KATANA

REFERENCE GUIDE

VERSION 1.4v1

Katana™ Reference Guide. Copyright © 2012 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 © 2012 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.

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.

Katana was brought to you by:
Andy Lomas, Andrew Bulhak, Andy Abgottspon, Brian Hall, Chris Beckford, Chris Hallam, Claire Connolly, Dan Hutchinson, Dan Lea, Davide Selmo, Eija Närvänen, Erica Cargle, Fayeez Ahmed, Gary Jones, Grant Bolton, Iulia Giurca, Jeremy Selan, João Montenegro, Joel Byrne, Jonathan Attfield, Konstantinos Stamos, Luke Titley, Örn Gunnarsson, Phil McAuliffe, Phillip Mullan, Richard Ellis, Simon Picard, Stefan Habel, Steve LaVietes, Tom Cowland.

The Foundry
6th Floor, Communications Building,
48 Leicester Square,
London
WC2H 7LT

Rev: April 24, 2013

Contents

PREFACE 8
About this Manual 8
Getting Help 8
Viewing Additional Help 8
Contacting Customer Support 9
NODES A-C 10
AimConstraint 10
Alembic_In 11
ArnoldGlobalSettings 12
ArnoldObjectSettings 13
ArnoldOutputChannelDefine 16
AttributeCopy 17
AttributeEditor 19
AttributeFile_In 21
AttributeModifierDefine 24
AttributeModifierResolve 26
AttributeScript 26
AttributeSet 29
BillboardConstraint 32
BoundsAdjust 33
CameraClippingPlaneEdit 34
CameraCreate 35
CameraImagePlaneCreate 36
CameraScreenWindowConstraint 38
ClippingConstraint 39
CollectionCreate 40
ConstraintListEdit 42
ConstraintResolve 42
CoordinateSystemDefine 43
NODES D-H 44
DependencyMerge 44
DollyConstraint 44
Dot 45

FaceSetCreate 46
Fork3D 46
FOVConstraint 47
Gaffer 48
Group 50
GroupMerge 50
GroupStack 51

NODES I-K 53

ImageAddMix 53
ImageAtop 54
ImageAverage 56
ImageBackgroundColor 59
ImageBlack 60
ImageBlur 61
ImageBrightness 62
ImageChannels 63
ImageClamp 65
ImageColor 67
ImageContrast 68
ImageCrop 69
ImageDifference 70
ImageDistort 73
ImageDivide 74
ImageExclusion 77
ImageExposure 79
ImageFade 81
ImageFrom 82
ImageGain 85
ImageGamma 86
ImageGeometric 87
ImageHypot 90
ImageIn 92
ImageInvert 95
ImageLevels 96
ImageMatte 97
ImageMatteMix 100
ImageMax 101

ImageMin 103
ImageMinus 106
ImageMix 108
ImageMultiply 109
ImageOrient 112
ImageOut 112
ImageOver 115
ImagePlus 117
ImagePosition 120
ImagePremultiply 121
ImageRead 122
ImageReformat 124
ImageSaturation 127
ImageScreen 128
ImageText 131
ImageTransform2D 133
ImageUnder 136
ImageUnpremultiply 139
ImageWhite 140
ImageWrite 141
Importomatic 147
InfoCreate 147
InteractiveRenderFilters 149
Isolate 149

NODES L-Q 151

LightCreate 151
LightLink 153
LightListEdit 155
LiveGroup 156
LiveGroupStack 156
LocationCreate 157
LodGroupCreate 158
LodSelect 159
LodValuesAssign 160
LookFileAssign 162
LookFileBake 162
LookFileGlobalsAssign 163

LookFileLightAndConstraintActivator 164
LookFileManager 165
LookFileMaterialsIn 165
LookFileMaterialsOut 166
LookFileResolve 166
Material 166
MaterialAssign 168
MaterialResolve 168
MaterialStack 169
Merge 169
NonpersistentSwitch 170
OCIOCDLTransform 171
OCIOColorSpace 172
OCIODisplay 172
OCIOFileTransform 173
OCIOLogConvert 174
OCIOLookTransform 174
OrientConstraint 174
ParentChildConstraint 175
PointConstraint 175
PrimitiveCreate 176
PrmanGlobalSettings 177
PrmanObjectSettings 177
PrmanOutputChannelDefine 178
PrmanShadingNode 178
Prune 178

NODES R-Z 180

ReflectionConstraint 180
Rename 180
Render 180
RenderOutputDefine 182
RenderScript 185
RenderSettings 186
ScaleConstraint 187
ScenegraphGeneratorResolve 187
ScenegraphGeneratorSetup 188
ScenegraphXml_In 188

ScreenCoordinateConstraint 188
ShadowBranch 189
TeapotCreate 190
Teleport 191
TimeOffset 192
Transform3D 192
VelocityApply 193
ViewerObjectSettings 193
VisibilityAssign 194

1 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.*

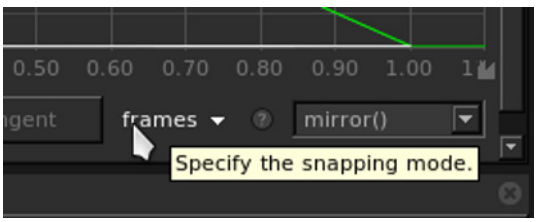
Getting Help

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.

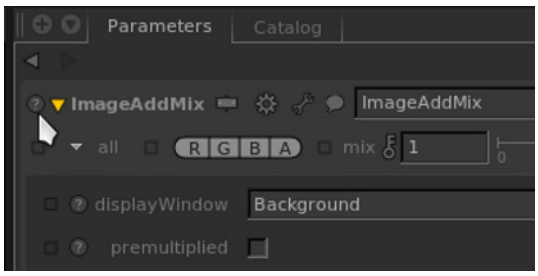
Viewing Additional Help

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 the in application help system fails to address, you can contact Customer Support directly via e-mail at **support@thefoundry.co.uk** or via telephone to our London office on **+44 (0)20 7968 6828** or to our Los Angeles office on **(310) 399 4555** during office hours.

2 Nodes A–C

AimConstraint

Applies an aim constraint to an object in the **Scene Graph**.

Control (UI)	Default Value	Function
basePath	none	The object to constrain.
basePath		
Adopt Scenegraph Selection	N/A	The currently selected **Scene Graph** location is used to populate the parameter.
Adopt Selected Nodegraph Node	N/A	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	N/A	Selects the location specified by this parameter in the **Scene Graph** tab.
targetPath	none	The object(s) to constrain the **basePath** object to.
targetPath > Add Locations		
Path	N/A	Adds another path to this parameter's list of paths.
Append Scenegraph Selection	N/A	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	N/A	Removes all existing paths and replaces them with paths populated with the currently selected **Scene Graph** locations.
Append Nodegraph Node Locations	N/A	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	N/A	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	N/A	Removes all paths from this parameter.
targetOrigin	Object	Sets how the center of the target object is calculated: • **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.

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 framerate to use when querying the alembic file.

Control (UI)	Default Value	Function
name	/root/world/geo/scene-graphGenerator	Specifies the **Scene Graph** location where the Alembic asset is to be placed.
name		
Parent to Scene Graph Selection	N/A	Sets the parent location of the Alembic asset to be the current Scene Graph selection.
Parent to /root/world/geo/...	N/A	Sets the parent location of the Alembic asset to be /root/world/geo.
Parent to /root/world/lgt/...	N/A	Sets the parent location of the Alembic asset to be /root/world/lgt.
Parent to /root/world/cam/...	N/A	Sets the parent location of the Alembic asset to be /root/world/cam.
Select In Scene Graph	N/A	Selects the location specified by this parameter in the **Scene Graph** tab.

Control (UI)	Default Value	Function
abcAsset	none	Specifies where to retrieve the asset, an Alembic (.abc) file.
abcAsset		
Browse...	N/A	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	N/A	Changes the name of the node to match the filename but without the path or extension.
addForceExpand	1	
fps	24	Sets how many frames constitute a second inside the Alembic file.
timing		
mode	Current Frame	Sets the timing mode to apply to the asset: • **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	globals.inTime (an expression)	The frame to retrieve from the Alembic asset.
timing > mode: Clamp Range		
inTime	globals.inTime (an expression)	The start frame for retrieving geometry from the Alembic asset.
outTime	globals.outTime (an expression)	The end frame for retrieving geometry from the Alembic asset.

ArnoldGlobalSettings

The ArnoldGlobalSettings node is populated by the XML file located at ${KATANA_HOME}/plugins/Resources/Arnold4.0/GenericAssign. These parameter names and defaults can change between Arnold versions and, as such, are provided for you to change. In Katana 1.4v1 the available options as shipped are:

ArnoldObjectSettings

The ArnoldObjectSettings node is populated by the XML files located at ${KATANA_HOME}/plugins/Resources/Arnold4.0/GenericAssign. These parameter names and defaults can change between Arnold versions and, as such, they are provided for you to change. In Katana 1.4v1 the available options as shipped are:

Control (UI)	Default Value	Function
Geometry		
invert_normals	Off	Flips normal direction.
smoothing	Off	Controls normal interpolation between polygons.
flip_t	none	Inverts (flips) the objects T coordinates.
Subdivision Surfaces		
subdiv_type	catclark	Only affects subD meshes. Sets the type of mesh subdivision.
iterations	0	Only affects subD meshes. Controls the maximum number of subD iterations. When set to 0, subD is off (plan polygons). Setting to an integer n gives n iterations (e.g. if iterations=10 there are 10 iterations).
subdiv_smooth_derivs	Off	Only affects subD meshes. When set to **on**, this enables calculation of dPdu and dPdv derivatives, allowing antsiotropic shaders to have a smoother appearance. This capability requires storing roughly 100 extra bytes of data per vertex, per keyframe, which is why it's disabled by default.
Adaptive		
subdiv_dicing_camera	None	Can adopt either Scenegraph selection, or Nodegraph selection.
pixel_error	0	Sets the acceptable tesselation error, in pixels, as seen from the camera. If there's no displacement, **pixel_error** represents the distance from the subdivided mesh to the *true* or *limit* surface. The smaller the error, the closer to the limit surface a mesh is (with less inter-frame popping when the subdivision level jumps). If displacement is enabled, **pixel_error** represents the size —in pixels— of the subdivided quads. A value of 0 disables adaptive subdivision.
subdiv_adaptive_metric	auto	When set to **auto** subdiv chooses between Edge Length and Flatness modes depending on the displacement of the mesh. If there is displacement, the Edge Length algorithm is used. If not —or if displacement is ignored in the global options— Flatness is used. You can also manually select **Edge Length** or **Flatness**. The subdiv_adaptive_metric attribute controls whether adaptive tessellation is guided by patch edge lengths, or the difference between the limit-surface, and control-surface (measured in pixels at the patch vertices). The **pixel_error** is either the edge-length size, or the error-tolerance.

Control (UI)	Default Value	Function
Nurbs		
nurbs_tesselate_u	5	Controls the amount of tesselation in the U direction.
nurbs_tesselate_v	5	Controls the amount of tesselation in the V direction.
Points & Curves		
point_mode	disk	Controls whether points are displayed as disks, or spheres.
curve_mode	ribbon	Sets curve display to ribbon, thick, or oriented.
curve_basis	b-spline	Sets the function used to approximate the curve.
min_pixel_width	0	Only applies to **curves** primitives. Enables pixel width enlargement, for improved antialiasing.
self_intersectable	On	Only applies to **curves** primitives. Makes curves invisible to rays that start on them.
Shading		
receive_shadows	On	Sets whether or not the object receives shadows.
self_shadows	On	Sets whether or not the object is able to cast shadows on itself.
Subsurface Scattering		
sss_sample_spacing	0.1	Sets the approximate distance —in world space units— between subsurface scattering sample points. It's important to have enough points in the sample cloud to get a clean result, so the sample spacing will probably need to be set a few times lower than the average scattering radius.
sss_max_samples	100000	Sets the maximum number of samples, regardless of the value of **sss_sample_spacing**.
sss_sample_distribution	blue_noise	Selects the method by which Arnold calculates the location of sub-surface scattering samples on the mesh.
sss_use_gi	On	Selects whether or not to account for global illumination when caching lighting information. This has no effect if **GI_diffuse_depth** is set to 0, so the default behavior is **on**. Advanced users may turn this off to reduce point cloud construction speed, or if diffuse color bleeding is not visually important.
sss_faceset	On	Toggles whether or not facesets defined in Katana are used in subsurface scattering.
Displacement		
disp_padding	0	Sets the amount of padding —or spacing— on displacement bounds.
disp_height	1	Sets the displacement scale. If disp_height is set without setting disp_padding, there could be clipping of the resulting displaced mesh.

Control (UI)	Default Value	Function
autobump	Off	Autobump puts the high frequencies of a displacement map into a bump attribute, so fewer subdivision iteration values are needed. With **autobump** set to **On**, the renderer makes a copy of all vertices in the mesh prior to displacement Before shading a point on the displaced surface, the equivalent Pref for that point on the original surface is found, and the displacement shader evaluated there.
omitDisplacement	Off	This is a Katana specific attribute, which —if enabled— blocks displacement shader creation, and connection to geometry.
zero_value	0	Sets the shift to the displacement amount, and varies depending on how the displacement map is generated. A value of 0 means displacement is always outwards. A value of 0.5 means that displacement values from 0.0 to 0.5 will push surface points inwards, and values greater than 0.5 will push outwards.
Ray Flags		
opaque	On	Sets whether the object is opaque or not.
trace_sets	None	A string of objects for which the geometry assigned in the node's CEL attribute won't evaluate, for selected types of rays. For this to work, the shader has to recognise, and use, an array of objects tagged **trace_sets**.
sidedness		
AI_RAY_CAMERA	On	Enable or disable double-sided visibility for camera rays.
AI_RAY_SHADOW	On	Enable or disable double-sided visibility for shadow rays.
AI_RAY_REFLECTED	On	Enable or disable double-sided visibility for reflections.
AI_RAY_REFRACTED	On	Enable or disable double-sided visibility for refractions.
AI_RAY_DIFFUSE	On	Enable or disable double-sided visibility for global illumination diffuse rays.
AI_RAY_SCATTERED	On	Enable or disable atmosphere and volumetric global illumination rays.
AI_RAY_GLOSSY	On	
AI_RAY_SUBSURFACE	On	
visibility		
AI_RAY_CAMERA	On	Enable or disable visibility of the object to the camera.
AI_RAY_SHADOW	On	Enables or disables inclusion of the object when calculating shadow rays.
AI_RAY_REFLECTED	On	Enables or disables reflection rays.
AI_RAY_DIFFUSE	On	Enables or disables diffuse global illumination rays.
AI_RAY_GLOSSY	On	Enable or disable blurred reflection rays.
AI_RAY_SCATTERED	On	Enable or disable atmosphere and volumetric global illumination rays.
AI_RAY_SUBSURFACE	On	

ArnoldOutputChannelDefine

Builds the parameters used by Arnold during render.

Note: driverParameters change depending on the selected driver.

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. 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.
driver	driver_exr	Sets the output channel driver to use. The controls available under **driverParameters** are dependant on the driver selected: • driver_display • driver_exr • driver_jpeg • driver_null • driver_png • driver_tiff
driver		
Browse...	N/A	Brings up the file browser or your studio's asset management browser and enables you to select the driver to use.
channel	RGBA	Sets the output channel.
type	RGBA	Sets the output channel type: BYTE, INT, LONG, BOOL, FLOAT, DOUBLE, RGB, RGBA, ABSRGB, VECTOR, POINT, POINT2, STRING, POINTER, ARRAY, MATRIX, and ENUM
filter	gaussian_filter	Sets the output channel filter. The controls available under filterParameters are dependant on the filter selected: box_filter, catrom2d_filter, catrom filter, closest_filter, cone_filter, cook_filter, cubic_filter, disk_filter, farthest_filter, gaussian_filter, heatmap_filter, mitnet_filter, sinc_filter, triangle_filter, variance_filter, video_filter
filter		
Browse...	N/A	Brings up the file browser or your studio's asset management browser and enables you to select the filter to use.
driverParameters - these controls are dependant on the driver selected		
filterParameters - these controls are dependant on the filter selected		

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 (e.g. geometry.point.P), the attribute is copied to the location specified by toAttr (e.g. 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	Defines the **copyFrom** location.
fromRoot		
Adopt Scenegraph Selection	N/A	The currently selected **Scene Graph** location is used to populate the parameter.
Adopt Selected Nodegraph Node	N/A	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	N/A	Selects the location specified by this parameter in the **Scene Graph** tab.
fromAttr	none	Defines the attribute that is copied.
toRoot	/root/world	Defines the copyTo location.
toRoot		
Adopt Scenegraph Selection	N/A	The currently selected **Scene Graph** location is used to populate the parameter.
Adopt Selected Nodegraph Node	N/A	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	N/A	Selects the location specified by this parameter in the **Scene Graph** tab.
toAttr	none	Defines the location where the attribute is copied to.

Control (UI)	Default Value	Function
toCEL	none	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. Specifies **Scene Graph** locations on which the operation or assignment acts. The **Scene Graph** locations are specified using the Collection Expression Language (CEL). For more information, see the CEL Reference document found on the documentation HTML page (accessed through the **Help** > **Documentation** menu option).
toCEL > Add Statements		
Paths	N/A	Adds a **Paths** list to this CEL parameter.
Collections	N/A	Adds a **Collections** list to this CEL parameter.
Custom	N/A	Adds a **Custom** parameter to this CEL parameter.
Append Scenegraph Selection	N/A	Adds a **Paths** list to this CEL parameter and places selected Scene Graph locations in the new list.
Replace With Scenegraph Selection	N/A	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	N/A	Copies this CEL statement to the clipboard.
Paste CEL Statement	N/A	Removes any parameters within this CEL parameter and pastes the CEL statement in the clipboard to this parameter.
Replace With Parameter Expression	N/A	Converts the current CEL parameter into an expression.
toCEL: Paths > Action		
Add Scenegraph Selection	N/A	Adds the currently selected Scene Graph location to this list.
Remove Scenegraph Selection	N/A	Removes the currently selected Scene Graph location from this list.
Remove Selected Paths	N/A	Removes the path(s), selected in this **Paths** list, from this list.
Select All	N/A	Selects all the paths in this list.
Select Selected Paths In Scenegraph	N/A	Selects the Scene Graph locations of the selected paths in this list.
Copy Selected Paths to Clipboard	N/A	Copies the selected paths from this list to the clipboard.
Show Extended View...	N/A	Brings up a dialog with the contents of this **Paths** list.
toCEL: Collections > Action		
Add Collections From Scenegraph Selection...	N/A	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.

Control (UI)	Default Value	Function
Add Scene Root Collections...	N/A	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	N/A	Removes the selected collection(s) from this list.
Select All	N/A	Selects all the collections in this list.
Copy Selected Paths to Clipboard	N/A	Copies all the selected collections and their paths to the clipboard.
"Find And Select" Selected Items...	N/A	

AttributeEditor

The AttributeEditor node is used to edit specific attributes of objects in the **Scene Graph**.

Control (UI)	Default Value	Function
exclusivity	none	Exclusivity locks the interactive **Viewer** tab edits of a location to this node. Specifies **Scene Graph** locations on which the operation or assignment acts. The **Scene Graph** locations are specified using the Collection Expression Language (CEL). For more information, see the CEL Reference document found on the documentation HTML page (accessed through the **Help** > **Documentation** menu option).
exclusivity > Add Statements		
Paths	N/A	Adds a **Paths** list to this CEL parameter.
Collections	N/A	Adds a **Collections** list to this CEL parameter.
Custom	N/A	Adds a **Custom** parameter to this CEL parameter.
Append Scenegraph Selection	N/A	Adds a **Paths** list to this CEL parameter and places selected Scene Graph locations in the new list.
Replace With Scenegraph Selection	N/A	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	N/A	Copies this CEL statement to the clipboard.
Paste CEL Statement	N/A	Removes any parameters within this CEL parameter and pastes the CEL statement in the clipboard to this parameter.
Replace With Parameter Expression	N/A	Converts the current CEL parameter into an expression.
exclusivity: Paths > Action		

Control (UI)	Default Value	Function
Add Scenegraph Selection	N/A	Adds the currently selected Scene Graph location to this list.
Remove Scenegraph Selection	N/A	Removes the currently selected Scene Graph location from this list.
Remove Selected Paths	N/A	Removes the path(s), selected in this **Paths** list, from this list.
Select All	N/A	Selects all the paths in this list.
Select Selected Paths In Scenegraph	N/A	Selects the Scene Graph locations of the selected paths in this list.
Copy Selected Paths to Clipboard	N/A	Copies the selected paths from this list to the clipboard.
Show Extended View...	N/A	Brings up a dialog with the contents of this **Paths** list.
exclusivity: Collections > Action		
Add Collections From Scenegraph Selection...	N/A	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...	N/A	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	N/A	Removes the selected collection(s) from this list.
Select All	N/A	Selects all the collections in this list.
Copy Selected Paths to Clipboard	N/A	Copies all the selected collections and their paths to the clipboard.
"Find And Select" Selected Items...	N/A	
Existing Edits Table	N/A	Attributes dragged onto the **Drop Attributes Here** hotbox, located below the table, are placed here for editing.
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.
Exisiting Edits Table > [right-click menu]		
Go To Location	N/A	Selects the Scene Graph location these edits affect.
Disable Overrides	N/A	Disables the edit

Control (UI)	Default Value	Function
Enable Overrides	N/A	Enables the edit
Move Overrides To Selected Scenegraph Location...	N/A	Moves the override to the Scene Graph location currently selected.
Copy Overrides To Selected Scenegraph Location...	N/A	Copies the override to the Scene Graph location currently selected.
Delete Overrides	N/A	Deletes the override.
Drop Attributes Here	N/A	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	Specifies **Scene Graph** locations on which the operation or assignment acts. The **Scene Graph** locations are specified using the Collection Expression Language (CEL). For more information, see the CEL Reference document found on the documentation HTML page (accessed through the **Help** > **Documentation** menu option).
CEL > Add Statements		
Paths	N/A	Adds a **Paths** list to this CEL parameter.
Collections	N/A	Adds a **Collections** list to this CEL parameter.
Custom	N/A	Adds a **Custom** parameter to this CEL parameter.
Append Scenegraph Selection	N/A	Adds a **Paths** list to this CEL parameter and places selected Scene Graph locations in the new list.
Replace With Scenegraph Selection	N/A	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	N/A	Copies this CEL statement to the clipboard.
Paste CEL Statement	N/A	Removes any parameters within this CEL parameter and pastes the CEL statement in the clipboard to this parameter.
Replace With Parameter Expression	N/A	Converts the current CEL parameter into an expression.

Control (UI)	Default Value	Function
CEL: Paths > Action		
Add Scenegraph Selection	N/A	Adds the currently selected Scene Graph location to this list.
Remove Scenegraph Selection	N/A	Removes the currently selected Scene Graph location from this list.
Remove Selected Paths	N/A	Removes the path(s), selected in this **Paths** list, from this list.
Select All	N/A	Selects all the paths in this list.
Select Selected Paths In Scenegraph	N/A	Selects the Scene Graph locations of the selected paths in this list.
Copy Selected Paths to Clipboard	N/A	Copies the selected paths from this list to the clipboard.
Show Extended View...	N/A	Brings up a dialog with the contents of this **Paths** list.
CEL: Collections > Action		
Add Collections From Scenegraph Selection...	N/A	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...	N/A	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	N/A	Removes the selected collection(s) from this list.
Select All	N/A	Selects all the collections in this list.
Copy Selected Paths to Clipboard	N/A	Copies all the selected collections and their paths to the clipboard.
"Find And Select" Selected Items...	N/A	
File Path	none	Describes the filepath to an Attributes File.
File Path		
Browse...	N/A	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	N/A	Changes the name of the node to match the filename but without the path or extension.
Custom File Parser	none	Specifies the .so file with the Attributes File parser. Leave it empty to use the default one.
Custom File Parser		
Browse...	N/A	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	N/A	Changes the name of the node to match the filename but without the path or extension.

Control (UI)	Default Value	Function
Attribute Group Name	attributeFile	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).
Apply When	immediate	Determines when the script runs: • **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 scenegraphLocationModifers 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 GraphLocationModifers 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.
timing		
mode	Current Frame	Sets the timing mode to apply to the asset: • **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	Specifies **Scene Graph** locations on which the operation or assignment acts. The **Scene Graph** locations are specified using the Collection Expression Language (CEL). For more information, see the CEL Reference document found on the documentation HTML page (accessed through the **Help** > **Documentation** menu option).
CEL > Add Statements		
Paths	N/A	Adds a **Paths** list to this CEL parameter.
Collections	N/A	Adds a **Collections** list to this CEL parameter.
Custom	N/A	Adds a **Custom** parameter to this CEL parameter.
Append Scenegraph Selection	N/A	Adds a **Paths** list to this CEL parameter and places selected Scene Graph locations in the new list.
Replace With Scenegraph Selection	N/A	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	N/A	Copies this CEL statement to the clipboard.
Paste CEL Statement	N/A	Removes any parameters within this CEL parameter and pastes the CEL statement in the clipboard to this parameter.
Replace With Parameter Expression	N/A	Converts the current CEL parameter into an expression.
CEL: Paths > Action		
Add Scenegraph Selection	N/A	Adds the currently selected Scene Graph location to this list.
Remove Scenegraph Selection	N/A	Removes the currently selected Scene Graph location from this list.
Remove Selected Paths	N/A	Removes the path(s), selected in this **Paths** list, from this list.
Select All	N/A	Selects all the paths in this list.
Select Selected Paths In Scenegraph	N/A	Selects the Scene Graph locations of the selected paths in this list.
Copy Selected Paths to Clipboard	N/A	Copies the selected paths from this list to the clipboard.
Show Extended View...	N/A	Brings up a dialog with the contents of this **Paths** list.

Control (UI)	Default Value	Function
CEL: Collections > Action		
Add Collections From Scenegraph Selection...	N/A	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...	N/A	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	N/A	Removes the selected collection(s) from this list.
Select All	N/A	Selects all the collections in this list.
Copy Selected Paths to Clipboard	N/A	Copies all the selected collections and their paths to the clipboard.
"Find And Select" Selected Items...	N/A	
applyWhen	immediate	Determines when the script runs: • **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	
recursiveEnable	No	
resolveIds	none	
applyWhen: deferred: modifierNameMode: specified		
modifierName	none	
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.

Control (UI)	Default Value	Function
args: Messer		
displacement	2	Sets the maximum amount the geometry is displaced using the Messer AMP plug-in.

AttributeModifierResolve

This node resolves Attribute Modifier Plug-ins (AMPs).

Control (UI)	Default Value	Function
resolveWithIds	all	Sets which attribute modifiers are resolved: • **all** - resolves all modifiers. • **specified** - resolves only those modifiers specified in the **specifiedResolveIds** field.
resolveWithIds: specified		
specifiedResolveIds	none	Determines which modifiers are resolved.

AttributeScript

This node runs Python scripts with the ability to query and modify attributes at **Scene Graph** locations specified by the CEL parameter.

These functions are available:

- **GetAttr**(attributePath, [inherit=False, asAttr=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.
- **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.
- **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])
  Returns the local or worldspace transformation as a 16-element list of float values representing a 4x4 transform matrix.
- **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 preexisting 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	Specifies **Scene Graph** locations on which the operation or assignment acts. The **Scene Graph** locations are specified using the Collection Expression Language (CEL). For more information, see the CEL Reference document found on the documentation HTML page (accessed through the **Help** > **Documentation** menu option).
toCEL > Add Statements		
Paths	N/A	Adds a **Paths** list to this CEL parameter.
Collections	N/A	Adds a **Collections** list to this CEL parameter.
Custom	N/A	Adds a **Custom** parameter to this CEL parameter.
Append Scenegraph Selection	N/A	Adds a **Paths** list to this CEL parameter and places selected Scene Graph locations in the new list.
Replace With Scenegraph Selection	N/A	Removes any parameters within this CEL parameter and creates a new Paths list and populates it with any selected Scene Graph locations.

Control (UI)	Default Value	Function
Copy CEL Statement As Text	N/A	Copies this CEL statement to the clipboard.
Paste CEL Statement	N/A	Removes any parameters within this CEL parameter and pastes the CEL statement in the clipboard to this parameter.
Replace With Parameter Expression	N/A	Converts the current CEL parameter into an expression.
toCEL: Paths > Action		
Add Scenegraph Selection	N/A	Adds the currently selected Scene Graph location to this list.
Remove Scenegraph Selection	N/A	Removes the currently selected Scene Graph location from this list.
Remove Selected Paths	N/A	Removes the path(s), selected in this **Paths** list, from this list.
Select All	N/A	Selects all the paths in this list.
Select Selected Paths In Scenegraph	N/A	Selects the Scene Graph locations of the selected paths in this list.
Copy Selected Paths to Clipboard	N/A	Copies the selected paths from this list to the clipboard.
Show Extended View...	N/A	Brings up a dialog with the contents of this **Paths** list.
toCEL: Collections > Action		
Add Collections From Scenegraph Selection...	N/A	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...	N/A	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	N/A	Removes the selected collection(s) from this list.
Select All	N/A	Selects all the collections in this list.
Copy Selected Paths to Clipboard	N/A	Copies all the selected collections and their paths to the clipboard.
"Find And Select" Selected Items...	N/A	

Control (UI)	Default Value	Function
applyWhen	immediate	Determines when the script is run: • **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 GraphLocationModifers 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 GraphLocationModifers group attribute and are evaluated by a LookFileResolve node or by the first implicit resolver if no LookFileResolve node is present. **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. • **during material resolve** - the script and its arguments are added as attributes under the **material.scenegraphLocationModifers** 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.		
recursiveEnable	No	
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 persistant 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 persistant 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: • paths • CEL
mode: paths		
paths	none	Sets the paths of the attribute. For example, /root/world/geo
mode:paths: paths > Add Locations		
Path	N/A	Adds another path to this parameter's list of paths.
Append Scenegraph Selection	N/A	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	N/A	Removes all existing paths and replaces them with paths populated with the currently selected **Scene Graph** locations.
Append Nodegraph Node Locations	N/A	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	N/A	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	N/A	Removes all paths from this parameter.
mode: CEL		
celSelection	none	Sets the attribute location to be overridden.
mode: CEL: celSelection > Add Statements		
Paths	N/A	Adds a **Paths** list to this CEL parameter.
Collections	N/A	Adds a **Collections** list to this CEL parameter.
Custom	N/A	Adds a **Custom** parameter to this CEL parameter.
Append Scenegraph Selection	N/A	Adds a **Paths** list to this CEL parameter and places selected **Scene Graph** locations in the new list.
Replace With Scenegraph Selection	N/A	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	N/A	Copies this CEL statement to the clipboard.
Paste CEL Statement	N/A	Removes any parameters within this CEL parameter and pastes the CEL statement in the clipboard to this parameter.
Replace With Parameter Expression	N/A	Converts the current CEL parameter into an expression.
mode: CEL: celSelection: Paths > Action		
Add Scenegraph Selection	N/A	Adds the currently selected **Scene Graph** location to this list.

Control (UI)	Default Value	Function
Remove Scenegraph Selection	N/A	Removes the currently selected **Scene Graph** location from this list.
Remove Selected Paths	N/A	Removes the path(s), selected in this **Paths** list, from this list.
Select All	N/A	Selects all the paths in this list.
Select Selected Paths In Scenegraph	N/A	Selects the **Scene Graph** locations of the selected paths in this list.
Copy Selected Paths to Clipboard	N/A	Copies the selected paths from this list to the clipboard.
Show Extended View...	N/A	Brings up a dialog with the contents of this **Paths** list.
mode: CEL: celSelection: Collections > Action		
Add Collections From Scenegraph Selection...	N/A	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...	N/A	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	N/A	Removes the selected collection(s) from this list.
Select All	N/A	Selects all the collections in this list.
Copy Selected Paths to Clipboard	N/A	Copies all the selected collections and their paths to the clipboard.
"Find And Select" Selected Items...	N/A	
action	Create/Override	Describes the action to apply to the Scene Graph attribute: • Create/Override • Delete • Force Default
attributeName	none	The attribute name to override.
action: Create/Override		
attributeType	double	The type of the attribute that is overridden: • 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.

Control (UI)	Default Value	Function
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		
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 hot-spot to use that attribute.

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	The object to constrain.
basePath		
Adopt Scenegraph Selection	N/A	The currently selected **Scene Graph** location is used to populate the parameter.
Adopt Selected Nodegraph Node	N/A	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	N/A	Selects the location specified by this parameter in the **Scene Graph** tab.
targetPath	none	The object(s) to constrain the object in **basePath** to.
targetPath > Add Locations		
Path	N/A	Adds another path to this parameter's list of paths.
Append Scenegraph Selection	N/A	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	N/A	Removes all existing paths and replaces them with paths populated with the currently selected **Scene Graph** locations.
Append Nodegraph Node Locations	N/A	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	N/A	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.

Control (UI)	Default Value	Function
Clear All	N/A	Removes all paths from this parameter.
targetOrigin	object	Sets how the center of the target object is calculated: • **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	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.
baseRotateAxis	0.0, 1.0, 0.0	The axis of the base object that is rotated to maintain orientation to the target. Adjusting these values changes the rotation of the base object, while keeping the aim constant.
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.

BoundsAdjust

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

Control (UI)	Default Value	Function
targetPath	none	Defines which geometry's bounding box is being adjusted.
targetPath		
Adopt Scenegraph Selection	N/A	The currently selected **Scene Graph** location is used to populate the parameter.
Adopt Selected Nodegraph Node	N/A	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	N/A	Selects the location specified by this parameter in the **Scene Graph** tab.
action	Strip Bounds	Sets the type of bounding adjustment to apply: • **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.

Control (UI)	Default Value	Function
action: Pad Bounds (%)		
percentage	0	Specify percentage to add to the original bounding box size. This is a keyable attribute.
when	immediate	Sets when the specified adjustment is applied: • **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)		
localSpace	0	Specifies the number of local space units to pad the bounding box.
when	immediate	Sets when the specified adjustment is applied: • **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.
cameraLocation		
Adopt Scenegraph Selection	N/A	The currently selected **Scene Graph** location is used to populate the parameter.
Adopt Selected Nodegraph Node	N/A	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	N/A	Selects the location specified by this parameter in the **Scene Graph** tab.
geometry		
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 product 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 ScenegraphLocations 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	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.
name		
Parent to Scenegraph Selection	N/A	Sets the parent location of the object created to be the current **Scene Graph** selection.
Parent to /root/world/geo/...	N/A	Sets the parent location of the object created to be /root/world/geo.
Parent to /root/world/lgt/...	N/A	Sets the parent location of the object created to be /root/world/lgt.
Parent to /root/world/cam/...	N/A	Sets the parent location of the object created to be /root/world/cam.
Select In Scenegraph	N/A	Selects the location specified by this parameter in the **Scene Graph** tab.
projection	perspective	Toggles the projection type of the camera: • **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.

Control (UI)	Default Value	Function
includeInCameraList	Yes	When enabled, the camera is visible in the camera list on the /root/world location, under **globals**.
transform		
interface	SRT Values	Sets the transform control layout: • **SRT Values** - exposes the scale, rotation, and translation controls. • **Transform Matrix** - exposes a matrix to control transformations.
transformOrder	Scale Rotate Translate	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	XYZ	Sets the order in which rotation is applied: • **XYZ** • **XZY** • **YXZ** • **YZX** • **ZXY** • **ZYX**
transform > interface: SRT Values		
translate	0.0, 0.0, 0.0	Controls camera translation on the xyz axes.
rotate	0.0, 0.0, 0.0	Controls camera rotation on the xyz axes.
scale	1.0, 1.0, 1.0	Controls camera scale on the xyz axes.
transform > interface: Transform Matrix		
matrix	1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 1.0	Controls transformations using a matrix in place of individual SRT controls.
makeInteractive	Yes	When set to **Yes**, you can drag objects in the Viewer and Katana writes the changes made to this object in this node.

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 panel. 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.
cameraLocation		
Adopt Scenegraph Selection	N/A	The currently selected **Scene Graph** location is used to populate the parameter.
Adopt Selected Nodegraph Node	N/A	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	N/A	Selects the location specified by this parameter in the **Scene Graph** tab.
name	plane	Sets the name of the image plane created.
imagePath	none	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. **Note:** Image sequences must contain a padded frame number.
imagePath		
Browse...	N/A	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	N/A	Changes the name of the node to match the filename but without the path or extension.
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	When set to **Yes**, this image plane is only displayed when looking through the camera it is attached to. When set to **No**, you can see the image plane in all views.
displayMode	RGBA	Sets the image plane display mode: • **None** • **Outline** • **RGB** • **RGBA**

Control (UI)	Default Value	Function
fit	Best	Controls how the image file fits into the image plane if there is a mis-match between aspect ratios: • **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. **Note**: To control the image's SRT values directly, enable the **manualPlacementSettings** checkbox.
crop		
left	0	Sets the amount of manual crop to apply to the edges of the image plane.
bottom	0	
right	1	
top	1	
manualPlacementSettings	No	When set to **Yes**, you can adjust the image manually using its SRT values.
manualPlacementSettings: Yes > size		
x	1	
y	1	
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	Describes the **Scene Graph** location of the object to constrain. This should be plane geometry.
basePath		
Adopt Scenegraph Selection	N/A	The currently selected **Scene Graph** location is used to populate the parameter.
Adopt Selected Nodegraph Node	N/A	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	N/A	Selects the location specified by this parameter in the **Scene Graph** tab.
targetPath	none	Describes the camera location to which the **basePath** object is constrained.
targetPath		
Adopt Scenegraph Selection	N/A	The currently selected **Scene Graph** location is used to populate the parameter.
Adopt Selected Nodegraph Node	N/A	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	N/A	Selects the location specified by this parameter in the **Scene Graph** tab.
Adjust Path Relative To 'basePath'	N/A	Converts the current **targetPath** to a path relative to the **basePath**. If the **targetPath** is an expression, it is converted to a constant.
distance	1	Sets distance from the camera at which the base **Scene Graph** location is constrained.
planeType	XY	The type of plane that is constrained: • **XY** • **XZ**

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	Describes the **Scene Graph** location of the object to constrain.
basePath		
Adopt Scenegraph Selection	N/A	The currently selected **Scene Graph** location is used to populate the parameter.
Adopt Selected Nodegraph Node	N/A	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	N/A	Selects the location specified by this parameter in the **Scene Graph** tab.
targetPath	none	Describes the object(s) location to which the **basePath** object is constrained.
targetPath > Add Locations		
Path	N/A	Adds another path to this parameter's list of paths.
Append Scenegraph Selection	N/A	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	N/A	Removes all existing paths and replaces them with paths populated with the currently selected **Scene Graph** locations.
Append Nodegraph Node Locations	N/A	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	N/A	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	N/A	Removes all paths from this parameter.
pad		
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.
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.

CollectionCreate

Collections are used to store a CEL statement. They are stored as attributes at /root. As they are simply

Control (UI)	Default Value	Function
name	none	Sets the name of the collection.
CEL	none	Specifies **Scene Graph** locations to store as part of this collection. The **Scene Graph** locations are specified using the Collection Expression Language (CEL). For more information, see the CEL Reference document found on the documentation HTML page (accessed through the **Help** > **Documentation** menu option).
CEL > Add Statements		
Paths	N/A	Adds a **Paths** list to this CEL parameter.
Collections	N/A	Adds a **Collections** list to this CEL parameter.
Custom	N/A	Adds a **Custom** parameter to this CEL parameter.
Append Scenegraph Selection	N/A	Adds a **Paths** list to this CEL parameter and places selected Scene Graph locations in the new list.
Replace With Scenegraph Selection	N/A	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	N/A	Copies this CEL statement to the clipboard.
Paste CEL Statement	N/A	Removes any parameters within this CEL parameter and pastes the CEL statement in the clipboard to this parameter.
Replace With Parameter Expression	N/A	Converts the current CEL parameter into an expression.
CEL: Paths > Action		
Add Scenegraph Selection	N/A	Adds the currently selected Scene Graph location to this list.
Remove Scenegraph Selection	N/A	Removes the currently selected Scene Graph location from this list.
Remove Selected Paths	N/A	Removes the path(s), selected in this **Paths** list, from this list.
Select All	N/A	Selects all the paths in this list.
Select Selected Paths In Scenegraph	N/A	Selects the Scene Graph locations of the selected paths in this list.
Copy Selected Paths to Clipboard	N/A	Copies the selected paths from this list to the clipboard.
Show Extended View...	N/A	Brings up a dialog with the contents of this **Paths** list.
CEL: Collections > Action		
Add Collections From Scenegraph Selection...	N/A	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...	N/A	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.

Control (UI)	Default Value	Function
Remove Selected Paths	N/A	Removes the selected collection(s) from this list.
Select All	N/A	Selects all the collections in this list.
Copy Selected Paths to Clipboard	N/A	Copies all the selected collections and their paths to the clipboard.
"Find And Select" Selected Items...	N/A	

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	Sets the **Scene Graph** location(s) to add to the constraint list.
locations > Add Locations		
Path	N/A	Adds another path to this parameter's list of paths.
Append Scenegraph Selection	N/A	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	N/A	Removes all existing paths and replaces them with paths populated with the currently selected **Scene Graph** locations.
Append Nodegraph Node Locations	N/A	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	N/A	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	N/A	Removes all paths from this parameter.
mode	add	Sets the constraint mode: • **add** - currently the only option, adds locations to 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: • **globally** • **relative scope**
coordinateSystemName	none	Specifies the unique name of the coordinate system to create.
referenceLocation	none	Specifies the Scene Graph location whose global transform defines the coordinate system.
scope: relative scoped		
baseLocation	none	
relativeLocation	none	
scope: relative scope: baseLocation		
Adopt Scenegraph Selection	N/A	The currently selected **Scene Graph** location is used to populate the parameter.
Adopt Selected Nodegraph Node	N/A	Creates an expression from the currently selected **Node Graph** node linking the **Scene Graph** location created by that node to this parameter.
scope: relative scope: relativeLocation		
Adopt Scenegraph Selection	N/A	The currently selected **Scene Graph** location is used to populate the parameter.
Adopt Selected Nodegraph Node	N/A	Creates an expression from the currently selected **Node Graph** node linking the **Scene Graph** location created by that node to this parameter.

3 Nodes D–H

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.

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
cueSettings		
outlineFile	none	Sets the location where the outline file is written.

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 47.

Control (UI)	Default Value	Function
basePath	none	Sets the location of the camera or light to dolly.
basePath		
Adopt Scenegraph Selection	N/A	The currently selected **Scene Graph** location is used to populate the parameter.
Adopt Selected Nodegraph Node	N/A	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	N/A	Selects the location specified by this parameter in the **Scene Graph** tab.
targetPath	none	Sets the location of the object(s) to fit within the field of view.
targetPath > Add Locations		

Control (UI)	Default Value	Function
Path	N/A	Adds another path to this parameter's list of paths.
Append Scenegraph Selection	N/A	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	N/A	Removes all existing paths and replaces them with paths populated with the currently selected **Scene Graph** locations.
Append Nodegraph Node Locations	N/A	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	N/A	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	N/A	Removes all paths from this parameter.
targetBounds	box	The type of bounds to use for the target object(s): • **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	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.

Dot

The Dot node performs no operation on the data passing through it. Its only purpose is to improve the appearance and layout of your Node Graph.

Tip: You can insert Dot nodes on-the-fly during link creation by pressing the **.** (**fullstop**) 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.

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 subset of faces on a single mesh.

Note: Face sets are only applicable with subdivision meshes.

Control (UI)	Default Value	Function
meshLocation	none	Describes the location of the mesh for which the set is created. **Note:** This is only applicable with subdivision meshes.
meshLocation		
Adopt Scenegraph Selection	N/A	The currently selected **Scene Graph** location is used to populate the parameter.
Adopt Selected Nodegraph Node	N/A	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	N/A	Selects the location specified by this parameter in the **Scene Graph** tab.
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.
selection		
Adopt Faces From Viewer	N/A	Stores the list of faces that are currently selected in the **Viewer** in this parameter.

Fork3D

Fork3D creates multiple instances of the upstream Scene Graph when the value of a fork variable is evaluated. The evaluation of upstream parameters varies depending on which branch of the fork node has been evaluated for a particular variable.

Fork3D should be treated as an advanced function and used with care, particularly nested Fork3D nodes, as they can result in performance issues.

Notes: This node accesses **getForkVar** (node, name, default) and retrieves a variable from the current branch of a downstream Fork3D node. The node argument must be fetched with the **getNode()** function. The function returns a string or float number, depending on the type of variable inside the Fork3D node.

Control (UI)	Default Value	Function
variables	output 0, output 1	By default the Fork3D node has two variable outputs. Additional outputs can be added by choosing the Add dropdown menu and choosing add output. A number variable or string variable can be added to each output. The **user** parameter can be used to add new parameters such as number arrays, string arrays, button, or toolbars.

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 44.

Control (UI)	Default Value	Function
basePath	none	Sets the object to constrain.
basePath		
Adopt Scenegraph Selection	N/A	The currently selected **Scene Graph** location is used to populate the parameter.
Adopt Selected Nodegraph Node	N/A	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	N/A	Selects the location specified by this parameter in the **Scene Graph** tab.
targetPath	none	Sets the object to constrain the **basePath** the object. You can add more locations using the **Add Locations** dropdown.
targetPath > Add Locations		
Path	N/A	Adds another path to this parameter's list of paths.
Append Scenegraph Selection	N/A	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	N/A	Removes all existing paths and replaces them with paths populated with the currently selected **Scene Graph** locations.
Append Nodegraph Node Locations	N/A	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.

Control (UI)	Default Value	Function
Replace with Nodegraph Node Locations	N/A	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	N/A	Removes all paths from this parameter.
targetBounds	box	Sets the type of bounds to use for the target object(s). • **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.
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.

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 is set using the KATANA_GAFFER_DEFAULT_PROFILE environement variable.
sync selection	Disabled	When enabled, selecting a gaffer light within the **Parameters** tab selects its location within the **Scene Graph** tab.

Control (UI)	Default Value	Function
Add Light	N/A	Adds a light to the gaffer table.
Add Rig	N/A	Adds a rig to the gaffer table.
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 constaints.
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.
[Gaffer object table]	gaffer	Displays a list of all objects controlled by this Gaffer node. The object table contains the following information: • **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. **Note:** The **Object**, **Material**, and **Linking** tabs under the object table are populated depending on the object selected.
[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 constaints.

Control (UI)	Default Value	Function
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.
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.		

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**.

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** 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)	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 Selectde 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.

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** 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 Selectde 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.

4 Nodes I–K

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
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the original image at 0 and the full effect at 1.
displayWindow	Background	The frame size to output in the event that the **fg** and **bg** inputs are different sizes: • **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		
[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: • **off** – no snapping is active. • **frames** – point movement snaps to the fr • **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
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the **bg** image at 0 and the full merge effect at 1.
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the merge is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the merge is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that, by default, the merge is limited to the fringe (semi-transparent areas).
operation	ImageAtop	If you don't want to layer the images together using the **ImageAtop** compositing operation, select the operation to use instead. The following conventions apply to the below operation descriptions: • **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
		The available operations (based on the Porter & Duff paper) are: • **ImageAtop** - Fb+B(1-f). This shows the shape of the background, with the foreground covering the background where the images overlap. • **ImageAverage** - (F+B)/2. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • **ImageDifference** - abs(F-B). This shows how much the pixels differ and is useful for comparing two very similar images. • **ImageDivide** - B/F. This divides the background values by the foreground values. • **ImageExclusion** - F+B-2FB. This is a more photographic form of **ImageDifference**. • **ImageFrom** - F-B. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see **ImageMinus**. • **ImageGeometric** - 2FB/(F+B). This is another way of averaging two images. Visually, it's close to **ImageMin**. • **ImageHypot** - sqrt(F*F+B*B). 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 is useful for adding reflections, as an alternative to **ImageScreen**. • **ImageIn** - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • **ImageMatte** - Ff*B(1-f). This is a premultiplied **ImageOver**. Use unpremultiplied images with this operation. • **ImageMax** - 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. • **ImageMin** - min(F,B). This takes the minimum values of both images. • **ImageMinus** - B-F. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see **ImageFrom**. • **ImageMultiply** - 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.

Control (UI)	Default Value	Function
		• **ImageOut** - 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. • **ImageOver** - 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. • **ImagePlus** - 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. • **ImageScreen** - F+B-FB. This is similar to **ImageHypot**, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • **ImageUnder** - F(1-b)+B. This is the reverse of the **ImageOver** 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	The frame size to output in the event that the **fg** and **bg** inputs are different sizes: • **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.

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 87 and ImageHypot on page 90.

Control (UI)	Default Value	Function
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.

Control (UI)	Default Value	Function
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the **bg** image at 0 and the full merge effect at 1.
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the merge is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the merge is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the merge is limited to the fringe (semi-transparent areas).
operation	ImageAverage	If you don't want layer the images together using the **ImageAverage** compositing operation, select the operation to use instead. The following conventions apply to the below operation descriptions: • **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. The available operations (based on the Porter & Duff paper) are: • **ImageAtop** – Fb+B(1-f). This shows the shape of the background, with the foreground covering the background where the images overlap. • **ImageAverage** – (F+B)/2. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • **ImageDifference** – abs(F-B). This shows how much the pixels differ and is useful for comparing two very similar images. • **ImageDivide** – B/F. This divides the background values by the foreground values. • **ImageExclusion** – F+B-2FB. This is a more photographic form of **ImageDifference**. • **ImageFrom** – F-B. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see **ImageMinus**.

Control (UI)	Default Value	Function
		• **ImageGeometric** - 2FB/(F+B). This is another way of averaging two images. Visually, it's close to **ImageMin**. • **ImageHypot** - sqrt(F*F+B*B). 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 is useful for adding reflections, as an alternative to **ImageScreen**. • **ImageIn** - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • **ImageMatte** - Ff*B(1-f). This is a premultiplied **ImageOver**. Use unpremultiplied images with this operation. • **ImageMax** - 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. • **ImageMin** - min(F,B). This takes the minimum values of both images. • **ImageMinus** - B-F. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see **ImageFrom**. • **ImageMultiply** - 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. • **ImageOut** - 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. • **ImageOver** - 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. • **ImagePlus** - 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. • **ImageScreen** - F+B-FB. This is similar to **ImageHypot**, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • **ImageUnder** - F(1-b)+B. This is the reverse of the **ImageOver** 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	The frame size to output in the event that the **fg** and **bg** inputs are different sizes: • **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.

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.
color > RGB		
red	0	Sets the red value of the background.
green	0	Sets the green value of the background.
blue	0	Sets the blue value of the background.
alpha	0	Sets the alpha value of the background.
color > HSL		
hue	0	Sets the hue of the background.
saturation	0	Sets the saturation of the background.
lightness	0	Sets the lightness of the background.
alpha	0	Sets the alpha value of the background.
color > HSV		
hue	0	Sets the hue of the background.
saturation	0	Sets the saturation of the background.
value	0	Sets the value of the background.
alpha	0	Sets the alpha value of the background.

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.
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.
color > RGB		
red	0	Sets the red value of the pixels.
green	0	Sets the green value of the pixels.
blue	0	Sets the blue value of the pixels.
alpha	1	Sets the alpha value of the pixels.
color > HSL		
hue	0	Sets the hue of the pixels.
saturation	0	Sets the saturation of the pixels.
lightness	0	Sets the lightness of the pixels.
alpha	1	Sets the alpha value of the pixels.
color > HSV		
hue	0	Sets the hue of the pixels.
saturation	0	Sets the saturation of the pixels.
value	0	Sets the value of the pixels.
alpha	1	Sets the alpha value of the pixels.

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
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the blur effect is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the blur effect is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the blur effect is limited to the fringe (semi-transparent areas).
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	Sets the vertical radius (in pixels) within which pixels are compared to calculate the blur. Higher values widen the compare area, producing more blur. By default, this value is the same as **xAmount**.
filter	Gaussian	Select the filtering algorithm to use: • **Box** • **Triangle** • **Gaussian** • **Bell** • **BSpline** • **Mitchell**

Control (UI)	Default Value	Function
borderExtend	Clamp	Select the border extend method for pixels required beyond the image borders: • **Mirror** • **Clamp** • **Background**
useOverscan	globals.compDefaults.useOverscan	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. However, if you are unsure of this procedure or the intergrity of overscanned areas is unknown, it's safer to leave this disabled. Overscan refers to image pixel data outside of the displayWindow and can be inspected using options in the Monitor. For information on explicitly manipulating these regions, see the ImageCrop node.
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
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the original image at 0 and the full brightness effect at 1.

Control (UI)	Default Value	Function
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the brightness change is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the brightness change is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the brightness effect is limited to the fringe (semi-transparent areas).
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: • **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.

Control (UI)	Default Value	Function
greenSource	i0	Select the input from which to take the green channel.
greenChannel	G	Select what to use as the green channel: • **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.
blueChannel	B	Select what to use as the blue channel: • **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	Select what to use as the alpha channel: • **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	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.

Control (UI)	Default Value	Function
constantColor > RGB		
red	0	Sets the red value of the pixels in any channels that you have set to **Const**.
green	0	Sets the green value of the pixels in any channels that you have set to **Const**.
blue	0	Sets the blue value of the pixels in any channels that you have set to **Const**.
alpha	1	Sets the alpha value of the pixels in any channels that you have set to **Const**.
constantColor > HSL		
hue	0	Sets the hue of the pixels in any channels that you have set to **Const**.
saturation	0	Sets the saturation of the pixels in any channels that you have set to **Const**.
lightness	0	Sets the lightness of the pixels in any channels that you have set to **Const**.
alpha	1	Sets the alpha value of the pixels in any channels that you have set to **Const**.
constantColor > HSV		
hue	0	Sets the hue of the pixels in any channels that you have set to **Const**.
saturation	0	Sets the saturation of the pixels for any channels that you have set to **Const**.
value	0	Sets the value of the pixels for any channels that you have set to **Const**.
alpha	1	Sets the alpha value of the pixels for any channels that you have set to **Const**.

ImageClamp

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

Control (UI)	Default Value	Function
common		
process	rgba	The channels to clamp.
mix	1	Dissolves between the original image at 0 and the full clamp effect at 1.

Control (UI)	Default Value	Function
common > mask		
channel	a	The channel from the **out_mask** input to use as a mask: • **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. By default, the clamp effect is limited to the non-black areas of the mask.
invert	No	When set to **Yes**, Katana inverts the use of the mask channel so that the clamp effect is limited to the non-white areas of the mask.
fringe	No	When set to **Yes**, the mask is modified so that by default the clamp effect is limited to the fringe (semi-transparent areas).
clamp	Both	Sets which values are use to clamp the input: • **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.
green	0	
blue	0	
alpha	0	
max		
red	16	Sets the maximum values at which the rgba channels are clamped.
green	16	
blue	16	
alpha	1	
inputs	Unpremultiplied	Select whether you are using a premultiplied or unpremultiplied input image: • **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.
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.
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 below RGB, HSL, or HSV controls to set the color.
color > RGB		
red	1	Sets the red value of the pixels.
green	1	Sets the green value of the pixels.
blue	1	Sets the blue value of the pixels.
alpha	1	Sets the alpha value of the pixels.
color > HSL		
hue	0	Sets the hue of the pixels.
saturation	0	Sets the saturation of the pixels.
lightness	1	Sets the lightness of the pixels.
alpha	1	Sets the alpha value of the pixels.
color > HSV		
hue	0	Sets the hue of the pixels.
saturation	0	Sets the saturation of the pixels.
value	1	Sets the value of the pixels.
alpha	1	Sets the alpha value of the pixels.

ImageContrast

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

Control (UI)	Default Value	Function
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the original image at 0 and the full contrast effect at 1.
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the contrast change is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the contrast change is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the contrast change is limited to the fringe (semi-transparent areas). 0.0 -> 0.0 0.5 -> 1.0 1.0 -> 0.0 out=4a*(1-a)
contrast		
rbg	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.

Control (UI)	Default Value	Function
fixedPoint > RGB		
red	0	Sets the red contrast pivot value.
green	0	Sets the green contrast pivot value
blue	0	Sets the blue contrast pivot value
alpha	1	Sets the alpha contrast pivot value
fixedPoint > HSL		
hue	0	Sets the hue contrast pivot value.
saturation	0	Sets the saturation contrast pivot value.
lightness	0	Sets the lightness contrast pivot value.
alpha	1	Sets the alpha contrast pivot value.
fixedPoint > HSV		
hue	0	Sets the hue contrast pivot value.
saturation	0	Sets the saturation contrast pivot value.
value	0	Sets the value contrast pivot value.
alpha	1	Sets the alpha contrast pivot value.
contrastFunction	Power	Select the contrast function to use: • **Power** • **Linear**
inputs	Unpremultiplied	Select whether you are using a premultiplied or unpremultiplied input image: • **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.

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 that 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.
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.
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
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the **bg** image at 0 and the full merge effect at 1.

Control (UI)	Default Value	Function
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the merge is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the merge is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the merge is limited to the fringe (semi-transparent areas).
operation	ImageDifference	If you don't want layer the images together using the **ImageDifference** compositing operation, select the operation to use instead. The following conventions apply to the below operation descriptions: • **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. The available operations (based on the Porter & Duff paper) are: • **ImageAtop** – Fb+B(1-f). This shows the shape of the background, with the foreground covering the background where the images overlap. • **ImageAverage** – (F+B)/2. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • **ImageDifference** – abs(F-B). This shows how much the pixels differ and is useful for comparing two very similar images. • **ImageDivide** – B/F. This divides the background values by the foreground values. • **ImageExclusion** – F+B-2FB. This is a more photographic form of **ImageDifference**. • **ImageFrom** – F-B. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see **ImageMinus**. • **ImageGeometric** – 2FB/(F+B). This is another way of averaging two images. Visually, it's close to **ImageMin**. • **ImageHypot** – sqrt(F*F+B*B). 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 is useful for adding reflections, as an alternative to **ImageScreen**.

Control (UI)	Default Value	Function
		• **ImageIn** - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • **ImageMatte** - Ff*B(1-f). This is a premultiplied **ImageOver**. Use unpremultiplied images with this operation. • **ImageMax** - 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. • **ImageMin** - min(F,B). This takes the minimum values of both images. • **ImageMinus** - B-F. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see **ImageFrom**. • **ImageMultiply** - 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. • **ImageOut** - 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. • **ImageOver** - F+B(1-f). This layers the foreground over the background according to the alpha of the foregound. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • **ImagePlus** - 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. • **ImageScreen** - F+B-FB. This is similar to **ImageHypot**, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • **ImageUnder** - F(1-b)+B. This is the reverse of the **ImageOver** 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	The frame size to output in the event that the **fg** and **bg** inputs are different sizes: • **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.

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 will come from pixel 50, 28 in the **input** image.

Control (UI)	Default Value	Function
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the original image at 0 and the full distortion effect at 1.
Mask		
channel	A	The channel from **out_mask** input to use as a mask: • **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. By default, the distortion effect is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the distortion effect is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the distortion effect is limited to the fringe (semi-transparent areas).
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: • **Mirror** • **Clamp** • **Background**

Control (UI)	Default Value	Function
useOverscan	globals.compDefaults.useOverscan	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. However, if you are unsure of this procedure or the intergrity of overscanned areas is unknown, it's safer to leave this disabled. Overscan refers to image pixel data outside of the displayWindow and can be inspected using options in the Monitor. For information on explicitly manipulating these regions, see the ImageCrop node.
processMode	Full Frame	Sets how to render this node: • **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. **Note:** Output in either mode is identical.
useBlueChannelAsMask	disabled	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.
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.
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	Filtering can introduce negative values and send values above 1.0. 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. 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.

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
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the **bg** image at 0 and the full merge effect at 1.
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the merge is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the merge is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the merge is limited to the fringe (semi-transparent areas).
operation	ImageDivide	If you don't want layer the images together using the **ImageDivide** compositing operation, select the operation to use instead. The following conventions apply to the below operation descriptions: • **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. The available operations (based on the Porter & Duff paper) are: • **ImageAtop** - Fb+B(1-f). This shows the shape of the background, with the foreground covering the background where the images overlap. • **ImageAverage** - (F+B)/2. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • **ImageDifference** - abs(F-B). This shows how much the pixels differ and is useful for comparing two very similar images. • **ImageDivide** - B/F. This divides the background values by the foreground values. • **ImageExclusion** - F+B-2FB. This is a more photographic form of **ImageDifference**. • **ImageFrom** - F-B. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see **ImageMinus**.

Control (UI)	Default Value	Function
		• **ImageGeometric** - 2FB/(F+B). This is another way of averaging two images. Visually, it's close to **ImageMin**. • **ImageHypot** - sqrt(F*F+B*B). 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 is useful for adding reflections, as an alternative to **ImageScreen**. • **ImageIn** - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • **ImageMatte** - Ff*B(1-f). This is a premultiplied **ImageOver**. Use unpremultiplied images with this operation. • **ImageMax** - 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. • **ImageMin** - min(F,B). This takes the minimum values of both images. • **ImageMinus** - B-F. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see **ImageFrom**. • **ImageMultiply** - 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. • **ImageOut** - 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. • **ImageOver** - F+B(1-f). This layers the foreground over the background according to the alpha of the foregound. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • **ImagePlus** - 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. • **ImageScreen** - F+B-FB. This is similar to **ImageHypot**, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • **ImageUnder** - F(1-b)+B. This is the reverse of the **ImageOver** 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	The frame size to output in the event that the **fg** and **bg** inputs are different sizes: • **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.

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
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the **bg** image at 0 and the full merge effect at 1.
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the merge is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the merge is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the merge is limited to the fringe (semi-transparent areas).

Control (UI)	Default Value	Function
operation	ImageExclusion	If you don't want layer the images together using the **ImageExclusion** compositing operation, select the operation to use instead. The following conventions apply to the below operation descriptions: • **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. The available operations (based on the Porter & Duff paper) are: • **ImageAtop** - Fb+B(1-f). This shows the shape of the background, with the foreground covering the background where the images overlap. • **ImageAverage** - (F+B)/2. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • **ImageDifference** - abs(F-B). This shows how much the pixels differ and is useful for comparing two very similar images. • **ImageDivide** - B/F. This divides the background values by the foreground values. • **ImageExclusion** - F+B-2FB. This is a more photographic form of **ImageDifference**. • **ImageFrom** - F-B. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see **ImageMinus**. • **ImageGeometric** - 2FB/(F+B). This is another way of averaging two images. Visually, it's close to **ImageMin**. • **ImageHypot** - sqrt(F*F+B*B). 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 is useful for adding reflections, as an alternative to **ImageScreen**. • **ImageIn** - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • **ImageMatte** - Ff*B(1-f). This is a premultiplied **ImageOver**. Use unpremultiplied images with this operation. • **ImageMax** - 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
		• **ImageMin** - min(F,B). This takes the minimum values of both images. • **ImageMinus** - B-F. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see **ImageFrom**. • **ImageMultiply** - 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. • **ImageOut** - 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. • **ImageOver** - F+B(1-f). This layers the foreground over the background according to the alpha of the foregound. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • **ImagePlus** - 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. • **ImageScreen** - F+B-FB. This is similar to **ImageHypot**, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • **ImageUnder** - F(1-b)+B. This is the reverse of the **ImageOver** 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	The frame size to output in the event that the **fg** and **bg** inputs are different sizes: • **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.

ImageExposure

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

Control (UI)	Default Value	Function
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the original image at 0 and the full exposure change at 1.
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the exposure change is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the exposure change is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the exposure effect is limited to the fringe (semi-transparent areas).
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.
gAlpha	1	Adjusts gain exposure in the alpha channel.
units	F-Stops	Select the units in which the exposure is altered: • **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
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the original image at 0 and the full fade effect at 1.
Mask		
channel	A	The channel from **out_mask** input to use as a mask: • **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. By default, the fade effect is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the fade effect is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the fade effect is limited to the fringe (semi-transparent areas).
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.
fadeToColor > RGB		
red	0	Sets the red value of the fade color.
green	0	Sets the green value of the fade color.
blue	0	Sets the blue value of the fade color.
alpha	1	Sets the alpha value of the fade color.
fadeToColor > HSL		
hue	0	Sets the hue of the fade color.
saturation	0	Sets the saturation of the fade color.
lightness	0	Sets the lightness of the fade color.
alpha	1	Sets the alpha value of the fade color.
fadeToColor > HSV		
hue	0	Sets the hue of the fade color.

Control (UI)	Default Value	Function
saturation	0	Sets the saturation of the fade color.
value	0	Sets the value of the fade color.
alpha	1	Sets the alpha value of the fade color.

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 136.

Control (UI)	Default Value	Function
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the **bg** image at 0 and the full merge effect at 1.
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the merge is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the merge is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the merge is limited to the fringe (semi-transparent areas).

Control (UI)	Default Value	Function
operation	ImageFrom	If you don't want layer the images together using the **ImageFrom** compositing operation, select the operation to use instead. The following conventions apply to the below operation descriptions: • **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. The available operations (based on the Porter & Duff paper) are: • **ImageAtop** - Fb+B(1-f). This shows the shape of the background, with the foreground covering the background where the images overlap. • **ImageAverage** - (F+B)/2. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • **ImageDifference** - abs(F-B). This shows how much the pixels differ and is useful for comparing two very similar images. • **ImageDivide** - B/F. This divides the background values by the foreground values. • **ImageExclusion** - F+B-2FB. This is a more photographic form of **ImageDifference**. • **ImageFrom** - F-B. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see **ImageMinus**. • **ImageGeometric** - 2FB/(F+B). This is another way of averaging two images. Visually, it's close to **ImageMin**. • **ImageHypot** - sqrt(F*F+B*B). 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 is useful for adding reflections, as an alternative to **ImageScreen**. • **ImageIn** - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • **ImageMatte** - Ff*B(1-f). This is a premultiplied **ImageOver**. Use unpremultiplied images with this operation. • **ImageMax** - 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
		• **ImageMin** - min(F,B). This takes the minimum values of both images. • **ImageMinus** - B-F. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see **ImageFrom**. • **ImageMultiply** - 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. • **ImageOut** - 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. • **ImageOver** - F+B(1-f). This layers the foreground over the background according to the alpha of the foregound. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • **ImagePlus** - 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. • **ImageScreen** - F+B-FB. This is similar to **ImageHypot**, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • **ImageUnder** - F(1-b)+B. This is the reverse of the **ImageOver** 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	The frame size to output in the event that the **fg** and **bg** inputs are different sizes: • **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.

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
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the original image at 0 and the full gain effect at 1.
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the gain adjustment is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the gain adjustment is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the gain adjustment is limited to the fringe (semi-transparent areas).
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: • **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 midtones.

Control (UI)	Default Value	Function
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the original image at 0 and the full gamma effect at 1.
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the gamma adjustment is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the gamma adjustment is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the gamma adjustment is limited to the fringe (semi-transparent areas).
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.

Control (UI)	Default Value	Function
fixedPoint		
fixedPoint	1.0000, 1.0000, 1.0000, 1.0000	The color (RGBA values) of ???. You can also use the below RGB, HSL, or HSV controls to set the color.
fixedPoint > RGB		
red	1	Sets the red value of ???
green	0	Sets the green value of ???
blue	0	Sets the blue value of ???
alpha	1	Sets the alpha value of ???
fixedPoint > HSL		
hue	0	Sets the hue of ???
saturation	0	Sets the saturation of ???
lightness	0	Sets the lightness of ???
alpha	1	Sets the alpha value of ???
fixedPoint > HSV		
hue	0	Sets the hue of ???
saturation	0	Sets the saturation of ???
value	0	Sets the value of ???
alpha	1	Sets the alpha value of ???
inputs	Unpremultiplied	Select whether you are using a premultiplied or unpremultiplied input image: • **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 56 and ImageHypot on page 90.

Control (UI)	Default Value	Function
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the **bg** image at 0 and the full merge effect at 1.
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the merge is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the merge is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the merge is limited to the fringe (semi-transparent areas.
operation	ImageGeometric	If you don't want layer the images together using the **ImageGeometric** compositing operation, select the operation to use instead. The following conventions apply to the below operation descriptions: • **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. The available operations (based on the Porter & Duff paper) are: • **ImageAtop** - Fb+B(1-f). This shows the shape of the background, with the foreground covering the background where the images overlap. • **ImageAverage** - (F+B)/2. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • **ImageDifference** - abs(F-B). This shows how much the pixels differ and is useful for comparing two very similar images. • **ImageDivide** - B/F. This divides the background values by the foreground values. • **ImageExclusion** - F+B-2FB. This is a more photographic form of **ImageDifference**. • **ImageFrom** - F-B. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see **ImageMinus**.

Control (UI)	Default Value	Function
		• **ImageGeometric** - 2FB/(F+B). This is another way of averaging two images. Visually, it's close to **ImageMin**. • **ImageHypot** - sqrt(F*F+B*B). 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 is useful for adding reflections, as an alternative to **ImageScreen**. • **ImageIn** - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • **ImageMatte** - Ff*B(1-f). This is a premultiplied **ImageOver**. Use unpremultiplied images with this operation. • **ImageMax** - 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. • **ImageMin** - min(F,B). This takes the minimum values of both images. • **ImageMinus** - B-F. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see **ImageFrom**. • **ImageMultiply** - 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. • **ImageOut** - 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. • **ImageOver** - F+B(1-f). This layers the foreground over the background according to the alpha of the foregound. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • **ImagePlus** - 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. • **ImageScreen** - F+B-FB. This is similar to **ImageHypot**, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • **ImageUnder** - F(1-b)+B. This is the reverse of the **ImageOver** 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	The frame size to output in the event that the **fg** and **bg** inputs are different sizes: • **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.

ImageHypot

This node layers images together using the **Hypot** compositing algorithm: **sqrt(F*F+B*B)**. 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 is 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 117 and ImageScreen on page 128.

Control (UI)	Default Value	Function
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the **bg** image at 0 and the full merge effect at 1.
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the merge is limited to the non-black areas of the mask.

Control (UI)	Default Value	Function
invert	disabled	Inverts the use of the mask channel so that the merge is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the merge is limited to the fringe (semi-transparent areas).
operation	ImageHypot	If you don't want layer the images together using the **ImageHypot** compositing operation, select the operation to use instead. The following conventions apply to the below operation descriptions: • **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. The available operations (based on the Porter & Duff paper) are: • **ImageAtop** - Fb+B(1-f). This shows the shape of the background, with the foreground covering the background where the images overlap. • **ImageAverage** - (F+B)/2. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • **ImageDifference** - abs(F-B). This shows how much the pixels differ and is useful for comparing two very similar images. • **ImageDivide** - B/F. This divides the background values by the foreground values. • **ImageExclusion** - F+B-2FB. This is a more photographic form of **ImageDifference**. • **ImageFrom** - F-B. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see **ImageMinus**. • **ImageGeometric** - 2FB/(F+B). This is another way of averaging two images. Visually, it's close to **ImageMin**. • **ImageHypot** - sqrt(F*F+B*B). 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 is useful for adding reflections, as an alternative to **ImageScreen**. • **ImageIn** - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • **ImageMatte** - Ff*B(1-f). This is a premultiplied **ImageOver**. Use unpremultiplied images with this operation. • **ImageMax** - 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
		• **ImageMin** - min(F,B). This takes the minimum values of both images. • **ImageMinus** - B-F. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see **ImageFrom**. • **ImageMultiply** - 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. • **ImageOut** - 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. • **ImageOver** - F+B(1-f). This layers the foreground over the background according to the alpha of the foregound. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • **ImagePlus** - 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. • **ImageScreen** - F+B-FB. This is similar to **ImageHypot**, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • **ImageUnder** - F(1-b)+B. This is the reverse of the **ImageOver** 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	The frame size to output in the event that the **fg** and **bg** inputs are different sizes: • **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.

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
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the **bg** image at 0 and the full merge effect at 1.
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the merge is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the merge is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the merge is limited to the fringe (semi-transparent areas).
operation	ImageIn	If you don't want layer the images together using the **ImageIn** compositing operation, select the operation to use instead. The following conventions apply to the below operation descriptions: • **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. The available operations (based on the Porter & Duff paper) are: • **ImageAtop** - Fb+B(1-f). This shows the shape of the background, with the foreground covering the background where the images overlap. • **ImageAverage** - (F+B)/2. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • **ImageDifference** - abs(F-B). This shows how much the pixels differ and is useful for comparing two very similar images. • **ImageDivide** - B/F. This divides the background values by the foreground values. • **ImageExclusion** - F+B-2FB. This is a more photographic form of **ImageDifference**. • **ImageFrom** - F-B. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see **ImageMinus**.

Control (UI)	Default Value	Function
		• **ImageGeometric** - 2FB/(F+B). This is another way of averaging two images. Visually, it's close to **ImageMin**. • **ImageHypot** - sqrt(F*F+B*B). 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 is useful for adding reflections, as an alternative to **ImageScreen**. • **ImageIn** - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • **ImageMatte** - Ff*B(1-f). This is a premultiplied **ImageOver**. Use unpremultiplied images with this operation. • **ImageMax** - 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. • **ImageMin** - min(F,B). This takes the minimum values of both images. • **ImageMinus** - B-F. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see **ImageFrom**. • **ImageMultiply** - 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. • **ImageOut** - 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. • **ImageOver** - F+B(1-f). This layers the foreground over the background according to the alpha of the foregound. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • **ImagePlus** - 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. • **ImageScreen** - F+B-FB. This is similar to **ImageHypot**, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • **ImageUnder** - F(1-b)+B. This is the reverse of the **ImageOver** 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	The frame size to output in the event that the **fg** and **bg** inputs are different sizes: • **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.

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
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the original image at 0 and the full invert effect at 1.
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the inversion is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the inversion is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the inversion is limited to the fringe (semi-transparent areas).
mode	Additive	Sets the invert calculation mode: • **Additive** • **Multiplicative**

ImageLevels

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

Control (UI)	Default Value	Function
common		
process	rgba	The channels to which you want to apply the levels adjustment.
mix	1	Dissolves between the original image at 0 and the full levels adjustment at 1.
common > mask		
channel	a	The channel from the **out_mask** input to use as a mask: • **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. By default, the levels adjustment is limited to the non-black areas of the mask.
invert	No	When set to **Yes**, Katana inverts the use of the mask channel so that the levels adjustment is limited to the non-white areas of the mask.
fringe	No	When set to **Yes**, the mask is modified so that by default the levels adjustment is limited to the fringe (semi-transparent areas).
inputMin		
red	0	Sets the minimum input level for the red channel.
green	0	Sets the minimum input level for the green channel.
blue	0	Sets the minimum input level for the blue channel.
alpha	0	Sets the minimum input level for the alpha channel.
inputMax		
red	1	Sets the maximum input level for the red channel.
green	1	Sets the maximum input level for the green channel.
blue	1	Sets the maximum input level for the blue channel.
alpha	1	Sets the maximum input level for the alpha channel.
gamma		
red	1	Sets the gamma level for the red channel.
green	1	Sets the gamma level for the green channel.
blue	1	Sets the gamma level for the blue channel.
alpha	1	Sets the gamma level for the alpha channel.

Control (UI)	Default Value	Function
outputMin		
red	0	Sets the minimum output level for the red channel.
green	0	Sets the minimum output level for the green channel.
blue	0	Sets the minimum output level for the blue channel.
alpha	0	Sets the minimum output level for the alpha channel.
outputMax		
red	1	Sets the maximum output level for the red channel.
green	1	Sets the maximum output level for the green channel.
blue	1	Sets the maximum output level for the blue channel.
alpha	1	Sets the maximum output level for the alpha channel.
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
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the **bg** image at 0 and the full merge effect at 1.

Control (UI)	Default Value	Function
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the merge is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the merge is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the merge is limited to the fringe (semi-transparent areas).
operation	ImageMatte	If you don't want layer the images together using the **ImageMatte** compositing operation, select the operation to use instead. The following conventions apply to the below operation descriptions: • **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. The available operations (based on the Porter & Duff paper) are: • **ImageAtop** – Fb+B(1-f). This shows the shape of the background, with the foreground covering the background where the images overlap. • **ImageAverage** – (F+B)/2. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • **ImageDifference** – abs(F-B). This shows how much the pixels differ and is useful for comparing two very similar images. • **ImageDivide** – B/F. This divides the background values by the foreground values. • **ImageExclusion** – F+B-2FB. This is a more photographic form of **ImageDifference**. • **ImageFrom** – F-B. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see **ImageMinus**. • **ImageGeometric** – 2FB/(F+B). This is another way of averaging two images. Visually, it's close to **ImageMin**. • **ImageHypot** – sqrt(F*F+B*B). 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 is useful for adding reflections, as an alternative to **ImageScreen**.

Control (UI)	Default Value	Function
		• **ImageIn** - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • **ImageMatte** - Ff*B(1-f). This is a premultiplied **ImageOver**. Use unpremultiplied images with this operation. • **ImageMax** - 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. • **ImageMin** - min(F,B). This takes the minimum values of both images. • **ImageMinus** - B-F. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see **ImageFrom**. • **ImageMultiply** - 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. • **ImageOut** - 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. • **ImageOver** - F+B(1-f). This layers the foreground over the background according to the alpha of the foregound. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • **ImagePlus** - 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. • **ImageScreen** - F+B-FB. This is similar to **ImageHypot**, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • **ImageUnder** - F(1-b)+B. This is the reverse of the **ImageOver** 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	The frame size to output in the event that the **fg** and **bg** inputs are different sizes: • **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.

ImageMatteMix

This node uses a matte to control the mixing of two image, background and foreground.

Control (UI)	Default Value	Function
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the background image at 0 and the foreground image at 1.
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the mix effect is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the mix effect is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the mix effect is limited to the fringe (semi-transparent areas).
amount	1	An optional bias for the mix operation. See **amountMath** for details.
amountMath	Normal	Select the mode to use: • **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.
displayWindow	Background	The frame size to output in the event that the **fg** and **bg** inputs are different sizes: • **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
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the **bg** image at 0 and the full merge effect at 1.
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the merge is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the merge is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the merge is limited to the fringe (semi-transparent areas).
operation	ImageMax	If you don't want layer the images together using the **ImageMax** compositing operation, select the operation to use instead. The following conventions apply to the below operation descriptions: • **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
		The available operations (based on the Porter & Duff paper) are: • **ImageAtop** - Fb+B(1-f). This shows the shape of the background, with the foreground covering the background where the images overlap. • **ImageAverage** - (F+B)/2. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • **ImageDifference** - abs(F-B). This shows how much the pixels differ and is useful for comparing two very similar images. • **ImageDivide** - B/F. This divides the background values by the foreground values. • **ImageExclusion** - F+B-2FB. This is a more photographic form of **ImageDifference**. • **ImageFrom** - F-B. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see **ImageMinus**. • **ImageGeometric** - 2FB/(F+B). This is another way of averaging two images. Visually, it's close to **ImageMin**. • **ImageHypot** - sqrt(F*F+B*B). 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 is useful for adding reflections, as an alternative to **ImageScreen**. • **ImageIn** - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • **ImageMatte** - Ff*B(1-f). This is a premultiplied **ImageOver**. Use unpremultiplied images with this operation. • **ImageMax** - 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. • **ImageMin** - min(F,B). This takes the minimum values of both images. • **ImageMinus** - B-F. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see **ImageFrom**. • **ImageMultiply** - 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.

Control (UI)	Default Value	Function
		• **ImageOut** - 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. • **ImageOver** - F+B(1-f). This layers the foreground over the background according to the alpha of the foregound. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • **ImagePlus** - 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. • **ImageScreen** - F+B-FB. This is similar to **ImageHypot**, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • **ImageUnder** - F(1-b)+B. This is the reverse of the **ImageOver** 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	The frame size to output in the event that the **fg** and **bg** inputs are different sizes: • **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.

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
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the **bg** image at 0 and the full merge effect at 1.

Control (UI)	Default Value	Function
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the merge is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the merge is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the merge is limited to the fringe (semi-transparent areas).
operation	ImageMin	If you don't want layer the images together using the **ImageMin** compositing operation, select the operation to use instead. The following conventions apply to the below operation descriptions: • **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. The available operations (based on the Porter & Duff paper) are: • **ImageAtop** – Fb+B(1-f). This shows the shape of the background, with the foreground covering the background where the images overlap. • **ImageAverage** – (F+B)/2. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • **ImageDifference** – abs(F-B). This shows how much the pixels differ and is useful for comparing two very similar images. • **ImageDivide** – B/F. This divides the background values by the foreground values. • **ImageExclusion** – F+B-2FB. This is a more photographic form of **ImageDifference**. • **ImageFrom** – F-B. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see **ImageMinus**. • **ImageGeometric** – 2FB/(F+B). This is another way of averaging two images. Visually, it's close to **ImageMin**. • **ImageHypot** – sqrt(F*F+B*B). 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 is useful for adding reflections, as an alternative to **ImageScreen**.

Control (UI)	Default Value	Function
		• **ImageIn** - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • **ImageMatte** - Ff*B(1-f). This is a premultiplied **ImageOver**. Use unpremultiplied images with this operation. • **ImageMax** - 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. • **ImageMin** - min(F,B). This takes the minimum values of both images. • **ImageMinus** - B-F. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see **ImageFrom**. • **ImageMultiply** - 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. • **ImageOut** - 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. • **ImageOver** - F+B(1-f). This layers the foreground over the background according to the alpha of the foregound. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • **ImagePlus** - 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. • **ImageScreen** - F+B-FB. This is similar to **ImageHypot**, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • **ImageUnder** - F(1-b)+B. This is the reverse of the **ImageOver** 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	The frame size to output in the event that the **fg** and **bg** inputs are different sizes: • **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.

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 82.

Control (UI)	Default Value	Function
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the **bg** image at 0 and the full merge effect at 1.
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the merge is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the merge is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the merge is limited to the fringe (semi-transparent areas).
operation	ImageMinus	If you don't want layer the images together using the **ImageMinus** compositing operation, select the operation to use instead. The following conventions apply to the below operation descriptions: • **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
		The available operations (based on the Porter & Duff paper) are: • **ImageAtop** - Fb+B(1-f). This shows the shape of the background, with the foreground covering the background where the images overlap. • **ImageAverage** - (F+B)/2. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • **ImageDifference** - abs(F-B). This shows how much the pixels differ and is useful for comparing two very similar images. • **ImageDivide** - B/F. This divides the background values by the foreground values. • **ImageExclusion** - F+B-2FB. This is a more photographic form of **ImageDifference**. • **ImageFrom** - F-B. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see **ImageMinus**. • **ImageGeometric** - 2FB/(F+B). This is another way of averaging two images. Visually, it's close to **ImageMin**. • **ImageHypot** - sqrt(F*F+B*B). 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 is useful for adding reflections, as an alternative to **ImageScreen**. • **ImageIn** - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • **ImageMatte** - Ff*B(1-f). This is a premultiplied **ImageOver**. Use unpremultiplied images with this operation. • **ImageMax** - 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. • **ImageMin** - min(F,B). This takes the minimum values of both images. • **ImageMinus** - B-F. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see **ImageFrom**. • **ImageMultiply** - 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.

Control (UI)	Default Value	Function
		• **ImageOut** - 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. • **ImageOver** - F+B(1-f). This layers the foreground over the background according to the alpha of the foregound. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • **ImagePlus** - 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. • **ImageScreen** - F+B-FB. This is similar to **ImageHypot**, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • **ImageUnder** - F(1-b)+B. This is the reverse of the **ImageOver** 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	The frame size to output in the event that the **fg** and **bg** inputs are different sizes: • **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.

ImageMix

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

Control (UI)	Default Value	Function
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the background image at 0 and the full mix effect at 1.

Control (UI)	Default Value	Function
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: • **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
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the **bg** image at 0 and the full merge effect at 1.
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the merge is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the merge is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the merge is limited to the fringe (semi-transparent areas).

Control (UI)	Default Value	Function
operation	ImageMultiply	If you don't want layer the images together using the **ImageMultiply** compositing operation, select the operation to use instead. The following conventions apply to the below operation descriptions: • **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. The available operations (based on the Porter & Duff paper) are: • **ImageAtop** - Fb+B(1-f). This shows the shape of the background, with the foreground covering the background where the images overlap. • **ImageAverage** - (F+B)/2. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • **ImageDifference** - abs(F-B). This shows how much the pixels differ and is useful for comparing two very similar images. • **ImageDivide** - B/F. This divides the background values by the foreground values. • **ImageExclusion** - F+B-2FB. This is a more photographic form of **ImageDifference**. • **ImageFrom** - F-B. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see **ImageMinus**. • **ImageGeometric** - 2FB/(F+B). This is another way of averaging two images. Visually, it's close to **ImageMin**. • **ImageHypot** - sqrt(F*F+B*B). 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 is useful for adding reflections, as an alternative to **ImageScreen**. • **ImageIn** - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • **ImageMatte** - Ff*B(1-f). This is a premultiplied **ImageOver**. Use unpremultiplied images with this operation. • **ImageMax** - 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
		• **ImageMin** - min(F,B). This takes the minimum values of both images. • **ImageMinus** - B-F. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see **ImageFrom**. • **ImageMultiply** - 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. • **ImageOut** - 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. • **ImageOver** - 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. • **ImagePlus** - 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. • **ImageScreen** - F+B-FB. This is similar to **ImageHypot**, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • **ImageUnder** - F(1-b)+B. This is the reverse of the **ImageOver** 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	The frame size to output in the event that the **fg** and **bg** inputs are different sizes: • **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
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the original image at 0 and the full rotation effect at 1.
Mask		
channel	A	The channel from **out_mask** input to use as a mask: • **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. By default, the rotation effect is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the rotation effect is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the rotation effect is limited to the fringe (semi-transparent areas).
orientation	No Change	Select how to rotate the input image: • **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
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the **bg** image at 0 and the full merge effect at 1.
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the merge is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the merge is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the merge is limited to the fringe (semi-transparent areas).
operation	ImageOut	If you don't want layer the images together using the **ImageOut** compositing operation, select the operation to use instead. The following conventions apply to the below operation descriptions: • **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. The available operations (based on the Porter & Duff paper) are: • **ImageAtop** - Fb+B(1-f). This shows the shape of the background, with the foreground covering the background where the images overlap. • **ImageAverage** - (F+B)/2. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • **ImageDifference** - abs(F-B). This shows how much the pixels differ and is useful for comparing two very similar images. • **ImageDivide** - B/F. This divides the background values by the foreground values. • **ImageExclusion** - F+B-2FB. This is a more photographic form of **ImageDifference**. • **ImageFrom** - F-B. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see **ImageMinus**.

Control (UI)	Default Value	Function
		• **ImageGeometric** - 2FB/(F+B). This is another way of averaging two images. Visually, it's close to **ImageMin**. • **ImageHypot** - sqrt(F*F+B*B). 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 is useful for adding reflections, as an alternative to **ImageScreen**. • **ImageIn** - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • **ImageMatte** - Ff*B(1-f). This is a premultiplied **ImageOver**. Use unpremultiplied images with this operation. • **ImageMax** - 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. • **ImageMin** - min(F,B). This takes the minimum values of both images. • **ImageMinus** - B-F. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see **ImageFrom**. • **ImageMultiply** - 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. • **ImageOut** - 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. • **ImageOver** - F+B(1-f). This layers the foreground over the background according to the alpha of the foregound. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • **ImagePlus** - 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. • **ImageScreen** - F+B-FB. This is similar to **ImageHypot**, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • **ImageUnder** - F(1-b)+B. This is the reverse of the **ImageOver** 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	The frame size to output in the event that the **fg** and **bg** inputs are different sizes: • **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.

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 136.

Control (UI)	Default Value	Function
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the **bg** image at 0 and the full merge effect at 1.
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the merge is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the merge is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the merge is limited to the fringe (semi-transparent areas).

Control (UI)	Default Value	Function
operation	ImageOver	If you don't want layer the images together using the **ImageOver** compositing operation, select the operation to use instead. The following conventions apply to the below operation descriptions: • **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. The available operations (based on the Porter & Duff paper) are: • **ImageAtop** - Fb+B(1-f). This shows the shape of the background, with the foreground covering the background where the images overlap. • **ImageAverage** - (F+B)/2. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • **ImageDifference** - abs(F-B). This shows how much the pixels differ and is useful for comparing two very similar images. • **ImageDivide** - B/F. This divides the background values by the foreground values. • **ImageExclusion** - F+B-2FB. This is a more photographic form of **ImageDifference**. • **ImageFrom** - F-B. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see **ImageMinus**. • **ImageGeometric** - 2FB/(F+B). This is another way of averaging two images. Visually, it's close to **ImageMin**. • **ImageHypot** - sqrt(F*F+B*B). 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 is useful for adding reflections, as an alternative to **ImageScreen**. • **ImageIn** - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • **ImageMatte** - Ff*B(1-f). This is a premultiplied **ImageOver**. Use unpremultiplied images with this operation. • **ImageMax** - 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
		• **ImageMin** - min(F,B). This takes the minimum values of both images. • **ImageMinus** - B-F. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see **ImageFrom**. • **ImageMultiply** - 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. • **ImageOut** - 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. • **ImageOver** - F+B(1-f). This layers the foreground over the background according to the alpha of the foregound. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • **ImagePlus** - 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. • **ImageScreen** - F+B-FB. This is similar to **ImageHypot**, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • **ImageUnder** - F(1-b)+B. This is the reverse of the **ImageOver** 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	The frame size to output in the event that the **fg** and **bg** inputs are different sizes: • **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.

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
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the **bg** image at 0 and the full merge effect at 1.
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the merge is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the merge is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the merge is limited to the fringe (semi-transparent areas).
operation	ImagePlus	If you don't want layer the images together using the **ImagePlus** compositing operation, select the operation to use instead. The following conventions apply to the below operation descriptions: • **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. The available operations (based on the Porter & Duff paper) are: • **ImageAtop** - Fb+B(1-f). This shows the shape of the background, with the foreground covering the background where the images overlap. • **ImageAverage** - (F+B)/2. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • **ImageDifference** - abs(F-B). This shows how much the pixels differ and is useful for comparing two very similar images. • **ImageDivide** - B/F. This divides the background values by the foreground values. • **ImageExclusion** - F+B-2FB. This is a more photographic form of **ImageDifference**. • **ImageFrom** - F-B. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see **ImageMinus**.

Control (UI)	Default Value	Function
		• **ImageGeometric** - 2FB/(F+B). This is another way of averaging two images. Visually, it's close to **ImageMin**. • **ImageHypot** - sqrt(F*F+B*B). 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 is useful for adding reflections, as an alternative to **ImageScreen**. • **ImageIn** - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • **ImageMatte** - Ff*B(1-f). This is a premultiplied **ImageOver**. Use unpremultiplied images with this operation. • **ImageMax** - 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. • **ImageMin** - min(F,B). This takes the minimum values of both images. • **ImageMinus** - B-F. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see **ImageFrom**. • **ImageMultiply** - 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. • **ImageOut** - 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. • **ImageOver** - F+B(1-f). This layers the foreground over the background according to the alpha of the foregound. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • **ImagePlus** - 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. • **ImageScreen** - F+B-FB. This is similar to **ImageHypot**, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • **ImageUnder** - F(1-b)+B. This is the reverse of the **ImageOver** 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	The frame size to output in the event that the **fg** and **bg** inputs are different sizes: • **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.

ImagePosition

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

If you are looking to do a transform with sub-pixel resampling, see ImageTransform2D on page 133 instead.

Control (UI)	Default Value	Function
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the original image at 0 and the full offset effect at 1.
Mask		
channel	A	The channel from **out_mask** input to use as a mask: • **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. By default, the offset effect is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the offset effect is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the offset effect is limited to the fringe (semi-transparent areas).

Control (UI)	Default Value	Function
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 139.

Control (UI)	Default Value	Function
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the original image at 0 and the full premultiplication effect at 1.
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the premultiplication is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the premultiplication is limited to the non-white areas of the mask.

Control (UI)	Default Value	Function
fringe	disabled	When enabled, the mask is modified so that by default the premultiplication is limited to the fringe (semi-transparent areas).
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.

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.
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].
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 > rawData: disabled		
colorspace	auto	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. 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.

Control (UI)	Default Value	Function
image > isProxy: enabled > fullResFrame		
[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	
bottom	timing.missingFrameBounds.bottom	
width	timing.missingFrameBounds.width	
height	timing.missingFrameBounds.height	
timing		
frame	frame	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. 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.
inMode	globals.compDefaults.fileIn.inMode	Sets what to do when a frame is required at a time value prior to **firstFrame**: • **Black** • **Freeze** • **Repeat** • **Mirror**
outMode	globals.compDefaults.fileIn.outMode	Sets what to do when a frame is required at a time value after **lastFrame**: • **Black** • **Freeze** • **Repeat** • **Mirror**
firstFrame	globals.inTime	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**. 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.

Control (UI)	Default Value	Function
lastFrame	globals.outTime	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**. 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.
lockSettings	disabled	When enabled, the **firstFrame**, **lastFrame**, **inMode**, and **outMode** values aren't automatically updated when a new file sequence is chosen.
missingFrames	globals.compDefaults.fileIn.missingFrames	Specifies what to do if a frame is not found: • **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. **Note:** If no frames in the image sequence are present, the render fails regardless of this control's setting.
advanced		
includeInErrorChecking	enabled	When enabled, Katana includes this node when it automatically checks ImageRead nodes for errors. When disabled, Katana excludes this node when it automatically checks ImageRead nodes for errors.

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
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the original image at 0 and the full reformat effect at 1. Mixing between different formats isn't usually desirable, so we recommend leaving this control set to 1.

Control (UI)	Default Value	Function
Mask - Masking a reformat isn't usually desirable, so we recommend not using these controls.		
channel	A	The channel from **out_mask** input to use as a mask: • **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. By default, the reformat effect is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the reformat effect is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the reformat effect is limited to the fringe (semi-transparent areas).
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.
preserveAspect: enabled		
center	enabled	When enabled, Katana pads the output image if any gaps remain after reformatting while preserving the original 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.

Control (UI)	Default Value	Function
filtering		
downFilter	Lanczos3	The filter kernel to use for downsampling: • **Gaussian** • **Triangle** • **Box** • **Bell** • **BSpline** • **Sinc** • **Lanczos2** • **Lanczos3** • **Lanczos5** • **Mitchell** • **Bilinear** • **Bicubic** • **Nearest**
upFilter	Mitchell	The filter kernel to use for upsampling: • **Gaussian** • **Triangle** • **Box** • **Bell** • **BSpline** • **Sinc** • **Lanczos2** • **Lanczos3** • **Lanczos5** • **Mitchell** • **Bilinear** • **Bicubic** • **Nearest**

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.
clampOutput	enabled	Filtering can introduce negative values and send values above 1.0. 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. 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.

ImageSaturation

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

Control (UI)	Default Value	Function
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the original image at 0 and the full saturation adjustment at 1.
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the inversion is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the inversion is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the inversion is limited to the fringe (semi-transparent areas).
saturation	1	Controls overall image saturation. Values less than 1 reduce saturation and vice-versa.

Control (UI)	Default Value	Function
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 on page 90.

Control (UI)	Default Value	Function
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the **bg** image at 0 and the full merge effect at 1.
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the merge is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the merge is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the merge is limited to the fringe (semi-transparent areas).

Control (UI)	Default Value	Function
operation	ImageScreen	If you don't want layer the images together using the **ImageScreen** compositing operation, select the operation to use instead. The following conventions apply to the below operation descriptions: • **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. The available operations (based on the Porter & Duff paper) are: • **ImageAtop** - Fb+B(1-f). This shows the shape of the background, with the foreground covering the background where the images overlap. • **ImageAverage** - (F+B)/2. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • **ImageDifference** - abs(F-B). This shows how much the pixels differ and is useful for comparing two very similar images. • **ImageDivide** - B/F. This divides the background values by the foreground values. • **ImageExclusion** - F+B-2FB. This is a more photographic form of **ImageDifference**. • **ImageFrom** - F-B. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see **ImageMinus**. • **ImageGeometric** - 2FB/(F+B). This is another way of averaging two images. Visually, it's close to **ImageMin**. • **ImageHypot** - sqrt(F*F+B*B). 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 is useful for adding reflections, as an alternative to **ImageScreen**. • **ImageIn** - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • **ImageMatte** - Ff*B(1-f). This is a premultiplied **ImageOver**. Use unpremultiplied images with this operation. • **ImageMax** - 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
		• **ImageMin** - min(F,B). This takes the minimum values of both images. • **ImageMinus** - B-F. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see **ImageFrom**. • **ImageMultiply** - 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. • **ImageOut** - 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. • **ImageOver** - F+B(1-f). This layers the foreground over the background according to the alpha of the foregound. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • **ImagePlus** - 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. • **ImageScreen** - F+B-FB. This is similar to **ImageHypot**, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • **ImageUnder** - F(1-b)+B. This is the reverse of the **ImageOver** 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	The frame size to output in the event that the **fg** and **bg** inputs are different sizes: • **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.

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	The width of the text frame in pixels. The default setting, **globals.width**, resizes the text frame to the width of the **resolution** indicated on the **Project Settings** tab. **Note:** You can only edit this field when the initial resolution is modified.
height	globals.height	The height of the text frame in pixels. The default setting, **globals.height**, resizes the text frame to the height of the **resolution** indicated on the **Project Settings** tab. **Note:** You can only edit this field when the initial resolution is modified.
text	none	Enter the text you want to display here. You can optionally query Scene Graph values from an incoming 3D scene by: • 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	Select: • **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**.

Control (UI)	Default Value	Function
size	18.0, size[0]	Sets the pixel size of the font. 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. No character ever renders larger than this size.
position	getDisplayWindow().width/2, getDisplayWindow().height/2	The pixel position at which the justified text is placed. 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. If you set **hjustify** to **Center** and **vjustify** to **Bottom**, the baseline of the last line of text is centered on this position horizontally.
hjustify	Center	Sets how to align the text horizontally: • **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	Sets how to align the text vertically: • **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**. The baseline is the imaginary line upon which most letters rest.
lineSpace	0	If you have several lines of text, this adjusts the spacing between each line. By using negative values, you can make the letters overlap.
wrapMode	None	Sets how to wrap long lines of text to fit inside the text frame: • **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		
wrapWidth	bounds.width	The width to use when calculating when to wrap the text.
wrapMode: Exact		
wrapWidth	bounds.width	The width to use when calculating when to wrap the text.

Control (UI)	Default Value	Function
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.
color > RGB		
red	1	Sets the red value of the rendered text.
green	1	Sets the green value of the rendered text.
blue	1	Sets the blue value of the rendered text.
alpha	1	Sets the alpha value of the rendered text.
color > HSL		
hue	0	Sets the hue of the rendered text.
saturation	0	Sets the saturation of the rendered text.
lightness	1	Sets the lightness of the rendered text.
alpha	1	Sets the alpha value of the rendered text.
color > HSV		
hue	0	Sets the hue of the rendered text.
saturation	0	Sets the saturation of the rendered text.
value	1	Sets the value of the rendered text.
alpha	1	Sets the alpha value of the rendered text.
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: • **Ignore** - the missing attributes are ignored. • **Error** - the missing attributes cause a render error.

ImageTransform2D

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

Control (UI)	Default Value	Function
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.

Control (UI)	Default Value	Function
mix	1	Dissolves between the original image at 0 and the full transform effect at 1.
Mask		
channel	A	The channel from **out_mask** input to use as a mask: • **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. By default, the transform is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the transform is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the transform is limited to the fringe (semi-transparent areas).
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.

Control (UI)	Default Value	Function
filtering		
downFilter	Lanczos3	The filter kernel to use for downsampling: • **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 artefacts at extreme zoom. • **Bicubic** - provides more rounded results, slightly blurrier but without the square artefacts. • **Nearest** - preserves edge detail, but gives quite "blocky" textures.
upFilter	Lanczos3	The filter kernel to use for upsampling: • **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 artefacts at extreme zoom. • **Bicubic** - provides more rounded results, slightly blurrier but without the square artefacts. • **Nearest** - preserves edge detail, but gives quite "blocky" textures.
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).

Control (UI)	Default Value	Function
clampOutput	enabled	Filtering can introduce negative values and send values above 1.0. 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. 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.
onlyApplyMotion	disabled	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. If you apply **onlyApplyMotion** to the incoming image, and then transform the result by the node (with motion blur disabled), the results will be similar (except for sampling differences).
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 lerped end points for each subframe 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 115.

Control (UI)	Default Value	Function
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the **bg** image at 0 and the full merge effect at 1.
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the merge is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the merge is limited to the non-white areas of the mask.
fringe	disabled	When enabled, the mask is modified so that by default the merge is limited to the fringe (semi-transparent areas).
operation	ImageUnder	If you don't want layer the images together using the **ImageUnder** compositing operation, select the operation to use instead. The following conventions apply to the below operation descriptions: • **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. The available operations (based on the Porter & Duff paper) are: • **ImageAtop** - Fb+B(1-f). This shows the shape of the background, with the foreground covering the background where the images overlap. • **ImageAverage** - (F+B)/2. This produces the average of the two images. The result is darker than the original images, but accentuates highlights. • **ImageDifference** - abs(F-B). This shows how much the pixels differ and is useful for comparing two very similar images. • **ImageDivide** - B/F. This divides the background values by the foreground values. • **ImageExclusion** - F+B-2FB. This is a more photographic form of **ImageDifference**. • **ImageFrom** - F-B. This subtracts the background from the foreground. For subtracting the foreground from the background instead, see **ImageMinus**.

Control (UI)	Default Value	Function
		• **ImageGeometric** - 2FB/(F+B). This is another way of averaging two images. Visually, it's close to **ImageMin**. • **ImageHypot** - sqrt(F*F+B*B). 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 is useful for adding reflections, as an alternative to **ImageScreen**. • **ImageIn** - Bf. This only shows the areas of the background that overlap with the alpha of the foreground. It can be useful for combining mattes. • **ImageMatte** - Ff*B(1-f). This is a premultiplied **ImageOver**. Use unpremultiplied images with this operation. • **ImageMax** - 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. • **ImageMin** - min(F,B). This takes the minimum values of both images. • **ImageMinus** - B-F. This subtracts the foreground from the background. For subtracting the background from the foreground instead, see **ImageFrom**. • **ImageMultiply** - 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. • **ImageOut** - 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. • **ImageOver** - F+B(1-f). This layers the foreground over the background according to the alpha of the foregound. This is the most commonly used operation. It's used when layering a foreground element over a background plate. • **ImagePlus** - 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. • **ImageScreen** - F+B-FB. This is similar to **ImageHypot**, but clamps pixel values to 1.0. This is mostly useful for combining mattes. • **ImageUnder** - F(1-b)+B. This is the reverse of the **ImageOver** 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	The frame size to output in the event that the **fg** and **bg** inputs are different sizes: • **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.

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 121.

Control (UI)	Default Value	Function
R	enabled	When enabled, affect the red channel of the image.
G	enabled	When enabled, affect the green channel of the image.
B	enabled	When enabled, affect the blue channel of the image.
A	enabled	When enabled, affect the alpha channel of the image.
mix	1	Dissolves between the original image at 0 and the full unpremultiplication effect at 1.
Mask		
channel	A	The channel from the **out_mask** input to use as a mask: • **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. By default, the unpremultiplication is limited to the non-black areas of the mask.
invert	disabled	Inverts the use of the mask channel so that the unpremultiplication is limited to the non-white areas of the mask.

Control (UI)	Default Value	Function
fringe	disabled	When enabled, the mask is modified so that by default the unpremultiplication is limited to the fringe (semi-transparent areas).
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		
[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	The width of the text frame in pixels. The default setting, **globals.width**, resizes the text frame to the width of the **resolution** indicated on the **Project Settings** tab. **Note:** You can only edit this field when the initial resolution is modified.
height	globals.height	The height of the text frame in pixels. The default setting, **globals.height**, resizes the text frame to the height of the **resolution** indicated on the **Project Settings** tab. **Note:** You can only edit this field when the initial resolution is modified.
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 below RGB, HSL, or HSV controls to set the color.
color > RGB		
red	1	Sets the red value of the pixels.
green	1	Sets the green value of the pixels.
blue	1	Sets the blue value of the pixels.
alpha	1	Sets the alpha value of the pixels.
color > HSL		
hue	0	Sets the hue of the pixels.

Control (UI)	Default Value	Function
saturation	0	Sets the saturation of the pixels.
lightness	1	Sets the lightness of the pixels.
alpha	1	Sets the alpha value of the pixels.
color > HSV		
hue	0	Sets the hue of the pixels.
saturation	0	Sets the saturation of the pixels.
value	1	Sets the value of the pixels.
alpha	1	Sets the alpha value of the pixels.

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.
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 speficy 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.

Control (UI)	Default Value	Function
inputs		
[identifier]	none	Defines short input identifiers. The identifier is included in the input/output input names and is used as a prefix for the output product rep. • %V is replaced with the view name. • %v is replaced with the appropriate product token. **Note:** Input identifiers have no effect on file names, only products.
mode	file	Sets whether to write a file or define a dependency: • **file** • **dependency**
file	none	Sets the file path and name for the rendered image(s). **Note:** If **mode** is set to **dependency**, this control is hidden.
inputs > mode: file > image		
channels	Input	Selects the channels to render: • **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).
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).
colorConvert	enabled	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. 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 reprojection of a log plate) and you want to name the output file log without a linear to log conversion.

Control (UI)	Default Value	Function
fileFormat	exr	Sets the file format to output: • **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 > rawData: disabled		
colorspace	Inf	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.
inputs > mode: file > image > fileFormat: exr		
exrCompression	Wavelet	Defines the exr compression method to use. All methods are lossless (with the exception of **Pixar 24**, which is lossless but quantisizes 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: • **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: • **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.

Control (UI)	Default Value	Function
exrTileWorldAlign	disabled	When enabled (in conjuction 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). Note: this does not guarantee that tiles are aligned - it merely attempts to meet this condition.
inputs > mode: file > image > fileFormat: rla		
rlaBitDepth	auto	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: • **8-bit** • **10-bit** • **16-bit** • **32-bit**
inputs > mode: file > image > fileFormat: png		
pngBitDepth	auto	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: • **8-bit** • **16-bit**
inputs > mode: file > image > fileFormat: tif or tiff		
tifCompression	LZW	The tiff compression method to use: • **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 compatability in the target reader.
tifBitDepth	auto	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: • **8-bit** • **16-bit** • **32-bit**
tifPredictor	None	The predictor type to use when **tifCompression** is enabled: • **None** - No prediction is used. • **Horizontal** - Horizontal prediction is used. This can result in smaller file sizes, but may present compatability 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	The frame size to write to the file: • **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	The image area to write to the file: • **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.
inputs > mode: file > bounds > overscan		
left	0	Overscan specifies the number of pixels to pad the render request in each direction during a hot 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. 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. 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.
bottom	0	
right	0	
top	0	

Control (UI)	Default Value	Function
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		
setActiveFrameRange	disabled	Sets how the active frame range for rendering is defined: • When enabled, the **activeFrameRange** controls are displayed which define the active frame range for rendering. • When disabled, Katana assumes that the active frame range is the same as the range between globals.inTime and globals.outTime. 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.
farmFileName	none	Defines the farm file name and path.
versionUp	Auto	Sets whether the outputs of this node are versioned up when rendered on the queue: • **Auto** - use the global setting specified in the outline file. • **Yes** - outputs version up. • **No** - outputs don't version up.
threadable	enabled	Determines whether the queue is allowed to assign multiple cores to a frame of this render. • When enabled, the queue may optionally thread the render. • When disabled, the queue must use only one core.
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.
excudeFromFarmOutput-Generation	disabled	When enabled, this node does not appear in any generated farm file (however, the node is still renderable if called directly). Enabling this control hides the **forceFarmOutputGeneration** control.
farmSettings > excudeFromFarmOutputGeneration: disabled		
forceFarmOutputGenera-tion	disabled	When enabled, this node always appears in generated farm files (regard-less of whether it has any valid outputs). Note: If **excludeFromFarmOutputGeneration** is also enabled, the node does not appear in the generated farm file (**excludeFromFarmOutputGen-eration** overrides **forceFarmOutputGeneration**).

Control (UI)	Default Value	Function
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.

Importomatic

The Importomatic is a super tool node with a custom interface to load and manage different geometry types, including assembly component, sprib, and Houdini geo. 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 will indicate 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 LookFiles (.klf) and are added as new Scene Graph locations in the scene when a LookFile is resolved. A common use of the InfoCreate node is to provide documentation and/or version specific information (either baked in a LookFile or as an InfoCreate node in a macro).

Control (UI)	Default Value	Function
leafName	info	If a **leafName** is populated, the info is created below each specified item in the locations parameter array. Common leaf names are: readme, info, and user.
locations	/root/world	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 array values can be modified by using the **Add Locations** menu options. **Note:** An empty or invalid location value generates a location of type **error** under /root.
locations > Add Locations		
Path	N/A	Adds another path to this parameter's list of paths.
Append Scenegraph Selection	N/A	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	N/A	Removes all existing paths and replaces them with paths populated with the currently selected **Scene Graph** locations.
Append Nodegraph Node Locations	N/A	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	N/A	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	N/A	Removes all paths from this parameter.
text (html editor)	view editor and preview	Sets the mode for the html editor: • **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).

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.

Note: 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 mode 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.

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** and select **Collect 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	This is a list of locations to keep while every other location is removed by the Isolate.
isolateLocations > Add Locations		
Path	N/A	Adds another path to this parameter's list of paths.
Append Scenegraph Selection	N/A	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	N/A	Removes all existing paths and replaces them with paths populated with the currently selected **Scene Graph** locations.
Append Nodegraph Node Locations	N/A	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	N/A	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.

Control (UI)	Default Value	Function
Clear All	N/A	Removes all paths from this parameter.
isolateFrom	/root/world/geo	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 (e.g. /root/world/geo/somecharacter), then set **isolateLocations** to the shape you'd like to keep.
Enable secondary (inverse) output	disabled	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: /root/world/geo /root/world/geo/box /root/world/geo/circle /root/world/lgts If /root/world/geo/box is isolated using isolateFrom **/root/world/geo**, the secondary output contains **/root/world/geo/circle**. **Note:** Any Scene Graph location that is a peer or ancestor of the isolationRoot is present on both outputs.

5 Nodes L–Q

LightCreate

This node is used to create a Scene Graph containing a light. LightCreate does not load the light from any file or product but instead builds an entirely novel light from the parameters you specify on this node. This node is not used generally, the Gaffer node is used instead.

Note: LightCreate requires a light shader to function properly.

LightCreate and CameraCreate are identical, except for the type of Scene GraphLocations they create, and the population of the lightList vs. cameraList.

Control (UI)	Default Value	Function
name	/root/world/lgt/light	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.
projection	perspective	Sets the light projection mode: • **perspective** • **othographic**
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 light.
orthographicWidth	30	Sets the orthographic projection width.
includeInCameraList	Yes	When enabled, the light is visible in the Camera List.
radius	1	Sets the light's radius.
previewColor		
previewColor	1, 1, 1	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.

Control (UI)	Default Value	Function
previewColor > RGB		
red	1	Sets the red value of the fade color.
green	1	Sets the green value of the fade color.
blue	1	Sets the blue value of the fade color.
previewColor > HSL		
hue	0	Sets the hue of the fade color.
saturation	0	Sets the saturation of the fade color.
lightness	1	Sets the lightness of the fade color.
previewColor > HSV		
hue	0	Sets the hue of the fade color.
saturation	0	Sets the saturation of the fade color.
value	1	Sets the value of the fade color.
manipulators > Add		
Add Entry	none	Adds entries to the manipulators list.
manipulators > manipulator0		
name	none	
preset	default	
selected	Handles	• Handles – • Outline – • None –
visible	Outline	• Handles – • Outline – • None –
transform		
interface	SRT Values	Sets the transform control layout: • **SRT Values** – exposes the scale, rotation, and translation controls. • **Transform Matrix** – exposes a matrix to control transformations.

Control (UI)	Default Value	Function
transformOrder	Scale Rotate Translate	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	XYZ	Sets the order in which rotation is applied: • XYZ • XZY • YXZ • YZX • ZXY • ZYX
transform > interface: SRT Values		
translate	0.0, 0.0, 0.0	Controls light translation on the xyz axes.
rotate	0.0, 0.0, 0.0	Controls light rotation on the xyz axes.
scale	1.0, 1.0, 1.0	Controls light scale on the xyz axes.
transform > interface: Transform Matrix		
matrix	1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 1.0	Controls transformations using a matrix in place of individual SRT controls.
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 light list 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.

Notes

- The Gaffer node uses a LightLink internally to provide light linking; the user interface there is substantially similar to the LightLink node.
- 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 will 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	Determines whether the link is acting upon the light's illumination or shadow visibility of the specified objects: • **illumination** • **shadow visibility** **Note:** Shadow visibility is only currently respected by Arnold renders.
action	exclusive on	Controls what the LightLink node's behavior: • **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 will be 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.
objects	none	Sets which object(s) to operate on.
objects > Add Statements		
Paths	N/A	Adds a **Paths** list to this CEL parameter.
Collections	N/A	Adds a **Collections** list to this CEL parameter.
Custom	N/A	Adds a **Custom** parameter to this CEL parameter.
Append Scenegraph Selection	N/A	Adds a **Paths** list to this CEL parameter and places selected **Scene Graph** locations in the new list.
Replace With Scenegraph Selection	N/A	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	N/A	Copies this CEL statement to the clipboard.
Paste CEL Statement	N/A	Removes any parameters within this CEL parameter and pastes the CEL statement in the clipboard to this parameter.

Control (UI)	Default Value	Function
Replace With Parameter Expression	N/A	Converts the current CEL parameter into an expression.
lightMode	CEL	Controls how you specify which lights to operate on: • **CEL** • **Paths** - included for backward compatibility.
lightMode: CEL		
lights	none	When **lightMode** is set to **CEL**, this CEL statement is used to select the lights to operate on.
lightMode: Paths		
lightPaths	none	When **lightMode** is set to **Paths**, this list of light path names is used as the set of lights to operate on.

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 extracts lights from components and makes them renderable from a LookFile.

Control (UI)	Default Value	Function
locations	none	Sets the locations of lights from a path, the Scene Graph, or the Node Graph by using the **Add Locations** dropdown menu. The Scene Graph and Node Graph can be replaced or appended from this location. Lights can be dragged to the locations box.
locations > Add Locations		
Path	N/A	Adds another path to this parameter's list of paths.
Append Scenegraph Selection	N/A	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	N/A	Removes all existing paths and replaces them with paths populated with the currently selected **Scene Graph** locations.
Append Nodegraph Node Locations	N/A	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	N/A	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	N/A	Removes all paths from this parameter.

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**.

LiveGroup

The LiveGroup node is similar to the Group node except the contents are loaded from an external file. The contents of the LiveGroup are locked (not editable), and used mainly during look development. To change the contents of a LiveGroup, you must modify the file it references.

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 will be 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.

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 page 156.

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.
[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.

Control (UI)	Default Value	Function
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 Selectde 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	Sets the path to load in the source (file) as the contents of the LiveGroup.
source		
Browse...	N/A	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	N/A	Changes the name of the node to match the filename but without the path or extension.

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	Describes one or more Scene Graph path(s) to the location(s) to be created.
locations > Add Locations		
Path	N/A	Adds another path to this parameter's list of paths.
Append Scenegraph Selection	N/A	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	N/A	Removes all existing paths and replaces them with paths populated with the currently selected **Scene Graph** locations.
Append Nodegraph Node Locations	N/A	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	N/A	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.

Control (UI)	Default Value	Function
Clear All	N/A	Removes all paths from this parameter.
attrs	None	Drag string or number attributes here to have them added to the Scene Graph location(s) created by this node.

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**.
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 > input0		
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	99999999999999997748 809823456034029568	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	99999999999999997748 809823456034029568	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	Collection Expression Language used to specify Scene Graph locations on which an operation or assignment acts. **Note:** CEL statement should match locations of type **level-of-detail group**.
CEL > Add Statements		
Paths	N/A	Adds a **Paths** list to this CEL parameter.
Collections	N/A	Adds a **Collections** list to this CEL parameter.
Custom	N/A	Adds a **Custom** parameter to this CEL parameter.
Append Scenegraph Selection	N/A	Adds a **Paths** list to this CEL parameter and places selected **Scene Graph** locations in the new list.
Replace With Scenegraph Selection	N/A	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	N/A	Copies this CEL statement to the clipboard.
Paste CEL Statement	N/A	Removes any parameters within this CEL parameter and pastes the CEL statement in the clipboard to this parameter.
Replace With Parameter Expression	N/A	Converts the current CEL parameter into an expression.
CEL: Paths > Action		
Add Scenegraph Selection	N/A	Adds the currently selected **Scene Graph** location to this list.
Remove Scenegraph Selection	N/A	Removes the currently selected **Scene Graph** location from this list.
Remove Selected Paths	N/A	Removes the path(s), selected in this **Paths** list, from this list.
Select All	N/A	Selects all the paths in this list.
Select Selected Paths In Scenegraph	N/A	Selects the **Scene Graph** locations of the selected paths in this list.
Copy Selected Paths to Clipboard	N/A	Copies the selected paths from this list to the clipboard.

Control (UI)	Default Value	Function
Show Extended View...	N/A	Brings up a dialog with the contents of this **Paths** list.
CEL: Collections > Action		
Add Collections From Scenegraph Selection...	N/A	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...	N/A	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	N/A	Removes the selected collection(s) from this list.
Select All	N/A	Selects all the collections in this list.
Copy Selected Paths to Clipboard	N/A	Copies all the selected collections and their paths to the clipboard.
"Find And Select" Selected Items...	N/A	
mode	by index	Sets the method used to specify levels of detail: • **by index** • **by tag** • **by weight**
mode : by index		
selectionIndex	0	Sets the index of which LOD child to keep.
mode : by tag		
selectionTag	hi	Sets the tag of which LOD child to keep.
mode : 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	Collection Expression Language used to specify Scene Graph locations on which an operation or assignment acts. **Note:** CEL statement should match locations of type **level-of-detail group**.
CEL > Add Statements		
Paths	N/A	Adds a **Paths** list to this CEL parameter.
Collections	N/A	Adds a **Collections** list to this CEL parameter.
Custom	N/A	Adds a **Custom** parameter to this CEL parameter.
Append Scenegraph Selection	N/A	Adds a **Paths** list to this CEL parameter and places selected **Scene Graph** locations in the new list.
Replace With Scenegraph Selection	N/A	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	N/A	Copies this CEL statement to the clipboard.
Paste CEL Statement	N/A	Removes any parameters within this CEL parameter and pastes the CEL statement in the clipboard to this parameter.
Replace With Parameter Expression	N/A	Converts the current CEL parameter into an expression.
CEL: Paths > Action		
Add Scenegraph Selection	N/A	Adds the currently selected **Scene Graph** location to this list.
Remove Scenegraph Selection	N/A	Removes the currently selected **Scene Graph** location from this list.
Remove Selected Paths	N/A	Removes the path(s), selected in this **Paths** list, from this list.
Select All	N/A	Selects all the paths in this list.
Select Selected Paths In Scenegraph	N/A	Selects the **Scene Graph** locations of the selected paths in this list.
Copy Selected Paths to Clipboard	N/A	Copies the selected paths from this list to the clipboard.
Show Extended View...	N/A	Brings up a dialog with the contents of this **Paths** list.
CEL: Collections > Action		
Add Collections From Scenegraph Selection...	N/A	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...	N/A	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	N/A	Removes the selected collection(s) from this list.
Select All	N/A	Selects all the collections in this list.

Control (UI)	Default Value	Function
Copy Selected Paths to Clipboard	N/A	Copies all the selected collections and their paths to the clipboard.
"Find And Select" Selected Items...	N/A	
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 > lod0		
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	99999999999999997748809823456034029568	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	99999999999999997748809823456034029568	When the bounding box is transformed to screen space, if its pixel count is less than the **maxVisible** parameter, the object is not displayed.

LookFileAssign

Assigns a LookFile to a Scene Graph location defined by a CEL statement.

Control (UI)	Default Value	Function
CEL	None	Specifies the Scene Graph location(s) where the LookFile is assigned.
asset	None	The LookFile that is assigned to the specified Scene Graph location(s).

LookFileBake

Bakes a LookFile for a Scene Graph location(s) specified in the **rootLocations** field.

Control (UI)	Default Value	Function
rootLocations		
rootLocations	/root/world	Sets the Scene Graph location(s) to bake the LookFile information for. Any location under **/root/world** can be used, but it is recommended that components or assembly locations are specified.
passes		
passes	None	Passes are typically render passes, but could also be auxiliary baking passes for generating pointclouds or brickmaps. A LookFile can have one or multiple passes. To add a pass, select **Add** > **Add Pass Input.** 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. **Note**: All pass names must be unique.
saveTo	None	Sets where to store the baked LookFile.
options		
includeGlobalAttributes	No	When set to **Yes**, GlobalAttributes are stored in the LookFile.
includeLodInfo	No	When set to **Yes**, the level of detail information is stored in the LookFile.
alwaysIncludeSelected-MaterialTrees	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.
Write Look File	N/A	Click to bake the LookFile.

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 165

Control (UI)	Default Value	Function
asset	none	The asset to assign to the LookFile.
resolveImmediately	No	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 LookFile assets This option has special behavior during LookFile baking. Instead of resolving, it appends the Look to the lookfile.referencedAssets attribute. This gets included in the resulting LookFile and maintains a live reference to it during subsequent Katana standard resolution.
Flush Look File Cache	N/A	Click to flush the LookFile cache and force a reload.

LookFileLightAndConstraintActivator

Katana maintains a list of lights, cameras, and constraints at **/root/world** within the Scene Graph. When a LookFile 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. 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 product, 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 product 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 LookFiles that have been set up in another scene. Each LookFile piece of imported geometry passed into this node must be assigned through a LookFileAssign node. Once the LookFile is assigned, LookFileManager decodes the LookFile into the passes set up by the look development artist using a LookFileBake node.

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

LookFileMaterialsIn

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

Control (UI)	Default Value	Function
lookfile	none	Sets the LookFile path and name.
passName	none	Sets the pass name to use from the LookFile.
asReference	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 LookFile from which they got their material.
locationForMaterials	Load at original location	Sets where in the Scene Graph to import the materials from: • **Load at original location** - the materials maintain the same location. • **Load at specified location** - provides a parameter, userLocation, that acts as a namespace for the material palette. For instance, a material at /root/materials/geo/chrome with userLocation default_pass is placed at /root/materials/lookfile/default_pass/geo/chrome.

LookFileMaterialsOut

Use this node to write incoming materials into a LookFile. 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 LookFile to contain the material.
Write Look File	N/A	Click to write the material to the specified LookFile.

LookFileResolve

This node applies a specific pass from assigned LookFiles 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 LookFile pass to use.
Flush Look FIle Cache	N/A	Click to flush the LookFile 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	Determines the node's behavior:
		• **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 making changes to when the original Material node which created this location is not within the current session or for multiple branches of a graph.
		• **override materials** - accepts drops of attributes beneath material attribute groups. This can be used in two ways:
		  • 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 **light material**, 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.

action: create new material		
namespace	none	
makeInteractive	Yes	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 render specific shader to the material. The shader available changes depending on the renderers installed.
action: create from LookFile		
namespace	none	
makeInteractive	Yes	When set to **Yes**, you can drag objects in the Viewer and Katana retains the information from the Viewer.
lookfile	none	
materialPath	N/A	
asReference	Yes	

Control (UI)	Default Value	Function
action: create child material		
makeInteractive	No	
Add shader	N/A	Click to add a render specific shader to the material. The shader available changes depending on the renderers installed.
location	none	
action: edit material		
makeInteractive	Yes	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 render specific shader to the material. The shader available changes depending on the renderers installed.
location	none	
action: override materials		
CEL	none	
attrs	none	

MaterialAssign

Assigns materials to geometry in the Scene Graph.

Control (UI)	Default Value	Function
CEL	none	Sets the CEL specification of Scene Graph locations on which the assignment will act.
materialAssign	none	Specifies the material to assign. Typically, you'll middle-mouse drag this from under /root/materials/geo in the Scene Graph.

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** panel. 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 in the Node Graph on the Merge node input ports to delete any unused ports.
- Right-click on a node while connecting a link in the Node Graph 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 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 will be hidden from the user inside a show macro so it's unlikely you'll need the advanced options.
showAdvancedOptions: Yes		
advanced		
sumBounds	No	When enabled, bound attributes are queried for each relevant input location and the unioned results are used. The output bounding box at each location will be expanded to be large enough to contain all the inputs at that location. This is important when merging renderable geometry together inside of components.

Control (UI)	Default Value	Function
preserveWorldSpaceX-form	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 will force 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.
preserveInheritedAttrib-utes	none	Displays a list of attribute names for which inheritance should be pre-served when choosing between inputs of the Merge. Whenever a child location's source input differs from that of its parent, these attributes will be queried globally and applied locally the child location. This is often used to preserve attributes like the source sprib for a piece of geometry, so that the hair procedural will work correctly on the leaf shapes.
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 will be given 'first crack' at providing the attribute in the result before the general rule is used. This is often used when merging two ver-sions of a component to form a single output model; the first input will provide most of the attributes, but a second input might provide cor-rectly 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.
inputs		
inputs	none	Allows you to name the inputs on the Merge node.

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, etc), that you don't want to mistakenly have set for a batch render. Using a NonpersistantSwitch ensures that batch renders always get the left-most input to the node.

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 coverts 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 covert 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	• **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.
g	0	Sets the exposure level for the green channel.
b	0	Sets the exposure level for the blue channel.
context		
key1	none	

Control (UI)	Default Value	Function
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	
content		
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: • **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.
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.

OrientConstraint

OrientConstraint matches the rotation (orientation) of the object in **basePath** to the object in **targetPath**. See also ParentChildConstraint on page 175 and PointConstraint on page 175.

Control (UI)	Default Value	Function
basePath	none	Sets the location of the object to constrain.
targetPath	none	Sets the location of the object(s) to constrain the object in **basePath** to.

Control (UI)	Default Value	Function
targetOrientation	Object	Sets the type of bounds to use for the target object(s): • **Object** • **Face**
xAxis	Yes	Constrains the x Axis.
yAxis	Yes	Constrains the y Axis.
zAxis	Yes	Constrains the z 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.

ParentChildConstraint

Constrains the translate, rotate, and scale values of one object (the parent) to another (the child). See also OrientConstraint on page 174 and PointConstraint on page 175.

Control (UI)	Default Value	Function
basePath	None	Defines the child object.
targetPath	None	Defines the parent object.

PointConstraint

Applies a constraint that translates the base object to a point defined by the target object(s). See also OrientConstraint on page 174 and ParentChildConstraint on page 175.

Control (UI)	Default Value	Function
basePath	None	Defines the location of the object to constrain.
targetPath	None	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.
baseOrigin	Object	Sets how the center of the base object is calculated: • **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.

Control (UI)	Default Value	Function
targetOrigin	Object	Sets how the center of the target object is calculated: • **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	When set to **Yes**, silently ignore the contraint if its target is not in the Scene Graph. When set to **No**, produce an error on contraint resolution if the target is missing.

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	Describes the Scene Graph location where the object is created.
type	sphere	Sets the type of primitive created (plane, sphere, etc.)
transform		
interface	SRT Values	Sets the transform control layout: • **SRT Values** – exposes the scale, rotation, and translation controls. • **Transform Matrix** – exposes a matrix to control transformations.
transformOrder	Scale Rotate Translate	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

Control (UI)	Default Value	Function
rotationOrder	XYZ	Sets the order in which rotation is applied: • XYZ • XZY • YXZ • YZX • ZXY • ZYX
interface: SRT Values		
translate	0.0, 0.0, 0.0	Controls light translation on the xyz axes.
rotate	0.0, 0.0, 0.0	Controls light rotation on the xyz axes.
scale	1.0, 1.0, 1.0	Controls light scale on the xyz axes.
interface: Transform Matrix		
matrix	1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 1.0	Controls transformations using a matrix in place of individual SRT controls.
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.

PrmanObjectSettings

The purpose of this node is to set prman attributes at levels of the Scene Graph hierarchy described by the given CEL statement. The important thing to note about this node is that due to the fact that it is changing different parts of the hierarchy it cannot display incoming values for each parameter. Only when a parameter of the node is set to Local or "Enable Assignment" is the parameter value valid and actually changing the attributes at the CEL statement defined Scene Graph locations.

To see the actual values in the hierarchy:

1. View the node whose Scene Graph you want to see.
2. Select the Scene Graph location whose attributes you want to examine.

3. In the **Attributes** panel, look under the **prmanStatements** section.

PrmanOutputChannelDefine

Control (UI)	Default Value	Function
name	none	
type	varying float	
params	none	

PrmanShadingNode

Control (UI)	Default Value	Function
name	none	
nodeType	varying float	
parameters		
publicInterface		
namePrefix		
pagePrefix		
nameRegExFind		
nameRegExReplace		
nameRegExFind		
pageRegExReplace		

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 will also have 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 149 and VisibilityAssign on page 194.

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 VisiblityAssign 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	The CEL statement to use to select locations to remove. For more information, see the CEL Reference document found on the documentation HTML page (accessed through the **Help** > **Documentation** menu option).

6 Nodes R–Z

ReflectionConstraint

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

Control (UI)	Default Value	Function
basePath	none	Sets the object to constrain.
targetPath	none	Sets the object(s) to constrain the object in **basePath** to.

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.
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 though \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 when rendering on the cue. Each Render node is intended to be a single invocation of Renderman or other renderer.

The Render node is really only used to track cue settings, product 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:

- Additional inputs to the Render node are dependency inputs and are only used when generating outline files for cue rendering.

Control (UI)	Default Value	Function
passName	Render	Sets the passName to identify this render node and is used to build the name of products written from the Render node. 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.
lock	disabled	When enabled, the product information for this Render node is no longer updated. 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 or render on the cue (because the product it produces is locked). It can be referenced in expressions with 'getRenderLocation'.
outputs		
outputs	none	Manages which available outputs are active.
farmGlobalSettings		
setActiveFrameRange	disabled	When enabled, activeFrameRange parameters are exposed to define the active frame range for rendering. When disabled, the active frame range is assumed to be the same as globals.inTime and globals.outTime. **Note:** The active frame range affects farm file generation and ShadowRenderAssign node assignments.
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.
dependAll	disabled	When enabled, farm dependencies wait until all frames of this node are rendered before rendering themselves.
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 will still be renderable if called directly).

Control (UI)	Default Value	Function
excludeFromFarmOutputGeneration: disabled		
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**).

RenderOutputDefine

Specifies output of an image (color, AOV, shadow map, etc.) 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, etc. This name appears in the Render node, along with (or as) the default **primary**.
type	color	Specifies the type of output. • **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 will setup 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.

Control (UI)	Default Value	Function
convertSettings		
fileExtension: exr		
exrCompression	Scanline ZIP	Defines the exr compression method to use. All methods are lossless (with the exception of **Pixar 24**, which is lossless but quantisizes 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: • **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: • **Tiled** - random tile access. • **Scanline** - random scanline access.
clampOutput	No	When set to **Yes**, post-render clamp negative rgb values to 0, and clamp alpha values to 0-1. **Note:** clampOutout has no effect on NaN and inf values.
colorConvert	Yes	When set to **Yes**, post-render convert rendered image data from linear to the output colorspace specified in the filename. The default value of Yes is suitable for nearly every situation, since the linear output of the render will be 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.
computeStats	None	Allows you to compute image statistics as a post process, appending as exr metadata. Select: • None • Raw • Depth **Note:** In depth mode, zero values and very large values are ignored. In both modes, only the region within the dataWindow is considered.
fileExtension: png		
pngBitDepth	16	Sets the bit depth of the rendered file: • **8-bit** • **16-bit**

Control (UI)	Default Value	Function
fileExtention: rla		
rlaBitDepth	16	Sets the bit depth of the rendered file: • **8-bit** • **10-bit** • **16-bit** • **32-bit**
fileExtention: tif		
tifBitDepth	16	The bit depth of the rendered file: • **8-bit** • **16-bit** • **32-bit**
tifCompression	LZW	The tif compression method to use: • **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 compatability in the target reader.
clampOutput	No	When set to **Yes**, post-render clamp negative rgb values to 0, and clamp alpha values to 0-1. **Note:** clampOutout has no effect on NaN and inf values.
colorConvert	Yes	When set to **Yes**, post-render convert rendered image data from linear to the output colorspace specified in the filename. The default value of Yes is suitable for nearly every situation, since the linear output of the render will be 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.
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.
cameraName	none	Describes the Scene Graph location of camera to render from. If empty, render from the camera specified in **renderSettings.cameraName** at /root.

RenderScript

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

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

Control (UI)	Default Value	Function
pythonImports		
pythonImports	none	Array of import statements to be added to the Python farm file. Example: from outline.modules.shell import Shell or import outline.module.shell
farmSettings		
setActiveFrameRange	disabled	When enabled, activeFrameRange parameters are exposed to define the active frame range for rendering. When disabled, the active frame range is assumed to be the same as globals.inTime and globals.outTime. **Note:** The active frame range affects farm file generation and ShadowRenderAssign node assignments.
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.
dependAll	disabled	When enabled, farm dependencies wait until all frames of this node are rendered before rendering themselves.
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 will still be renderable if called directly).
excludeFromFarmOutputGeneration: disabled		
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**).

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 at path to the camera's location in the Scene Graph.
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 ouput 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 normalizated coordinates: xmin xmax ymin ymax, starting in the upper leftahand 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.

ScaleConstraint

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

Control (UI)	Default Value	Function
basePath	None	Sets the object to constrain.
targetPath	None	Sets the object(s) to constrain the object in **basePath** to.

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.

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.

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.
asset	none	Describes the file path and name of the XML scene file.
timing		
mode	Current Frame	Sets the timing mode to apply to the asset: • **Current Frame** • **Hold Frame** • **Clamp Range**

ScreenCoordinateConstraint

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

Control (UI)	Default Value	Function
basePath	None	Sets the object to constrain.
targetPath	None	Sets the object(s) to constrain the object in **basePath** to.

Control (UI)	Default Value	Function
allowMissingTargets	No	When set to **Yes**, silently ignore the contraint if its target is not in the Scene Graph. When set to **No**, produce an error on contraint resolution if the target is missing.
respectMotionBlur	Yes	When set to **Yes**, constraints are adjusted to allow for the target's motion within the time the shutter is open.
targetBounds	Box	The type of bounds to use for the target object(s): • **box** - the screen window is constrained to fit the bounding box of the target object(s). • **sphere** - the screen window is constrained to fit a sphere that encloses the bounding box of the target objects(s).
keepAspectRatio	Yes	When set to **Yes**, maintain the aspect ratio of the screen window. When set to **No**, modify the aspect ratio of the screen window to fill as much of the frame as possible.

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.

Notes

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	Sets the objects to prune: • **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	Acts like a render ouput device with override capabilities. Select: • **no override** • **primary shadow** • **primary deepshad** • **primary shadow, secondary deepshad** • **primary color, secondary deepshad.**
shadowBranch		
resolution	512x512	Sets the resolution of the image(s).
pixelSamples	1.0, 1.0	Sets the subpixel sampling rate (X x Y samples)
zthreshold	0.9960, 0.9960, 0.9960	Sets the minimum opacity for objects to appear in shadow maps.
deepshadowerror	0.01	Sets the allowable error in deep shadow generation, where 0 equals lossless.
depthfilter	min	Sets the filter to use on depth (Z) values: • **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 176.

Control (UI)	Default Value	Function
name	/root/world/geo/primitive	Describes the Scene Graph location where the object is created.
type	teapot	Sets the type of primitive created (plane, sphere, etc.)

Control (UI)	Default Value	Function
transform		
interface	SRT Values	Sets the transform control layout: • **SRT Values** – exposes the scale, rotation, and translation controls. • **Transform Matrix** – exposes a matrix to control transformations.
transformOrder	Scale Rotate Translate	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	XYZ	Sets the order in which rotation is applied: • XYZ • XZY • YXZ • YZX • ZXY • ZYX
interface: SRT Values		
translate	0.0, 0.0, 0.0	Controls light translation on the xyz axes.
rotate	0.0, 0.0, 0.0	Controls light rotation on the xyz axes.
scale	1.0, 1.0, 1.0	Controls light scale on the xyz axes.
interface: Transform Matrix		
matrix	1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 1.0	Controls transformations using a matrix in place of individual SRT controls.
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.

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 will appear 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.

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.
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 will happen.
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.

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	Collection Expression Language. Used to specify Scene Graph locations on which an operation or assignment will act. For more information, see the CEL Reference document found on the documentation HTML page (accessed through the **Help** > **Documentation** menu option).
fps	24	Defines frames per second. Used to detmine the amount of the velocity (which is defined in units per second) to apply to the geometry attributes.
velocityScale	1	Defines a muliplier on the velocity attribute, where 1 = no change.

ViewerObjectSettings

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

Control (UI)	Default Value	Function
CEL	none	Collection Expression Language. Used to specify Scene Graph locations on which an operation or assignment will act. For more information, see the CEL Reference document found on the documentation HTML page (accessed through the **Help** > **Documentation** menu option).
drawOptions		
hide	No	Sets whether the object should be hidden in the Viewer.
fill	inherit	Sets how the object is displayed, as: • **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.

Control (UI)	Default Value	Function
light	inherit	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: • **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.
smoothing	inherit	When the objects referenced by the CEL statement are beign displayed as points or lines, this parameter sets whether they should be anti-aliased. The options are: • **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 considerd 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.
annotation		
text	None	Sets the text to display with the geometry. When empty, no tag is displayed.
color	0.4, 0.4, 0.4	Sets the default background color for any annotation text.
pickable	Yes	Sets whether the object is pickable or not.

VisibilityAssign

The VisibilityAssign node changes the visiblility 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 will pick up the parent's visibility setting of 0, but the item to render will use its explicitly set value of 1.

The Scene Graph displays visibility of each Scene Graph item as icons.

Control (UI)	Default Value	Function
CEL	none	Collection Expression Language. Used to specify Scene Graph locations on which an operation or assignment will act. Specifies what part of the Scene Graph to assign this attribute to. For more information, see the CEL Reference document found on the documentation HTML page (accessed through the **Help** > **Documentation** menu option).
visible	1	Sets the visibility of objects in the render. 0 specifies not visible in render and anything else specifies visible in render.