KATANA

REFERENCE GUIDE

VERSION 1.0v3

Katana software engineering: Andy Lomas, Jonathan Attfield, Chris Hallam, João Montenegro, Örn Gunnarsson, Andrew Bulhak, Steve LaVietes, Jeremy Selan, Brian Hall
Product testing: Phillip Mullan, Richard Ellis
Katana customer support engineer and supplementary testing: Iulia Giurca
Initial writing: Sony Pictures Imageworks
Main reference guide authors: Charles Quinn, Joel Byrne, Eija Närvänen, Tytti Pohjola, Gary Jones
Proof reading: Charles Quinn, Eija Närvänen, Gary Jones

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

Rev: January 6, 2012

Contents

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.*

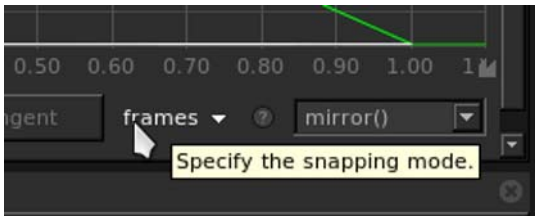
Getting Help

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.

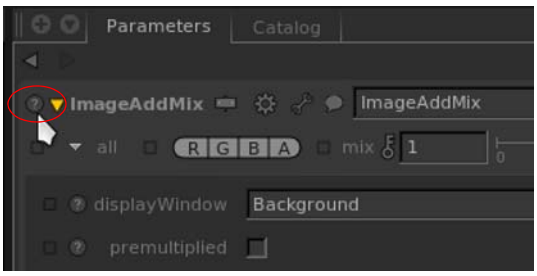
Viewing Online Help

Katana features several forms of online 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** panels 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:
  - **Documentation –** this reference guide and the Katana user guide.
  - **Quick Reference** – a list of easy access reference documents.
  - **Node Reference** – the HTML node reference guide.
  - **API Reference** – information on Katana APIs.

Contacting Customer Support

Should questions arise that this manual or the online 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 a scene.

Control (UI)	Default Value	Function
basePath	None	The object to constrain.
targetPath	None	The object(s) to constrain the object in **basePath** to.
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.
baseAimAxis	0.0, 0.0, -1.0	The axis of the base object that is pointed at the target. Adjusting these values changes the side of the object that is aimed at the target.
baseUpAxis	0.0, 1.0, 0.0	The axis of the base object that is pointed upwards relative to the target. Adjusting these values changes the rotation of the base object, while keeping the aim constant.
targetUpAxis	0.0, 1.0, 0.0	The 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-graphGeneraror	Specifies the Scene Graph location where the Alembic asset is to be placed.
abcAsset	none	Specifies where to retrive the asset input for an Alembic (.abc) file.
fps	24	
timing		
mode	Current Frame	Sets the timing mode to apply to the asset: • Current Frame • Hold Frame • Clamp Range

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
channel	RGBA	Sets the output channel.

Control (UI)	Default Value	Function
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
driverParameters - these controls are dependant on the driver selected		

ArnoldGlobalSettings

The ArnoldGlobalSettings node is populated by the XML file located at PLUGINS/Resources/Renderers/Arnold3.3/GenericAssign. These parameter names and defaults can change between Arnold versions and, as such, they are provided for you to change.

Control (UI)	Default Value	Function
Anti-aliasing Settings		
AA_samples	1	The number of antialiasing samples. Arnold will fire (AA_samples * AA_samples) camera rays per pixel.
AA_samples_progressive_begin	1	A negative value means that the renderer generates a single camera ray for NxN pixels, so it is basically just shooting 1 ray instead of NxN. So for -5 you get 1 ray for every 25 pixels. Therefore if you were to set a -2, you get 1 ray for every 4 pixels.
AA_samples_progressive_step	2	Increment steps for progressive sampling (example: begin -1, step 2, AA 5: = render at -1, 1, 3, 5)
AA_filter	gaussian	An enumerated type which specifies the pixel filtering kernel: box, disk, triangle, cone, cubic, catrom, cook, mitnet, gaussian, catrom2, closest
AA_filter_width	2	The width in pixels of the multi-pixel filtering kernel.
AA_sample_clamp	10000000150474662198 76688855040	Clamp pixel samples so that they don't get bigger than a maximum value. This can sometimes help with antialiasing extremely bright pixels for motion blur, depth of field, etc, by effectively clamping the dynamic range of the pixel samples.
Lighting		

Control (UI)	Default Value	Function
GI_hemi_samples	2	Number of samples for indirect diffuse (aka global illumination, aka final gather, aka radiosity etc). For the HDRI shader, 3 is fast, 4-5 good quality.
GI_specular_samples	2	Like **GI_hemi_samples**, but for soft blurry reflection.
Ray Depth		
GI_total_depth	10	Ultimate limit on ray depth, regardless of ray type.
GI_diffuse_depth	0	The number of bounces considered for global illumination (0 = off). **Note:** Expect substantial increases in render times each time GI_diffuse_depth is raised by 1.
GI_glossy_depth	0	For glossy reflections, number of times a ray can bounce off a glossy surface.
GI_reflection_depth	2	For reflection rays (AI_RAY_REFLECTED), number of times a ray can bounce off a reflective surface.
GI_refraction_depth	2	For refraction rays (AI_RAY_REFRACTED), number of times a ray can pass through a refractive object.
Auto Transparency		
auto_transparency_mode	shadow_only	Sets the auto transparency mode: • always • shadow_only • never
auto_transparency_depth	10	Sets the depth value for auto transparency.
auto_transparency_threshold	0.99	Sets the threshold value for auto transparency.
Environment		
background	none	If you have defined or inherited a material to be used as a background shader, such as a sky_HDRI shader, insert the path to the material here.
background_visibility		
AI_RAY_CAMERA	disabled	Object/s visible to camera?
AI_RAY_SHADOW	enabled	Object/s considered for shadow rays?
AI_RAY_REFLECTED	enabled	Toggle to turn on/off for reflection rays
AI_RAY_REFRACTED	enabled	Toggle to turn on/off for refraction rays
AI_RAY_DIFFUSE	disabled	Toggle to turn on/off for diffuse global illumination rays
AI_RAY_GLOSSY	disabled	Toggle to turn on/off for blurred reflection rays (very slow)
AI_RAY_SCATTERED	disabled	Toggle to turn on/off for atmosphere and volumetric GI rays (very slow)
atmosphere	none	If you have defined or inherited a material to be used as an atmosphere shader, insert the path to the material here.

Control (UI)	Default Value	Function
GI_falloff_start_dist	999999995904	Controls the trace distance of GI rays in the scene. GI_falloff_start_dist specifies when the GI rays begin the decay, at which distance the color begins to blend to the background color.
GI_falloff_stop_dist	999999995904	Controls the trace distance of GI rays in the scene. GI_falloff_stop_dist specifies when the GI rays end the decay (GI intensity = zero), at which distance the color is fully blended to the background color.
SSS		
GI_sss_hemi_samples	2	Number of samples for sss (aka global illumination, aka final gather, aka radiosity etc)
sss_lazy_evaluation	disabled	With this option on, all points for the sss point cloud are generated, but the renderer doesn't calculate them until the mesh is encountered at render time. This results in a faster render, but is more expensive in memory.
sss_subpixel_cache	disabled	Only does sss lookups once per pixel. This can result in faster render times, but can result in artifacts in motion blurred renders or scenes where the lighting is changing. This parameter is planned to be deprecated soon.
single_scatter_samples	0	Controls the amount of samples in the single scatter method of subsurface lighting.
Shadows		
shadow_bias	1.00e-06	Shadow maps (depth maps), overwrites light shader.
shadows_obey_light_linking	enabled	If set to 'yes' light group(light link) will be used for shadowing to determine which object to cast shadow.
Camera		
camera_type	from camera location	Sets the camera type used: • Cyl • fisheye • ortho • persp
filter_map	none	This is a place to plug a shader for masking effects like vignettes. If the shader returns black the ray won't even be traced, otherwise it will just multiply the result.
Motion Blur		
shutter_type	box	Controls the filter used by the shutter: • box - on/off. • triangle - ramped.
Depth of Field		
focal_distance	0	The distance the rear principal plane of the lens is separated from the "film". This is prinically used when creating images with Depth of Field.

Control (UI)	Default Value	Function
aperture_size	0	The aperture is usually specified as an f-number, the ratio of focal length to effective aperture diameter. A lower f-number denotes a greater aperture opening. Reducing the aperture size increases the depth of field, which describes the extent to which subject matter lying closer than or farther from the actual plane of focus appears to be in focus. In general, the smaller the aperture (the larger the number), the greater the distance from the plane of focus the subject matter may be while still appearing in focus.
aperture_nsides	0	Instead of a circular aperture, specifies an n-sided polygon. This mimics the aperture in a real-world camera, which is usually not perfectly circular but has sides as a result of the leafed aperture.
aperture_ratio	1	nstead of a circular aperture, creates an elliptical aperture. Used for anamorphic depth-of-field.
aperture_relax	0	If an n-sided aperture is specified, then this parameter gradually smooths the shape and blends the aperture into a circle.
aperture_rotation	0	Rotates the aperture. This will be used if aperture_nsides or aperture_ratio is specified.
Scene Generation		
forceExpand	disabled	Geometry is looked at procedurally but horizontally expanded and inserted flat (as a flat object in the scene).
checkpointRender	disabled	If enabled, Katana will watch for the Arnold render to be killed and will copy intermediate data in exr files to the output render location. If enabled on a subsequent render, the unrendered regions of the image are rendered and stitched together with the previously rendered image data to produce the final frame. Only works for exr output, including vshads (does not work with deepshads).
assInclude	none	Path to an ass file which is source immediately prior to rendering.
assIncludeWhen	at start up	Determines when the ass file is included. This is relevant if the ass file contains light linking information. • at start up • following lights
strict_procedural_compatibility	disabled	If set to 'yes' arnold will perform procedural DSO compatibility check.
Error Settings		
error_color_bad_map	1.0000, 0.0000, 0.0000	Sets the display color for the specified error. Use the RGB, HSL, and HSV sliders or color picker to alter the color.
error_color_bad_mesh	0.0000, 1.0000, 0.0000	Sets the display color for the specified error. Use the RGB, HSL, and HSV sliders or color picker to alter the color.
error_color_bad_pix	0.0000, 0.0000, 1.0000	Sets the display color for the specified error. Use the RGB, HSL, and HSV sliders or color picker to alter the color.

Control (UI)	Default Value	Function
error_color_bad_shader	1.0000, 0.0000, 1.0000	Sets the display color for the specified error. Use the RGB, HSL, and HSV sliders or color picker to alter the color.
Debug Settings		
Ass File Generation		
assFile	none	Sets the name of the ass_file to which Arnold should save the scene. If there exists a value for this entry, then Arnold will not render the scene; It will write the scene's contents to an .ass file and then exit. If user adds **.gz** at the end of file name, the output ass file will be automatically compressed.
assFileContents	all	Allows you to limit what nodes are written to an assFile: • all • geometry • geometry and materials
Disable Features		
ignore_tmaps	disabled	When enabled, tmaps are are ignored.
ignore_shaders	disabled	When enabled, shaders are ignored.
ignore_atm_shaders	disabled	When enabled, atm shaders are are ignored.
ignore_lights	disabled	When enabled, lights are are ignored.
ignore_shadows	disabled	When enabled, shadows are are ignored.
ignore_subdivision	disabled	When enabled, subdviisions are are ignored.
ignore_displacement	disabled	When enabled, displacement is ignored.
ignore_smoothing	disabled	When enabled, smoothing is ignored.
ignore_sss	disabled	When enabled, subsurface scattering is ignored.
GI_direct_lighting	enabled	Enables or disables Direct lighting in a GI render. When disabled, use only the GI contribution of lights.
Misc Settings		
bucket_size	64	The size of a rendering bucket (in a single dimension). The total size of the rendering bucket is bucket_size * bucket_size pixels.
bucket_scanning	top	Sets the bucket scanning order.
arbitrary		
none	none	Allows the user to set new parameters that haven't yet been built into the Katana GUI. This is done using the AttributeSet node.

ArnoldObjectSettings

The ArnoldObjectSettings node is populated by the XML files located at PLUGINS/Resources/Renderers/Arnold3.3/GenericAssign. These parameter names and defaults can change between Arnold versions and, as such, they are provided for you to change.

Control (UI)	Default Value	Function
CEL Statement		
CEL	none	Collection Expression Language. Used to specify Scene Graph locations on which an operation or assignment acts. For more information, see the CEL Reference document in the **Help** > **Quick Reference** menu.
Geometry		
inv_normals	disabled	When enabled, the object's normals are inverted.
smoothing	disabled	Controls normal interpolation between polys.
Subdivision Surfaces		
iterations	0	Controls the maximum subdiv iterations. 0 = no subd, plain poly, n = catclark subdiv with n iterations.
NURBS		
nurbs_tesselate_u	5	Controls for tesselation of nurbs objects in u.
nurbs_tesselate_v	5	Controls for tesselation of nurbs objects in v.
Points & Curves		
min_pixel_width	0	Enables pixel width enlargement for improved anti-aliasing. Currently only applies to "curves" primitives.
self_intersectable	enabled	When enabled, curves are invisible to rays that start on them. Currently only applies to "curves" primitives.
Shading		
receive_shadows	enabled	When enabled, the object receives shadows.
self_shadows	enabled	When enabled, the object can shadow itself.
Subsurface Scattering		
sss_sample_spacing	0.1	Controls the average distance (in world space units) between two sample points. Note that the current code is not able to achieve this exactly - so this should be treated as a rough estimate.Although the sample generation settings are now decoupled from the SSS lookup settings, it is still important to have enough points in the pointcloud to get a clean result. This usually means having a sample spacing a few times lower than the average scattering radius.

Control (UI)	Default Value	Function
sss_max_samples	100000	Controls the maximum number of samples that can be generated on this shape. This is a helpful way to avoid generating excessive points by accidentally specifying too small of a threshold. Another way to use this is to always leave the spacing very small and use this as the effective number of samples to generate.
sss_use_gi	enabled	When enabled, global illumination is accounted for when caching lighting information. This has no effect if GI_diffuse_depth is 0, so it is usually a good idea to always leave it on. Advanced users can turn it off if they feel the point cloud generation is taking too long, or if the diffuse color bleeding is not visually important to the subsurface scattering term. Note that this only affects point cloud construction speed. Lookups simply access cached information.
sss_faceset	enabled	When enabled, facesets in Katana are utilized for sss.
Displacement		
displacement_padding	0	Sets the height of the displacement bounds.
omitDisplacement	disabled	When enabled, the displacement shader is not created or connected to geometry.
Ray Flags		
opaque	enabled	When enabled, the object is opaque.
trace_sets	none	Describes a string of objects for which the geometry listed in the CEL statement won't evaluate for selected types of rays. This functionality requires the shader to recognize and utilize an array of objects tagged for trace_sets
sidedness		
AI_RAY_CAMERA	enabled	Enables or disables double-sided visibility for camera rays.
AI_RAY_SHADOW	enabled	Enables or disables double-sided visibility for shadow rays.
AI_RAY_REFLECTED	enabled	Enables or disables double-sided visibility for reflections.
AI_RAY_REFRACTED	enabled	Enables or disables double-sided visibility for reflections.
AI_RAY_DIFFUSE	enabled	Enables or disables double-sided visibility for GI diffuse rays.
AI_RAY_SCATTERED	enabled	Enables or disables for atmosphere and volumetric GI rays.
visibility		
AI_RAY_CAMERA	enabled	Enables or disables the object's visibility to the camera.
AI_RAY_SHADOW	enabled	Enables or disables whether the object is considered for shadow rays.
AI_RAY_REFLECTED	enabled	Enables or disables reflection rays.
AI_RAY_REFRACTED	enabled	Enables or disables refraction rays.
AI_RAY_DIFFUSE	enabled	Enables or disables diffuse global illumination rays.

Control (UI)	Default Value	Function
AI_RAY_SCATTERED	enabled	Enables or disables blurred reflection rays (very slow).
AI_RAY_GLOSSY	enabled	Enables or disables atmosphere and volumetric GI rays (very slow).

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 will begin traversing 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 will be 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' will be evaluated. If 'toCEL' is empty, all locations in the destination scene are evaluated

Notes

This node will begin traversing the 'copyFrom' scene at location 'fromRoot', and the 'copyTo' 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 will be copied to the location specified by 'toAttr' (e.g. geometry.point.Pref) on the 'copyTo' location.

Control (UI)	Default Value	Function
fromRoot	/root/world	Defines the 'copyFrom' location.
fromAttr	none	Defines the attribute to be copied.
toRoot	/root/world	Defines the 'copyTo' location.
toAttr	none	Defines the location where the attribute will be copied to.
toCEL	none	Allows you to filter the evaluation of this node. Only locations in the destination scene that match 'toCEL' will be evaluated. If 'toCEL' is empty, all locations in the destination scene are evaluated.

AttributeEditor

The AttributeEditor node is used to edit specific attributes of objects in the Scene Graph. When viewing an AttributeEditor node, changes can be made to the Attributes tab (which is usually read-only) that are stored and applied in the AttributeEditor node.

Control (UI)	Default Value	Function
AttributeEditor		
exclusivity	none	Exclusivity locks the interactive (via Attribute Panel or maya viewer) edits of a location to a specific node. This is so you can drag around things made with LightCreate and PrimitiveCreate and have the edits go back to the original node which created it. Currently, this applies to all attributes on a location. If these locations are not in the "exclusivity" section, by default they cannot be edited in the AttributeEditor. Entries to the Exclusivity section are CEL statements; they can be paths, collections, or custom entries. For more information, see the CEL Reference document in the **Help** > **Quick Reference** menu.
Existing Edits Table	n/a	This is where edits made in the Attributes tab while viewing the node are stored. From here, you can delete or disable overrides. You can also move certain overrides back up the graph to the node that is being edited, and jump to that location; however, additions to this section must be made in the Attributes tab.
Existing Edits	none	The name of the attribute, grouped under Scene Graph location, is displayed in the Existing Edits column.
Index	none	When the edited attribute is a number array 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 Value column. E.g., if the surface color of a material was edited red, the value would be [32.0, 0.0, 0.0], and if the RenderSettings of the /root node were changed to hd, the value would be 'hd'. Clicking on the value pops up a simple string or number field allowing you to change the value (it won't be the custom ui typically associated with the attribute).

AttributeFile_In

This node reads in an Attribute file from a specified location.

Control (UI)	Default Value	Function
CEL Statement		
CEL	none	Collection Expression Language. Used to specify Scene Graph locations on which an operation or assignment acts. For more information, see the CEL Reference document in the **Help** > **Quick Reference** menu.
File Path	None	Describes the filepath to an Attributes File.
Custom File Parser	None	Specifies the **.so** file with the Attributes File parser. Leave it empty to use the default one.
Attribute Group Name	attributeFile	Specifies the Name for the Group Attribute where the attributes from the file are stored. Leave it empty to store the attributes directly under the location (without a group attribute).
Apply When	immediate	Determines when the script will run: • **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 Scene GraphLocationModifers 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 filters in the Scene Graph panel 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 • Hold Frame • Clamp Range

AttributeModifierDefine

Control (UI)	Default Value	Function
CEL Statement		
CEL	none	Collection Expression Language. Used to specify Scene Graph locations on which an operation or assignment acts. For more information, see the CEL Reference document in the **Help** > **Quick Reference** menu.
applyWhen	deferred	Sets when the modifier is applied: • immediate • deferred
modifierType	none	Sets the modifier type to apply. The **modifierType** determines whicj arguments are available in the **args** dropdown.
args: AttributeFile		
filepath	none	Describes the AttributeFile filepath.
sofilepath	none	Describes the **.so** filepath.
groupAttr	none	Describes the groupAttr filepath.

AttributeModifierResolve

This node resolves both AMPs and SLMs.

Control (UI)	Default Value	Function
resolveWithIds	all	Sets which AttributeModifiers are resolved: • **all** – resolves all modifiers. • **specified** – resolves only those modifiers specified in the **specifiedResolveIds** field.
specifiedResolveIds	none	When **resolveWithIds** is set to **specified**, 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 path (i.e. "visible" or "material.surfaceParams.Surf_Col"), this will return a list of values. If inherit is set to True, the attribute will be queried globally. If an attribute does not exist at the given attribute path, None is returned.
  - If asAttr is True, a successful query will return 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 can be useful in two ways:
    - They can 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.
- **SetAttr**(attributePath, [attrValue])
  - Given an attribute path, this will override 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 will be 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.
- **GetName**()
  - Returns the base name of the current Scene Graph location.
- **GetFullName**()
  - Returns the full path of the current Scene Graph location. This is useful as a random seed.
- **GetType**()
  - Returns the type of the current Scene Graph location.
- **GetFrameTime**()
  - Returns the current frame time as a float.
- **GetXform**([worldSpace=False])
  - Returns the local or worldspace transformation as a 16-element list of float values representing a 4x4 transform matrix.
- **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 Statements		
CEL	none	Collection Expression Language. Used to specify Scene Graph locations on which an operation or assignment acts. For more information, see the CEL Reference document in the **Help** > **Quick Reference** menu.

Control (UI)	Default Value	Function
applyWhen	immediate	Determines when the script will 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:** Plugin resolve and look file resolve may be tested either by enabling implicit Scene Graph filters in the Scenegraph panel 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 will evaluate 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.
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
script	none	Contains the Python scripts to run as specified.

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 overriden: • paths • CEL
mode: paths		
paths	none	Sets the paths of the attribute. For example, /root/world/geo
mode: CEL		
CEL	none	Sets the attribute location to be overridden.

Control (UI)	Default Value	Function
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.
attributeType	double	The type of the attribute that is overridden: • integer • double • float • string • group
groupInherit	Yes	Decides whether or not implicity-created groups are inherited lower in the Scene Graph hierarchy. For instance, creating **foo.bar** implicity creates the group **foo**. This group is either inherited or not, depending on this parameter.
numberValue	0.0	Sets the override value.

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.
targetPath	None	The object(s) to constrain the object in **basePath** to.
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.
baseAimAxis	0.0, 0.0, -1.0	The axis of the base object that is pointed at the target. Adjusting these values changes the side of the object that is aimed at the target.

Control (UI)	Default Value	Function
baseRotateAxis	0.0, 1.0, 0.0	The rotation axis of the object in **basePath**.
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.

BoundsAdjust

Allows you to adjust the bounding box of a geo.

Control (UI)	Default Value	Function
targetPath	none	Defines which geo's bounding is adjusted.
action	Strip Bounds	Sets the type of bounding adjustment to apply: • **Strip Bounds** - removes the bounding box from the geo entirely. • **Pad Bounds (%)** - allows you to pad the bounding box by a user-defined percentage. Selecting this option opens further **padAmount** attributes. • **Pad Bounds (local)** - allows you to pad the bounding box by an amount specified in units. Selecting this option opens further attributes:
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 on every cook of the Scene Graph (the result is visible in the attribute editor). • **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 local space units. This is a keyable attribute.
when	immediate	Sets when the specified adjustment is applied: • **immediate** - pad the bounds on every cook of the Scene Graph (the result is visible in the attribute editor). • **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. Its interface shows incoming values when found.

Control (UI)	Default Value	Function
cameraLocation	/root/world/cam/camera	Describes the location of the camera.
Geometry		
near	1	Sets the near clipping plane for the specified camera.
far	10000	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.

Notes: LightCreate and CameraCreate are identical, except for the type of ScenegraphLocations they create, and the population of the lightList vs. cameraList

Tips: 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.
projection	perspective	Toggles the projection type of the camera: • perspective projection • orthographic projection.
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		

Control (UI)	Default Value	Function
left	-1	This set of four number parameters controls the screen window placement on the imaging plane. They are, in order, left, right, bottom and top bounds of the screen window.
right	1	
bottom	-1	
top	1	
centerOfInterest	20	Offsets the center of interest of the camera.
orthographicWidth	30	Sets the orthographic projection width.
includeInCameraList	Yes	When enabled, the camera is visible in the Camera List.
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 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.
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.

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.
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.
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 mismatch 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.

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.
targetPath	none	Describes the camera location to which the **basePath** object is constrained.

Control (UI)	Default Value	Function
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).

Control (UI)	Default Value	Function
basePath	none	Describes the Scene Graph location of the object to constrain. This should be plane geometry.
targetPath	none	Describes the object location to which the **basePath** object is constrained.
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 and can be used to make the CEL expressions local to a branch of the Scene Graph (or kept global under /root). They are stored as attributes at the location defined by the location parameter in the CollectionCreate node. As they are simply attributes within the Scene Graph, Collections can be included within Katana Look Files.

Notes

When you have nothing attached in this node you will get an error in the scene graph warning that 3D input is required.
Collections are relative to the location they are store at. This is very useful for components loaded from disk that store collections of the child Scene Graph locations (e.g. shapes) within them. As you can see in the example screenshots below, a collection at /root/world/geo references /root/world/geo/primitive as /primitive. If you create the CEL statement using the "Paths" or "Collections" panels, the gui will assist you in getting the paths correct.

Control (UI)	Default Value	Function
location	none	Defines where you want the collection to appear in the Scene Graph. **Note:** If this is left empty, the location defaults to /root.
name	none	Sets the name of the collection.
CEL Statement		
CEL	none	Collection Expression Language. Used to specify Scene Graph locations on which an operation or assignment acts. For more information, see the CEL Reference document in the **Help** > **Quick Reference** menu.

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 LookFile. Only constraints on shapes listed in the constraintList are resolved at render time.

Control (UI)	Default Value	Function
locations	none	Sets the Scene Graph location of the constraint.
mode	add	Sets the constraint mode: • **add** – currently the only option, adding shapes to the constraintList.

ConstraintResolve

Produces a Scene Graph with all constraints resolved.

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 whether the coordinate system is defined globally or scoped to a specific location.
name	none	Specifies the unique name of the coordinate system to create.
path	none	Specifies the Scene Graph location whose global transform defines the coordinate system.

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.

Disparity

This node generates a vector distortion image from an unsorted list of **source** and **destination** points. Radial basis interpolation is used to generate intermediate values. The resulting vector image is suitable for use as an input to the ImageDistort node, and has the net result of warping pixels from the **src** locations to the **dst** locations.

Control (UI)	Default Value	Function
bounds		
resolution	Dependent on Project Settings	Sets 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.

Control (UI)	Default Value	Function
kernelType	Cubic	Sets the kernel type: • Linear • Quadratic • Cubic
locations - specifies the scattered points to->from		
name	p_0, p_1, p_2, and so on	Sets the name of the locations used to create the vector distortion image.
x, y, x, y	0, 40, 0, -40	Set the locations' source and destination coordinates on the xy axes.

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 37.

Control (UI)	Default Value	Function
basePath	none	Sets the location of the camera or light to dolly.
targetPath	none	Sets the location of the object(s) to fit within the field of view.
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 **D** 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.
faceSetName	faceset	Defines the name of the new set.
selection	none	Records the list of faces as an array. You can adopt the faces present in the Viewer by using the dropdown menu on the right.

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 pulldown 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 35.

Control (UI)	Default Value	Function
basePath	None	Sets the object to constrain.
targetPath	None	Sets the object to constrain the **basePath** the object. You can add more locations using the **Add Locations** dropdown.
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: • default • prman • arnold-c
actions	n/a	Click to display a dropdown containing actions available in the Gaffer: • **Add Light** - adds a light and material. • **Add Rig** - adds a rig. • **Add Aim Target** - adds an aim constraint. • **Add Master Material** - adds a master material. • **Add Sky Dome** - adds a sky dome and material. • **Import Rig** - allows you to import an existing **.gprig** file. **Note:** All the add options in the actions menu are addable on an individual basis in the Node Graph. For example, **Add Light** is the same as adding a LightCreate node manually.
[Gaffer object list]	gaffer	Displays a list of all objects controlled by this Gaffer node. The object list 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 the render. **Note:** Checking solo only solos the light for interactive renders, not batch render.s • **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. **Note:** The **Object**, **Material**, and **Linking** tabs under the object list are populated depending on the object selected.

Control (UI)	Default Value	Function
object list: light		
Object Tab		
geometry		
radius	1	Sets the radius of the light.
fov	70	Sets the field of view angle.
centerOfInterest	20	Sets the center of interest.
near	0.1	Sets the distance of the near clipping plane.
far	100000	Sets the distance of the far clipping plane.
screenWindow	-1.0, 1.0, -1.0, 1.0	Controls the screen window placement on the imaging plane. They are the left, right, bottom, and top bounds of the screen window.
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		

Control (UI)	Default Value	Function
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.
enable aim constraint	disabled	When enabled, specify **aim constraint options**.
enable aim constraint: enabled		
targetPath	none	Specifies the object or objects to constrain to. If you want to aim a light to point at a target, this would be the target. If you set multiple targets, then the constraint aims at the average center of the objects.
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.
baseAimAxis	0.0, 0.0, -1.0	The axis of the base object that is pointed at the target. Adjusting these values changes the side of the object that is aimed at the target.
baseUpAxis	0.0, 1.0, 0.0	The axis of the base object that is pointed upwards relative to the target. Adjusting these values changes the rotation of the base object, while keeping the aim constant.
targetUpAxis	0.0, 1.0, 0.0	The axis of the target object that is pointed upwards relative to the base object. Adjusting these values changes the rotation of the target object, while maintaining the aim constant.
display		
text	none	Places a label in the Viewer containing the string entered in the text field.
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.
Material Tab		
useLookFileMaterial	disabled	When enabled, the material from an associated LookFile is used.
Add Shader	n/a	Click to add a shader from the dropdown list. The **Material Tab** is populated with controls appropriate to the shader selected.
Linking Tab		
light linking		

Control (UI)	Default Value	Function
defaultLink	on	When set to **on**, the light is active for all objects except those listed under exceptions. When set to **off**, the light is inactive for all objects except those listed under exceptions.
enable exceptions	disabled	When enabled, enter Scene Graph object locations to create an exception list.
shadow linking		
defaultShadowLink	on	When set to **on**, shadows are active for all objects except those listed under exceptions. When set to **off**, shadows are inactive for all objects except those listed under exceptions.
enable exceptions	disabled	When enabled, enter Scene Graph object locations to create an exception list.
object list: rig		
Object Tab		
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 rig translation on the xyz axes.
rotate	0.0, 0.0, 0.0	Controls rig rotation on the xyz axes.

Control (UI)	Default Value	Function
scale	1.0, 1.0, 1.0	Controls rig 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.
enable point constraint	disabled	When enabled, specify **aim constraint options**.
enable point constraint: enabled		
targetPath	none	Specifies the object or objects to constrain to. If you want to aim a rig to point at a target, this would be the target. If you set multiple targets, then the constraint aims at the average center of the objects.
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.
baseOrigin	Object	Sets how the center of the base 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.
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.
orient constraint options	disabled	When enabled, specify **aim constraint options**.
orient constraint options: enabled		
targetPath	none	Specifies the object or objects to constrain to. If you want to aim a rig to point at a target, this would be the target. If you set multiple targets, then the constraint aims at the average center of the objects.
targetOrientation	Object	Sets how the orientation of the target object is calculated: • Object - uses the local origin of the object as the target.
allowMissingTargets	No	When set to **Yes**, silently ignore the constraint if its target is not in the Scene Graph. When set to **No**, produce an error on constraint resolution if the target is missing.
xAxis	Yes	When set to **Yes**, orientation is constrained on the x axis.

Control (UI)	Default Value	Function
yAxis	Yes	When set to **Yes**, orientation is constrained on the y axis.
zAxis	Yes	When set to **Yes**, orientation is constrained on the z axis.
object list: aim		
Object Tab		
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 aim constraint translation on the xyz axes.
rotate	0.0, 0.0, 0.0	Controls aim constraint rotation on the xyz axes.
scale	1.0, 1.0, 1.0	Controls aim constraint 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.
object list: material		
Material Tab		
useLookFileMaterial	disabled	When enabled, the material from an associated LookFile is used.
Add shader	n/a	Click to add a shader from the dropdown list. The **Material Tab** is populated with controls appropriate to the shader selected.
object list: sky		

Control (UI)	Default Value	Function
Object Tab		
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 sky dome translation on the xyz axes.
rotate	0.0, 0.0, 0.0	Controls sky dome rotation on the xyz axes.
scale	1.0, 1.0, 1.0	Controls sky dome 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.
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.
latlongOrientation	Exterior	Determines where the environment map is projected from onto the sky dome: • **Interior** - produces more realistic results. • **Exterior** - the default setting.
Material Tab		
useLookFileMaterial	disabled	When enabled, the material from an associated LookFile is used.

Control (UI)	Default Value	Function
Add Shader	n/a	Click to add a shader from the dropdown list. The **Material Tab** is populated with controls appropriate to the shader selected.
Linking Tab		
light linking		
defaultLink	on	When set to **on**, the sky dome light is active for all objects except those listed under exceptions. When set to **off**, the sky dome light is inactive for all objects except those listed under exceptions.
enable exceptions	disabled	When enabled, enter Scene Graph object locations to create an exception list.
shadow linking		
defaultShadowLink	on	When set to **on**, shadows are active for all objects except those listed under exceptions. When set to **off**, shadows are inactive for all objects except those listed under exceptions.
enable exceptions	disabled	When enabled, enter Scene Graph object locations to create an exception list.

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 would be that of 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 drag

that node into the GroupMerge's stacklist using the **Shift**+**MMB**+drag technique. At that point, the GroupMerge is permanently typed as a group of the type of node that was dragged in.

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.

Notes: When the GroupStack node is first created, its type is not defined. You can create a node and drag that node into the GroupStack's stacklist using the **Shift**+**MMB**+drag technique. At that point, the GroupStack is permanently typed as a group of the type of node that was dragged in.

Tips:

- Clicking the **+** icon at the top of the stacklist of a type-seeded GroupStack creates a new node of the same type at the bottom of the stacklist.
- Pressing the delete key removes the currently selected node(s) from the stacklist.
- Pressing the **i** key toggles the ignore state of the selected nodes(s) in the stacklist.
- Select and middle-mouse drag allows you to rearrange the order of the nodes in the stacklist, effectively changing the order in which they are evaluated when the Scene Graph bakes.
- **Shift** or **Ctrl**+click in the stacklist to select and edit multiple nodes.
- Right-clicking a node in the stacklist opens a convenient menu giving you access to several operations to perform on that node.

  These include: ignore, view at, delete, duplicate, cut, copy, paste, and tear-off parameters.

4 Nodes I–K

IBLCoordinateCreate

This node defines Image Based Lighting (IBL) coordinates within the Scene Graph, combining low and high contrast to simulate directional light.

Control (UI)	Default Value	Function
name	/root/world/lgt/gaffer/ibl-rig	Describes the Scene Graph location of the target rig.
geometry		
latlongOrientation	Interior	Determines where the environment map is projected from: • **Interior** • **Exterior**
pts		
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 translation on the xyz axes.
rotate	0.0, 0.0, 0.0	Controls rotation on the xyz axes.

Control (UI)	Default Value	Function
scale	1.0, 1.0, 1.0	Controls 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.
claimExclusivity	Yes	When set to **No**, the camera is not controlled by another node in the Scene Graph and is effectively locked.

IBLQuadCreate

Control (UI)	Default Value	Function
IBLCoordinateCreate Tab		
geometry		
stSpace	Lower Latlong	Specifies what coordinate system to use for st texture coordinate generation: • **Lower** - the geometry looks correct with the lower texture applied. • **Upper** - the geometry looks correct with the upper texture applied. • **Basic** - do not do any IBL coordinate generate. Just make a normal quad tex coords.
flipT	Enabled	When enabled, flip the t texture coordinate on the generated geometry. Typically, for Arnold-C shaders it should be enabled.
shape	Quad	Specifies the shape to create: • **Quad** - an arbitrary quadrilateral (4 points on the same plane). • **Rectangle** - contsrain the corner positions to create a rectangular shape (90 degree corners).
extractLocal	Enabled	When enabled, extract a local coordinate transform, set it as the xform mtx, and create a local coordinate system. When disabled, create the quad with points directly in world space. **Note:** Both options produce the same image, but generate different attribute hierarchies.
creation	Immediate	Specify when geometry is created: • **Immediate** - immediately create the geometry. • **Deferred** - create geometry during Scene GraphLocationModifier resolve time. The advantage of deferred creation is that all the settings (such as subdivision) can be further controlled live at render time. This option is rarely needed.
compatibilityXFormOrder	Disabled	When enabled, the legacy xform order is enforced. In normal usage, it is recommended that this option remain disabled.
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
interface: SRT Values		
translate	0.0, 0.0, 0.0	Controls translation on the xyz axes.
rotate	0.0, 0.0, 0.0	Controls rotation on the xyz axes.
scale	1.0, 1.0, 1.0	Controls 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.

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.

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.
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. • **grid** – point movement snaps to the grid in the curves display.

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.

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	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. 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 **ImageSubtract**. • **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.

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 85 and ImageHypot on page 87.

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	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.

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 **ImageSubtract**. • **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.

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.
RGB		
red	0	Sets the red value of the background.
green	0	Sets the green value of the background.

Control (UI)	Default Value	Function
blue	0	Sets the blue value of the background.
alpha	0	Sets the alpha value of the background.
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.
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.
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.

Control (UI)	Default Value	Function
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.
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.
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.
mix	1	Dissolves between the original image at 0 and the full blur 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 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.

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.
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.

Control (UI)	Default Value	Function
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.
greenSource	i0	Select the input from which to take the green channel.

Control (UI)	Default Value	Function
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
constantRGB		
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**.
constantHSL		
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**.
constantHSV		
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
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.
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.
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.
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.
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.

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).
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.

Control (UI)	Default Value	Function
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.

A good example of this is a roto shape. Only the shape itself needs to be contained in the data window. The area outside the shape extending to the display window (frame) is understood to be a constant color (0,0,0,0 by default) and need not be explicitly processed and stored.

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.

Control (UI)	Default Value	Function
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.
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	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 **ImageSubtract**. • **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 **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**
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.

Control (UI)	Default Value	Function
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.

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	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 **ImageSubtract**.

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 **ImageSubtract**. • **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		
glntensity	1	Adjusts gain intensity.
units: F-Stops		
fsColor		
red	0	Adjusts f-stop exposure in the red channel.
green	0	Adjusts f-stop exposure in the green channel.
blue	0	Adjusts f-stop exposure in the blue channel.
fsAlpha	0	Adjusts f-stop exposure in the alpha channel.
units: Gain		
gColor		
red	1	Adjusts gain exposure in the red channel.
green	1	Adjusts gain exposure in the green channel.
blue	1	Adjusts gain exposure in the blue channel.

Control (UI)	Default Value	Function
gAlpha	1	Adjusts gain exposure in the alpha channel.
units	F-Stops	Select the units in which the exposure is altered: • **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.

Control (UI)	Default Value	Function
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.
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 135.

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	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 **ImageSubtract**. • **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.

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.

Control (UI)	Default Value	Function
green	1	Adjusts gain in the green channel.
blue	1	Adjusts gain in the blue channel.
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.

Control (UI)	Default Value	Function
b	1	Adjusts gamma in the blue channel.
a	1	Adjusts gamma in the alpha channel.
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 54 and ImageHypot on page 87.

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.

Control (UI)	Default Value	Function
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 **ImageSubtract**. • **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.

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 116 and ImageScreen on page 127.

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	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.

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 **ImageSubtract**. • **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.

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.

Control (UI)	Default Value	Function
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 **ImageSubtract**.

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.

Control (UI)	Default Value	Function
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.
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.

Control (UI)	Default Value	Function
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.
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.
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.
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.

Control (UI)	Default Value	Function
		• **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 **ImageSubtract**. • **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.

Control (UI)	Default Value	Function
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.

Control (UI)	Default Value	Function
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.

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	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. 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 **ImageSubtract**. • **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.

Control (UI)	Default Value	Function
ignoreZero	disabled	When enabled, zero values are ignored. This is useful with Z depth passes where infinity or no-coverage is represented by a zero value. In this case, the zero is a false-positive for the minimum and should be ignored.
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.
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	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 **ImageSubtract**. • **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.

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 80.

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. 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.

Control (UI)	Default Value	Function
		• **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 **ImageSubtract**. • **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.

Control (UI)	Default Value	Function
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.
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).
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.

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 **ImageSubtract**. • **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.

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.

Control (UI)	Default Value	Function
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.

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	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 **ImageSubtract**.

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 135.

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 **ImageSubtract**. • **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.

Control (UI)	Default Value	Function
		• **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 **ImageSubtract**. • **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.

Control (UI)	Default Value	Function
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.

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 132 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.

Control (UI)	Default Value	Function
fringe	disabled	When enabled, the mask is modified so that by default the offset effect is limited to the fringe (semi-transparent areas).
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 138.

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.

Control (UI)	Default Value	Function
invert	disabled	Inverts the use of the mask channel so that the premultiplication is limited to the non-white areas of the mask.
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].
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
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.
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.
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.
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.

Control (UI)	Default Value	Function
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 87.

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.

Control (UI)	Default Value	Function
fringe	disabled	When enabled, the mask is modified so that by default the merge is limited to the fringe (semi-transparent areas).
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 **ImageSubtract**. • **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.
font	Arial	Lets you select a font for the text when **fontSource** is set to **Builtin**: either **Arial** or **Courier**.
fontFile	none	Lets you select a font for the text when **fontSource** is set to **File**.
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.

Control (UI)	Default Value	Function
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.
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.

Control (UI)	Default Value	Function
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.
mix	1	Dissolves between the original image at 0 and the full transform effect at 1.

Control (UI)	Default Value	Function
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 114.

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.

Control (UI)	Default Value	Function
		• **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 **ImageSubtract**. • **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.

Control (UI)	Default Value	Function
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.

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 120.

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.
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.
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.
HSL		
hue	0	Sets the hue of the pixels.

Control (UI)	Default Value	Function
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.
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.

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 products and 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 a product: • **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.
image – If mode is set to dependency, these controls are hidden.		
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).
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.
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**
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.
exrType: Tiled		
exrTileWidth	256	Sets the tile width to use when writing to tiled exr files.
exrTileHeight	256	Sets the tile height to use when writing to tiled exr files.
exrTileWorldAlign	disabled	When enabled (in 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.
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.

Control (UI)	Default Value	Function
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**
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**
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.
Inputs > 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.

Control (UI)	Default Value	Function
Inputs > 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.
displayWindow: manual > displayWindowResolution		
resolution	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	
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.

Control (UI)	Default Value	Function
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	
postScripts		
Add	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.
setActiveFrameRange: enabled		
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.
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.

Control (UI)	Default Value	Function
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.
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**).

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 filesystem).

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: • Appending or replacing with Scene Graph Selection grabs path locations directly from the Scene Graph selection • Appending or replacing with Node Graph Node Locations adds expressioned node references which will evaluate to a Scene Graph location being generated by the selected node(s). This expression will update as the referenced node changes. **Note:** An empty or invalid location value generates a location of type **error** under /root.
text (html editor)	view editor and preview	Using the popup menu, set the mode for the html editor. • **view editor and preview** - the top section of editor is html source and the bottom section of the editor is rendered preview. • **view only editor** - shows onlythe 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 will remove everything else in the scene. This is useful because you can scope this operation to only apply to a specific branch of the scene. For example, you could isolate a character or two out of all the geometry in your scene.

Isolate cannot take a collection. The workaround is to select the collection contents in the Scene Graph (using a Collect and Select command), then use the Replace with Scene Graph Selection menu item on the isolateLocations.

If you want to remove some specific locations, use a Prune node instead.

Control (UI)	Default Value	Function
isolateLocations	none	This is a list of locations to keep while every other location is removed by the Isolate.

Control (UI)	Default Value	Function
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 will be modified by the Isolate. 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		
manipulators	none	Adds entries to the manipulators list.
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.

Control (UI)	Default Value	Function
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.

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.
- 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 visiblity** **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 CEL statements that select which object to operate on.
lightMode	CEL	Controls how you specify which lights to operate on: • **CEL** • **Paths**
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** pulldown menu. The Scene Graph and Node Graph can be replaced or appended from this location. Lights can be dragged to the locations box.
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 will be picked up automatically. If the source file cannot be read or no longer exists, a copy stored in the scene file will be 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 154.

Control (UI)	Default Value	Function
stack contents	none	Contains the list of nodes to manage.
source	none	Sets the path to load in the source (file) as the contents of the LiveGroup.

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.
attrs	None	Drag string or number attributes here to have them added to the Scene Graph location(s) created by this node.

LodSelect

This node removes all but one LOD (level-of-detail) beneath selected level-of-detail groups.

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**.
mode	by index	Sets the method used to specify levels of detail: • **by index** • **by tag** • **by weight**
selectionIndex	0	Sets the index of which LOD child to keep.

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 auxillary 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 157

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 product 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. LookFileLightAndConstraintActivator nodes activate LookFile lights and constraints by updating the respective lists.

Control (UI)	Default Value	Function
Action	none	Searches the Scene Graph for lights and constraints brought in by LookFiles 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** - ceates 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.

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 leftmost 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 leftmost 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.
preserveWorldSpaceXform	No	When enabled, all inherited xform attributes (preceded by an origin statement) are applied at each location whose source input differs from that of its parent. This is only necessary in exceptional situations where there are conflicting transformations on overlapping locations of the merge inputs. Basically, this 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.
preserveInheritedAttributes	none	Displays a list of attribute names for which inheritance should be preserved when choosing between inputs of the Merge. Whenever a child location's source input differs from that of its parent, these attributes 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 versions of a component to form a single output model; the first input will provide most of the attributes, but a second input might provide correctly deformed geometry or other attributes that should be used in preference to the first input. Again, this is typically rolled into a show macro, so it's unlikely you'll need to work with this setting directly.
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.

Control (UI)	Default Value	Function
in	0	Sets the index of the input port to pass through. The left-most input port is index 0, and port indices increase from left-to-right. This control accepts expressions. **Note:** This value is **not** saved to the .katana scene file.

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.

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.

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.
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.

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, .vf
cccid	none	When **src** points to a .ccc file, specify the id to lookup. OpenColorIO::Contexts (envvars) are obeyed.

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 166 and PointConstraint on page 166.

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 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.

ParentChildConstraint

Constrains the translate, rotate, and scale values of one object (the parent) to another (the child). See also OrientConstraint on page 165 and PointConstraint on page 166.

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 165 and ParentChildConstraint on page 166.

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.

Control (UI)	Default Value	Function
camera		
depthOfField		
fStopInfinite	Yes	When set to **Yes**, depth of field is disabled.
fStop	22	Sets the camera f-stop vlaue (focal length/lens diameter).
focalLen	44	Sets the lens focal length.
focalDist	0	Sets the distance in camera units at which objects are in focus.
imagerBackground	0.0000, 0.0000, 0.0000	Sets the background color
options		
pixelSamples	2.0, 2.0	Sest the subpixel sampling rate (X x Y samples)

Control (UI)	Default Value	Function
pixelFilter		
filterFunc	gaussian	Sets the filter used for producing final pixels from pixel samples: • **box** • **triangle** • **catmull-rom** • **mitchell** • **blackman-harris** • **sinc** • **gaussian**
width	2	Sets the filter width, in pixels.
height	2	Sets the filter height, in pixels.
hider		
hider	hidden	Sets the hidden surface elimination method: • **hidden** - the default suitable for most situations. • **photon** - used to generate photon maps. • **raytrace**
hidden		
jitter	1	When set to **1**, stochastic sampling is enabled which is useful for motion blur or depth-of-field.
mpcache	Yes	When set to **Yes**, micropolygons are cached to disk.
samplemotion	Yes	This control should be set to **Yes** for normal motion blur and **No** for post-process (2d) motion-blur.
mpmemory	6144	Sets the memory (KB) available for micropolygon caching.
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.
sigma	0	Toggles alternate hiding designed for very small, thin, or transparent geometry, for example hair.
sigmablur	1	Sest the quality of antialiasing for sigma hiding (0 to 1). Lower values create a sharper look that may alias more readily and higher values closer to 1.0 result in a softer, more filtered look.
shutteropening	0.0, 1.0	Specifies the timing of the opening and closing of the camera shutter. The default values represent instantaneous open/close.

Control (UI)	Default Value	Function
aperture	0.0, 0.0, 0.0, 0.0	Controls lens aperture shape for special depth-of-field effects. The four values represent: • **nsides** - the number of sides of the aperture. When set to less than 3, use a circular aperture with legacy stochastic sampling pattern. • **angle** - the polygon orientation (0 degrees puts a vertex on the x axis). • **roundness** - set to 0 for a regular polygon with straight sides, 1 for a circle, 0-1 for curved edges bowed out, 0 to -1 for edges bowed in. • **density** - set to 0 for constant density. When negative (resp. positive), the aperture is brighter near the center (resp. near the rim.). Range [-1,1].
maxvpdepth	-1	Controls the maximum number of visible points considered for compositing or deep shadow map creation. The default (-1) is disabled, meaning no limit.
occlusionbound	0	Specifies how far a point can be displaced in screen space before it must be shaded prior to hiding. The default (zero) causes all displaced objects to be shaded before hiding. Set higher for faster renders at the risk of possible errors on displaced objects.
photon		
emit	1	Sets the total number of photons emitted from all light sources.
shadingInterpolation	constant	Controls how values are interpolated across a micropolygon. If constant, the color and opacity of all the pixels inside the micropolygon are the same. If smooth, the colors and opacities are interpolated from the calculated values. Note that this must be set to smooth for use with C.entered Derivatives and Normals. Note also that shading interpolation is really an attribute, not an option, so it can be set on a per-object basis.
shadingRate	1	Sets the frequency of shading, in pixel area per shade. For example, 1.0 means at least once per pixel; .25 means approximately 4 times per pixel. 4.0 would mean micropolygons could be 2x2 pixels or so in size.
ribInclude	none	Specifies a RIB file (must end in .rib) which is read into the output RIB stream up front with the Options (before WorldBegin).
outputChannels		
outputChannels	none	Controls output channels.
declarations		
groupings		

Control (UI)	Default Value	Function
membership	none	In recent versions of PRman, nothing within a procedural or delayed rib archive may be included within a **grouping.membership** group unless the enclosing scope is itself defined to be within that group. Because Katana makes heavy use of procedurals throughout its hierarchy, you must pre-declare the names of any groups you want to use within your scene. These are specified here as either a space-or-comma-delimited list of names.
commandArguments		
numThreadModeProcs	1	Sets the number of threads to use while rendering.
prmanVersion	none	When specified, this version string is substituted for the PRman version in environment variables $RMANTREE, $PATH, $RMAN_SHADERPATH, and $RMAN_DSOPATH.
prmanPluginVersion	none	Sest the version string to be substituted for plug-in paths in the **prman-Version** environment variables.
riFilters	none	Adds Ri Filter args to the prman command line. Each string in the array represents one filter to be added. The first token in each string is the name of the filter dso, and any additional space-delimited tokens are added as filter args. Examples: **riFilters string** = > **prman command line args** filter = > -rif filter filter arg1 arg2 = > -rif filter -rifargs arg1 arg2 -rifend

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.

Control (UI)	Default Value	Function
CEL	none	Collection Expression Language used to specify Scene Graph locations on which an operation or assignment acts.
attributes		
cull		
backfacing	1	Attribute "cull" "backfacing" [1] • **backfacing** - when turned off, surfaces are shaded even if they have one side and their back side is facing the camera. Attribute "cull" "hidden" [1] • **hidden** - when turned off, surfaces are shaded even if they are behind other surfaces. These attributes are convenient to force shading to happen, for example for baking ambient occlusion or indirect illumination. For regular rendering, these attributes should be left at their default value (1) so that surfaces that do not contribute to an image are culled before wasting time shading them.
hidden	1	
dice		
binary	No	Attribute "dice" "int binary" [0] This is a flag that indicates whether the lowest level patches must be diced into a grid of micropolygons with power-of-two dimensions. This attribute is typically used to prevent patch cracking on high-curvature patches.
hair	No	Attribute "dice" "int hair" [0] This Attribute specifically used for RiCurves geometry makes thin hair-like curves much cheaper to shade, and more memory efficient when dealing with long hair which spans multiple buckets. When it is enabled by setting to 1 (it is disabled by default), Du() of any quantity is 0, with the trivial exception of Du(u) = 1. In other words, derivatives across the width of the curve are ill defined. Because of this, this attribute should only be used for curves which are truly hair-like (very thin in screen space).
rasterorient	Yes	Attribute "dice" "int rasterorient" [1] PRMan supports (as of version 10) a new mode of dicing on NURBs that computes micropolygon sizes using an unoriented raster space metric rather than the standard screen aligned raster metric. This is useful in situations where it is important that the dicing rate on an object doesn't change due to camera orientation. For example, it can help in situations where objects which are viewed edge on result in large micropolygons (but small in screen space), which are then displaced leading to micropolygons which are large in screen space. The ability to turn off the oriented dicing metric has been implemented for NURB, subdivision surface, and bilinear and bicubic primitives.

Control (UI)	Default Value	Function
displacementbound		
sphere	0	Attribute "displacementbound" "float sphere" [0.0] Controls the sizes of calculated bounding boxes on primitives in order to account for the effects of displacement mapping. The size is specified by identifying a single floating-point value which is the radius of a sphere which is guaranteed to contain the maximum possible displacement, and the name of the coordinate system in which this sphere resides. This value should be as tight as possible. It is extremely inefficient, both in terms of memory usage and calculation time, to specify a bounding sphere which is larger than the actual displacement. Therefore, this sphere should be as small as possible without permitting points on the object to displace farther than the sphere's radius.
coordinatesystem	object	Attribute "displacementbound" "string coordinatesystem" ["object"] The coordinate system identified can be: • any of the standard coordinate systems ("screen", "camera", "world", "object") • "surface" or "displacement", which specifies the shader space of the appropriate shader on this object • "shader", which is identical to "displacement" if there is a displacement shader on this object, or to "surface" if there is not; • "current", which is the Shading Language current space (identical to "camera" in PhotoRealistic RenderMan); • "null", which specifies the coordinate system at the time of the RiAttribute call; or, • any user-defined coordinate system created with RiCoordinateSystem.
irradiance		
maxerror	0.5	Attribute "irradiance "float maxerror" [0.5] An error metric to control quality/speed tradeoff for irradiance calculations. If set to zero, the irradiance will be computed at every shading point; no attempt at interpolation will be done. The larger 'maxerror' is, the more time can be saved by interpolating nearby irradiance values.
maxpixeldist	30	Attribute "irradiance "float maxpixeldist" [30.0] An alternative backup metric to control quality/speed tradeoff for irradiance. If there are visible discontinuities in occlusion or irradiance, typically on curved objects far from other objects, it is often helpful to reduce 'maxpixeldist' to smaller values -- for example 10 or 15. This ensures that interpolation is only done over shorter distances.
photon		
estimator	50	Attribute "photon" "int estimator" [50] Controls the number of photons to consider when estimating caustic effects.

Control (UI)	Default Value	Function
causticmap	none	Attribute "photon" "string causticmap" [""] Specifies the caustic photonmap to store photons in during photon tracing; this attribute allows you to deposit photons into a per-primitive map. Also specifies which caustic photon map to use when a shader calls the caustic shade-op.
globalmap	none	Attribute "photon" "string globalmap" [""] Specifies the global photonmap to store photons in during photon tracing; this attribute allows you to deposit photons into a per-primitive map.
shadingmodel	matte	Attribute "photon" "string shadingmodel" ["matte"] Specifies a simplified shading model to employ when tracing photons. If empty, the shader will be analyzed to determine photon scattering.
sides		
backfacetolerance	0	Attribute "sides" "float backfacetolerance" [0.0] The backface culling tolerance angle is a floating-point number, measured in degrees, which specifies the angle that the primitive must exceed, beyond the "silhouette normal", before it may be culled prior to shading. The default value is 0 degrees. For example an angle of 20.0 will cause the renderer to not cull backfacing objects until their surface normals point more than 110 degrees away from the viewing vector. Note that this does not affect the fact that by the end of the rendering pipeline, backface culling is exact and occurs at 90 degrees.
doubleshaded	No	Attribute "sides" "int doubleshaded" [0] When this Attribute is enabled, two-sided primitives will be shaded twice, once on each side. The surface normals will face in the standard direction on one pass, and in the inverted, or reversed-orientation, direction on the other pass (roughly speaking, towards the viewer once and away from the viewer once). This permits, for example, two-sided objects which are laminar sheets or ribbons to have different colors on each side, or to be displaced in both directions in order to give it a shader-generated thickness.
trace		
maxdiffusedepth	1	Attribute "trace" "int maxdiffusedepth" [1] This attribute limits the number of diffuse bounces for indirect illuminance relative to the associated primitive. To resolve the interaction between per-primitive values, we pass the current max down the ray tree and apply MIN(ray.depthMax, primitive.depthMax + ray.depth). The maxdiffusedepth option is used to limit the number of diffuse and specular bounces of photons in the photon tracing pass., while maxspeculardepth controls the number of reflection and refraction bounces.

Control (UI)	Default Value	Function
maxspeculardepth	2	Attribute "trace" "int maxspeculardepth" [2] This attribute limits the number of specular bounces for indirect illuminance relative to the associated primitive. To resolve the interaction between per-primitive values, we pass the current max down the ray tree and apply MIN(ray.depthMax, primitive.depthMax + ray.depth). The maxspeculardepth controls the number of reflection and refraction bounces.
displacements	No	Attribute "trace" "int displacements" [0] Controls whether true displacements appear in ray traced results. When false, the displacement will be disregarded for the purposes of ray-primitive intersection tests, but shading will take the displacements into account effectively resulting in a bump mapped appearance.
bias	0.01	Attribute "trace" "float bias" [0.01] This bias value affects transmission/shadow rays as well as trace/environment rays. It is an offset applied to the ray origin, moving it slightly away from the surface launch point in the ray direction. This offset can prevent blotchy artifacts resulting from the ray immediately finding an intersection with the surface it just left.
samplemotion	0	Attribute "trace" "int samplemotion" [0] Controls whether motion blurred objects appear in ray traced results. When 0, the motion blur of other objects hit by rays launched from the object with this attribute will be ignored. When non-zero, motion blur will be taken into account by rays launched from an object with this attribute.
user		
user	none	Attribute "user" "uniform string myattribute" ["foo"] PRMan (subsequent to version 10.0) supports the use of arbitrarily defined token/value pairs for the user attribute. These token/value pairs may be arbitrarily defined and set, and then queried with the attribute shadeop or via the RxAttribute mechanism. Like other attributes, all token/value pairs associated with this attribute will be pushed and popped on the graphics stack with each RiAttributeBegin and RiAttributeEnd call.
visibility		
camera	Yes	Attribute "visibility" "int camera" [1] Controls the visibility of subsequent primitives to the camera. By default all primitives are visible.
photon	No	Attribute "visibility" "int photon" [0] Controls the visibility of the current primitive to photons. Since photon map generation is a separate rendering pass, this attribute is usually superfluous. We support it to facilitate the use of the same RIB archive in both photon generation and final rendering passes.

Control (UI)	Default Value	Function
transmission	No	Attribute "visibility" "string transmission" ["opaque"] Controls the visibility of primitives to transmission (shadow) rays. Since transmission rays can accumulate contributions of semi-transparent objects, for colored shadows, this attribute provides several levels of control over how opacity is handled: • **transparent** primitives are ignored by transmission rays. • **opaque** hit primitives are simply considered to be opaque, no additional shader execution is required during transmission ray intersection. • **Os** the current Opacity attribute is used as the object's opacity, without reference to the surface shader. • **shader** the full surface shader on intersected objects is executed to determine (colored) opacity, Oi. In some circumstances, shader analysis may determine that Oi does not depend on complete shader execution.
trace	No	Attribute "visibility" "int trace" [0] Controls the visibility of the current primitive to trace rays. These are reflection and refraction rays cast by gather, trace, and environment. This is a prman11.5 attribute that is still honored but has been replaced by other attributes.
grouping		
membership	none	Attribute "grouping" "string membership" [""] Controls the group membership of subsequent primitives. A single primitive can be a member of several groups. Membership is used to precisely control trace relationships between objects. Ray-tracing shaders on one object can limit their ray intersections to members of specific groups by using the optional "subset" parameter of the tracing operators. The name_spec supports relative and absolute specification: • "name" • "list,of,names" or "list of names" • "+additional,names" • "-without,names"

Control (UI)	Default Value	Function
shade		
strategy	grids	Attribute "shade" "string strategy" ["grids"] Using the long-standing, default, "grids" strategy, all shaders attached to an object are executed as soon as it becomes shadeable. Starting with PRMan 12.0, when Attribute "shade" "strategy" ["vpvolumes"] is attached to a primitive, its associated Atmosphere and Interior shaders are instead executed only after all of the final depth samples ("visible points") for that object, and others, have been determined. This "post-visibility computation" volume shading of camera samples allows Interior shaders to execute on these depth samples directly, without requiring ray-tracing or analytic techniques to determine the "back side" of an object. However, ray-traced Interior volumes are still valuable for combining volumetric and refractive effects. In some cases, the "vpvolumes" strategy can also be used to compensate for motion blur and depth of field "dragging" of volume samples seen by the camera. Using the "vpvolumes" strategy usually increases the number of points at which volumes are executed (it is a function of PixelSamples, rather than ShadingRate). However, there are some cases where it may actually reduce the number of volume shading points, such as when a very complex object is very small on screen. It will also not run Atmosphere shaders on parts of objects which are completely hidden. Interior shaders allow effects such as the smoke inside a closed object to be described by a volume shader which is independent of the object's surface shader. The surface must be partially transparent for the interior to be exposed.
volumeintersectionstrategy	exclusive	Attribute "shade" "string volumeintersectionstrategy" ["exclusive"] This attribute allows regions with overlapping Interior shaders to combine their effects in either "additive" or "exclusive". There is also a "float volumeintersectionpriority" which controls the combining order, and also which of "exclusive" or "additive" apply when both are present in the same region.
volumeintersectionpriority	0	Attribute "shade" "float volumeintersectionpriority" [0.0] Controls the combining order of volumeintersectionstrategy.

Control (UI)	Default Value	Function
diffusehitmode	primitive	Attribute "shade" "string diffusehitmode" ["primitive"] Possible values: "primitive" or "shader". The hitmode attributes provide limited control over whether or not shaders will actually execute at the attached object's shading points ("hits"). The renderer generates shading points for objects seen directly by the camera, and also at the points where rays from ray-tracing functions intersect objects. These attributes are typically used to control shading expense at ray hit points. Full shading is typically desirable in very coherent (specular) secondary reflections; if the same objects are also sampled by very diffuse rays, then expensive full shading may not be necessary in those contexts. Currently, these attributes only affect rays shot due to these shading functions: occlusion, indirectdiffuse, transmission, and shadow. These functions also accept an explicit "hitmode" parameter which can override the hit object's attribute setting. Rays originating from trace and environment currently always cause shading to occur. The behavior of gather rays depends on modifiers supplied in the shader for each gathered variable. When the mode is "primitive", then the Color and Opacity attributes associated with the hit object will be used as the the color and opacity of the hit point, without further shading computation. This optimization can reduce render time since potentially expensive shaders are not being run on the hit object. If the mode is "shader", then the shader attached to the hit primitive will be run to compute the color and opacity.
specularhitmode	shader	Attribute "shade" "string specularhitmode" ["shader"] See diffusehitmode
transmissionhitmode	shader	Attribute "shade" "string transmissionhitmode" ["shader"] See diffusehitmode
camerahitmode	shader	Attribute "shade" "string camerahitmode" ["shader"] The "primitive" mode for "camerahitmode" is unique in that it does not actually disable shader execution on objects visible to the camera. Instead, shaders are always run on camera-visible objects, but the "primitive" setting causes them to always be considered opaque. So both the Opacity attribute attached to these objects, and their shader's output Oi, are ignored, and opacity is forced to be opaque, color(1,1,1). This simplification can reduce render time by causing shading to be skipped on all objects that fall behind the marked objects, since they are guaranteed to be completely hidden.

Control (UI)	Default Value	Function
relativeshadingrate	1	Attribute "shade" "relativeshadingrate" [1.0] The renderman default for this parameter is actually none in that the absence of the attribute means something different than having the attribute set with any value. The katana default is 1.0 but, like other PrmanObjectSetting paramters, is not emitted to the rib unless it is enabled. Sets the value of the relative shading rate equal to the relativeshadingrate value times any other relativeshadingrate values in the Attribute stack. The final ShadingRate used on an object will be the accumulated relativeshadingrate multiplied by the actual shadingrate. This attribute was introduced in prman13.
resetrelativeshadingrate	0	Attribute "shade" "resetrelativeshadingrate" [0.0] The renderman default for this parameter is actually none in that the absence of the attribute means something different than having the attribute set with any value. The katana default is 1.0 but, like other PrmanObjectSetting paramters, is not emitted to the rib unless it is enabled. Attribute "shade" "resetrelativeshadingrate" resets the relativeshadingrate to the given value. The final ShadingRate used on an object will be the accumulated relativeshadingrate multiplied by the actual shadingrate. This attribute was introduced in prman13.
attributes > stochastic		
sigma	0	Attribute "stochastic" "int sigma" [0] The "stochastic" "sigma" attribute has no effect unless the "sigma" option to RiHider has been enabled. If that is the case, setting this attribute to a value of 1 indicates that the geometry to follow should undergo sigma hiding. When both are enabled alternate hiding algorithms designed especially for use with very small, thin, or transparent geometry are used. When dealing with such geometry, the usage of sigma hiding generally has a better balance of antialiasing, speed, and memory properties than that achieved by simply raising the PixelSamples setting globally. In RPS 13.0 beta 1, only curves with Attribute "dice" "hair" [1] and Attribute "stochastic" "int sigma" [1] will undergo sigma hiding. This attribute was introduced in prman13.

Control (UI)	Default Value	Function
geometricApproximation		
motionfactor	0	GeometricApproximation "float motionfactor" [0.0] A special geometric approximation type, motionfactor, is available with RiGeometricApproximation to provide a processing/quality tradeoff for motion blurred objects whose motion can be large in screen space. This will cause the renderer to check the length of the motion blur on the screen for all motion blurred objects and if the distance is large, raise the effective shading rate value of the blurred objects. Since the objects will be blurred across the screen, fine shading detail would be lost anyway. This can save large amounts of processing if many objects in the scene have large blurs. A motionfactor factor value of 0.0 will turn this feature off. Values greater than 1.0 will cause motion blurred objects to have their effective shading rate raised even higher.
flatness	1	GeometricApproximation "float flatness" [1.0] The predefined geometric approximation is "flatness." Flatness is expressed as a distance from the true surface to the approximated surface in pixels. Flatness is sometimes called chordal deviation.

Control (UI)	Default Value	Function
orientation	rh	The handedness of a coordinate system is referred to as its orientation. The initial "camera" coordinate system is left-handed: x points right, y point up, and z points in. Transformations, however, can flip the orientation of the current coordinate system. An example of a transformation that does not preserve orientation is a reflection. (More generally, a transformation does not preserve orientation if its Jacobian is negative.) Similarly, geometric primitives have an orientation, which determines whether their surface normals are defined using a right-handed or left-handed rule in their object coordinate system. Defining the orientation of a primitive to be opposite that of the object coordinate system causes the primitive to be turned inside-out. If a primitive is inside-out, its normal will be computed so that it points in the opposite direction. This has implications for culling, shading, and solids (see the section on Solids and Spatial Set Operations). The outside surface of a primitive is the side from which the normal points outward; the inside surface is the opposite side. The interior of a solid is the volume that is adjacent to the inside surface and the exterior is the region adjacent to the outside. This is discussed further in the section on Geometric Primitives. The current orientation of primitives is maintained as part of the graphics state independent of the orientation of the current coordinate system. The current orientation is initially set to match the orientation of the initial coordinate system, and always flips whenever the orientation of the current coordinate system flips. It can also be modified directly with RiOrientation and RiReverseOrientation. If the current orientation is not the same as the orientation of the current coordinate system, geometric primitives are turned inside out, and their normals are automatically flipped. This procedure sets the current orientation to be either: • **lh** - explicit left-handed orientation • **rh** - explicit right-handed orientation. • **outside** - match the current coordinate system • **inside** - the inverse of the current coordinate system
shadingRate	1	The number of shading calculations per primitive is controlled by the current shading rate. The shading rate is expressed in pixel area. If geometric primitives are being broken down into polygons and each polygon is shaded once, the shading rate is interpreted as the maximum size of a polygon in pixels. A rendering program will shade at least at this rate, although it may shade more often. Whatever the value of the shading rate, at least one shading calculation is done per primitive. A shading rate of RI_INFINITY specifies that shading need only be done once per polygon. A shading rate of 1 specifies that shading is done at least once per pixel. This second case is often referred to as Phong shading.

Control (UI)	Default Value	Function
sides	2	Objects can be two-sided or one-sided. Both the inside and the outside surface of two-sided objects are visible, whereas only the outside surface of a one-sided object is visible. If the outside of a one-sided surface faces the viewer, the surface is said to be frontfacing, and if the outside surface faces away from the viewer, the surface is backfacing. Normally closed surfaces should be defined as one-sided and open surfaces should be defined as two-sided. The major exception to this rule is transparent closed objects, where both the inside and the outside are visible. If sides is 2, subsequent surfaces are considered two-sided and both the inside and the outside of the surface will be visible. If sides is 1, subsequent surfaces are considered one-sided and only the outside of the surface will be visible.
shadinginterpolation	smooth	Shading calculations are performed at discrete positions on surface elements or in screen space (at a frequency determined by the shading rate). The results can then either be interpolated or constant over some region of the screen or the interior of a surface element corresponding to one shading sample. This function controls how values are interpolated between shading samples (usually across a polygon). • **constant** - the color and opacity of all the pixels inside the polygon are the same. This is often referred to as flat or facetted shading. • **smooth** - the color and opacity of all the pixels between shaded values are interpolated from the calculated values. This is often referred to as Gouraud shading.
matte	No	Matte objects are the functional equivalent of three-dimensional holdout mattes. Matte objects are not shaded and are set to be completely opaque so that they hide objects behind them. However, regions in the output image where a matte object is visible are treated as transparent.
opacity	1.0000, 1.0000, 1.0000	Sets the current opacity. The color component values must be in the range [0,1]. Normally there are three components in the color (red, green, and blue), but this may be changed with RiColorSamples. If the opacity is 1, the object is completely opaque; if the opacity is 0, the object is completely transparent.
color	1.0000, 1.0000, 1.0000	Sets the current color. Normally there are three components in the color (red, green, and blue), but this may be changed with the colorsamples request.

Control (UI)	Default Value	Function
scopedCoordinateSystem	none	Shaders often need to perform calculations in non-standard coordinate systems. The coordinate systems with predefined names are: "raster," "screen," "camera," "world," and "object." At any time, the current coordinate system can be marked for future reference. This function marks the coordinate system defined by the current transformation with the indicated name and saves it. But the marked transformation is saved on a separate stack, independent of the global list maintained by RiCoordinateSystem. This stack is pushed and popped by RiAttributeBegin and RiAttributeEnd calls (but not by RiTransformBegin and RiTransformEnd). Scoped coordinate system can then be referred to by name in subsequent shaders, or in RiTransformPoints and RiCoordSysTransform, just like global coordinate systems.
ribinclude	none	This is a place where you can add any arbitrary rib code to the section of the rib defined by the cel statement. If the string entered appears to be a rib file (ends with .rib), the contents of the file will be included as a rib archive. Otherwise, the text itself is entered into the rib stream.
enableStrictVisibilityTesting	No	When set to **Yes**, prman visibility options (camera, diffuse, specular, transmission, photon, midpoint) are not re-enableable beneath a bounded location where they had been previously disabled. This is acutally PRman's default behavior (at least since prman 12.5). Normally, to emulate the behavior artists prefer, Katana plays some tricks (see the note below) with the prman attribute stack. However, this convenience is at the expense of memory and performance. Thus it is often desirable to disable this behavior, such as during instancing. Once this attribute is enabled in the Scene Graph, subsequence calls to unset it (at child locations) are ignored. **Note:** To allow the traditional re-enabling of visiblity, Katana resets the visibility to all on before all subdivide calls. Following the subdivide, the orginial attributes are restored. (Katana keeps its own record of the visibility attribute stack internally).

ProxyMakeRenderable

Control (UI)	Default Value	Function
mode	Explicit Paths	Determines whether to use explicit paths vs. CEL statement for specifying the locations to promote. CEL is recommended in all cases. Explicit Paths is merely provided for backwards compatibility. For more information, see the CEL Reference document in the **Help** > **Quick Reference** menu.
mode: Explicit Paths		
locations		
locations	none	Specify the location of the proxy.
mode: CEL		
CEL		
CEL	none	Collection Expression Language used to specify Scene Graph locations on which an operation or assignment acts.
proxyType	viewer	Determines the type of proxy to use for promotional purposes. Proxies are listed as children of the proxies attribute. For example, to load the proxies.pit proxy, set proxyType to "pit".
mode: CEL		
pruneOriginalChildren	Yes	When set to **Yes**, removes any other children (e.g. actual geometry) of the location being promoted. This ensures that the promoted proxy geometry is the only child.

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 148 and VisibilityAssign on page 199.

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 hit the little arrow on the CEL parameter and choose Collect and Select in Scene Graph....

  This will process for a time, then select all objects that are pruned in the Scene Graph panel.

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 in the **Help** > **Quick Reference** menu.

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:

- Don't delete the port 'input' on the render node, or the node will become unusable.
- 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.

Control (UI)	Default Value	Function
excludeFromFarmOutput-Generation	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		
forceFarmOutputGenera-tion	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 (**excludeFromFarmOutputGenera-tion** 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 con-version 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 'renderOut-puts.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 dif-ferent 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.

Control (UI)	Default Value	Function
channel	rgba	Sets the channels to output. You can also set a user-defined channel from a PrmanDisplayChannelDefine node.
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 2D 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 used.
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	Sest 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.

StickyNote

While not a node as such, you can insert StickyNotes into the Node Graph to make context sensitive notes or record useful information. Double-click on the node to display a properties dialog box.

TeapotCreate

This node creates a specific type of PrimitiveCreate node rendering a teapot instead of a sphere or cube. See also PrimitiveCreate on page 167.

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.)
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).

Control (UI)	Default Value	Function
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 in the **Help** > **Quick Reference** menu.
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.

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 in the **Help** > **Quick Reference** menu.
visible	1	Sets the visibility of objects in the render. 0 specifies not visible in render and anything else specifies visible in render.