

Goblin Node User Manual

Practical workflows and precise node reference

App version	1.7.2
Documentation revision	b3b8e5bfad5d
Generated	2026-10-01
Canonical manual	https://rainerweston.github.io/goblin-node-docs/manual/

Contents

- Start Here
- Goblin Node Manual
- Make your first texture
- Core Workflows
- Manage graphs in the Project Browser
- Import assets
- Connect nodes
- Preview graph results
- Save a graph
- Import or export a graph package
- Export textures
- Concepts
- How graph flow works
- Ports and image channels
- Resolution inheritance
- Bit depth
- Animation and Time
- Local persistence
- Recipes
- First procedural texture
- Organic noise material
- Hard-surface panel
- Stylized retro texture
- Basic animated texture
- Troubleshooting
- The preview is blank
- Node Reference
- Image
- Checker
- Gradient
- Polygon
- Shape
- Solid Color
- Text
- Paint
- Atlas Pack
- Atlas Split
- Shape Extrude
- Fractal Noise
- Noise
- Perlin Noise

Voronoi
Simplex Noise
Gabor Noise
Anisotropic Noise
Crystal Noise
BnW Spots
Splatter Circular
Tile Generator
Auto Levels
Combine RGBA
Combine HSB
Combine Alpha
Gradient (Dynamic)
Gradient Map
Grayscale
Histogram Range
Histogram Scan
Histogram Select
HSL Adjustment
Invert
Levels
Palette Extract
Palettize
Quantize
Separate RGBA
Separate HSB
Separate Alpha
Threshold
Curves
Box Blur
Directional Blur
Radial Blur
Blur HQ
Non-Uniform Blur
Anisotropic Kuwahara
Highpass
Channel Swizzle
FXAA
Median
Sharpen
Slope Blur
Coordinates Transform
Directional Warp
Multi Directional Warp
Kaleidoscope

Mirror
Quick Tile
Clone Stamp
Quad Transform
Skew
Trapezoid Transform
Transform
Safe Transform
Image Sample
Vector Warp
Warp
Swirl
Spherize
Bevel
Curvature
Distance
Dilate / Erode
Edge Detect
Emboss
Flood Fill
Flood Fill Random Grayscale
Flood Fill to Grayscale/Color
Flood Fill to BBox Size
Flood Fill To Index
Flood Fill to Gradient
Ambient Occlusion
Auto Crop
Normal Combine
Normal Map
Blend
Height Blend
Average Blend
Switch
Multi Switch
Input
Instance
Output
Float
Bool
Int
Time
Enum
Vec2
Math
Dot

Frame
Comment

Start Here

Goblin Node Manual

Applies to ios · Since 1.6.0

Use the manual

- **Learn by making.** Start with a working texture. Change one parameter to see how the graph responds.
 - **Find exact behavior.** Use node and concept pages for ports, defaults, and limits.
 - **Recover quickly.** Troubleshooting pages start with a visible symptom and lead to a testable fix.
-

Make your first texture

Applies to ios · Since 1.6.0

Build a checker texture and view it in the Output preview. You will add nodes, make a connection, and change a parameter.

Before you start

Open Goblin Node and create a graph. Keep the node library and Output preview visible.

Build the graph

1. Find **Checker** under **Generators > Pattern**.
2. Drag **Checker** into the graph.
3. Find **Graph Output** under **Graph**.
4. Drag **Graph Output** to the right of **Checker**.
5. Drag from the **out** output port on **Checker**.
6. Release the connection on the input port of **Graph Output**.

The checker pattern now appears in the Output preview.

Change the pattern

1. Select **Checker**.
2. Set **Squares** to 16.

The preview now shows more, smaller squares. Goblin Node recalculates downstream nodes after a parameter changes.

Try one change

Set **Squares** to 4. Compare the larger squares with the previous result.

Common mistakes

- If the preview is blank, check that the connection ends at **Graph Output**.
- If the wrong controls appear, select **Checker** again.
- If a panel is closed, use its toolbar control to reopen it.

Next steps

- [Connect nodes](#)
- [How graph flow works](#)
- [Checker reference](#)
- [The preview is blank](#)

Core Workflows

Manage graphs in the Project Browser

Applies to ios · Since 1.6.0

Use the Project Browser to manage graphs, assets, collections, and Trash.

Create and organize a graph

1. Open the Project Browser.
2. Create a graph.
3. Enter a distinct name.
4. Move the graph into a collection when you need catalog grouping.

Recover a graph

Items moved to Trash remain available for restoration. Open **Trash**, select the graph, and use the restore action.

CAUTION: A purge permanently removes the selected item. Confirm the target before you purge it.

Import assets

Applies to ios · Since 1.6.0

Add an asset that an Image node or palette workflow can use.

1. Open **Assets** in the Project Browser.
2. Open the asset picker.
3. Choose a supported source.
4. Review the new asset in the catalog.

iOS can offer file, camera, or photo-library choices when the operating system permits access.

Palette imports accept palette assets that use PNG, HEX, or Adobe ASE data.

Connect nodes

Applies to ios · Since 1.6.0

Create a connection so one node can send data to another node.

Procedure

1. Find the source node.
2. Drag from its output port.
3. Move the pointer to a compatible input port.
4. Release the pointer.

The graph shows a connection between the two ports. The downstream result updates in the preview.

If the connection does not appear

1. Check that you started from an output port.
2. Check that the destination is an input port.
3. Try another compatible input port.

See [How graph flow works](#) for the direction and purpose of data flow.

Preview graph results

Applies to ios · Since 1.6.0

Choose the preview that matches the result you need to inspect.

1. Use the Selected preview to inspect the selected node.
2. Use the Output preview to inspect a configured Graph Output result.
3. Use the 3D preview to inspect assigned material maps on a mesh.

The size and bit-depth badge describes the current two-dimensional preview result. Preview controls do not change exported pixels unless they change graph or export settings.

Save a graph

Applies to ios · Since 1.6.0

Save after a meaningful graph change or before leaving the editor.

1. Press the save control.
2. Wait for the saving state to finish.
3. Confirm that no save-failed notice appears.

Autosave can preserve later changes. A failed save keeps the graph dirty and permits another manual or automatic attempt.

If saving fails, keep the editor open and retry after the connection or storage problem clears.

Import or export a graph package

Applies to ios · Since 1.6.0

Use a `.gnproj` file for a graph without bundled assets. Use `.gnpack` when the export includes assets or dependencies.

Use a `.gnpackage` file to share a collection as a custom node library. Collection package transfer requires Pro.

Export

1. Select a graph in the Project Browser.
2. Open the graph transfer action.
3. Choose the required package options.
4. Export the package.

Import

1. Open the Project Browser.
2. Open the package importer.
3. Choose a `.gnproj` or `.gnpack` file.
4. Review the package details.
5. Confirm the import.

Graph package import and export require Pro. Package import creates a graph rather than replacing an open graph.

Export a custom node library

1. Mark a collection as a node library.
2. Open the collection action menu.
3. Select **Export .gnpackage**.
4. Select optional unreferenced assets from the collection.
5. Export the package.

The package always includes nested collections, graph dependencies, and referenced assets.

Import a custom node library

1. Open the Project Browser.
2. Select **Import .gnpackage**.
3. Choose a .gnpackage file.
4. Review the collection, graph, and asset counts.
5. If items are invalid, select the option to import valid items only.
6. Confirm the import.

Import creates a new top-level collection and does not replace an existing library. The library is visible on the importing device by default.

Export textures

Applies to ios · Since 1.6.0

Export a configured Graph Output result from the editor.

1. Open the Output preview.
2. Select the required graph output.
3. Display the export dialog.
4. Set the file format.
5. Set the naming options.
6. Confirm the operation.

A single encoded file uses the iOS save or share flow.

Multiple output files create a ZIP archive. Batch export and animation export require Pro.

Concepts

How graph flow works

Applies to ios · Since 1.6.0

A graph is a procedural recipe made from connected nodes. Each node creates data or changes data from an earlier node.

An output port sends data. An input port receives data. A connection carries the data from the output port to the input port.

Flow matters because Goblin Node evaluates upstream sources before downstream results. A **Checker** node can feed **Box Blur**, which can then feed **Graph Output**.

```
Checker -> Box Blur -> Graph Output
```

Graph Output identifies a result that the Output preview and export tools can use. A branch that does not reach an output can still appear in the Selected preview.

Learn how to [connect nodes](#) or [make your first texture](#).

Ports and image channels

Applies to ios · Since 1.6.0

An input port receives data, and an output port sends data. Compatible ports form a connection.

Image data can contain one grayscale channel, red-green-blue channels, or red-green-blue-alpha channels. Alpha stores transparency.

Channel nodes separate or combine image channels. Value nodes send typed parameter data, such as a float, integer, Boolean value, enumeration, or two-component vector.

Goblin Node can promote data to a wider channel layout when a node contract supports it. Some nodes intentionally return grayscale or opaque color.

Resolution inheritance

Applies to ios · Since 1.6.0

Resolution inheritance lets a node get its output size from the graph or a connected input.

Relative to Parent starts from the graph resolution. **Relative to Input** starts from the node's authoritative input. A scale then changes that base size.

Some nodes define a special dimension source. Image Sample uses **Coordinates** when connected, then **Image**, then the graph dimensions.

Pixel-based modes use explicit output pixels. Their relative footprint changes when the graph resolution changes.

Bit depth

Applies to ios · Since 1.6.0

Bit depth controls how precisely a graph stores image samples. More precision can reduce banding and preserve a wider numeric range.

Eight-bit processing stores ordinary bounded samples. Native 16-bit processing preserves more steps for supported nodes. The 16F mode can preserve finite negative and above-one values through supported paths.

Unsupported nodes can downgrade a branch to eight-bit processing. The preview is a bounded display projection and is not proof that an export preserves the same numeric range.

PNG and TIFF support unsigned 16-bit grayscale export. OpenEXR preserves 16F values in the current floating-point export path.

Animation and Time

Applies to ios · Since 1.6.0

Animation settings store frames per second, an inclusive start frame, an inclusive end frame, and the seamless-loop choice.

The default range is frame 0 through frame 59 at 30 frames per second. The current preview frame is session state and does not make the graph dirty.

The **Time** node can provide seconds, the current frame, or normalized progress. A one-frame range produces normalized progress of 0.

Animation export supports at most 1000 rendered frames and requires Pro. Seamless Loop reserves an extra unrendered endpoint so the last authored frame remains below normalized progress 1.

Local persistence

Applies to ios · Since 1.6.0

Goblin Node uses local IndexedDB storage on iOS for graphs, assets, and catalog data.

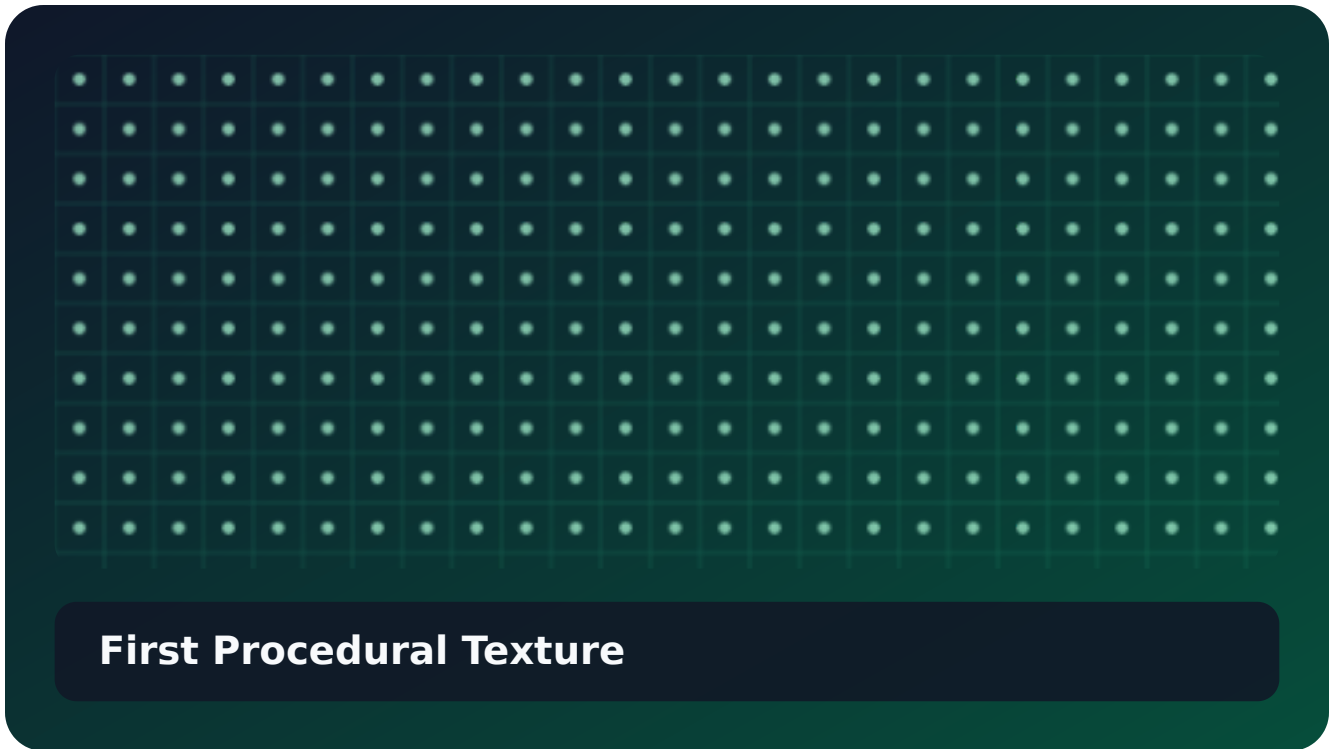
Graph history is local to the device. It does not currently have a restore interface.

Graph packages provide the portable transfer path. Dormant cloud-sync code does not make current graphs available across devices.

Recipes

First procedural texture

Applies to ios · Since 1.6.0



Checker texture preview

This 512 × 512 example connects **Checker** to **Graph Output**. It uses a fixed graph seed of 0.

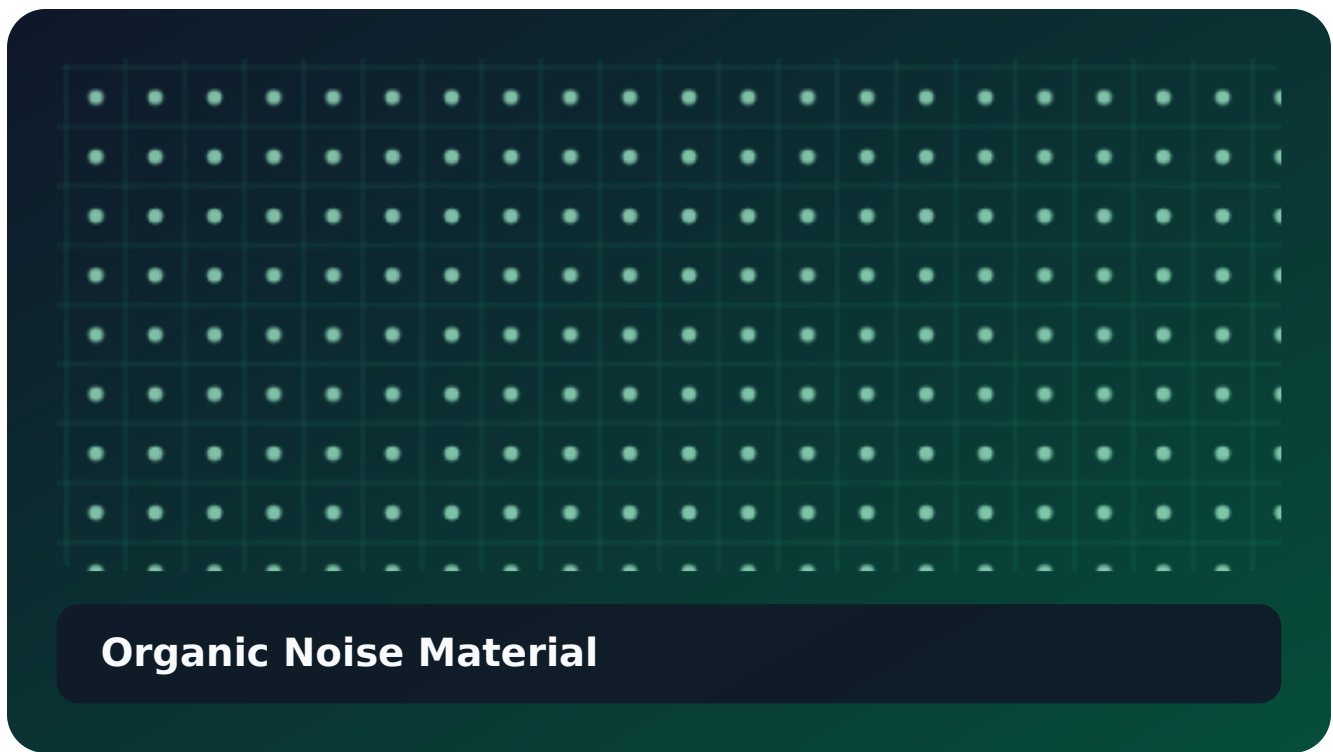
1. Download the [graph package](#).
2. Open the Project Browser.
3. Open the package importer.
4. Choose the downloaded package.
5. Open the imported graph.

The Output preview shows a repeating checker pattern. Change **Squares** to compare larger and smaller cells.

NOTE: Graph package import requires Pro.

Organic noise material

Applies to ios · Since 1.6.0



Organic noise preview

This 512×512 example uses **Perlin Noise**, **Box Blur**, and **Levels**. Its fixed seed is 12345.

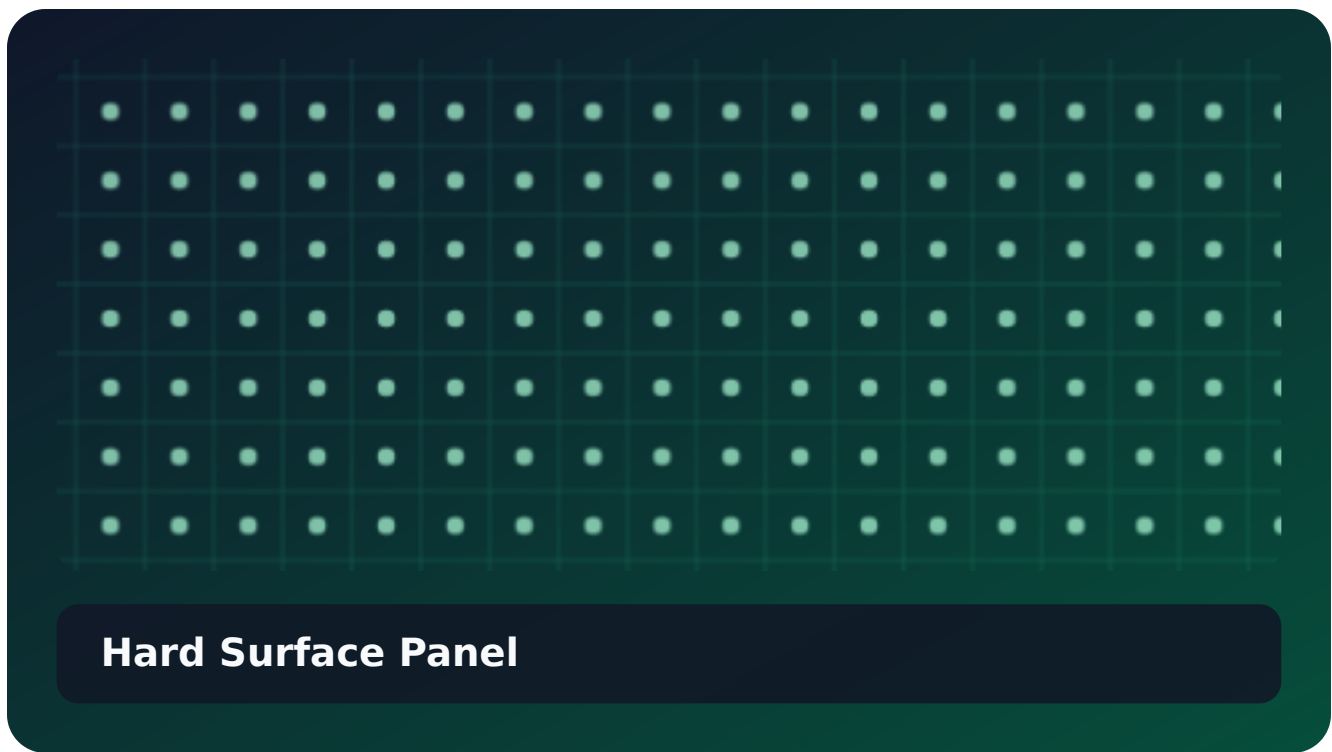
1. Download the [graph package](#).
2. Import the package from the Project Browser.
3. Open the imported graph.
4. Select **Perlin Noise**.
5. Change **Frequency** by one step.

The texture keeps its structure while its feature scale changes.

NOTE: Graph package import requires Pro.

Hard-surface panel

Applies to ios · Since 1.6.0



Hard-surface panel preview

This 512×512 example sends **Checker** through **Bevel**. The result provides a geometric height base.

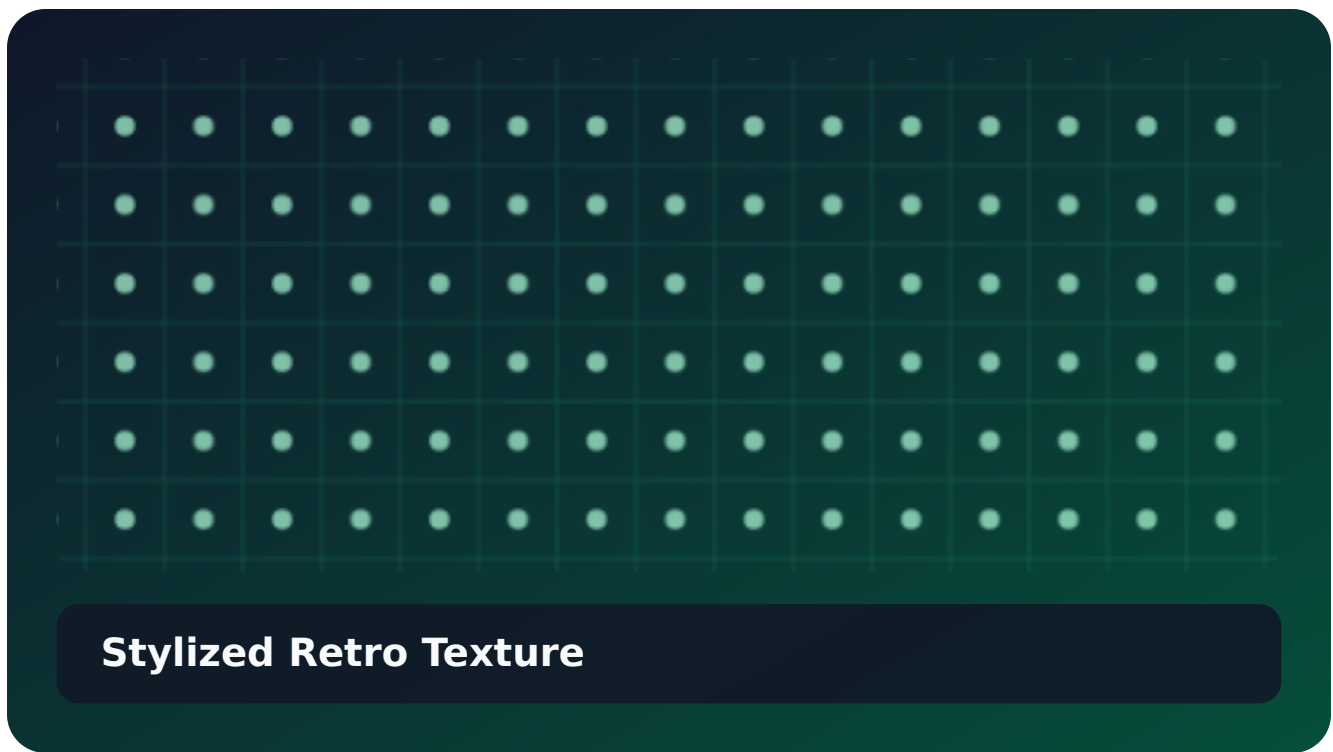
1. Download the [graph package](#).
2. Import the package from the Project Browser.
3. Open the imported graph.
4. Select **Bevel**.
5. Adjust **Distance**.

The panel edges become wider or narrower. **Bevel** requires Pro.

NOTE: Graph package import requires Pro.

Stylized retro texture

Applies to ios · Since 1.6.0



Stylized retro preview

This 512×512 example sends **Gradient** through **Quantize** to create six bands.

1. Download the [graph package](#).
2. Import the package from the Project Browser.
3. Open the imported graph.
4. Select **Quantize**.
5. Set **Steps** to 4.

The Output preview now shows four broader bands.

NOTE: Graph package import requires Pro.

Basic animated texture

Applies to ios · Since 1.6.0



Basic Animated Texture

Animated texture preview

This example connects **Time** to **Graph Output**. It uses frames 0 through 59 at 30 frames per second.

1. Download the [graph package](#).
2. Import the package from the Project Browser.
3. Open the imported graph.
4. Start preview playback.
5. Watch the output change across the frame range.

The normalized Time value progresses from the first frame toward the final frame.

NOTE: Graph package import and animation export require Pro.

Troubleshooting

The preview is blank

Applies to ios · Since 1.6.0

The Selected preview or Output preview shows no texture.

Most likely cause

The preview has no valid source. A missing connection or the wrong preview target commonly causes this symptom.

Fix the Output preview

1. Find **Graph Output** in the graph.
2. Check its input for a connection.
3. Connect an image-producing node to the input.
4. Select that output in the Output preview.

The connected texture should appear.

Fix the Selected preview

1. Select an image-producing node.
2. Open the Selected preview.

If the preview stays blank, reload the graph and repeat the check. Preserve the graph package before reporting a repeatable rendering failure.

See [Connect nodes](#) for the connection procedure.

Node Reference

Image

Use Image to bring a bitmap asset into the graph as a primary source or reference layer. It commonly feeds cleanup nodes like Levels or Blur before blending into procedural branches.

Availability: Free feature set

Common uses

- Start graphs with one or more generator nodes, then route them into filters or mixing nodes.
- Combine with Levels, Curves, or Blend to shape the final look quickly.

Input ports

- None
- Usually no required image input; parameters define the output pattern.

Output ports

- out
- Primary output is an image signal used by downstream nodes.

Parameters

This node has no shared processing parameters.

Behavior

- Generator nodes define the initial signal that propagates through the graph.
- Most generator outputs can be treated as grayscale unless the node is color-focused.

Common mistakes

- Large or high-contrast sources can amplify artifacts in downstream blur or warp chains.
- Keep source scale consistent before combining multiple generators.

Checker

Use Checker to generate high-contrast repeating blocks for mask testing, UV validation, or tiling previews. It is often combined with warp or threshold nodes to build stylized panel patterns.

Availability: Free feature set

Common uses

- Start graphs with one or more generator nodes, then route them into filters or mixing nodes.
- Combine with Levels, Curves, or Blend to shape the final look quickly.

Input ports

- None
- Usually no required image input; parameters define the output pattern.

Output ports

- out
- Primary output is an image signal used by downstream nodes.

Parameters

Parameter	Type	Default	Range or values
Squares	int	8	1 to 128

Behavior

- Generator nodes define the initial signal that propagates through the graph.
- Most generator outputs can be treated as grayscale unless the node is color-focused.

Common mistakes

- Large or high-contrast sources can amplify artifacts in downstream blur or warp chains.
- Keep source scale consistent before combining multiple generators.

Gradient

Use Gradient to create controlled value ramps for masks, falloffs, and directional shading inputs. It typically drives blend factors, height transitions, or color-mapping chains.

Availability: Free feature set

Common uses

- Start graphs with one or more generator nodes, then route them into filters or mixing nodes.
- Combine with Levels, Curves, or Blend to shape the final look quickly.

Input ports

- None
- Usually no required image input; parameters define the output pattern.

Output ports

- out
- Primary output is an image signal used by downstream nodes.

Parameters

Parameter	Type	Default	Range or values
Type	enum	linear	linear, reflected, radial, circular
Rotation	enum	0	0, 90, 180, 270, custom
Free Rotation	float	0	0 to 360
Center X	float	0.5	0 to 1
Center Y	float	0.5	0 to 1

Behavior

- Generator nodes define the initial signal that propagates through the graph.
- Most generator outputs can be treated as grayscale unless the node is color-focused.

Common mistakes

- Large or high-contrast sources can amplify artifacts in downstream blur or warp chains.
- Keep source scale consistent before combining multiple generators.

Polygon

Use Polygon to produce clean geometric masks with adjustable side count and edge shape. It works well as a base for decals, icons, and hard-surface cutout workflows.

Availability: Free feature set

Common uses

- Start graphs with one or more generator nodes, then route them into filters or mixing nodes.
- Combine with Levels, Curves, or Blend to shape the final look quickly.

Input ports

- None
- Usually no required image input; parameters define the output pattern.

Output ports

- out
- Primary output is an image signal used by downstream nodes.

Parameters

Parameter	Type	Default	Range or values
Sides	int	6	3 to 32
Explode	float	0	0 to 1
Triangle Size	float	1	0 to 1
Scale	float	0.8	0 to 2
Auto Scale	bool	true	-
Flat Orientation	bool	false	-
Rotation	float	0	0 to 1

Parameter	Type	Default	Range or values
Gradient	bool	false	-
Grad Invert	bool	false	-

Behavior

- Generator nodes define the initial signal that propagates through the graph.
- Most generator outputs can be treated as grayscale unless the node is color-focused.

Common mistakes

- Large or high-contrast sources can amplify artifacts in downstream blur or warp chains.
- Keep source scale consistent before combining multiple generators.

Shape

Use Shape to block out primitive silhouettes quickly when prototyping forms or mask logic. It is commonly layered with blur, bevel, or warp nodes to build richer structures.

Availability: Free feature set

Common uses

- Start graphs with one or more generator nodes, then route them into filters or mixing nodes.
- Combine with Levels, Curves, or Blend to shape the final look quickly.

Input ports

- None
- Usually no required image input; parameters define the output pattern.

Output ports

- out
- Primary output is an image signal used by downstream nodes.

Parameters

Parameter	Type	Default	Range or values
Type	enum	square	square, disc, paraboloid, bell, gaussian, thorn, pyramid, hemisphere, cone, capsule
Scale	float	0.9	0.01 to 5
Size X	float	1	0 to 1
Size Y	float	1	0 to 1
Rotation	float	0	0 to 360
Offset X	float	0	-1 to 1
Offset Y	float	0	-1 to 1
Capsule Length	float	0.5	0 to 2
Shape as alpha	bool	false	-

Behavior

- Generator nodes define the initial signal that propagates through the graph.
- Most generator outputs can be treated as grayscale unless the node is color-focused.

Common mistakes

- Large or high-contrast sources can amplify artifacts in downstream blur or warp chains.
- Keep source scale consistent before combining multiple generators.

Solid Color

Use Solid Color to provide a constant fill for backgrounds, tint passes, or blend backplates. It is useful as a neutral baseline when testing downstream color and contrast operations.

Availability: Free feature set

Common uses

- Start graphs with one or more generator nodes, then route them into filters or mixing nodes.
- Combine with Levels, Curves, or Blend to shape the final look quickly.

Input ports

- None
- Usually no required image input; parameters define the output pattern.

Output ports

- out
- Primary output is an image signal used by downstream nodes.

Parameters

This node has no shared processing parameters.

Behavior

- Generator nodes define the initial signal that propagates through the graph.
- Most generator outputs can be treated as grayscale unless the node is color-focused.

Common mistakes

- Large or high-contrast sources can amplify artifacts in downstream blur or warp chains.
- Keep source scale consistent before combining multiple generators.

Text

Use Text to render procedural labels or glyph masks directly inside the graph. It is typically transformed, beveled, or blended to create decals and signage details.

Availability: Free feature set

Common uses

- Start graphs with one or more generator nodes, then route them into filters or mixing nodes.
- Combine with Levels, Curves, or Blend to shape the final look quickly.

Input ports

- None
- Usually no required image input; parameters define the output pattern.

Output ports

- out
- Primary output is an image signal used by downstream nodes.

Parameters

Parameter	Type	Default	Range or values
Font	enum	sans-serif	sans-serif, serif, monospace, cursive, fantasy, Arial, Calibri, Consolas, Courier New, DM Sans, Fira Code, Geist, Georgia, Helvetica, IBM Plex Sans, Inter, JetBrains Mono, Lora, Menlo, Merriweather, Montserrat, Noto Sans, Open Sans, Outfit, Playfair Display, Poppins, Roboto, Segoe UI, Source Code Pro, Times New Roman, Trebuchet MS, Verdana
Size	float	0.25	0 to 10
X	float	0.5	0 to 1
Y	float	0.5	0 to 1
Rotation	float	0	0 to 360

Behavior

- Generator nodes define the initial signal that propagates through the graph.
- Most generator outputs can be treated as grayscale unless the node is color-focused.

Common mistakes

- Large or high-contrast sources can amplify artifacts in downstream blur or warp chains.
- Keep source scale consistent before combining multiple generators.

Paint

Use Paint to add hand-painted detail, masks, or colour corrections directly onto your texture. Choose a black, white, BG Input, or empty background; strokes are composited over that background.

Availability: Free feature set

Common uses

- Use BG Input when you want to paint over a connected image; Black is the default background.
- Select the Paint node and open the Selected preview tab to see the brush overlay.

Input ports

- input
- brush
- Optional BG image, used only when Background is set to BG Input. Brush remains an optional stamp mask.

Output ports

- out
- Black and white backgrounds with grayscale strokes stay grayscale; colour strokes promote the output to RGB. Empty backgrounds and RGBA BG Input preserve RGBA.

Parameters

Parameter	Type	Default	Range or values
Background	enum	black	black, white, bgInput, empty
Brush Size	float	20	1 to 500
Opacity	float	1	0 to 1
Hardness	float	0.8	0 to 1
Brush Spacing	float	0.35	0.05 to 1
Grid Size	int	16	1 to 256

Parameter	Type	Default	Range or values
Blend Mode	enum	normal	normal, multiply, color_burn, linear_burn, min, subtract_clamp, screen, color_dodge, add, max, addSub, overlay, softlight, hardlight, vivid_light, linear_light, pinlight, hard_mix, difference, exclusion, divide, color, hue, saturation, luminosity

Behavior

- Paint strokes are stored as serialised point data in the node.
- The brush overlay appears automatically when the node is selected.

Common mistakes

- Stroke data is stored in the graph file and may increase file size with many strokes.
- Undo is limited to clearing all strokes via the node reset action.

Atlas Pack

Use Atlas Pack to combine multiple textures into a single atlas for efficient GPU storage or sprite-sheet workflows. Each input is placed into a grid cell defined by Grid X and Grid Y dimensions, with cells filled left-to-right, top-to-bottom.

Availability: Pro

Common uses

- Connect images to the numbered input ports. Unused cells remain transparent.
- Adjust Input Count to control how many ports are visible on the node.

Input ports

- None
- Up to 16 image inputs arranged as a numbered list.

Output ports

- out

Port names can depend on the graph or node configuration.

- Single output containing all input images arranged on the grid.

Parameters

Parameter	Type	Default	Range or values
Grid X	int	4	1 to 8
Grid Y	int	4	1 to 8
Input Count	int	4	1 to 16

Behavior

- Inputs are placed into a flat grid without overlap or padding.
- Grid position is determined by input index modulo grid width.

Common mistakes

- All inputs are placed into cells without scaling — input resolutions should match the target cell size for best results.
- The output resolution is determined by the graph resolution, not the inputs.

Atlas Split

Use Atlas Split to extract individual textures from an atlas or sprite-sheet. Each cell of the grid becomes a separate output port, with the cell content scaled to full output resolution.

Availability: Pro

Common uses

- Connect an atlas texture to the input port.
- Adjust Grid X and Grid Y to match the atlas layout.
- Connect downstream nodes to the individual cell outputs you need.

Input ports

- input
- Single atlas image input to be split into grid cells.

Output ports

- None

Port names can depend on the graph or node configuration.

- Each grid cell becomes a separate numbered output port with the cell content scaled to full resolution.

Parameters

Parameter	Type	Default	Range or values
Grid X	int	4	1 to 8
Grid Y	int	4	1 to 8

Behavior

- Outputs are generated left-to-right, top-to-bottom matching standard atlas layout.
- Only connected output ports are computed; disconnected outputs are skipped.

Common mistakes

- Only connected outputs are rendered — unused outputs are skipped for performance.
- Cell content is stretched to fill the full output resolution; source cell aspect ratio is not preserved.

Shape Extrude

Use Shape Extrude to generate heightmaps from primitive or custom shapes. The shape is extruded along its axis to create a volumetric heightfield. Built-in Cube and Cylinder shapes provide clean base geometry, while Custom mode accepts an external shape mask. Bevel softens edges, profile gradients modulate the extrusion cross-section, and rotation tilts the result in 3D space.

Availability: Pro

Common uses

- Use Cube or Cylinder for clean primitive extrusions.
- Connect a grayscale mask to the Shape input for custom extrusion profiles.
- Adjust Bevel to soften the extrusion edges.
- Use Rotation X/Y to tilt the extruded shape in 3D space.

Input ports

- shape
- profileGradient
- profileMask
- Shape: optional grayscale mask for custom extrusion profile.
- Profile Gradient: optional image sampled along the V-axis to modulate cross-section.
- Profile Mask: optional image sampled at pixel position to selectively mask the extrusion.

Output ports

- out
- Outputs a single-channel grayscale heightmap representing the extruded 3D volume.

Parameters

Parameter	Type	Default	Range or values
Shape Type	enum	cube	cube, cylinder, custom
Shape Size	float	0.5	0.01 to 1
Extrude Height	float	0.5	0 to 1
Extrude Depth	float	0.5	0 to 1
Scale	float	0.5	0.01 to 1
Profile Type	enum	straight	straight, gradient, mask
Bevel	float	0.5	0 to 1
Rotation X	float	0	-90 to 90
Rotation Y	float	0	-90 to 90

Behavior

- Height values range from black (extrude depth base) to white (extrude height peak).
- Bevel uses smoothstep interpolation for clean edge falloff.
- Profile gradient samples the connected image along the extrusion V-axis.

Common mistakes

- Extreme bevel values (>0.5) may over-soften small shapes.
- Custom shape input should be a binary or soft grayscale mask.
- Large rotation angles may produce clipped results near edges.

Fractal Noise

Use Fractal Noise for multi-scale breakup where both broad forms and fine detail are required. It often drives roughness masks, terrain-like height variation, and layered distortion.

Availability: Pro

Common uses

- Use low-frequency noise for broad breakup and high-frequency noise for detail.
- Drive distortion, blend masks, or thresholding with noise outputs.

Input ports

- None
- No external image input required; parameters and seed control output.

Output ports

- out
- Outputs grayscale or multichannel noise depending on node settings.

Parameters

Parameter	Type	Default	Range or values
Type	enum	fbm	fbm, ridged

Parameter	Type	Default	Range or values
Frequency	int	16	1 to 128
Seed	int	12345	0 to 65535
Octaves	int	4	1 to 8
Lacunarity	float	2	1 to 4
Gain	float	0.5	0 to 1
Disorder	float	0	0 to 1
Multichannel	bool	false	-
Non-Square Expansion	bool	true	-

Behavior

- Noise nodes are deterministic for a given seed and parameter set.
- Different noise models create different value distributions and edge behavior.
- Disorder animates the underlying noise ingredients on a perfect loop, so 1.0 returns to the same result as 0.0.

Common mistakes

- Very high frequency values can alias at lower output resolutions.
- When using seed-heavy graphs, keep seed strategy organized for reproducibility.

Noise

Use Noise for fast random variation when you need grain, dither, or stochastic masking. It is commonly filtered and remapped before being used as blend or warp control.

Availability: Free feature set

Common uses

- Use low-frequency noise for broad breakup and high-frequency noise for detail.
- Drive distortion, blend masks, or thresholding with noise outputs.

Input ports

- None
- No external image input required; parameters and seed control output.

Output ports

- out
- Outputs grayscale or multichannel noise depending on node settings.

Parameters

Parameter	Type	Default	Range or values
Seed	int	0	0 to 65535
Disorder	float	0	0 to 1
Multichannel	bool	false	-
Non-Square Expansion	bool	true	-

Behavior

- Noise nodes are deterministic for a given seed and parameter set.
- Different noise models create different value distributions and edge behavior.
- Disorder animates the underlying noise ingredients on a perfect loop, so 1.0 returns to the same result as 0.0.

Common mistakes

- Very high frequency values can alias at lower output resolutions.
- When using seed-heavy graphs, keep seed strategy organized for reproducibility.

Perlin Noise

Use Perlin when you want smooth organic variation without harsh cell boundaries. It is effective for clouds, soft erosion masks, and flowing material modulation.

Availability: Free feature set

Common uses

- Use low-frequency noise for broad breakup and high-frequency noise for detail.
- Drive distortion, blend masks, or thresholding with noise outputs.

Input ports

- None
- No external image input required; parameters and seed control output.

Output ports

- out
- Outputs grayscale or multichannel noise depending on node settings.

Parameters

Parameter	Type	Default	Range or values
Frequency	int	8	1 to 128
Seed	int	12345	0 to 65535
Disorder	float	0	0 to 1
Multichannel	bool	false	-
Non-Square Expansion	bool	true	-

Behavior

- Noise nodes are deterministic for a given seed and parameter set.
- Different noise models create different value distributions and edge behavior.
- Disorder animates the underlying noise ingredients on a perfect loop, so 1.0 returns to the same result as 0.0.

Common mistakes

- Very high frequency values can alias at lower output resolutions.
- When using seed-heavy graphs, keep seed strategy organized for reproducibility.

Voronoi

Voronoi divides the image into seeded cells. Distance keeps the original smooth field. Cell walls produces a binary crack mask, Flat cells assigns one seeded value per region, and Sampled cells takes one RGB color from the optional Image input at each cell's feature point.

Availability: Pro

Common uses

- Use Cell walls and Wall width to make adjustable cracks directly.
- Use Flat cells as a clear input for Edge Detect or Flood Fill.
- Connect a color image to Sampled cells to assign one color per cell.

Input ports

- image
- Image is optional and used by Sampled cells; other modes ignore it.

Output ports

- out
- Distance retains the existing grayscale or multichannel result. Cell walls is grayscale; Flat cells can be grayscale or multichannel; Sampled cells is RGB.

Parameters

Parameter	Type	Default	Range or values
Output	enum	distance	distance, walls, flat, sampled
Wall Width	float	0.06	0.005 to 0.3
Frequency	int	10	1 to 128
Randomness	float	1	0 to 1
Seed	int	12345	0 to 999999
Disorder	float	0	0 to 1
Distance	enum	euclidean	euclidean, manhattan, chebyshev
Invert	bool	false	-
Multichannel	bool	false	-
Non-Square Expansion	bool	true	-

Behavior

- Noise nodes are deterministic for a given seed and parameter set.
- Different noise models create different value distributions and edge behavior.
- Disorder animates the underlying cell layout on a perfect loop, so 1.0 returns to the same result as 0.0.

Common mistakes

- Very high frequency values can alias at lower output resolutions.
- Sampled cells uses seeded colors when Image is disconnected.
- GPU rendering is available when supported; the CPU path remains the fallback.

Simplex Noise

Use Simplex Noise when you need smooth noise with better isotropy than Perlin. Simplex noise has fewer directional artifacts and is well-suited for terrain, cloud, and flow-like procedural textures where directional bias would be visible.

Availability: Pro

Common uses

- Use low-frequency noise for broad breakup and high-frequency noise for detail.
- Drive distortion, blend masks, or thresholding with noise outputs.

Input ports

- None
- No external image input required; parameters and seed control output.

Output ports

- out
- Outputs grayscale or multichannel noise depending on node settings.

Parameters

Parameter	Type	Default	Range or values
Frequency	int	8	1 to 128
Seed	int	12345	0 to 65535
Disorder	float	0	0 to 1
Multichannel	bool	false	-
Non-Square Expansion	bool	true	-

Behavior

- Noise nodes are deterministic for a given seed and parameter set.
- Simplex noise uses a skewing grid to produce fewer directional artifacts than Perlin.
- Disorder animates the underlying noise ingredients on a perfect loop, so 1.0 returns to the same result as 0.0.

Common mistakes

- Very high frequency values can alias at lower output resolutions.
- When using seed-heavy graphs, keep seed strategy organized for reproducibility.

Gabor Noise

Use Gabor Noise to generate textured patterns based on Gabor wavelets — sinusoidal carriers windowed by Gaussian envelopes. Each cell contains one random wavelet impulse with hash-driven position and weight, evaluated over a 3×3 neighborhood for smooth continuity. Supports Fractal Brownian Motion (FBM) via octaves for multi-scale detail. The direction vector (from Angle) gives oriented fibrous textures.

Availability: Pro

Common uses

- Scale controls the cell (feature) size in pixels.
- Frequency controls the underlying wave rate of the Gabor kernel.
- Alpha controls the Gaussian envelope width (higher = more localized impulses).
- Angle sets the direction of the Gabor wave (orientation of fibers).
- Octaves enables fractal FBM stacking for multi-scale detail.
- Persistence controls amplitude falloff per octave (higher = more high-frequency detail).

Input ports

- None
- No external image input required; parameters and seed control output.

Output ports

- out
- Outputs grayscale or multichannel noise depending on node settings.

Parameters

Parameter	Type	Default	Range or values
Bandwidth	float	64	1 to 512
Impulse Density	float	16	1 to 128
Frequency	float	0.1	0.01 to 10
Octave Range	float	3	0 to 8
Orientation	float	0	0 to 360
Isotropy	float	1	0 to 1
Disorder	float	0	0 to 1
Seed	int	12345	0 to 65535
Multichannel	bool	false	-
Non-Square Expansion	bool	true	-

Behavior

- Gabor noise is deterministic for a given seed and parameter set.
- Each cell has one random impulse; the 3×3 neighborhood ensures smooth transitions.
- FBM mode stacks octaves with 2× frequency multiplier per octave.

Common mistakes

- Very high frequency with low alpha may produce visible ringing or aliasing.

Anisotropic Noise

Anisotropic Noise creates an elongated random-value field. X Amount and Y Amount set the number of regions on each axis; the default makes horizontal strips. Rotate turns the field 90 degrees. Smoothness sets how much of each region fades into the next, while Smoothness Interpolation blends a linear fade toward a smoother quintic curve. Disorder and Disorder Speed bend the bands. A seed makes the pattern reproducible. Goblin Node also supports independent tiling axes, non-square expansion, and multichannel RGB output.

Availability: Pro

Common uses

- Start with X Amount 1 and Y Amount 18 for horizontal bands; turn on Rotate for vertical bands.
- Increase X Amount to break the strips into shorter segments.
- Use Smoothness 0 for sharp boundaries or 1 for fades spanning each region.
- Enable Multichannel for independently seeded red, green, and blue fields.

Input ports

- None
- No image input is required; graph and node seeds control the generated field.

Output ports

- out
- Outputs grayscale by default or independent RGB channels when Multichannel is enabled.

Parameters

Parameter	Type	Default	Range or values
X Amount	int	1	1 to 128
Y Amount	int	18	1 to 128
Y Amount by Resolution	bool	false	-
Rotate 90°	bool	false	-
Smoothness	float	0.75	0 to 1
Smoothness Interpolation	float	0.5	0 to 1
Disorder	float	0	0 to 1
Disorder Speed	float	0.5	0 to 2
Seed	int	12345	0 to 65535
Non-Square Expansion	bool	false	-
Multichannel	bool	false	-

Behavior

- Values are deterministic for a given graph seed, node seed, and parameters.
- When tiling is enabled, the field and displacement repeat across the selected edges.
- Non-Square Expansion keeps regions closer to square on rectangular images; Y Amount by Resolution adjusts Y count only when expansion is off.

Common mistakes

- A single region on a tiled axis is constant along that axis; increase its amount to add variation.
- Very high counts at a small output resolution can alias.
- Disorder Speed scales the displacement amplitude, not animation time.

Crystal Noise

Use Crystal Noise to create sharp, angular crack patterns that resemble crystalline structures, shattered glass, or mineral veins. Each cell randomly generates 2–6 branch segments radiating outward with hash-driven angles and lengths. Disorder controls branch angle jitter and spread. Supports multichannel output for RGB variation across seeds.

Availability: Pro

Common uses

- Frequency controls the number of cells (lower = larger crystals).
- Disorder adds randomness to branch angles and spacing.
- Invert flips black and white.
- Multichannel produces independent RGB crystal patterns.

Input ports

- None
- No external image input required; parameters and seed control output.

Output ports

- out
- Outputs grayscale or multichannel noise depending on node settings.

Parameters

Parameter	Type	Default	Range or values
Frequency	int	8	1 to 128
Disorder	float	0.5	0 to 1
Seed	int	12345	0 to 999999
Invert	bool	false	-
Multichannel	bool	false	-
Non-Square Expansion	bool	true	-

Behavior

- Crystal noise is deterministic for a given seed and parameter set.
- Each cell evaluates a 3×3 neighborhood for smooth branch continuity across cell boundaries.
- RGB multichannel uses three independent seed offsets for varied patterns per channel.

Common mistakes

- Very low frequency with high disorder may produce overlapping branches.

BnW Spots

Use BnW Spots to create organic black-and-white spot patterns with multi-layer stacking. The algorithm places spots at jittered grid positions across multiple layers with randomized sizes. Scale controls overall density, Disorder adds positional randomness, Roughness controls edge softness, and Layers determines how many stacked spot layers are multiplied together. Supports multichannel output for independent RGB spot patterns.

Availability: Pro

Common uses

- Scale controls the grid density (1 = coarse, 8 = fine).
- Disorder adds randomness to spot positions within each cell.
- Roughness controls the edge falloff of spots (0 = hard, 1 = normal, 2 = very soft).
- Layers stacks multiple spot layers at different scales for richer patterns.
- Invert flips black and white.
- Multichannel produces independent RGB spot patterns.

Input ports

- None
- No external image input required; parameters and seed control output.

Output ports

- out
- Outputs grayscale or multichannel noise depending on node settings.

Parameters

Parameter	Type	Default	Range or values
Scale	int	4	1 to 8
Disorder	float	0.5	0 to 1
Roughness	float	1	0 to 2
Layers	int	3	1 to 8
Seed	int	12345	0 to 999999
Invert	bool	false	-
Multichannel	bool	false	-
Non-Square Expansion	bool	true	-

Behavior

- BnW Spots is deterministic for a given seed and parameter set.
- Each layer independently places spots at jittered grid positions.
- Layers are combined via multiplicative blending (dark spots accumulate).
- RGB multichannel uses three independent seed offsets for varied patterns per channel.

Common mistakes

- Very high scale with many layers may produce dense, nearly solid output.
- Low roughness with high disorder may create overlapping hard-edged spots.

Splatter Circular

Use Splatter Circular to distribute instances of an input pattern radially. You can control radius, rotation, scale, masking, and randomization per instance. Excellent for radial motifs, lens effects, and decorative rosettes.

Availability: Pro

Common uses

- Enable Center Orientation to align instances radially towards the pivot.
- Tweak radius variation and scale variation for natural scattering.

Input ports

- pattern
- background
- Takes a Pattern image to splatter and a Background image.

Output ports

- out
- Outputs the scattered composite.

Parameters

Parameter	Type	Default	Range or values
Number	int	12	1 to 64
Radius	float	0.25	0 to 1
Radius Rand	float	0	0 to 1
Ring Amount	int	1	1 to 10
Ring Spacing	float	1	0 to 1
Spread	float	1	0 to 2
Spiral	float	0	0 to 1
Ring Rotation Offset	float	0	0 to 1
Size X	float	0.1	0.01 to 2
Size Y	float	0.1	0.01 to 2

Parameter	Type	Default	Range or values
Size X Rand	float	0	0 to 1
Size Y Rand	float	0	0 to 1
Scale	float	1	0 to 2
Scale Rand	float	0	0 to 1
Rotation	float	0	0 to 1
Rot Rand	float	0	0 to 1
Center Orientation	bool	false	-
Mask Rand	float	0	0 to 1
Lum Rand	float	0	0 to 1
Blend Mode	enum	max	max, add, normal
Seed	int	12345	0 to 999999
Interpolation	enum	nearest	nearest, bilinear

Behavior

- Distributes instances based on angular intervals derived from the number of instances.

Common mistakes

- High instance counts with complex patterns can slow down rendering.
- If scaling is large, instances may clip canvas borders if tiling is disabled.

Tile Generator

Use Tile Generator to lay out repeatable brick or panel motifs with explicit spacing control. It serves as a structural base that can be aged with noise, warp, and edge wear.

Availability: Pro

Common uses

- Use with masks, gradients, or floods to create patterned materials quickly.
- Combine with transform nodes to break perfect repetition.

Input ports

- tile
- pattern2
- pattern3
- pattern4
- scaleMap
- maskMap
- rotationMap
- vectorMap
- distributionMap
- lumaMap
- colorMap
- Inputs vary by node; many pattern controls are parameter-driven.

Output ports

- out
- Outputs pattern maps for color, mask, or height pipelines.

Parameters

Parameter	Type	Default	Range or values
Pattern Number	int	1	1 to 4
Pattern Select	enum	random	random, index, distribution
Pattern Symmetry	float	0	0 to 1
Symmetry Random Mode	enum	horizontalVertical	horizontalVertical, horizontal, vertical
Pattern Rotation Random	float	0	0 to 1
Mask Random	float	0	0 to 1
Scale Map Influence	float	0	0 to 1
Vector Map Rotation	float	0	0 to 1

Parameter	Type	Default	Range or values
Mask Map Threshold	float	0	0 to 1
Luma Map Influence	float	0	0 to 1
Color Map Influence	float	0	0 to 1
X Count	int	4	1 to 128
Y Count	int	4	1 to 128
Size Mode	enum	normal	normal, keepRatio, absolute, pixel
Size X	float	1	0.01 to 10
Size Y	float	1	0.01 to 10
Size X Var	float	0	0 to 1
Size Y Var	float	0	0 to 1
Overall Scale	float	0.9	0 to 10
Overall Scale Var	float	0	0 to 1
Overall Pos Var	float	0	0 to 5
Rotation	float	0	0 to 1
Rotation Var	float	0	0 to 1
Pos X Var	float	0	0 to 5
Pos Y Var	float	0	0 to 5
Offset X	float	0	0 to 1
Offset Y	float	0	0 to 1
Lum Var	float	0	0 to 1
Hue Var	float	0	0 to 1
Blend Mode	enum	max	max, add, normal
Seed	int	12345	0 to 999999
Interpolation	enum	nearest	nearest, bilinear

Behavior

- Pattern nodes synthesize repeated structures from parameterized placement rules.
- Pattern outputs are often strongest when paired with mask-aware blends.

Common mistakes

- Over-randomization can make tiles hard to art direct.
- High tile counts can increase render cost in complex graphs.

Auto Levels

Use Auto Levels to quickly normalize tonal range before further grading or masking. It is useful early in a chain to recover contrast from flat or compressed inputs.

Availability: Free feature set

Common uses

- Place channel nodes after generators and before final output shaping.
- Use histogram and levels tools to isolate useful ranges for masks.

Input ports

- `input`
- Usually expects one image input, with optional secondary inputs for some nodes.

Output ports

- `out`
- Outputs modified image or channel data for downstream processing.

Parameters

This node has no shared processing parameters.

Behavior

- These nodes remap values rather than creating major new structure.
- Channel operations are useful for converting outputs into mask-friendly ranges.

Common mistakes

- Stacking many remap nodes can flatten contrast unexpectedly.
- Watch channel assumptions when mixing grayscale and RGBA signals.

Combine RGBA

Use Combine RGBA to pack independently authored channels into one texture output. It is commonly used for mask packing workflows and custom data-map assembly.

Availability: Free feature set

Common uses

- Place channel nodes after generators and before final output shaping.
- Use histogram and levels tools to isolate useful ranges for masks.

Input ports

- r
- g
- b
- a
- Usually expects one image input, with optional secondary inputs for some nodes.

Output ports

- out
- Outputs modified image or channel data for downstream processing.

Parameters

This node has no shared processing parameters.

Behavior

- These nodes remap values rather than creating major new structure.
- Channel operations are useful for converting outputs into mask-friendly ranges.

Common mistakes

- Stacking many remap nodes can flatten contrast unexpectedly.
- Watch channel assumptions when mixing grayscale and RGBA signals.

Combine HSB

Use Combine HSB to rebuild color from independently processed HSV channels. Colored channel inputs are converted to luminance, and missing inputs contribute zero.

Availability: Free feature set

Common uses

- Connect normalized grayscale masks to H, S, and B.
- Pair with Separate HSB for channel-specific color processing.

Input ports

- h
- s
- b
- H, S, and B inputs are normalized grayscale signals; missing inputs default to zero.

Output ports

- out
- Outputs an opaque RGB texture.

Parameters

This node has no shared processing parameters.

Behavior

- HSB uses standard HSV conversion with brightness equal to the maximum RGB channel.

Common mistakes

- Hue wraps across the black-to-white mask range.
- Alpha is not carried through; the output is opaque RGB.

Combine Alpha

Use Combine Alpha to pair color content with a separately authored opacity or mask signal. It is useful when transparency needs independent control from RGB edits.

Availability: Free feature set

Common uses

- Place channel nodes after generators and before final output shaping.
- Use histogram and levels tools to isolate useful ranges for masks.

Input ports

- color
- alpha
- Usually expects one image input, with optional secondary inputs for some nodes.

Output ports

- out
- Outputs modified image or channel data for downstream processing.

Parameters

This node has no shared processing parameters.

Behavior

- These nodes remap values rather than creating major new structure.
- Channel operations are useful for converting outputs into mask-friendly ranges.

Common mistakes

- Stacking many remap nodes can flatten contrast unexpectedly.
- Watch channel assumptions when mixing grayscale and RGBA signals.

Gradient (Dynamic)

Use Gradient (Dynamic) to art-direct value-to-color mapping with editable stops in context. It is often placed after grayscale generators to create stylized palette looks.

Availability: Pro

Common uses

- Place channel nodes after generators and before final output shaping.
- Use histogram and levels tools to isolate useful ranges for masks.

Input ports

- `input`
- `gradient`
- Usually expects one image input, with optional secondary inputs for some nodes.

Output ports

- `out`
- Outputs modified image or channel data for downstream processing.

Parameters

Parameter	Type	Default	Range or values
Orientation	enum	horizontal	horizontal, vertical
Gradient Pos	float	0.5	0 to 1

Behavior

- These nodes remap values rather than creating major new structure.
- Channel operations are useful for converting outputs into mask-friendly ranges.

Common mistakes

- Stacking many remap nodes can flatten contrast unexpectedly.
- Watch channel assumptions when mixing grayscale and RGBA signals.

Gradient Map

Use Gradient Map to recolor luminance data using a fixed ramp palette. It is effective for turning height or mask data into readable albedo-style previews.

Availability: Free feature set

Common uses

- Place channel nodes after generators and before final output shaping.
- Use histogram and levels tools to isolate useful ranges for masks.

Input ports

- `input`
- Usually expects one image input, with optional secondary inputs for some nodes.

Output ports

- `out`
- Outputs modified image or channel data for downstream processing.

Parameters

This node has no shared processing parameters.

Behavior

- These nodes remap values rather than creating major new structure.
- Channel operations are useful for converting outputs into mask-friendly ranges.

Common mistakes

- Stacking many remap nodes can flatten contrast unexpectedly.
- Watch channel assumptions when mixing grayscale and RGBA signals.

Grayscale

Use Grayscale to reduce color images to intensity data for mask-driven operations. It is often a prep step before thresholding, histogram selection, or gradient mapping.

Availability: Free feature set

Common uses

- Place channel nodes after generators and before final output shaping.
- Use histogram and levels tools to isolate useful ranges for masks.

Input ports

- `input`
- Usually expects one image input, with optional secondary inputs for some nodes.

Output ports

- `out`
- Outputs modified image or channel data for downstream processing.

Parameters

This node has no shared processing parameters.

Behavior

- These nodes remap values rather than creating major new structure.
- Channel operations are useful for converting outputs into mask-friendly ranges.

Common mistakes

- Stacking many remap nodes can flatten contrast unexpectedly.
- Watch channel assumptions when mixing grayscale and RGBA signals.

Histogram Range

Use Histogram Range to isolate a tonal window and stretch it into usable contrast. It helps extract targeted detail bands from noisy or broad-value sources.

Availability: Pro

Common uses

- Place channel nodes after generators and before final output shaping.
- Use histogram and levels tools to isolate useful ranges for masks.

Input ports

- `input`
- Usually expects one image input, with optional secondary inputs for some nodes.

Output ports

- `out`
- Outputs modified image or channel data for downstream processing.

Parameters

Parameter	Type	Default	Range or values
Range	float	0.5	0 to 1
Position	float	0.5	0 to 1

Behavior

- These nodes remap values rather than creating major new structure.
- Channel operations are useful for converting outputs into mask-friendly ranges.

Common mistakes

- Stacking many remap nodes can flatten contrast unexpectedly.
- Watch channel assumptions when mixing grayscale and RGBA signals.

Histogram Scan

Use Histogram Scan to sweep through value ranges and reveal specific luminance slices. It is useful for animated mask reveals and precise band selection.

Availability: Pro

Common uses

- Place channel nodes after generators and before final output shaping.
- Use histogram and levels tools to isolate useful ranges for masks.

Input ports

- `input`
- Usually expects one image input, with optional secondary inputs for some nodes.

Output ports

- `out`
- Outputs modified image or channel data for downstream processing.

Parameters

Parameter	Type	Default	Range or values
Position	float	0.5	0 to 1
Contrast	float	0.5	0 to 1
Invert Position	bool	false	-

Behavior

- These nodes remap values rather than creating major new structure.
- Channel operations are useful for converting outputs into mask-friendly ranges.

Common mistakes

- Stacking many remap nodes can flatten contrast unexpectedly.
- Watch channel assumptions when mixing grayscale and RGBA signals.

Histogram Select

Use Histogram Select to pick a narrow value band around a chosen center. It is commonly used to isolate material features for selective blending.

Availability: Pro

Common uses

- Place channel nodes after generators and before final output shaping.
- Use histogram and levels tools to isolate useful ranges for masks.

Input ports

- `input`
- Usually expects one image input, with optional secondary inputs for some nodes.

Output ports

- `out`
- Outputs modified image or channel data for downstream processing.

Parameters

Parameter	Type	Default	Range or values
Position	float	0.5	0 to 1
Range	float	0.5	0 to 1
Contrast	float	0	0 to 1

Behavior

- These nodes remap values rather than creating major new structure.
- Channel operations are useful for converting outputs into mask-friendly ranges.

Common mistakes

- Stacking many remap nodes can flatten contrast unexpectedly.
- Watch channel assumptions when mixing grayscale and RGBA signals.

HSL Adjustment

Use HSL Adjustment for fast hue, saturation, and lightness correction without rebuilding upstream content. It is ideal for palette iteration and final artistic tuning.

Availability: Free feature set

Common uses

- Place channel nodes after generators and before final output shaping.
- Use histogram and levels tools to isolate useful ranges for masks.

Input ports

- `input`
- Usually expects one image input, with optional secondary inputs for some nodes.

Output ports

- `out`
- Outputs modified image or channel data for downstream processing.

Parameters

Parameter	Type	Default	Range or values
Hue	float	0	-180 to 180
Saturation	float	1	0 to 2
Lightness	float	1	0 to 2

Behavior

- These nodes remap values rather than creating major new structure.
- Channel operations are useful for converting outputs into mask-friendly ranges.

Common mistakes

- Stacking many remap nodes can flatten contrast unexpectedly.
- Watch channel assumptions when mixing grayscale and RGBA signals.

Invert

Use Invert to flip channel polarity when a workflow expects opposite mask semantics. It is a quick fix for switching between cutout and keep regions.

Availability: Free feature set

Common uses

- Place channel nodes after generators and before final output shaping.
- Use histogram and levels tools to isolate useful ranges for masks.

Input ports

- `input`
- Usually expects one image input, with optional secondary inputs for some nodes.

Output ports

- `out`
- Outputs modified image or channel data for downstream processing.

Parameters

This node has no shared processing parameters.

Behavior

- These nodes remap values rather than creating major new structure.
- Channel operations are useful for converting outputs into mask-friendly ranges.

Common mistakes

- Stacking many remap nodes can flatten contrast unexpectedly.
- Watch channel assumptions when mixing grayscale and RGBA signals.

Levels

Use Levels for manual black, midtone, and white-point control with predictable grading behavior. It is typically used to tighten masks or rebalance contrast before export.

Availability: Free feature set

Common uses

- Place channel nodes after generators and before final output shaping.
- Use histogram and levels tools to isolate useful ranges for masks.

Input ports

- `input`
- Usually expects one image input, with optional secondary inputs for some nodes.

Output ports

- `out`
- Outputs modified image or channel data for downstream processing.

Parameters

Parameter	Type	Default	Range or values
Input Low	float	0	0 to 1
Input High	float	1	0 to 1
Gamma	float	1	0.01 to 5

Parameter	Type	Default	Range or values
Output Low	float	0	0 to 1
Output High	float	1	0 to 1
R Input Low	float	0	0 to 1
R Input High	float	1	0 to 1
R Gamma	float	1	0.01 to 5
R Output Low	float	0	0 to 1
R Output High	float	1	0 to 1
G Input Low	float	0	0 to 1
G Input High	float	1	0 to 1
G Gamma	float	1	0.01 to 5
G Output Low	float	0	0 to 1
G Output High	float	1	0 to 1
B Input Low	float	0	0 to 1
B Input High	float	1	0 to 1
B Gamma	float	1	0.01 to 5
B Output Low	float	0	0 to 1
B Output High	float	1	0 to 1
A Input Low	float	0	0 to 1
A Input High	float	1	0 to 1
A Gamma	float	1	0.01 to 5
A Output Low	float	0	0 to 1
A Output High	float	1	0 to 1

Behavior

- These nodes remap values rather than creating major new structure.
- Channel operations are useful for converting outputs into mask-friendly ranges.

Common mistakes

- Stacking many remap nodes can flatten contrast unexpectedly.
- Watch channel assumptions when mixing grayscale and RGBA signals.

Palette Extract

Use Palette Extract to derive dominant colors from an input image automatically. It provides a starting palette for stylization, quantization, or downstream remapping.

Availability: Pro

Common uses

- Place channel nodes after generators and before final output shaping.
- Use histogram and levels tools to isolate useful ranges for masks.

Input ports

- input
- Usually expects one image input, with optional secondary inputs for some nodes.

Output ports

- out
- Outputs modified image or channel data for downstream processing.

Parameters

Parameter	Type	Default	Range or values
Colors	int	8	2 to 64
Method	enum	kmeans	kmeans, medianCut, histogram
Color Space	enum	lab	rgb, lab
Priority	enum	frequency	frequency, vibrance, contrast
Sort	enum	luminance	none, luminance, hue

Behavior

- These nodes remap values rather than creating major new structure.
- Channel operations are useful for converting outputs into mask-friendly ranges.

Common mistakes

- Stacking many remap nodes can flatten contrast unexpectedly.
- Watch channel assumptions when mixing grayscale and RGBA signals.

Palettize

Use Palettize to constrain output to a limited color set for stylized or retro rendering. It is effective after gradient mapping or painting-style preprocessing.

Availability: Pro

Common uses

- Place channel nodes after generators and before final output shaping.
- Use histogram and levels tools to isolate useful ranges for masks.

Input ports

- image
- palette
- Usually expects one image input, with optional secondary inputs for some nodes.

Output ports

- out
- Outputs modified image or channel data for downstream processing.

Parameters

Parameter	Type	Default	Range or values
Dithering	enum	none	none, bayer2, bayer4, bayer8, noise, floydSteinberg, atkinson
Strength	float	0.5	0 to 1
Bayer Scale	int	1	1 to 16

Behavior

- These nodes remap values rather than creating major new structure.
- Channel operations are useful for converting outputs into mask-friendly ranges.

Common mistakes

- Stacking many remap nodes can flatten contrast unexpectedly.
- Watch channel assumptions when mixing grayscale and RGBA signals.

Quantize

Use Quantize to reduce continuous values into discrete steps and intentional banding. It is useful for toon shading, posterization, and compressed data-map outputs.

Availability: Free feature set

Common uses

- Place channel nodes after generators and before final output shaping.
- Use histogram and levels tools to isolate useful ranges for masks.

Input ports

- image
- curve
- Usually expects one image input, with optional secondary inputs for some nodes.

Output ports

- out
- Outputs modified image or channel data for downstream processing.

Parameters

Parameter	Type	Default	Range or values
Link Steps	bool	true	-
Quantize Red	int	8	1 to 256

Parameter	Type	Default	Range or values
Quantize Green	int	8	1 to 256
Quantize Blue	int	8	1 to 256
Quantize Alpha	int	8	1 to 256
Offset	float	0	-1 to 1
Slope	float	0	0 to 1
Slope Curve	enum	linear	linear, smoothstep, curveInput

Behavior

- These nodes remap values rather than creating major new structure.
- Channel operations are useful for converting outputs into mask-friendly ranges.

Common mistakes

- Stacking many remap nodes can flatten contrast unexpectedly.
- Watch channel assumptions when mixing grayscale and RGBA signals.

Separate RGBA

Use Separate RGBA to break a packed texture into editable per-channel outputs. It enables independent processing before optional recombination.

Availability: Free feature set

Common uses

- Place channel nodes after generators and before final output shaping.
- Use histogram and levels tools to isolate useful ranges for masks.

Input ports

- input
- Usually expects one image input, with optional secondary inputs for some nodes.

Output ports

- r
- g
- b
- a
- Outputs modified image or channel data for downstream processing.

Parameters

This node has no shared processing parameters.

Behavior

- These nodes remap values rather than creating major new structure.
- Channel operations are useful for converting outputs into mask-friendly ranges.

Common mistakes

- Stacking many remap nodes can flatten contrast unexpectedly.
- Watch channel assumptions when mixing grayscale and RGBA signals.

Separate HSB

Use Separate HSB to isolate HSV color properties for independent processing. Achromatic pixels emit hue zero, and brightness is the maximum RGB channel.

Availability: Free feature set

Common uses

- Process the three masks independently before recombining them with Combine HSB.

Input ports

- input
- Accepts grayscale, RGB, or RGBA image input.

Output ports

- h
- s
- b
- H is normalized hue, S is saturation, and B is HSV brightness/value.

Parameters

This node has no shared processing parameters.

Behavior

- Outputs normalized grayscale H, S, and B masks and discards source alpha.

Common mistakes

- Hue is circular, so values near black and white in the hue mask are adjacent colors.

Separate Alpha

Use Separate Alpha to isolate transparency from color for targeted edits. It is commonly used to clean masks while preserving RGB detail.

Availability: Free feature set

Common uses

- Place channel nodes after generators and before final output shaping.
- Use histogram and levels tools to isolate useful ranges for masks.

Input ports

- input
- Usually expects one image input, with optional secondary inputs for some nodes.

Output ports

- out
- Outputs modified image or channel data for downstream processing.

Parameters

This node has no shared processing parameters.

Behavior

- These nodes remap values rather than creating major new structure.
- Channel operations are useful for converting outputs into mask-friendly ranges.

Common mistakes

- Stacking many remap nodes can flatten contrast unexpectedly.
- Watch channel assumptions when mixing grayscale and RGBA signals.

Threshold

Use Threshold to convert grayscale into a hard binary mask around a cutoff. It is ideal for stencils, region gating, and crisp selection masks.

Availability: Free feature set

Common uses

- Place channel nodes after generators and before final output shaping.
- Use histogram and levels tools to isolate useful ranges for masks.

Input ports

- input
- Usually expects one image input, with optional secondary inputs for some nodes.

Output ports

- out
- Outputs modified image or channel data for downstream processing.

Parameters

Parameter	Type	Default	Range or values
Threshold	float	0.5	0 to 1

Behavior

- These nodes remap values rather than creating major new structure.
- Channel operations are useful for converting outputs into mask-friendly ranges.

Common mistakes

- Stacking many remap nodes can flatten contrast unexpectedly.
- Watch channel assumptions when mixing grayscale and RGBA signals.

Curves

Use Curves for fine non-linear tonal shaping when simple levels are not enough. It is effective for sculpting contrast response in masks and height-like inputs.

Availability: Free feature set

Common uses

- Place channel nodes after generators and before final output shaping.
- Use histogram and levels tools to isolate useful ranges for masks.

Input ports

- input
- Usually expects one image input, with optional secondary inputs for some nodes.

Output ports

- out
- Outputs modified image or channel data for downstream processing.

Parameters

This node has no shared processing parameters.

Behavior

- These nodes remap values rather than creating major new structure.
- Channel operations are useful for converting outputs into mask-friendly ranges.

Common mistakes

- Stacking many remap nodes can flatten contrast unexpectedly.
- Watch channel assumptions when mixing grayscale and RGBA signals.

Box Blur

Use Box Blur to soften harsh transitions and average local detail quickly. It is often used to precondition masks before blending or displacement.

Availability: Free feature set

Common uses

- Use blur before thresholding to smooth noisy masks.
- Use sharpen and highpass to recover or isolate detail bands.

Input ports

- input
- Requires an image input signal.

Output ports

- out
- Outputs a filtered image that preserves graph connectivity.

Parameters

Parameter	Type	Default	Range or values
Radius	float	2	0 to 10

Behavior

- Detail filters operate on neighboring pixels and are sensitive to resolution.
- Different blur and sharpen algorithms produce distinct edge behavior.

Common mistakes

- Large radii can wipe out fine structure.
- Repeated detail filters can create halos or ringing.

Directional Blur

Use Directional Blur to smear detail along a chosen axis for motion-like effects. It works well for streaking textures and anisotropic mask shaping.

Availability: Pro

Common uses

- Use blur before thresholding to smooth noisy masks.
- Use sharpen and highpass to recover or isolate detail bands.

Input ports

- input
- Requires an image input signal.

Output ports

- out
- Outputs a filtered image that preserves graph connectivity.

Parameters

Parameter	Type	Default	Range or values
Intensity	int	8	0 to 128
Angle	float	0	0 to 360

Behavior

- Detail filters operate on neighboring pixels and are sensitive to resolution.
- Different blur and sharpen algorithms produce distinct edge behavior.

Common mistakes

- Large radii can wipe out fine structure.
- Repeated detail filters can create halos or ringing.

Radial Blur

Use Radial Blur to push detail around or away from a center point. It is useful for spin, zoom, and lens-inspired distortion treatments.

Availability: Pro

Common uses

- Use blur before thresholding to smooth noisy masks.
- Use sharpen and highpass to recover or isolate detail bands.

Input ports

- input
- Requires an image input signal.

Output ports

- out
- Outputs a filtered image that preserves graph connectivity.

Parameters

Parameter	Type	Default	Range or values
Mode	enum	spin	spin, zoom
Intensity	float	0.35	0 to 2
Center X	float	0.5	0 to 1
Center Y	float	0.5	0 to 1
Samples	int	16	2 to 64

Behavior

- Detail filters operate on neighboring pixels and are sensitive to resolution.
- Different blur and sharpen algorithms produce distinct edge behavior.

Common mistakes

- Large radii can wipe out fine structure.
- Repeated detail filters can create halos or ringing.

Blur HQ

Use Blur HQ when smoother blur quality is needed with reduced sampling artifacts. It is suited to final passes where regular blur appears too coarse.

Availability: Pro

Common uses

- Use blur before thresholding to smooth noisy masks.
- Use sharpen and highpass to recover or isolate detail bands.

Input ports

- input
- Requires an image input signal.

Output ports

- out
- Outputs a filtered image that preserves graph connectivity.

Parameters

Parameter	Type	Default	Range or values
Strength	float	5	0 to 50
Quality	int	8	1 to 32
Direction	enum	both	horizontal, vertical, both

Behavior

- Detail filters operate on neighboring pixels and are sensitive to resolution.
- Different blur and sharpen algorithms produce distinct edge behavior.

Common mistakes

- Large radii can wipe out fine structure.
- Repeated detail filters can create halos or ringing.

Non-Uniform Blur

Use Non-Uniform Blur to vary blur intensity across an image using a Blur Map. It allows pushing the blur into directional shapes using Anisotropy and biasing the sampling direction using Asymmetry, replicating complex lens effects and directional smears.

Availability: Pro

Common uses

- Connect a mask to the Blur Map input to drive variable blur strength.
- Increase Blades to simulate polygonal aperture shapes.

Input ports

- input
- blurMap
- Requires an image input signal and optionally a grayscale Blur Map.

Output ports

- out
- Outputs a variably filtered image that preserves graph connectivity.

Parameters

Parameter	Type	Default	Range or values
Intensity	float	10	0 to 50
Anisotropy	float	0	0 to 1
Asymmetry	float	0	0 to 1
Angle	float	0	0 to 360
Samples	int	8	1 to 32
Blades	int	3	1 to 16

Behavior

- Detail filters operate on neighboring pixels and are sensitive to resolution.
- Performance scales with the product of Blades and Samples.

Common mistakes

- High sample counts heavily impact performance on high resolutions.
- Without a Blur Map, it acts as a uniform directional or radial blur depending on parameters.

Anisotropic Kuwahara

Use Anisotropic Kuwahara to smooth local detail in directionally coherent sectors. It computes or consumes a direction map and favors low-variance sectors, producing flattened brush-like regions that still follow the image structure.

Availability: Pro

Common uses

- Use a direction map input for explicit flow control.
- Increase anisotropy and radius for stronger painterly abstraction.

Input ports

- input
- anisotropyAngleMap
- slopeMap
- radiusMap
- directionMap
- Requires an image input signal; angle/slope/radius/direction maps are optional controls.

Output ports

- out
- Outputs a stylized filtered image that preserves graph connectivity.

Parameters

Parameter	Type	Default	Range or values
Radius	float	6	0 to 32
Smoothness	float	1	0 to 4
Sharpness	float	8	0 to 32
Anisotropy	float	0.75	0 to 1
Use Input Direction Map	bool	false	-
Tensor Smoothness	float	1	0 to 8
Anisotropy Angle	float	0	-1 to 1

Parameter	Type	Default	Range or values
Anisotropy Angle Map Multiplier	float	1	-4 to 4
Slope Map Input Multiplier	float	1	0 to 4
Ignore Alpha	bool	true	-

Behavior

- Direction-map steering still affects kernel orientation even when anisotropy is low.
- Radius map and angle map inputs can spatially vary blur footprint and direction.

Common mistakes

- Large radius and high sharpness can produce over-flattened regions.
- High resolutions with large kernels are computationally expensive.

Highpass

Use Highpass to isolate fine detail by subtracting broad low-frequency information. It is commonly used in sharpening and detail-overlay workflows.

Availability: Pro

Common uses

- Use blur before thresholding to smooth noisy masks.
- Use sharpen and highpass to recover or isolate detail bands.

Input ports

- input
- Requires an image input signal.

Output ports

- out
- Outputs a filtered image that preserves graph connectivity.

Parameters

Parameter	Type	Default	Range or values
Radius	float	10	0.1 to 100

Behavior

- Detail filters operate on neighboring pixels and are sensitive to resolution.
- Different blur and sharpen algorithms produce distinct edge behavior.

Common mistakes

- Large radii can wipe out fine structure.
- Repeated detail filters can create halos or ringing.

Channel Swizzle

Use Channel Swizzle to reorder, duplicate, clear, or fill channels without splitting and recombining a texture. Each output channel independently selects R, G, B, A, 0, or 1.

Availability: Free feature set

Common uses

- Use identity mapping for a pass-through RGBA texture.
- Map constants to alpha when an explicitly opaque or transparent output is needed.

Input ports

- `input`
- Accepts one grayscale, RGB, or RGBA texture.

Output ports

- `out`
- Outputs RGBA at the inherited resolution and bit depth.

Parameters

Parameter	Type	Default	Range or values
Output R	enum	r	r, g, b, a, zero, one
Output G	enum	g	r, g, b, a, zero, one
Output B	enum	b	r, g, b, a, zero, one
Output A	enum	a	r, g, b, a, zero, one

Behavior

- The output is always RGBA.
- Missing or invalid mappings normalize to identity defaults.

Common mistakes

- Channel Swizzle copies channels directly; it does not calculate luminance or weighted channel mixes.
- RGB and grayscale inputs without alpha are treated as fully opaque.

FXAA

FXAA applies a Fast Approximate Anti-Aliasing pass that detects local contrast edges and blends across them. It is particularly useful after Shape or Disc-like generators when their silhouettes look pixelated.

Availability: Pro

Common uses

- Place FXAA near the end of a graph after the shapes or masks whose edges need smoothing.
- Use it as a lightweight one-node finishing pass for aliasing rather than a general blur.

Input ports

- input
- Requires one Color or Grayscale image input.

Output ports

- out
- Outputs a Color or Grayscale image matching the input format.

Parameters

This node has no shared processing parameters.

Behavior

- The effect has no authored parameters and preserves the input channel format.
- Tiling controls whether edge samples wrap across texture boundaries.

Common mistakes

- FXAA can soften intentional one-pixel detail and sharp mask boundaries.
- It works from contrast, so low-contrast edges may receive little or no smoothing.

Median

Use Median to suppress speckle noise while keeping key edges more stable than blur. It is useful for cleanup before edge detection or thresholding.

Availability: Pro

Common uses

- Use blur before thresholding to smooth noisy masks.
- Use sharpen and highpass to recover or isolate detail bands.

Input ports

- input
- Requires an image input signal.

Output ports

- out
- Outputs a filtered image that preserves graph connectivity.

Parameters

Parameter	Type	Default	Range or values
Radius	int	1	1 to 5

Behavior

- Detail filters operate on neighboring pixels and are sensitive to resolution.
- Different blur and sharpen algorithms produce distinct edge behavior.

Common mistakes

- Large radii can wipe out fine structure.
- Repeated detail filters can create halos or ringing.

Sharpen

Use Sharpen to increase local edge contrast after softening or resampling operations. It helps recover crispness in final texture outputs.

Availability: Free feature set

Common uses

- Use blur before thresholding to smooth noisy masks.
- Use sharpen and highpass to recover or isolate detail bands.

Input ports

- input
- Requires an image input signal.

Output ports

- out
- Outputs a filtered image that preserves graph connectivity.

Parameters

Parameter	Type	Default	Range or values
Intensity	float	1	0 to 4
Algorithm	enum	unsharp	unsharp, laplacian, highBoost

Behavior

- Detail filters operate on neighboring pixels and are sensitive to resolution.
- Different blur and sharpen algorithms produce distinct edge behavior.

Common mistakes

- Large radii can wipe out fine structure.
- Repeated detail filters can create halos or ringing.

Slope Blur

Use Slope Blur to drag pixels along a gradient field for directional erosion effects. It is frequently used with noise or height inputs to create natural breakup.

Availability: Pro

Common uses

- Use blur before thresholding to smooth noisy masks.
- Use sharpen and highpass to recover or isolate detail bands.

Input ports

- input
- slope
- Requires an image input signal.

Output ports

- out
- Outputs a filtered image that preserves graph connectivity.

Parameters

Parameter	Type	Default	Range or values
Samples	float	8	0 to 32
Intensity	float	8	0 to 16
Mode	enum	blur	blur, min, max

Behavior

- Detail filters operate on neighboring pixels and are sensitive to resolution.
- Different blur and sharpen algorithms produce distinct edge behavior.

Common mistakes

- Large radii can wipe out fine structure.
- Repeated detail filters can create halos or ringing.

Coordinates Transform

Use Coordinates Transform to remap sampling space with explicit scale, rotation, and offset control. It is a foundational step for repositioning patterns non-destructively.

Availability: Pro

Common uses

- Apply distortions after establishing a stable base pattern.
- Use mild distortion early and stronger effects near the end for control.

Input ports

- input
- Typically expects an image input plus optional control inputs.

Output ports

- out
- Provides transformed image output plus a grayscale mask output for downstream seam-aware blending.

Parameters

Parameter	Type	Default	Range or values
From	enum	cartesian	cartesian, polar, logPolar
To	enum	polar	cartesian, polar, logPolar
Center X	float	0.5	0 to 1
Center Y	float	0.5	0 to 1
Angle Offset	float	0	-360 to 360
Radius Scale	float	1	0.01 to 4
Log Exponent	float	4	0.1 to 10

Behavior

- Distortion nodes remap sampling coordinates rather than direct color values.
- Input continuity strongly influences final distortion quality.

Common mistakes

- Aggressive warps can stretch artifacts and expose seams.
- Coordinate-heavy chains can become hard to debug without checkpoints.

Directional Warp

Use Directional Warp to displace content along a controlled vector direction map. It is useful for flow-like deformation and directional wear patterns.

Availability: Pro

Common uses

- Apply distortions after establishing a stable base pattern.
- Use mild distortion early and stronger effects near the end for control.

Input ports

- input
- warpMap
- Typically expects an image input plus optional control inputs.

Output ports

- out
- Outputs transformed image data in the same graph branch.

Parameters

Parameter	Type	Default	Range or values
Intensity	float	0.1	0 to 1
Angle	float	0	0 to 360
Non-Square Expansion	bool	true	-

Behavior

- Distortion nodes remap sampling coordinates rather than direct color values.
- Input continuity strongly influences final distortion quality.

Common mistakes

- Aggressive warps can stretch artifacts and expose seams.
- Coordinate-heavy chains can become hard to debug without checkpoints.

Multi Directional Warp

Use Multi Directional Warp to apply complex directional displacement in 2, 4, or 8 directions simultaneously. Results are blended using Average, Max, or Min modes, or chained sequentially for extreme distortion. It is ideal for breaking up rigid shapes and creating ‘stay-in-place’ noise breakup.

Availability: Pro

Common uses

- Use ‘Average’ mode for subtle, centered breakup that doesn’t drift the texture.
- Use ‘Chain’ mode for organic, liquid-like distortions and heavy smearing.
- Combine with high Intensity values for marble-like appearances.

Input ports

- input
- intensityInput
- Requires a primary image input and a mandatory grayscale intensity map.

Output ports

- out
- Outputs the multi-directionally warped composite image.

Parameters

Parameter	Type	Default	Range or values
Intensity	float	0.1	0 to 1
Angle	float	0	0 to 360
Mode	enum	average	average, max, min, chain
Directions	enum	1	1, 2, 4
Non-Square Expansion	bool	true	-

Behavior

- Directions 1 applies warp at ‘angle’ and ‘angle + 180’.
- Directions 2 adds ‘angle + 90’ and ‘angle + 270’.
- Directions 4 adds 45-degree intermediate steps for a total of 8 directions.

Common mistakes

- High direction counts (8 directions) combined with large intensity can lead to significant blurring in ‘Average’ mode.
- Chain mode is noticeably more expensive than other modes as it cannot be performed in a single parallel accumulation.

Kaleidoscope

Use Kaleidoscope to build radial symmetry from a source wedge. It is effective for ornamental motifs, decals, and mandala-like structures.

Availability: Free feature set

Common uses

- Apply distortions after establishing a stable base pattern.
- Use mild distortion early and stronger effects near the end for control.

Input ports

- input
- Typically expects an image input plus optional control inputs.

Output ports

- out
- Outputs transformed image data in the same graph branch.

Parameters

Parameter	Type	Default	Range or values
Mode	enum	mirror	mirror, rotate, pinwheel
Slices	int	6	2 to 32
Rotation	float	0	0 to 360
Scale	float	1	0.1 to 4
Center X	float	0.5	0 to 1
Center Y	float	0.5	0 to 1
Offset X	float	0	-1 to 1
Offset Y	float	0	-1 to 1
Repeat	int	1	1 to 8
Twist	float	0	0 to 4

Behavior

- Distortion nodes remap sampling coordinates rather than direct color values.
- Input continuity strongly influences final distortion quality.

Common mistakes

- Aggressive warps can stretch artifacts and expose seams.
- Coordinate-heavy chains can become hard to debug without checkpoints.

Mirror

Use Mirror to enforce bilateral or axis symmetry for seamless designs. It is often applied before detail layering to keep both sides coherent.

Availability: Free feature set

Common uses

- Apply distortions after establishing a stable base pattern.
- Use mild distortion early and stronger effects near the end for control.

Input ports

- `input`
- Typically expects an image input plus optional control inputs.

Output ports

- `out`
- Outputs transformed image data in the same graph branch.

Parameters

Parameter	Type	Default	Range or values
Mode	enum	axisX	axisX, axisY, corner
Axis X Offset	float	0.5	0 to 1
Axis Y Offset	float	0.5	0 to 1

Parameter	Type	Default	Range or values
Invert Axis X	bool	false	-
Invert Axis Y	bool	false	-
Corner Type	enum	topLeft	topLeft, topRight, bottomLeft, bottomRight

Behavior

- Distortion nodes remap sampling coordinates rather than direct color values.
- Input continuity strongly influences final distortion quality.

Common mistakes

- Aggressive warps can stretch artifacts and expose seams.
- Coordinate-heavy chains can become hard to debug without checkpoints.

Quick Tile

Use Quick Tile to repeat an input reliably with controls aimed at tileable outputs. It is a fast route to seamless patterns before additional variation passes.

Availability: Pro

Common uses

- Apply distortions after establishing a stable base pattern.
- Use mild distortion early and stronger effects near the end for control.

Input ports

- `input`
- Typically expects an image input plus optional control inputs.

Output ports

- `out`
- `mask`
- Outputs transformed image data in the same graph branch.

Parameters

Parameter	Type	Default	Range or values
Mask Warping H	float	0	-100 to 100
Mask Warping V	float	0	-100 to 100
Mask Size H	float	0.15	0 to 1
Mask Size V	float	0.15	0 to 1
Mask Precision H	float	0.5	0 to 1
Mask Precision V	float	0.5	0 to 1

Behavior

- Distortion nodes remap sampling coordinates rather than direct color values.
- Input continuity strongly influences final distortion quality.

Common mistakes

- Aggressive warps can stretch artifacts and expose seams.
- Coordinate-heavy chains can become hard to debug without checkpoints.

Clone Stamp

Use Clone Stamp to remove defects or repeated scan artifacts by copying a clean source region into a target area. An optional brush mask can shape the copied region.

Availability: Free feature set

Common uses

- Pick a source point with similar texture direction and lighting.
- Use a soft brush or mask for less visible repairs.

Input ports

- input
- brushMask
- Requires an image input. Brush Mask is optional and is sampled as grayscale influence.

Output ports

- out
- Outputs the repaired image using the input image format.

Parameters

Parameter	Type	Default	Range or values
Source X	float	0.25	0 to 1
Source Y	float	0.5	0 to 1
Target X	float	0.75	0 to 1
Target Y	float	0.5	0 to 1
Brush Size	float	1	0.01 to 1
Hardness	float	0.75	0 to 1
Opacity	float	1	0 to 1

Behavior

- Copies source pixels by target-relative offset and blends them through brush falloff, opacity, and optional mask input.

Common mistakes

- Large hard brushes can create obvious repeated patches.
- Copying from mismatched texture direction can leave visible repair marks.

Quad Transform

Use Quad Transform to project an image into an arbitrary four-corner shape. It is useful for perspective correction and decal placement tasks.

Availability: Pro

Common uses

- Apply distortions after establishing a stable base pattern.
- Use mild distortion early and stronger effects near the end for control.

Input ports

- input
- Typically expects an image input plus optional control inputs.

Output ports

- out
- Outputs transformed image data in the same graph branch.

Parameters

Parameter	Type	Default	Range or values
P00	vec2	0,0	-
P01	vec2	0,1	-
P10	vec2	1,0	-
P11	vec2	1,1	-
Culling	enum	frontOnly	frontOnly, backOnly, frontOverBack, backOverFront
Background	float	0	0 to 1
Sampling	enum	bilinear	bilinear, nearest

Behavior

- Distortion nodes remap sampling coordinates rather than direct color values.
- Input continuity strongly influences final distortion quality.

Common mistakes

- Aggressive warps can stretch artifacts and expose seams.
- Coordinate-heavy chains can become hard to debug without checkpoints.

Skew

Use Skew to shear geometry for angled perspective and directional bias. It is commonly used to fake tilt before further distortion.

Availability: Free feature set

Common uses

- Apply distortions after establishing a stable base pattern.
- Use mild distortion early and stronger effects near the end for control.

Input ports

- input
- Typically expects an image input plus optional control inputs.

Output ports

- out
- Outputs transformed image data in the same graph branch.

Parameters

Parameter	Type	Default	Range or values
Axis	enum	horizontal	horizontal, vertical
Amount	float	0	-1 to 1
Align	enum	center	center, topLeft, bottomRight

Behavior

- Distortion nodes remap sampling coordinates rather than direct color values.
- Input continuity strongly influences final distortion quality.

Common mistakes

- Aggressive warps can stretch artifacts and expose seams.
- Coordinate-heavy chains can become hard to debug without checkpoints.

Trapezoid Transform

Use Trapezoid Transform to taper one side and simulate perspective squeeze. It is helpful for signage, panel, and projection-style adjustments.

Common uses

- Apply distortions after establishing a stable base pattern.
- Use mild distortion early and stronger effects near the end for control.

Input ports

- input
- Typically expects an image input plus optional control inputs.

Output ports

- out
- Outputs transformed image data in the same graph branch.

Parameters

Parameter	Type	Default	Range or values
Top Stretch	float	1	0 to 1
Bottom Stretch	float	1	0 to 1
Sampling	enum	bilinear	bilinear, nearest

Behavior

- Distortion nodes remap sampling coordinates rather than direct color values.
- Input continuity strongly influences final distortion quality.

Common mistakes

- Aggressive warps can stretch artifacts and expose seams.
- Coordinate-heavy chains can become hard to debug without checkpoints.

Transform

Use Transform for general translate, rotate, and scale adjustments in a single node. It is the standard positioning utility for most graph branches.

Availability: Free feature set

Common uses

- Apply distortions after establishing a stable base pattern.
- Use mild distortion early and stronger effects near the end for control.

Input ports

- input
- Typically expects an image input plus optional control inputs.

Output ports

- out
- Outputs transformed image data in the same graph branch.

Parameters

Parameter	Type	Default	Range or values
Offset X	float	0	-10 to 10
Offset Y	float	0	-10 to 10
Rotation	float	0	0 to 360
Scale	float	1	-5 to 5
Scale X	float	1	-5 to 5
Scale Y	float	1	-5 to 5
Repetitions	int	1	1 to 20
Wrap Tiling	bool	true	-
Affect Alpha	bool	false	-

Behavior

- Distortion nodes remap sampling coordinates rather than direct color values.
- Input continuity strongly influences final distortion quality.

Common mistakes

- Aggressive warps can stretch artifacts and expose seams.
- Coordinate-heavy chains can become hard to debug without checkpoints.

Safe Transform

Use Safe Transform when a tiled texture must stay seamless while offsets and rotation remain crisp. It wraps sampling on both axes, snaps offsets to whole pixels, and always constrains rotation to quarter turns for exact texel preservation.

Availability: Free feature set

Common uses

- Use Repetitions to tile the input down before transforming it.
- Rotation is always limited to 0°, 90°, 180°, or 270° for exact tile continuity.

Input ports

- `input`
- Accepts any image input and preserves its output format.

Output ports

- `out`
- Produces a same-size, seamless image using nearest-neighbor sampling.

Parameters

Parameter	Type	Default	Range or values
Offset X	float	0	-1 to 1
Offset Y	float	0	-1 to 1
Rotation	float	0	0 to 270
Scale	float	1	-4 to 4
Scale X	float	1	-4 to 4
Scale Y	float	1	-4 to 4
Repetitions	int	1	1 to 20

Behavior

- Offsets are rounded to source pixels.
- The node always wraps both axes so its output remains tileable.

Common mistakes

- Nearest-neighbor sampling deliberately keeps hard pixel edges instead of smoothing them.

Image Sample

Image Sample is a low-level UV remapping primitive. Red supplies U and green supplies V for every output pixel, while addressing and filtering determine how the source image is read.

Availability: Pro

Common uses

- Build coordinate maps with red for horizontal U and green for vertical V.
- Use Bilinear for smooth transforms and Nearest for masks, atlases, or pixel art.
- Choose OpenGL when the coordinate map stores V upward instead of downward.

Input ports

- `image`
- `coordinates`
- Image accepts grayscale, RGB, or RGBA data.
- Coordinates expects U in red and V in green; when absent, an identity UV grid is generated.

Output ports

- `out`
- Preserves the Image input pixel format and supports native 8-bit and 16-bit processing.

Parameters

Parameter	Type	Default	Range or values
Address Mode	enum	repeat	repeat, clamp, mirror
Filtering	enum	bilinear	bilinear, nearest
Coordinate Format	enum	directx	directx, opengl

Behavior

- Coordinates controls the default output dimensions; Image controls output format and bit depth.
- Repeat, Clamp, and Mirror addressing are applied independently to U and V.

Common mistakes

- Blue and alpha in the Coordinates input are ignored.
- Repeat maps integer UV boundaries back to the start of the texture.

Vector Warp

Vector Warp uses a color vector map instead of a grayscale driver. Red controls horizontal movement and green controls vertical movement around a neutral 50% value, which makes it suitable for flow maps, normal-map-derived vectors, and authored directional deformation.

Availability: Pro

Common uses

- Use RGBA Merge or a normal-map conversion to build a vector map.
- Keep red and green near 50% where no displacement is wanted.
- Switch Vector Format when the source flow or normal map uses the opposite green-channel convention.

Input ports

- `input`
- `vectorMap`
- Input is the image to distort.
- Vector Map is a color map with 50% red/green as the neutral vector.

Output ports

- out
- Preserves the primary input image format and dimensions.

Parameters

Parameter	Type	Default	Range or values
Intensity	float	0.1	0 to 1
Non-Square Expansion	bool	true	-
Vector Format	enum	directx	directx, opengl

Behavior

- Only the red (X) and green (Y) channels are sampled.
- DirectX and OpenGL formats differ only by their green-channel direction.

Common mistakes

- Blue and alpha do not affect the displacement.
- High intensity can stretch detail and reveal non-tiled edges when tiling is disabled.

Warp

Use Warp to bend sampling coordinates with a warp map or procedural driver. It breaks repetition and adds organic deformation to otherwise rigid patterns.

Availability: Free feature set

Common uses

- Apply distortions after establishing a stable base pattern.
- Use mild distortion early and stronger effects near the end for control.

Input ports

- input
- warpMap
- Typically expects an image input plus optional control inputs.

Output ports

- out
- Outputs transformed image data in the same graph branch.

Parameters

Parameter	Type	Default	Range or values
Intensity	float	0.1	0 to 1
Non-Square Expansion	bool	true	-

Behavior

- Distortion nodes remap sampling coordinates rather than direct color values.
- Input continuity strongly influences final distortion quality.

Common mistakes

- Aggressive warps can stretch artifacts and expose seams.
- Coordinate-heavy chains can become hard to debug without checkpoints.

Swirl

Use Swirl to create angular rotational distortion around a configurable center. The twist is strongest near the center and falls off with distance, producing smooth spiral effects useful for water, galaxy, and organic form deformation.

Availability: Pro

Common uses

- Lower intensities produce subtle twisting; high intensities create tight spirals.
- Adjust center point to offset the swirl origin from the canvas center.

Input ports

- input
- Requires an image input to distort.

Output ports

- out
- Outputs the swirl-distorted image with the same dimensions.

Parameters

Parameter	Type	Default	Range or values
Intensity	float	0.25	0 to 10
Center X	float	0.5	0 to 1
Center Y	float	0.5	0 to 1
Direction	enum	cw	cw, ccw

Behavior

- Swirl applies a radial rotation offset proportional to distance from center.
- The twist angle is $intensity * (1 - distance_from_center / max_distance)$.

Common mistakes

- Very high intensity can produce extreme twisting at the center that may alias.
- Tiling may introduce seam artifacts at the edges with strong swirl.

Spherize

Use Spherize to create spherical distortion effects, pushing pixels outward for a bulge effect or pulling them inward for a pinch effect. Positive intensity values expand (bulge), negative values pinch (contract). Ideal for stylized warp effects and lens simulation.

Availability: Pro

Common uses

- Positive intensity bulges the image outward; negative intensity pinches inward.
- Combine with masks to apply spherical distortion only to specific regions.

Input ports

- input
- Requires an image input to distort.

Output ports

- out
- Outputs the spherized image with the same dimensions.

Parameters

Parameter	Type	Default	Range or values
Intensity	float	0.25	-10 to 10
Center X	float	0.5	0 to 1
Center Y	float	0.5	0 to 1

Behavior

- Spherize remaps UVs using a spherical projection function centered at the given point.
- Intensity controls the radius of the spherical effect with negative values inverting the direction.

Common mistakes

- High intensity values can produce extreme stretching at the edges.
- Tiling may cause visible seams when used with strong spherical distortion.

Bevel

Use Bevel to turn masks into height and normal-style edge relief data. It is commonly used to add chamfers to decals, trims, and hard-surface details.

Availability: Pro

Common uses

- Use analysis outputs as masks or support maps in blend chains.
- Pair with levels or threshold nodes to isolate useful map ranges.

Input ports

- input
- curve
- Usually expects a structured grayscale or height-like input.

Output ports

- height
- normal
- Outputs one or more analysis maps depending on node type.

Parameters

Parameter	Type	Default	Range or values
Distance	float	0.5	-1 to 1
Corner Type	enum	round	round, angular
Smoothing	float	0	0 to 5
Use Non-Uniform Blur	bool	false	-
Use Custom Curve	bool	false	-
Normal Intensity	float	5	0 to 50
Normal Format	enum	opengl	opengl, directx

Behavior

- These nodes compute derived data from shape and height information.
- Outputs are often best consumed by downstream utility and mixing nodes.

Common mistakes

- Analysis nodes can be sensitive to noisy or low-contrast inputs.
- Map interpretation changes with downstream color-space assumptions.

Curvature

Use Curvature to extract convex and concave response from shape or height information. It is valuable for smart masks such as edge wear and cavity dirt.

Availability: Pro

Common uses

- Use analysis outputs as masks or support maps in blend chains.
- Pair with levels or threshold nodes to isolate useful map ranges.

Input ports

- input
- Usually expects a structured grayscale or height-like input.

Output ports

- out
- Outputs one or more analysis maps depending on node type.

Parameters

Parameter	Type	Default	Range or values
Radius	int	1	1 to 8
Intensity	float	1	0 to 5
Smoothness	int	0	0 to 10
Invert	bool	false	-
Mode	enum	both	both, convex, concave

Behavior

- These nodes compute derived data from shape and height information.
- Outputs are often best consumed by downstream utility and mixing nodes.

Common mistakes

- Analysis nodes can be sensitive to noisy or low-contrast inputs.
- Map interpretation changes with downstream color-space assumptions.

Distance

Use Distance to compute distance fields from features for controlled falloffs. It is useful for outlines, expansion masks, and procedural borders.

Availability: Pro

Common uses

- Use analysis outputs as masks or support maps in blend chains.
- Pair with levels or threshold nodes to isolate useful map ranges.

Input ports

- mask
- source
- Usually expects a structured grayscale or height-like input.

Output ports

- out
- Outputs one or more analysis maps depending on node type.

Parameters

Parameter	Type	Default	Range or values
Max Distance	float	0.25	0 to 1
Distance	enum	euclidean	euclidean, manhattan, chebyshev

Parameter	Type	Default	Range or values
Combine	enum	combine	combine, source

Behavior

- These nodes compute derived data from shape and height information.
- Outputs are often best consumed by downstream utility and mixing nodes.

Common mistakes

- Analysis nodes can be sensitive to noisy or low-contrast inputs.
- Map interpretation changes with downstream color-space assumptions.

Dilate / Erode

Use Dilate / Erode to expand white mask regions with positive amounts or contract them with negative amounts. Distance mode gives hard binary morphology; Blur mode uses a softened neighborhood method before restoring a binary mask.

Availability: Pro

Common uses

- Connect a binary or thresholded grayscale mask to Input.
- Use positive values to dilate and negative values to erode.

Input ports

- input
- Expects a grayscale binary mask on the Input port.

Output ports

- out
- Outputs a binary grayscale mask on Out.

Parameters

Parameter	Type	Default	Range or values
Amount	float	0	-64 to 64
Blur Scale	float	0	-64 to 64
Mode	enum	distance	distance, blur

Behavior

- The signed amount controls direction and radius.
- Distance and Blur modes share the same signed amount contract and have CPU/GPU implementations.

Common mistakes

- Non-binary inputs are thresholded at 50%.
- Large radii can be expensive on the CPU path.

Edge Detect

Use Edge Detect to isolate luminance boundaries as clean line masks. It is often paired with blur or threshold for stylized contour effects.

Availability: Pro

Common uses

- Use analysis outputs as masks or support maps in blend chains.
- Pair with levels or threshold nodes to isolate useful map ranges.

Input ports

- input
- Usually expects a structured grayscale or height-like input.

Output ports

- out
- Outputs one or more analysis maps depending on node type.

Parameters

Parameter	Type	Default	Range or values
Edge Width	float	1	1 to 16
Edge Roundness	float	0	0 to 16
Tolerance	float	0.05	0 to 1
Invert	bool	false	-

Behavior

- These nodes compute derived data from shape and height information.
- Outputs are often best consumed by downstream utility and mixing nodes.

Common mistakes

- Analysis nodes can be sensitive to noisy or low-contrast inputs.
- Map interpretation changes with downstream color-space assumptions.

Emboss

Use Emboss to simulate directional relief shading from grayscale detail. It helps preview form depth and generate stamped surface looks.

Availability: Pro

Common uses

- Use analysis outputs as masks or support maps in blend chains.
- Pair with levels or threshold nodes to isolate useful map ranges.

Input ports

- `input`
- Usually expects a structured grayscale or height-like input.

Output ports

- out
- Outputs one or more analysis maps depending on node type.

Parameters

Parameter	Type	Default	Range or values
Azimuth	float	0	0 to 360
Elevation	float	45	0 to 90
Amount	float	1	0 to 5

Behavior

- These nodes compute derived data from shape and height information.
- Outputs are often best consumed by downstream utility and mixing nodes.

Common mistakes

- Analysis nodes can be sensitive to noisy or low-contrast inputs.
- Map interpretation changes with downstream color-space assumptions.

Flood Fill

Use Flood Fill to identify connected regions and output per-island metadata. It is useful for region-based masking and procedural selection workflows.

Availability: Pro

Common uses

- Use analysis outputs as masks or support maps in blend chains.
- Pair with levels or threshold nodes to isolate useful map ranges.

Input ports

- input
- Usually expects a structured grayscale or height-like input.

Output ports

- bbox
- info
- Outputs one or more analysis maps depending on node type.

Parameters

Parameter	Type	Default	Range or values
Wrap Edges	bool	true	-

Behavior

- These nodes compute derived data from shape and height information.
- Outputs are often best consumed by downstream utility and mixing nodes.

Common mistakes

- Analysis nodes can be sensitive to noisy or low-contrast inputs.
- Map interpretation changes with downstream color-space assumptions.

Flood Fill Random Grayscale

Use Flood Fill Random Grayscale to assign unique random values to each connected island. It creates quick variation masks across repeated regions.

Availability: Pro

Common uses

- Use analysis outputs as masks or support maps in blend chains.
- Pair with levels or threshold nodes to isolate useful map ranges.

Input ports

- input
- Usually expects a structured grayscale or height-like input.

Output ports

- out
- Outputs one or more analysis maps depending on node type.

Parameters

Parameter	Type	Default	Range or values
Seed	int	0	0 to 999999

Behavior

- These nodes compute derived data from shape and height information.
- Outputs are often best consumed by downstream utility and mixing nodes.

Common mistakes

- Analysis nodes can be sensitive to noisy or low-contrast inputs.
- Map interpretation changes with downstream color-space assumptions.

Flood Fill to Grayscale/Color

Use Flood Fill to Grayscale/Color with Flood Fill’s Info output and a grayscale or color map. It samples the base map at each region’s bounds center, fills that region with the sampled value, then applies independent stable luminance and hue adjustment/variation.

Availability: Pro

Common uses

- Connect Flood Fill’s Info output to Flood Fill Info and a base grayscale or color map to Grayscale/Color Input.
- Luminance changes brightness; Color controls rotate hue. All four controls default to zero, so regions retain their sampled input-map values by default.

Input ports

- floodFill
- input
- Flood Fill Info expects the Info output from Flood Fill.
- Grayscale/Color Input supplies the per-region base tones.

Output ports

- out
- Outputs one grayscale or color swatch per flood-filled region.

Parameters

Parameter	Type	Default	Range or values
Luminance Adjustment	float	0	-1 to 1
Luminance Random	float	0	-1 to 1
Color Adjustment	float	0	-1 to 1
Color Random	float	0	-1 to 1

Behavior

- Every connected flood-filled region receives a uniform result.
- Output format follows the Grayscale/Color Input so grayscale maps remain grayscale and color maps retain color.

Common mistakes

- The Flood Fill Info input is required; the node cannot derive regions from a plain mask.
- Very large positive or negative luminance values can clamp colors to black or white.

Flood Fill to BBox Size

Use Flood Fill to BBox Size with Flood Fill's BBox output to measure each region relative to the canvas. It can output width, height, or the larger of the two dimensions.

Availability: Pro

Common uses

- Connect Flood Fill's BBox output to this node.
- Use the Max mode for a single size mask that catches wide or tall regions.

Input ports

- input
- BBox Map expects the BBox output from Flood Fill.

Output ports

- out
- Outputs a grayscale map with one size value per flood-filled region.

Parameters

Parameter	Type	Default	Range or values
Output	enum	max	max, x, y

Behavior

- X reads the BBox width, Y reads the BBox height, and Max returns their larger value.
- GPU acceleration uses a single shader pass and falls back to the matching CPU calculation.

Common mistakes

- Values are relative to the entire canvas, so small regions can have low contrast.
- The node expects Flood Fill BBox data, not its Info output.

Flood Fill To Index

Use Flood Fill To Index with Flood Fill's Info output to assign every detected region an index, ordered from the top-left scan direction.

Availability: Pro

Common uses

- Connect Flood Fill Info to this node.
- Normalized output is best for ordinary grayscale workflows.

Input ports

- input
- Flood Fill Info expects the Info output from Flood Fill.

Output ports

- out
- Outputs a grayscale region-index map.

Parameters

Parameter	Type	Default	Range or values
Output	enum	normalized	normalized, integer

Behavior

- Normalized mode maps region indices from 0 to 1.
- Integer mode emits the zero-based index clamped to 8-bit.

Common mistakes

- Integer mode is clamped to the editor’s current 8-bit output range.
- The app uses Flood Fill Info rather than Substance’s BBox/Special Shape Info pair.

Flood Fill to Gradient

Use Flood Fill to Gradient to generate directional ramps inside segmented regions. It is effective for per-island shading and directional blend control.

Availability: Pro

Common uses

- Use analysis outputs as masks or support maps in blend chains.
- Pair with levels or threshold nodes to isolate useful map ranges.

Input ports

- input
- info
- angleInput
- slopeInput
- Usually expects a structured grayscale or height-like input.

Output ports

- out
- Outputs one or more analysis maps depending on node type.

Parameters

Parameter	Type	Default	Range or values
Seed	int	0	0 to 999999
Angle	float	0	0 to 1
Angle Variation	float	0	0 to 1
Multiply by BBox Size	float	0	0 to 1
Angle Input Mult	float	0	0 to 1
Slope Input Mult	float	0	0 to 1
Slope Intensity	float	1	0 to 1
Flat Slope Color	float	0	0 to 1

Behavior

- These nodes compute derived data from shape and height information.
- Outputs are often best consumed by downstream utility and mixing nodes.

Common mistakes

- Analysis nodes can be sensitive to noisy or low-contrast inputs.
- Map interpretation changes with downstream color-space assumptions.

Ambient Occlusion

Use Ambient Occlusion to estimate contact shadowing from height-like inputs. It adds depth cues that improve readability in previews and material maps.

Availability: Free feature set

Common uses

- Use analysis outputs as masks or support maps in blend chains.
- Pair with levels or threshold nodes to isolate useful map ranges.

Input ports

- input
- Usually expects a structured grayscale or height-like input.

Output ports

- out
- Outputs one or more analysis maps depending on node type.

Parameters

Parameter	Type	Default	Range or values
Radius	float	10	1 to 50
Strength	float	1	0 to 10
Bias	float	0.1	0 to 1
Samples	int	16	4 to 128
Noise	float	1	0 to 1

Behavior

- These nodes compute derived data from shape and height information.
- Outputs are often best consumed by downstream utility and mixing nodes.

Common mistakes

- Analysis nodes can be sensitive to noisy or low-contrast inputs.
- Map interpretation changes with downstream color-space assumptions.

Auto Crop

Auto Crop finds non-black content bounds and recenters the result while preserving aspect ratio.

Availability: Pro

Common uses

- Use after generating isolated shapes or decals.

Input ports

- input
- Accepts grayscale or color input.

Output ports

- out
- Preserves the input format.

Parameters

This node has no shared processing parameters.

Behavior

- The current implementation uses fit-keep-ratio behavior.

Common mistakes

- Transparent color images use non-black color or non-opaque alpha as content.

Normal Combine

Layers a detail normal over a base normal without treating normal vectors as ordinary colors.

Availability: Pro

Common uses

- Connect base normal to Normal 1 and detail normal to Normal 2.

Input ports

- normal1
- normal2
- Expects two RGB normal maps.

Output ports

- out
- Outputs an RGB normal map.

Parameters

Parameter	Type	Default	Range or values
Technique	enum	detail	whiteout, channelMixer, detail

Behavior

- Whiteout is fast; Channel Mixer and Detail Oriented are higher quality.

Common mistakes

- Use matching normal-map orientations.

Normal Map

Use Normal Map to convert height information into tangent-space lighting vectors. It is a core step for producing game-ready normal textures.

Availability: Free feature set

Common uses

- Use analysis outputs as masks or support maps in blend chains.
- Pair with levels or threshold nodes to isolate useful map ranges.

Input ports

- input
- Usually expects a structured grayscale or height-like input.

Output ports

- out
- Outputs one or more analysis maps depending on node type.

Parameters

Parameter	Type	Default	Range or values
Strength	float	1	0.01 to 10
Format	bool	false	-

Behavior

- These nodes compute derived data from shape and height information.
- Outputs are often best consumed by downstream utility and mixing nodes.

Common mistakes

- Analysis nodes can be sensitive to noisy or low-contrast inputs.
- Map interpretation changes with downstream color-space assumptions.

Blend

Use Blend to combine two branches with selectable compositing behavior. It is the primary node for layering color, mask, and detail passes.

Availability: Free feature set

Common uses

- Normalize inputs before mixing for predictable blends.
- Use masks and height maps to control which regions dominate.

Input ports

- fg
- bg
- mask
- Expects two or more input branches depending on node type.

Output ports

- out
- Outputs the mixed result and optional helper maps for some nodes.

Parameters

Parameter	Type	Default	Range or values
Mode	enum	normal	normal, alpha_over, multiply, color_burn, linear_burn, min, subtract_clamp, screen, color_dodge, add, max, addSub, overlay, softlight, hardlight, vivid_light, linear_light, pinlight, hard_mix, difference, exclusion, divide, color, hue, saturation, luminosity
Opacity	float	1	0 to 1

Behavior

- Mixing nodes merge, select, or route among multiple incoming signals.
- Blend behavior depends on selected mode and input value ranges.

Common mistakes

- Input range mismatches can make blend results look clipped or washed out.
- Switch-based nodes can hide inactive branches during debugging.

Height Blend

Use Height Blend to mix materials based on relative height influence instead of simple opacity. It preserves relief hierarchy in layered surface workflows.

Availability: Pro

Common uses

- Normalize inputs before mixing for predictable blends.
- Use masks and height maps to control which regions dominate.

Input ports

- heightTop
- heightBottom
- mask
- Expects two or more input branches depending on node type.

Output ports

- height
- mask
- Outputs the mixed result and optional helper maps for some nodes.

Parameters

Parameter	Type	Default	Range or values
Height Offset	float	0.5	0 to 1
Contrast	float	0.5	0 to 1
Mode	enum	balanced	balanced, bottomPriority
Opacity	float	1	0 to 1

Behavior

- Mixing nodes merge, select, or route among multiple incoming signals.
- Blend behavior depends on selected mode and input value ranges.

Common mistakes

- Input range mismatches can make blend results look clipped or washed out.
- Switch-based nodes can hide inactive branches during debugging.

Average Blend

Use Average Blend to merge multiple sources into an even composite signal. It is useful for stabilizing noisy inputs or balancing detail contributions.

Availability: Pro

Common uses

- Normalize inputs before mixing for predictable blends.
- Use masks and height maps to control which regions dominate.

Input ports

- None
- Expects two or more input branches depending on node type.

Output ports

- out

Port names can depend on the graph or node configuration.

- Outputs the mixed result and optional helper maps for some nodes.

Parameters

Parameter	Type	Default	Range or values
Input Count	int	2	1 to 6
Auto Level	bool	false	-
Intensity 1	float	1	0 to 1
Intensity 2	float	1	0 to 1
Intensity 3	float	1	0 to 1
Intensity 4	float	1	0 to 1
Intensity 5	float	1	0 to 1
Intensity 6	float	1	0 to 1

Behavior

- Mixing nodes merge, select, or route among multiple incoming signals.
- Blend behavior depends on selected mode and input value ranges.

Common mistakes

- Input range mismatches can make blend results look clipped or washed out.
- Switch-based nodes can hide inactive branches during debugging.

Switch

Use Switch to toggle between two inputs for quick A/B variants. It is useful for testing alternate branches without rewiring the graph.

Availability: Free feature set

Common uses

- Normalize inputs before mixing for predictable blends.
- Use masks and height maps to control which regions dominate.

Input ports

- a
- b
- Expects two or more input branches depending on node type.

Output ports

- out
- Outputs the mixed result and optional helper maps for some nodes.

Parameters

Parameter	Type	Default	Range or values
Use B	bool	false	-

Behavior

- Mixing nodes merge, select, or route among multiple incoming signals.
- Blend behavior depends on selected mode and input value ranges.

Common mistakes

- Input range mismatches can make blend results look clipped or washed out.
- Switch-based nodes can hide inactive branches during debugging.

Multi Switch

Use Multi Switch to route one of several inputs to a single output. It supports mode-driven pipelines and indexed variant selection.

Availability: Free feature set

Common uses

- Normalize inputs before mixing for predictable blends.
- Use masks and height maps to control which regions dominate.

Input ports

- None
- Expects two or more input branches depending on node type.

Output ports

- out

Port names can depend on the graph or node configuration.

- Outputs the mixed result and optional helper maps for some nodes.

Parameters

Parameter	Type	Default	Range or values
Selected Input	int	1	1 to 8

Behavior

- Mixing nodes merge, select, or route among multiple incoming signals.
- Blend behavior depends on selected mode and input value ranges.

Common mistakes

- Input range mismatches can make blend results look clipped or washed out.
- Switch-based nodes can hide inactive branches during debugging.

Input

Use Input to define the public entry ports of a reusable subgraph. These ports become the contract parents must supply when instanting the graph.

Availability: Free feature set

Common uses

- Add clear labels to ports so upstream graphs can connect inputs correctly.
- Treat input ports as a stable contract for reusable graph assets.

Input ports

- None
- No upstream image input required; this node declares interface ports.

Output ports

- None

Port names can depend on the graph or node configuration.

- Outputs declared interface values into the current graph.

Parameters

This node has no shared processing parameters.

Behavior

- Ports are normalized on load to ensure required defaults exist.
- Graph input handles map to instance input handles by ID.

Common mistakes

- Changing port IDs can break existing graph instance wiring.
- Missing defaults can cause unexpected empty inputs in instances.

Instance

Use Instance to execute another saved graph as a nested reusable node. It is ideal for modular authoring where shared logic is maintained centrally.

Availability: Free feature set

Common uses

- Use instances to build reusable graph libraries and layered workflows.
- Expose only the parameters needed for parent-level control.

Input ports

- None
- Accepts linked graph inputs plus optional exposed parameter inputs.

Output ports

- out

Port names can depend on the graph or node configuration.

- Outputs the linked graph outputs mapped by port IDs.

Parameters

This node has no shared processing parameters.

Behavior

- Texture engine instance execution requires the project list to be configured first.
- Inputs and outputs are rebuilt from linked graph contracts.

Common mistakes

- Broken project references prevent the instance from rendering.
- Changing linked graph ports requires parent graph wiring updates.

Output

Use Output to publish named results from a graph for preview, export, or instancing. Well-labeled outputs make downstream integration predictable.

Availability: Free feature set

Common uses

- Name outputs for downstream readability and export clarity.
- Use multiple ports when a graph should expose several outputs.

Input ports

- None
- Receives final branch signals that should be exported from the graph.

Output ports

- None

Port names can depend on the graph or node configuration.

- Publishes graph outputs to preview, export, and graph instances.

Parameters

This node has no shared processing parameters.

Behavior

- Graph output handles are normalized to include default out handle semantics.
- Export and preview surfaces consume graph output ports directly.

Common mistakes

- Unconnected outputs return empty results.
- Changing output IDs can break parent graph connections.

Float

Use Float to emit a constant scalar that drives numeric parameters. It centralizes tuning values so multiple nodes can share one control source.

Availability: Pro

Common uses

- Use value nodes to centralize controls for exposed parameters.
- Name value nodes clearly when they drive multiple graph sections.

Input ports

- None
- No image input required. Values are set directly on the node.

Output ports

- out
- Outputs typed scalar or vector data.

Parameters

Parameter	Type	Default	Range or values
Value	float	0.5	0 to 1
Min	float	0	-
Max	float	1	-
Snap To Int	bool	false	-

Behavior

- Value nodes output non-image data used by typed parameter inputs.
- They are useful for reusable graph controls and presets.

Common mistakes

- Type mismatches can prevent expected connections.
- Overusing disconnected value nodes can clutter the canvas quickly.

Bool

Use Bool to emit a stable true/false control signal for toggles. It simplifies branch and feature switches in parameterized graphs.

Availability: Pro

Common uses

- Use value nodes to centralize controls for exposed parameters.
- Name value nodes clearly when they drive multiple graph sections.

Input ports

- None
- No image input required. Values are set directly on the node.

Output ports

- out
- Outputs typed scalar or vector data.

Parameters

Parameter	Type	Default	Range or values
Value	bool	false	-

Behavior

- Value nodes output non-image data used by typed parameter inputs.
- They are useful for reusable graph controls and presets.

Common mistakes

- Type mismatches can prevent expected connections.
- Overusing disconnected value nodes can clutter the canvas quickly.

Int

Use Int to emit whole-number control values for counts and discrete indices. It is useful for selecting options that should not interpolate.

Availability: Pro

Common uses

- Use value nodes to centralize controls for exposed parameters.
- Name value nodes clearly when they drive multiple graph sections.

Input ports

- None
- No image input required. Values are set directly on the node.

Output ports

- out
- Outputs typed scalar or vector data.

Parameters

Parameter	Type	Default	Range or values
Value	int	0	0 to 100
Min	int	0	-
Max	int	100	-

Behavior

- Value nodes output non-image data used by typed parameter inputs.
- They are useful for reusable graph controls and presets.

Common mistakes

- Type mismatches can prevent expected connections.
- Overusing disconnected value nodes can clutter the canvas quickly.

Time

Use Time with Math nodes to drive procedural parameters on every preview or exported animation frame.

Availability: Pro

Common uses

- Use Seconds for rate-based motion.
- Use Normalized for effects spanning the saved frame range.

Input ports

- None
- No input required.

Output ports

- out
- Outputs an int in Frame mode and a float otherwise.

Parameters

This node has no shared processing parameters.

Behavior

- Reads render context and does not dirty saved graph data as the playhead moves.

Common mistakes

- Large frame ranges can produce large exports.
- Frame mode emits an integer while other modes emit floats.

Enum

Use Enum to emit a named mode selection for enum-based parameters. It keeps mode switching explicit and centralized in larger graphs.

Availability: Pro

Common uses

- Use value nodes to centralize controls for exposed parameters.
- Name value nodes clearly when they drive multiple graph sections.

Input ports

- None
- No image input required. Values are set directly on the node.

Output ports

- out
- Outputs typed scalar or vector data.

Parameters

Parameter	Type	Default	Range or values
Selection	int	0	0 to 64

Behavior

- Value nodes output non-image data used by typed parameter inputs.
- They are useful for reusable graph controls and presets.

Common mistakes

- Type mismatches can prevent expected connections.
- Overusing disconnected value nodes can clutter the canvas quickly.

Vec2

Use Vec2 to emit paired numeric controls such as offsets and coordinates. It is a convenient source for transform and direction parameters.

Availability: Pro

Common uses

- Use value nodes to centralize controls for exposed parameters.
- Name value nodes clearly when they drive multiple graph sections.

Input ports

- None
- No image input required. Values are set directly on the node.

Output ports

- out
- Outputs typed scalar or vector data.

Parameters

This node has no shared processing parameters.

Behavior

- Value nodes output non-image data used by typed parameter inputs.
- They are useful for reusable graph controls and presets.

Common mistakes

- Type mismatches can prevent expected connections.
- Overusing disconnected value nodes can clutter the canvas quickly.

Math

The Math node applies a selected arithmetic, trigonometric, or comparison operation to two input values. Values can be provided via incoming connections or slider parameters.

Availability: Pro

Common uses

- Use clamp to keep results between 0 and 1.
- Connect grayscale maps to apply operations per-pixel.

Input ports

- a
- b
- Takes two input values (either parameters or connections).

Output ports

- out
- Outputs the calculated result (r channel).

Parameters

Parameter	Type	Default	Range or values
Operation	enum	add	add, subtract, multiply, divide, power, logarithm, sqrt, inversesqrt, absolute, exponent, minimum, maximum, lessThan, greaterThan, sign, round, floor, ceil, truncate, fraction, modulo, snap, pingpong, sine, cosine, tangent, arcsine, arccosine, arctangent, arctan2, radians, degrees
Value 1	float	0.5	-10 to 10
Value 2	float	0.5	-10 to 10
Clamp	bool	false	-

Behavior

- Evaluates standard math operations.
- If map inputs are provided, runs per-pixel.

Common mistakes

- Division by zero defaults to 0 safely.
- Using non-grayscale maps might yield unexpected results.

Dot

Use Dot to reroute long connections and keep wiring readable across dense canvases. Labeled dots also document signal intent at handoff points.

Availability: Free feature set

Common uses

- Name dots to clarify routing intent in dense graphs.
- Use portal pairing when routing signals across distant graph regions.

Input ports

- `input`
- Receives one incoming signal.

Output ports

- `out`
- Forwards the incoming signal from its output handle.

Parameters

This node has no shared processing parameters.

Behavior

- Dot supports optional portal pairing by named target dot IDs.
- Read-only labels appear when not selected for quick scanning.

Common mistakes

- Unnamed dots are hard to maintain in larger graphs.
- Portal pairing confusion can hide where a signal originates.

Frame

Use Frame to visually group related nodes into functional sections. It improves navigation and maintenance in large production graphs.

Availability: Free feature set

Common uses

- Use frame labels to describe the purpose of each region.
- Resize frames to keep related node clusters visually contained.

Input ports

- None
- No signal input; this is an organizational node.

Output ports

- None
- No signal output; this is an organizational node.

Parameters

This node has no shared processing parameters.

Behavior

- Frame nodes use editor-specific drag and resize behavior.
- Frame styling and opacity are editable directly on the node chrome.

Common mistakes

- Overlapping frames can make layout interactions harder.
- Very large frames with vague labels reduce clarity.

Comment

Use Comment to record intent, constraints, or TODOs beside relevant logic. It helps collaborators understand decisions without opening external docs.

Availability: Free feature set

Common uses

- Keep comments close to the node groups they describe.
- Use concise notes that explain intent, not obvious settings.

Input ports

- None
- No signal input; documentation only.

Output ports

- None
- No signal output; documentation only.

Parameters

This node has no shared processing parameters.

Behavior

- Comments are non-processing helper nodes for documentation only.
- They help preserve graph authorship intent during collaboration.

Common mistakes

- Outdated comments can mislead future edits.
- Large text blocks can clutter graph navigation.