the source (file) as the contents of the LiveGroup. For more information, refer to the Asset and File Path Widget Types in the Common Parameter Widgets on page 13.
	N/A	Opens the Parameters tab.
[Right-click menu] (when non-editable)		
Load...	N/A	Opens the Load LiveGroup dialog to allow you to select a LiveGroup source for the node.
Revert	N/A	Reloads the LiveGroup from its current source , thus discarding any changes made while it was unlocked. Revert is only available when a source has been set.
Edit Contents	N/A	Changes the state of the node to be editable, so that contents of the node can be modified.
Convert to Group	N/A	Converts the LiveGroup node to a Group node.
<i>Other</i> Show Parameters	N/A	Opens the Parameters tab.
[Right-click menu] (when non-editable)		

Control (UI)	Default Value	Function
Load...	N/A	Opens the Load LiveGroup dialog to allow you to select a LiveGroup source for the node.
Revert	N/A	Reloads the LiveGroup from its current source , thus discarding any changes made while it was unlocked. Revert is only available when a source has been set.
Publish...	N/A	Opens the Publish LiveGroup dialog for publishing the parameters and contents of the LiveGroup node as a LiveGroup source file or asset. This leaves the node in an editable state for further changes.
Publish and Load...	N/A	Opens the Publish and Load LiveGroup dialog for publishing the parameters and contents of the LiveGroup node as a LiveGroup source file or asset, and for loading the published source file or asset. This sets the node to be non-editable and locks the node against further changes.
Convert to Group	N/A	Converts the LiveGroup node to a Group node.
<i>Other</i> Show Parameters	N/A	Opens the Parameters tab.

LiveGroupStack

This is a convenience node for managing a stack of nodes of the same type. When initially created, it is typeless. Drag a node into the list to define the type of the stack.



NOTE: This node is a live reference to an externally stored file specified using the **source** parameter. As such, its internal nodes are read-only. See also [LiveGroup](#) on the previous page.

Control (UI)	Default Value	Function
[stack contents]	none	Contains the list of nodes to manage.
	N/A	Brings up a searchable list to aid in selection.
 / 	N/A	 Hides all nodes.  Unhides all nodes.

Control (UI)	Default Value	Function
[Right-click menu]		
Ignore Selected Entries	N/A	Disabled for this node while it is a LiveGroupStack.
View At Location	N/A	Sets the current view node to the selected node
Delete Selected Entries	N/A	Disabled for this node while it is a LiveGroupStack.
Duplicate Selected Entries	N/A	Disabled for this node while it is a LiveGroupStack.
Cut Selected Entries	N/A	Disabled for this node while it is a LiveGroupStack.
Copy Selected Entries	N/A	Copies the selected node to the clipboard.
Paste	N/A	Disabled for this node while it is a LiveGroupStack.
Tearoff Parameters Of Selected Entries...	N/A	Create a new floating window with the parameters of this node on a tab inside.
source	none	<p>Sets the path to load in the source (file) as the contents of the LiveGroup.</p> <p>For more information, refer to the Asset and File Path Widget Types in the Common Parameter Widgets on page 13.</p>

LookFileAssign

Assigns a Look File to a scene graph location defined by a CEL statement.

Control (UI)	Default Value	Function
CEL	None	<p>Specifies the scene graph location(s) where the Look File is assigned.</p> <p>The scene graph locations are specified using the Collection Expression Language (CEL). The CEL parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>
asset	None	<p>The Look File that is assigned to the specified scene graph location(s).</p> <p>For more information, refer to the Asset and File Path Widget Types in the Common Parameter Widgets on page 13.</p>

LookFileGlobalsAssign

LookFileGlobalsAssign nodes associate a look file with the **/root** location and is designed to repeat the changes made to that location (LookFiles for assets are assigned to the location of the asset).

Also see [LookFileManager](#) on page 219.

Control (UI)	Default Value	Function
asset	none	<p>The asset to assign to the Look File.</p> <p>For more information, refer to the Asset and File Path Widget Types in the Common Parameter Widgets on page 13.</p>

Control (UI)	Default Value	Function
resolveImmediately	No	<p>When set to Yes, LookFileResolve runs on the root of the scene as part of this node. This is useful for overriding or layering scene root attributes from published Look File assets</p> <p>This option has special behavior during Look File baking. Instead of resolving, it appends the Look to the lookfile.referencedAssets attribute. This gets included in the resulting Look File and maintains a live reference to it during subsequent Katana standard resolution.</p>
Flush Look File Cache	N/A	Click to flush the Look File cache and force a reload.

MaterialAssign

Assigns materials to geometry in the scene graph.

Control (UI)	Default Value	Function
CEL	none	<p>Sets the CEL specification of scene graph locations on which the assignment acts.</p> <p>For more information, see the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL widget parameters in Common Parameter Widgets on page 13.</p>
materialAssign	none	<p>Specifies the material to assign. Typically, you middle-mouse drag this from under /root/materials/geo in the scene graph. The materialAssign parameter options are available by clicking the ▼ dropdown menu.</p> <p>For more information, refer to the Path Selection Widget Types in Common Parameter Widgets on page 13.</p>

NonpersistentSwitch

This node is identical to the Switch node, except that the in control is reset to 0 whenever the file is loaded (the value you set it to in your current session is never saved to the **.katana** file).

This is useful for switches you may want to use interactively (low-quality settings, for example), that you don't want to mistakenly have set for a batch render. Using a NonpersistentSwitch ensures that batch renders always get the left-most input to the node.

PonyFarm

The PonyFarm node is an example SuperTool that allows you to create ponies and cows. You can also group ponies and cows under a herd. Herds create a scene graph group complete with transform attributes. Ponies and cows created below the herd inherit its transformations, which enables you to move the ponies and cows around as one.

Control (UI)	Function
	Specifies the cows, ponies, and herds scene graph locations. For more information, refer to the Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.
 > Show Incoming Scene	Displays all incoming lights, rigs, and master materials from upstream Gaffer-type nodes.
sync selection	When enabled, selecting a Gaffer light within the Parameters tab selects its location within the Scene Graph tab: <ul style="list-style-type: none"> • off - no syncing is performed (the default). • out - selection of a light in the GafferThree node is mirrored in the Scene Graph tab, but not the other way around. • in/out - selecting in either the Scene Graph tab or GafferThree node results in the corresponding entry in the other also being selected.
Right-click in the [PonyFarm object table]	
Add > Pony	Adds a pony to the scene.
Add > Cow	Adds a cow to the scene.
Add > Herd	Adds a herd to the scene to create a scene graph group of entities.

Control (UI)	Function
Add > Herd of Ponies	Adds a herds of ponies to the scene.
Delete	Deletes the selected entity in the object table.
Duplicate	Creates a copy of the currently selected entity.
Adopt for Editing	Allows you to make edits on the entity that has been shown from an incoming scene, which can be any upstream Gaffer-type node.
Delete Edit Package	If you adopted an entity for editing, you can revert back to a read-only state and reverse the changes that you applied.
Randomize Color	Assigns a random color to the selected entity.
Expand All	<p>Expands the selected branch in the object table to reveal all children. If the selected branch does not have any children, nothing happens when attempting to expand.</p> <p> NOTE: In the menu, Expand All becomes Expand Branch whenever there is more than one item in the Gaffer object table.</p>
Expand All To	<p>Expand the branch to a specific type, either assembly, component, or level-of-detail group. This method of expansion applies specifically to the scene graph, and has limited use for the GafferThree.</p> <p> NOTE: In the menu, Expand All To becomes Expand Branch To whenever there is more than one item in the Gaffer object table.</p>
Collapse All	<p>Collapses the selected branch in the object table to hide all children. If the selected branch does not have any children, nothing happens when attempting to collapse.</p> <p> NOTE: In the menu, Collapse All becomes Collapse Branch whenever there is more than one item in the Gaffer object table.</p>

Control (UI)	Function
Collapse All To	<p>Collapse the branch to a specific type, either assembly, component, or level-of-detail group. This method of collapse applies specifically to the scene graph, and has limited use for the GafferThree.</p> <p> NOTE: In the menu, Collapse All To becomes Collapse Branch To whenever there is more than one item in the Gaffer object table.</p>
Expand Location	Expands the selected location to only the children and leaves directly below it in the hierarchy.
Collapse Location	Collapses the selected location and any children and leaves directly below it, but not any entities higher than the location in the hierarchy.
Add a Cow	Adds a cow to the scene.

Control (UI)	Default Value	Function
XForm Tab > transform		
transform	N/A	<p>Transforms the entity according to the Scale, Rotation, and Translation (SRT) or matrix controls.</p> <p>For more information, refer to the Transform Controls Widget Group in the Common Parameter Widgets on page 13.</p>
transform > Tools ▾	N/A	<p>Adjusts the entity to match selected scene graph selection options in the dropdown menu.</p> <p>For more information, refer to the Transform Tools Widget Group in the Common Parameter Widgets on page 13.</p>
AnimalInfo Tab		
age	1	The age parameter is an example that shows you that you can set up any type of attributes for this node, for instance, the age of a cow or a pony.

PonyStack

The PonyStack node is an example node that allows you to create multiple pony geometry locations with differing transform attributes.

Control (UI)	Default Value	Function
location	/root/world/geo	Specifies the ponies scene graph location. For more information, refer to the Create Scene Graph Location Widget Type in the Common Parameter Widgets on page 13.
+	N/A	Creates a new pony in the object list.
transform		
transform	N/A	Transforms each of the ponies according to the Scale, Rotation, and Translation (SRT), or matrix controls. For more information, refer to the Transform Controls Widget Type in the Common Parameter Widgets on page 13.

PrmanLiveRenderSettings

The PrmanLiveRenderSettings node provides parameters for use by PRMan when live rendering. The parameters are defined in a GenericAssign **.xml** file, such as **PrmanGlobalSettings.xml** and **PrmanObjectSettings.xml**.



NOTE: The parameters that are available for this node are dependent on which version of PRMan you are using and whether you are using the PRMan plug-in that ships with Katana or RenderMan Studio for Katana.

For more information on some of the other parameters, please refer to the documentation that ships with RenderMan.

RendererProceduralAssign

The RendererProceduralAssign node allows you to assign renderer procedurals to specific locations in the scene.

Control (UI)	Default Value	Function
CEL	N/A	<p>Sets the CEL specification of scene graph locations on which the assignment acts.</p> <p>The scene graph locations are specified using the Collection Expression Language (CEL). The CEL parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>
rendererProceduralAssign	N/A	<p>Specifies which procedural to assign. Typically, you'll middle-mouse drag this from under /root/materials/proc in the scene graph. The rendererProceduralAssign parameter options are available by clicking the ▼ dropdown menu.</p> <p>For more information, refer to the Scene Graph Location Widget Type in Common Parameter Widgets on page 13.</p>

RenderScript

This node generates a user-specified command in the outline script, following the same dependency rules as the Render node. This node is not renderable interactively or via --batch command.

Control (UI)	Default Value	Function
command	shell	<p>Sets the outline function to generate. Default value of 'shell' expects a single commandArgument which is the shell command to run on the farm.</p> <p>Example:</p> <pre>command("nodeName", "commandArg1", "commandArg2", "keywordName1" => "keywordValue1", "keywordName2" => "keywordValue2",)</pre>
commandArguments		
commandArguments	none	<p>Array of positional arguments added to the outline function.</p> <p>Example:</p> <pre>command("nodeName", "commandArg1", "commandArg2", "keywordName1" => "keywordValue1", "keywordName2" => "keywordValue2",)</pre>
keywordArguments		

Control (UI)	Default Value	Function
keywordArguments	none	<p>Array of keyword arguments (name => value pairs) added to the outline function.</p> <p>Example:</p> <pre>command("nodeName", "commandArg1", "commandArg2", "keywordName1" => "keywordValue1", "keywordName2" => "keywordValue2",)</pre>
pythonImports		
pythonImports	none	<p>Array of import statements to be added to the Python farm file.</p> <p>Example:</p> <pre>from outline.modules.shell import Shell</pre> <p>or</p> <pre>import outline.module.shell</pre>
farmSettings		
setActiveFrameRange	disabled	<p>When enabled, activeFrameRange parameters are exposed to define the active frame range for rendering.</p> <p>When disabled, the active frame range is assumed to be the same as <code>globals.inTime</code> and <code>globals.outTime</code>.</p> <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;">  <p>NOTE: The active frame range affects farm file generation and ShadowRenderAssign node assignments.</p> </div>

Control (UI)	Default Value	Function
setActiveFrameRange: enabled		
start	1	When setActiveFrameRange is enabled, sets the start of the active frame range.
end	1	When setActiveFrameRange is enabled, sets the end of the active frame range.
farmSettings continued		
dependAll	disabled	When enabled, farm dependencies wait until all frames of this node are rendered before rendering themselves.
threadable	enabled	
farmFileName		
excludeFromFarmOutputGeneration	disabled	When enabled, this node does not appear in any generated farm file (however, the node is still renderable if called directly).
forceFarmOutputGeneration	disabled	When enabled, this node always appears in a generated farm file (regardless of whether it has any valid outputs). <div data-bbox="857 1163 932 1224" data-label="Image"></div> NOTE: If excludeFromFarmOutputGeneration is also set, the node does not appear in the generated farm file (excludeFromFarmOutputGeneration overrides forceFarmOutputGeneration).
inputs		
inputs		

RenderSettings

The RenderSettings node defines the 3D render output settings (camera to use, renderer, size of output image) for an image. For Renderman renders, these settings affect the Format, ScreenWindow, CropWindow statements in your

output RIB, as well as the camera transform and motion samples generated.

Control (UI)	Default Value	Function
cameraName	/root/world/cam/camera	Specifies the camera that the scene should be rendered through. The field contains a path to the camera's location in the scene graph. The cameraName parameter options are available by clicking the ▼ dropdown menu. For more information, refer to the Scene Graph Location Widget Type in Common Parameter Widgets on page 13.
renderer	prman	Specifies the renderer to use.
resolution	512x512	Sets the size of the output image.
overscan	0	Pads the data window of the resulting render by the specified pixel amount on each side. The frame window is unchanged.
adjustScreenWindow	No adjustment	Adjusts the pixel aspect ratio to match one of the Device aspect ratio's dimensions. Either the height or the width of the screen window is adjusted to match the output resolution.
maxTimeSamples	1	Sets how many times a point is sampled when the shutter is open. For animated parameters within Katana (such as transforms), this is how many samples are evaluated from shutter open to close. The higher the number, the more accurate the motion blur.
shutterOpen	0	Specifies the timing of the opening and closing of the camera shutter.
shutterClose	0	
cropWindow	0.0, 1.0, 0.0, 1.0	Specifies the render crop window in normalized coordinates: xmin xmax ymin ymax, starting in the upper left-hand corner. The part of the image that renders has a dotted red line around it. Note: The dotted red line isn't displayed unless you are viewing the RenderSettings node.

Control (UI)	Default Value	Function
interactiveOutputs	all	Specifies whether all the AOVs are rendered during Preview Render, or whether only the primary pass is rendered. If you set the output to primary , the Local Assignment box turns yellow to indicate that you are using local values.
adjustScreenWindowWhen	deferred	

ScenegraphObjectSettings

The ScenegraphObjectSettings allows you to add attributes, used by the **Scene Graph** tab, to locations according to the CEL match.

Control (UI)	Default Value	Function
CEL	N/A	<p>Specifies the scene graph location(s).</p> <p>The scene graph locations are specified using the Collection Expression Language (CEL). The CEL parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>
stopExpand	No	The Scene Graph tab "Expand..." actions do not traverse beneath locations at which this enabled.

ShadowManager

The Shadow Manager is a Super Tool that automates the process of creating Shadow Maps. For more details on the ShadowManager node, please consult the *Katana Technical Guide*, in the *Groups, Macros, and Super Tools* chapter.

Control (UI)	Default Value	Function
Render Passes +	N/A	Creates a new pass of the type associated with this node and adds it to the pass list.

Once you have added a pass, with pass selected

ShadowBranch	N/A	See ShadowBranch node information.
Add light +	N/A	Allows you to add the available lights in the scene.

Once you have added a light, with light selected

renderLocation	default location dependant on scene graph location of light	Specifies the location on disk for shadow map.
resolution	square_512	Sets the resolution of the shadow map.
shadowType	zfile	Specifies the shadow map type.

Control (UI)	Default Value	Function
Pass list [Right-click menu]		
Delete	N/A	Deletes the selected pass from the pass list.
View From ShadowBranch Node	N/A	Allows you to view the scene graph for the selected shadow map pass.

Switch

This node allows you to switch between multiple input nodes. Only the portion of the node graph connected to the selected input port on the Switch node is evaluated. You can select which input to choose using the **in** parameter on the node.

Control (UI)	Default Value	Function
in	0	Selects the index of the input port (starting at zero) to pass through to the output.

Teleport

This node can be used to visually clean up a scene by hiding the lines between nodes. In order to attach more than one node to the Teleport node, click **Add > Add Pass Input**. Then, in the Node Graph, you can drag a line from any node to connect it. When inputs are not shown, each connected node appears on the Teleport node as an output arrow.

Control (UI)	Default Value	Function
name	output	Sets the name for each input.
Show inputs	disabled	When enabled, the connector between this node and the inputs are not shown

TimeOffset

In Katana the current time used in parameter evaluation is a property that flows up the graph and is referenced as a frame in parameter expression. This node modifies that time in upstream nodes. Common uses are to offset or lock data loaded from an upstream input.

Control (UI)	Default Value	Function
inputFrame	1	Sets the value of frame in input nodes. This can be an expression, for example frame + 10.

VariableDelete

This node deletes the locally-set graph state variable listed in the node's **variableName** parameter.

Control (UI)	Default Value	Function
variableName	var1	Specify the name of the local graph state variable that is to be deleted by the node.

VariableEnabledGroup

This node works like a group, but allows you to enable or disable a variable pattern to which other variable nodes attempt to match.

Control (UI)	Default Value	Function
variableName	var1	Specify the name of the graph state variable to which you want to match a specific pattern.
pattern	N/A	Specify the pattern input that the graph state variable should match in order to enable the variable group. A pattern can be any input. The pattern parameter is a CEL statement widget. For more information about CEL statements, refer to Common Parameter Widgets on page 13.

VariableSet

This node sets a graph state variable pattern to which other variable nodes attempt to match.

Control (UI)	Default Value	Function
variableName	var1	Specify the name of the graph state variable to which you want to match a specific pattern.
variableValue	N/A	Specify the value input that the graph state variable should match. A value must be a string.

VariableSwitch

This is a specialized Switch node that selects an input based on the value of a Graph State Variable.

Control (UI)	Default Value	Function
variableName	var1	The name of the Graph State Variable used to select an input port.
patterns	N/A	<p>A set of patterns linked to input ports, used to select an input port according to the value of the Graph State Variable. Patterns may be added and deleted using the wrench  dropdown menu for the group and existing patterns, respectively. Input ports for which no pattern is defined match their exact port name.</p> <p>A pattern takes the form of a CEL statement, where the {@ [name]=="value"} syntax may be used to specify requirements of additional Graph State Variables.</p> <p>The patterns parameter is a CEL statement widget. For more information about CEL statements, refer to Common Parameter Widgets on page 13.</p>



NOTE: If a VariableSwitch node defines no patterns, input select is performed using a faster look-up operation. This may be useful for nodes with a large number of input ports.

ViewerMaterialEdit

ViewerMaterialEdit allows you to put Viewer shader code into the scene rather than in a separate file.

Control (UI)	Default Value	Function
location	N/A	<p>Sets the scene graph location of the material you want to edit. The location parameter options are available by clicking the  dropdown menu.</p> <p>For more information, refer to the Scene Graph Location Widget Type in Common Parameter Widgets on page 13.</p>
viewerShaderSource	N/A	Specifies the GLSL shader code that defines how objects are drawn in the Viewer.

ViewerObjectSettings

Adjusts how objects are displayed in the **Viewer** tab.

Control (UI)	Default Value	Function
CEL	none	<p>The scene graph locations are specified using the Collection Expression Language (CEL). The CEL parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>
drawOptions		
hide	No	Sets whether the object should be hidden in the Viewer.
fill	inherit	<p>Sets how the object is displayed, as:</p> <ul style="list-style-type: none"> • points - display the object using points at the vertices or control points. • wireframe - display the object using wireframe mode. • solid - display the object as a solid. If the display style for the object uses a 3D lighting model, then display the object using that lighting model, whereas if the Viewer tab's display style is points or wireframe, display the object using a single solid color. • inherit - no change to the object's display style, use the default.
light	inherit	<p>Sets the lighting model for the object. This setting doesn't influence the object when it is drawn using wireframe or points. You can set it to:</p> <ul style="list-style-type: none"> • default - uses the simple shaded lighting model. • shaded - uses the viewer shader assigned to the object (or the default viewer shader if one isn't assigned). • inherit - don't override the Viewer tab display style.

Control (UI)	Default Value	Function
smoothing	inherit	<p>When the objects referenced by the CEL statement are being displayed as points or lines, this parameter sets whether they should be anti-aliased. The options are:</p> <ul style="list-style-type: none"> • off - no anti-aliasing. • lines - when displayed as a wireframe, the objects are anti-aliased. • points - when displayed using points, the objects are anti-aliased. • both - when displayed as a wireframe or using points, the objects are anti-aliased. • inherit - no object specific override, use the current default.
windingOrder	inherit	Sets whether the object has a clockwise or counterclockwise winding order. The winding order determines which direction is considered out from an object and which direction is in.
pointSize	4	Sets the size of the points when the object is rendered as a series of points.
color	0.4, 0.4, 0.4	<p>Sets the default draw color.</p> <p>For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.</p>
annotation		
text	None	Sets the text to display with the geometry. When empty, no tag is displayed.
color	0.4, 0.4, 0.4	<p>Sets the default background color for any annotation text.</p> <p>For more information, refer to the Color Widget Type in the Common Parameter Widgets on page 13.</p>
ViewerObjectSettings parameters continued		
pickable	Yes	Sets whether the object is pickable or not.

VisibilityAssign

The VisibilityAssign node changes the visibility setting of objects in the scene. The attribute is inherited, thus large sections of the scene graph can be made visible/invisible by assigning to common parents

A child can be explicitly set to visible even if its parent is not visible. For example, to render just one of several siblings, set the parent's visibility to 0, and set the item to render's visibility to 1. All siblings that are not explicitly marked picks up the parent's visibility setting of 0, but the item to render uses its explicitly set value of 1.

The **Scene Graph** tab displays visibility of each scene graph item as icons.

Control (UI)	Default Value	Function
CEL	none	<p>Specifies what part of the scene graph to assign this attribute to.</p> <p>The scene graph locations are specified using the Collection Expression Language (CEL). The CEL parameter options are available by clicking Add Statements.</p> <p>For more information, refer to the CEL Reference document found on the documentation HTML page (accessed through Help > Documentation) or the CEL Statement Widget Type in Common Parameter Widgets on page 13.</p>
visible	1	Sets the visibility of objects in the render. 0 specifies not visible in render and anything else specifies visible in render.