Material Definition Language
Core definitions

11 December 2018
Version 1.4

Material Definition Language — Core definitions

Copyright Information

Document build number 312200.1281

Contents

Material Definition Language — Core definitions

1   Introduction

The Material Definition Language (MDL) module `nvidia::core_definitions` contains a collection of MDL materials that can either be used independently (*simple materials*) or in combination with other materials through the use of *material combiners* and *material modifiers*. *Texturing functions* provide further control and refinement of material parameter values. Together, materials, combiners, modfiers and the texturing functions can simulate complex, real-world models of appearance.

2   Materials and building blocks

2.1   Simple materials

Simple materials can either be used directly to model real world materials with matching behavior or can be used as components when creating more complex materials using material combiners or material modifiers.

2.1.1   Simple Diffuse

A basic opaque, diffuse reflective material.

Parameters:

name	type	default	description
Color	`color`	80% grey	The color (reflectivity) of the material. Realistic values are between 4% black and 90% white.
Diffuse Roughness	`float`	0.0	Higher roughness values lead a powdery appearance. Values must be between 0.0 and 1.0.
Bumps	`float3`	no bumps	Attach bump or normal maps here.

2.1.2   Metal

A metallic material, supports stretched (anisotropic) reflections.

Parameters:

name	type	default	description
Color	`color`	90% white	The color of the metal.
Roughness	`float`	0.05	Higher roughness values lead to bigger highlights and blurry reflections. Values must be between 0.0 and 1.0.
Reflection weight	`float`	0.9	Intensity of highlights and glossy reflections. Values must be between 0.0 and 1.0.
Anisotropy	`float`	0.0	Higher values will stretch the highlight, simulating microscopic scratches. Values must be between 0.0 and 1.0.
Anisotropy rotation	`float`	0.0	Changes the orientation of the anisotropy. A value of 1 will rotate the orientation 360 ° Values must be between 0.0 and 1.0.

| Bumps | `float3` | no bumps | Attach bump or normal maps here. |

2.1.3   Plastic

A basic dielectric, works for not just plastic but every material opaque that is not metallic. Supports stretched highlights

Parameters:

name	*type*	*default*	*description*
Color	`color`	50% grey	The color of the material.
Roughness	`float`	0.05	Higher roughness values lead to bigger highlights and blurry reflections. Values must be between 0.0 and 1.0.
Reflection weight	`float`	1.0	Additional control for the reflectivity. Values must be between 0.0 and 1.0.
Anisotropy	`float`	0.0	Higher values will stretch the highlight, simulating microscopic scratches. Values must be between 0.0 and 1.0.
Anisotropy rotation	`float`	0.0	Changes the orientation of the anisotropy. A value of 1 will rotate the orientation 360°. Values must be between 0.0 and 1.0.
IOR	`float`	1.5	Determines reflectivity. Typical plastics have an index of refraction of around 1.5. Realistic values are between 1.0 and 4.0
Bumps	`float3`	no bumps	Attach bump or normal maps here.

2.1.4   Retroreflective

A material with a retroreflective component, works well for road signs and retroreflective stickers.

Parameters:

name	*type*	*default*	*description*
Color	`color`	color(.2,.03,.03)	The color of the material.
Reflection color	`color`	color(.8,.8,.03)	The color of the material.
Roughness	`float`	0.05	Higher roughness values lead to bigger highlights and blurry reflections. Values must be between 0.0 and 1.0.
Reflection weight facing	`float`	0.05	Reflectivity control for geometry facing the viewer. Values must be between 0.0 and 1.0.
Reflection weight edge	`float`	0.05	Reflectivity control for the reflectivity at geometry edges. Values must be between 0.0 and 1.0.
Bumps	`float3`	no bumps	Attach bump or normal maps here.

2.1.5 Thin glass

A basic transmissive dielectric without refraction or volume. Suitable for all cases were transparent materials are modeled as a single polygon.

Parameters:

name	type	default	description
Transmission color	color	95% transmissive	The color of the material.
Roughness	float	0.0	Higher roughness values lead to bigger highlights and blurry reflections. Values must be between 0.0 and 1.0.
IOR	float	1.4	Determines reflectivity. Typical glass materials have an index of refraction of around 1.5. Realistic values are between 1.0 and 4.0.
Bumps	float3	no bumps	Attach bump or normal maps here.

2.1.6 Thin translucent material

A diffuse transmissive dielectric material. Suitable as a basis for creating paper, fabric or leaves.

Parameters:

name	type	default	description
Diffuse color	color	95% white	The color of the material.
Translucence color	color	95% white	The color of light passing through the material.
Translucence weight	float	0.5	the amount of translucence vs. diffuse reflection. Values must be between 0.0 and 1.0.
Roughness	float	0.0	Higher roughness values lead to bigger highlights and blurry reflections. Values must be between 0.0 and 1.0.
IOR	float	1.5	Determines reflectivity. Realistic values are between 1.0 and 4.0.
Bumps	float3	no bumps	Attach bump or normal maps here.

2.1.7 Thick glass

A basic transmissive dielectric with refraction and coloring in the volume.

Parameters:

name	type	default	description
Transmission color	color	100% white	Colors the light entering the volume. Think stained glass.

Volume color	`color`	95% white	The color of the glass body. The actual color in the rendering will depend also on thickness of the model and "Volume reference distance".
Volume reference distance	`float`	0.1	Should be set to the typical thickness of objects made from this material. "Volume color" will be reached at this distance. The unit for this distance is meter.
Roughness	`float`	0.0	Higher roughness values lead to bigger highlights and blurry reflections. Values must be between 0.0 and 1.0.
IOR	`float`	1.5	Determines reflectivity and refraction strength. Typical glasses have an index of refraction of around 1.5. Realistic values are between 1.0 and 4.0.
Abbe number	`float`	0.0	Controls dispersion. 0.0 switches dispersion off, dispersive materials have abbe numbers between 25.0 and 85.0 Realistic values are between 1.0 and 4.0.
Bumps	`float3`	no bumps	Attach bump or normal maps here.

2.1.8 Thick translucent material

A subsurface scattering material. Can be used to create, for example, wax or milk.

Parameters:

name	*type*	*default*	*description*
Transmission color	`color`	100% white	Colors the light entering the volume. Think stained glass.
Volume color	`color`	95% white	The color of the object body. The actual color in the rendering will depend also on thickness of the model and "Volume reference distance".
Volume scattering	`float`	0.5	Amount of sub surface scattering at "Volume reference distance". Values must be between 0.0 and 1.0.
Volume reference distance	`float`	0.1	Should be set to the typical thickness of objects made from this material. "Volume color" and "Volume scattering" will be reached at this distance. The unit for this distance is meter.
Reflection roughness	`float`	0.0	Higher roughness values lead to bigger highlights and blurry reflections. Values must be between 0.0 and 1.0.
Reflection weight	`float`	1.0	Overall reflectivity of the material. Values must be between 0.0 and 1.0.
IOR	`float`	1.4	Determines reflectivity. Realistic values are between 1.0 and 4.0.

Abbe number	`float`	0.0	Controls dispersion. 0.0 switches dispersion off, dispersive materials have abbe numbers between 25.0 and 85.0 Realistic values are between 1.0 and 4.0.
Bumps	`float3`	no bumps	Attach bump or normal maps here.

2.1.9   Flexible material model

A complex material that can be configured to recreate a wide variety of looks.

Parameters:

name	type	default	description
Base color	`color`	50% grey	The color of the material.
Diffuse roughness	`float`	0.0	Higher roughness values lead to a more "powdery" look. Values must be between 0.0 and 1.0.
Metallic material	`bool`	false	If true, reflection will be colored and independent of view direction. If false, reflection will be white and direction dependent. Directional dependence is in this case based on the index of refraction (Fresnell effect).
Reflection weight	`float`	1.0	Controls the amount of (glossy or mirror-like) reflection. Values must be between 0.0 and 1.0.
Reflection roughness	`float`	0.1	Higher roughness values lead to more blurry reflections. Values must be between 0.0 and 1.0.
Reflection anisotropy	`float`	0.0	Higher values will stretch the highlight. Values must be between 0.0 and 1.0.
Anisotropy rotation	`float`	0.0	Changes the orientation of the anisotropy. A value of 1 will rotate the orientation 360 ° Values must be between 0.0 and 1.0.
Transmission color	`color`	100% white	Color effect for transmission independent of thickness of the object. Think stained glass.
Volume color	`color`	100% white	Only applies if not "Thin walled". "Volume color" will be reached at "Volume reference distance" (in meter).
Volume reference distance	`float`	0.1	Only applies if not "Thin walled". Should be set to the typical thickness of objects made from this material. "Volume color" and "Volume scattering" will be reached at this distance. The unit for this distance is meter.
Transmission roughness	`float`	0.0	Higher values lead to objects seen through the material to appear blurry. Values must be between 0.0 and 1.0.

Transmission weight	`float`	0.0	Weights how much light passes through the object vs it's diffuse reflectivity Values must be between 0.0 and 1.0.
IOR	`float`	1.5	Determines reflectivity and refraction strength. Realistic values are between 1.0 and 4.0.
Abbe number	`float`	0.0	Controls dispersion. 0.0 switches dispersion off, dispersive materials have abbe numbers between 25.0 and 85.0
Thin walled	`bool`	false	Thin walled materials do not refract and do not have volume effects. Good for soap bubbles or window glass.
Bumps	`float3`	no bumps	Attach bump or normal maps here.

2.1.10 Flaky Paint

A multi layer paint material featuring metallic flakesflakes. It is made up of a diffuse base, a flake layer and a clearcoat layer.

Parameters:

name	*type*	*default*	*description*
Base color	`color`	rgb 0.3, 0.01, 0.01	The color of the base paint.
Flake color	`color`	rgb 0.6, 1.0, 0.1	The color of the Flakes.
Flake roughness	`float`	0.15	Determines roughness of the metallic flakes. Values must be between 0.0 and 1.0.
Flake size	`float`	1.0	Determines size of the metallic flakes, in mm.
Flake amount	`float`	0.4	Determines amount of visible metallic flakes. Values must be between 0.0 and 1.0.
Flake weight	`float`	0.8	Determines visibility of the metallic flakes. Values must be between 0.0 and 1.0.
Flake orientation randomness	`float`	1.0	Larger numbers will increase the sparkle radius around highlights.
IOR	`float`	1.6	termines reflectivity of the clear coat. Realistic values are between 1.0 and 4.0.
Coat roughness	`float`	0.0	Determines roughness of the clear coat.
Coat bump	`float3`	no bumps	Attach bump or normal maps here.

2.2 Modifier materials

Modifier materials can be used to create new materials based on already created materials. They either combine multiple materials into a new one or add additional features to an existing material.

2.2.1 Apply clear coating

This modifier applies an additional (dielectric) clear coat to an existing material. The clear coat can have a dedicated bump map, this way its, for example, possible to simulate the typical "orange peel" effect of paints.

Parameters:

name	type	default	description
Base Material	`material`	Simple diffuse	The material that will get a clear coating applied.
Coat filter color	`color`	100% white	For simulating coatings with colored resins that modulate the color of underlying layers
Reflection roughness	`float`	0.0	Determines roughness of the clear coat. Values must be between 0.0 and 1.0.
Coat visibility	`float`	1.0	Determines visibility of the clear coat.
IOR	`float`	1.6	Determines reflectivity of the clear coat. Realistic values are between 1.0 and 4.0.
Bumps	`float3`	no bumps	Attach bump or normal maps here.

2.2.2 Apply thin metal coating

Apply metal coat to an existing material.

Parameters:

name	type	default	description
Base Material	`material`	Simple diffuse	The material that will get a coating applied.
Reflection color	`color`	95% white	The color of the metallic coat
Reflection roughness	`float`	0.0	Determines roughness of the coat. Values must be between 0.0 and 1.0.
Reflection weight	`float`	0.3	The oacity of the metallic coat. Values must be between 0.0 and 1.0.
Bumps	`float3`	no bumps	Attach bump or normal maps here.

2.2.3 Apply a cover of dust

A diffuse dust cover thats more visible towards the edges of an object.

Parameters:

name	type	default	description
Base Material	`material`	Simple diffuse	The material that will get a coating applied.
Dust color	`color`	70% grey	The color of the dust.
Dust density	`float`	1.0	The opacity of the dust cover. Values must be between 0.0 and 1.0.

Dust amount	`float`	0.5	Overall dust amount. Values must be between 0.01 and 1.0.
Bumps	`float3`	no bumps	Attach bump or normal maps here.

2.2.4 Apply a color falloff

This applies a view dependent color filter to the underlying materials reflection. Note that this is applied at microscopic levels, therefore the color transitions will be blurred if the material is of high roughness. A color falloff can be used for example to increase the realism of metallic reflections.

Parameters:

name	*type*	*default*	*description*
Base Material	`material`	Metal	The material that will get modified.
Color 1	`color`	"red"	Falloff color 1 (facing direction)
Color 2	`color`	"green"	Falloff color 2
Color 3	`color`	"blue"	Falloff color 3
Color 4	`color`	"red"	Falloff color 4
Color 5	`color`	"green"	Falloff color 5 (object edges)

2.2.5 Apply Flake coating

Apply layer of metallic flakes to an existing material. Flakes have a random variation to their orientation resulting in sparkling effects around highlights. Parameters:

name	*type*	*default*	*description*
Base Material	`material`	Simple diffuse	The material that will get a flake layer applied to.
Color	`color`	rgb 0.9, 0.7, 0.5	The color of the Flakes.
Roughness	`float`	0.0	Determines roughness of the metallic flakes.
Flake size	`float`	1.0	Determines size of the metallic flakes, given in mm.
Flake amount	`float`	0.5	Determines amount of visible metallic flakes. Values must be between 0.0 and 1.0.
Flake opacity	`float`	0.5	Determines visibility of the metallic flakes. Values must be between 0.0 and 1.0.
Flake orientation randomness	`float`	1.0	Larger numbers will increase the sparkle radius around highlights.

2.2.6   Apply thin film

Apply thin film coating to an existing material. Thin film effects for example change the color of lenses or are responsible for heat coloring of metals or the rainbow colors in soap bubbles. Parameters:

name	type	default	description
Base Material	`material`	Simple diffuse	The material that will get a flake layer applied to.
IOR	`float`	1.6	The index of refraction of the thin film interface.
thickness	`float`	400	Thickness of the thin film in nm.

2.2.7   Add cut-outs

Cuts out a shape from an existing material. Also forces material to be thin-walled since cut-outs do not work with volumetric effects. Suitable for modeling leaves, grass or fences without actually creating exact geometry.

Parameters:

name	type	default	description
Base Material	`material`	Plastic	The material that you want to cut out your object from.
Cutout	`float`	1.0	Attach a texturing function to define the extent of the object. Note that while the parameter is defined as "float", it is not meant to be used as a substitute of opacity.

2.2.8   Add simple sticker

A quick way for adding stickers to a material. The sticker is a simple dielectric and needs a mask to define it's extent

Parameters:

name	type	default	description
Base Material	`material`	Plastic	The material that will get a sticker added.
Sticker Color	`color`	50% grey	The color of the material.
Sticker Roughness	`float`	0.05	Higher roughness values lead to bigger highlights and blurry reflections. Values must be between 0.0 and 1.0.
Sticker Reflectivity	`float`	1.0	Additional control for the reflectivity. Values must be between 0.0 and 1.0.
Sticker mask	`float`	0.0	Determines extent of the sticker. Values must be between 0.0 and 1.0.

| Sticker IOR | `float` | 1.5 | Determines reflectivity. Typical plastics have an index of refraction of around 1.5. Realistic values are between 1.0 and 4.0 |
| Sticker Bumps | `float3` | no bumps | Attach bump or normal maps here. |

2.2.9   Add global bumpmap

Adds global bumpmap to existing materials. Local bump map of the base material is preserved and work on top of the global bump map.

Parameters:

name	*type*	*default*	*description*
Base Material	`material`	Plastic	The material that will get a global bump map applied.
Bumps	`float3`	no bumps	Attach bump or normal maps here.

2.2.10   Add displacement

Adds displacement to an existing material. Note that the object might need special setup to show the displacement correctly (configure the maximum displacement for the object and add subdivisions for low polygonal geometry).

Parameters:

name	*type*	*default*	*description*
Base Material	`material`	Plastic	The material that will get a global bump map applied.
Displacement amount	`float`	0.0	Attach displacement texture here. Note that the object needs to be set up correctly to have good displacement results.
Displacement scale	`float`	1.0	A global scale factor for the displacement amount.

2.2.11   Add emission

Adds emission to a material.

Parameters:

name	*type*	*default*	*description*
Base Material	`material`	Simple diffuse	The material that will get emission added.
Color	`color`	100% white	The color of the Light.
Intensity	`float`	1000	the brightness of the emitted light
Unit for emission	`emmission unit type`	lumen/m2	supported units are lumen/m2,lumen,candela, nit (candela/m2)

2.2.12 Add thermal emission

Adds emission to a material. The color is determined by a color temperature. The default is 6500K daylight.

Parameters:

name	type	default	description
Base Material	material	Simple diffuse	The material that will get emission added.
Temperature	float	6500	The color temperature of the Light in Kelvin.
Intensity	float	1000	the brightness of the emitted light
Unit for emission	emmission unit type	lumen/m2	supported units are lumen/m2,lumen,candela, nit (candela/m2)

2.2.13 Surface Blender

Blend surface characteristics of two materials or mask them using a texture. Volumetric and geometric characteristics of the "Base Material" will be retained.

Parameters:

name	type	default	description
Base Material	material	Plastic	The material the blend is based on.
Blend Material	material	Metal	Surface properties to be used for the blend are taken from this material.
Blend weight	float	0.0	Weight of the "Blend material" to be used in the blend. Values must be between 0.0 and 1.0.

2.3 Emissive materials

In MDL, there is no distinction between regular objects and lightsources. Any object can have emissive as well as reflective/transmissive behavior. The following set of materials replicate the behavior of standard "light shaders".

2.3.1 Diffuse emission

Emissive material emitting in all directions.

Parameters:

name	type	default	description
Color	color	100% white	The color of the Light.
Intensity	float	1000.0	The brightness of the light source
Unit for emission	emmission unit type	lumen/m2	supported units are lumen/m2,lumen,candela, nit (candela/m2)

2.3.2 Spotlight emission

Emissive material emitting focused in one direction (y direction in object space).

Parameters:

name	type	default	description
Color	color	100% white	The color of the Light.
Intensity	float	1000.0	The brightness of the light source.
Spot focus	float	30.0	Larger values lead to more focused spotlights.
Unit for emission	emmission unit type	lumen/m2	supported units are lumen/m2,lumen,candela, nit (candela/m2)

2.3.3 IES file based emission

Emissive material emitting as described in an IES file (y direction in object space).

Parameters:

name	type	default	description
IES light profile data	IES file	no profile	Data describing the distribution of the light.
Color	color	100% white	The color of the Light.
Intensity	float	1000.0	The brightness of the light source.

3 Texturing functions

Texturing functions can be used to add variation to many features of a material. They can for example be used to change the color across an object, the intensities of reflections or add variations to an otherwise smooth surface through bump maps. nvidia::core_definitions suppports texturing using bitmap textures as well as a selection of procedural textures to add variation. Most texturing functions come in 2 flavours: 1 variant suitable to add bumps to a material and one variant to add color or vary roughness or reflectivity.

3.1 Bitmap texture (color/float variant)

This function allows texturing using image files of various file formates. It is suitable for adding variation to color or float type parameters.

Parameters:

name	type	default	description
Bitmap file	file texture		The image file to be used for texturing
Scalar mode	enum	"Average"	Defines what should happen if a color texture is used on a "float" type parameter. By defaul, the average value is used.
Brightness	float	1.0	A controll to vary the brightness of the image before use.

Contrast	`float`	1.0	A controll to vary the contrast of the image before use.
UV space index	`int`	0	Allows the use of a specific UV channel if the object has multiple UV sets defined.
Invert image	`bool`	false	Allows inverting of the texture.
Rotation	`float`	0.0	Rotation angle of the texture in degrees
Offset	`float2`	0.0, 0.0	Controls position of the texture on the object
Tiling	`float2`	1.0, 1.0	Controls the scale of the texture on the object. Higher values result on higher repetition of the image.
Clip	`bool`	false	If set to true, texture will not repeat. Outside of the texture, color will be black and the scalar value will be 0.

3.2 Bitmap texture (bump variant)

Allows texturing using image files of various file formates. The image is interpreted as a hight map to compute the bumpiness.

Parameters:

name	*type*	*default*	*description*
Bitmap file	`file texture`		The image file to be used for texturing
Bump mode	`enum`	"Average"	Defines what should happen if a color texture is used on a "float" type parameter. By defaul, the average value is used.
Bump strength	`float`	1.0	Scales the virtual heightmap, higher values lead to stronger bumps.
UV space index	`int`	0	Allows the use of a specific UV channel if the object has multiple UV sets defined.
Clip	`bool`	false	If set to true, texture will not repeat. Outside of the texture the surface will be flat.
Rotation	`float`	0.0	Rotation angle of the texture in degrees
Offset	`float2`	0.0, 0.0	Controls position of the texture on the object
Tiling	`float2`	1.0, 1.0	Controls the scale of the texture on the object. Higher values result on higher repetition of the image.

3.3 Normalmap texture

Allows the use of tangent space normal maps.

Parameters:

name	type	default	description
Normalmap file	`file texture`		The image file containing the normal map.
Strength	`float`	1.0	Controls the strength of the normal map.
UV space index	`int`	0	Allows the use of a specific UV channel if the object has multiple UV sets defined.
Clip	`bool`	false	If set to true, texture will not repeat. Outside of the texture the surface will be flat.
Rotation	`float`	0.0	Rotation angle of the texture in degrees
Offset	`float2`	0.0, 0.0	Controls position of the texture on the object
Tiling	`float2`	1.0, 1.0	Controls the scale of the texture on the object. Higher values result on higher repetition of the image.

3.4   3d checker texture (color/float variant)

Allows texturing using a 3 dimensional checkerboard pattern. 3D textures use all not just uv but uvw or object space (xyz) coordinates. If "w" coordinates are not explicitely provided they are set to 0.

Parameters:

name	type	default	description
Color 1	`color`	100% white	1st checker color.
Color 2	`color`	0% black	2nd checker color.
Blur	`floar`	0.0	Higher values lead to a blurring of the checker tiles.
Use Object Space	`bool`	false	If off, UVW space will be used. If on, 3d texturing in object space will apply. For applications that do not support object space, world space will be used.
UV space index	`int`	0	Allows the use of a specific UV channel if the object has multiple UV sets defined. Only applies if "Use Object Space" is off.
Rotation	`float3`	0.0, 0.0, 0.0	Rotation angle of the texture in degrees. To just rotate in the "uv" plane provide the the rotation value as the 3rd component.
Offset	`float3`	0.0, 0.0, 0.0	Controls position of the texture on the object
Tiling	`float3`	1.0, 1.0, 1.0	Controls the scale of the texture on the object. Higher values result on higher repetition of the image.

3.5   3d checker texture (bump variant)

Allows texturing using a checkerboard pattern.

Parameters:

name	type	default	description
Bump strength	float	1	The strength of the bump mapping effect.
Blur	floar	0.0	Higher values lead to a blurring of the checker tiles.
Use Object Space	bool	false	If off, UVW space will be used. If on, 3d texturing in object space will apply. For applications that do not support object space, world space will be used.
UV space index	int	0	Allows the use of a specific UV channel if the object has multiple UV sets defined. Only applies if "Use Object Space" is off.
Rotation	float3	0.0, 0.0, 0.0	Rotation angle of the texture in degrees. To just rotate in the "uv" plane provide the the rotation value as the 3rd component.
Offset	float3	0.0, 0.0, 0.0	Controls position of the texture on the object.
Tiling	float3	1.0, 1.0, 1.0	Controls the scale of the texture on the object. Higher values result on higher repetition of the image.

3.6   Perlin noise texture (color/float variant)

Allow texturing with a random noise pattern.

Parameters:

name	type	default	description
Color 1	color	100% white	1st noise color.
Color 2	color	0% black	2nd noise color.
Levels	int	3	Higher amounts will add detail to the noise. Value must be between 1 and 6.
Billowing appearance	bool	false	The noise will take on a more cloud like shape if set to true.
Lower threshold	float	0.0	Increasing this value will create bigger areas uniformly colored with Color 2.
Upper threshold	float	1.0	Lowering this value will create bigger areas uniformly colored with Color 1.
Use Object Space	bool	false	If off, UVW space will be used. If on, 3d texturing in object space will apply. For applications that do not support object space, world space will be used.

UV space index	`int`	0	Allows the use of a specific UV channel if the object has multiple UV sets defined. Only applies if "Use Object Space" is off.
Rotation	`float3`	0.0, 0.0, 0.0	Rotation angle of the texture in degrees. To just rotate in the "uv" plane provide the the rotation value as the 3rd component.
Offset	`float3`	0.0, 0.0, 0.0	Controls position of the texture on the object.
Tiling	`float3`	1.0, 1.0, 1.0	Controls the scale of the texture on the object. Higher values result on higher repetition of the image.

3.7   Perlin noise texture (bump variant)

Allow texturing with a random noise pattern.

Parameters:

name	*type*	*default*	*description*
Bump strength	`float`	1	The strength of the bump mapping effect.
Levels	`int`	1	Higher amounts will add detail to the noise. Value must be between 1 and 6.
Billowing appearance	`bool`	false	The noise will take on a more cloud like shape if set to true.
Lower threshold	`float`	0.0	Increasing this value will create bigger uniformly flat low areas.
Upper threshold	`float`	1.0	Lowering this value will create bigger uniformly flat high areas.
Use Object Space	`bool`	false	If off, UVW space will be used. If on, 3d texturing in object space will apply. For applications that do not support object space, world space will be used.
UV space index	`int`	0	Allows the use of a specific UV channel if the object has multiple UV sets defined. Only applies if "Use Object Space" is off.
Rotation	`float3`	0.0, 0.0, 0.0	Rotation angle of the texture in degrees. To just rotate in the "uv" plane provide the the rotation value as the 3rd component.
Offset	`float3`	0.0, 0.0, 0.0	Controls position of the texture on the object.
Tiling	`float3`	1.0, 1.0, 1.0	Controls the scale of the texture on the object. Higher values result on higher repetition of the image.

3.8   Cellular noise texture (color/float variant)

This function creates patterns of randomly positioned cells. Also known as "Worley noise"

Parameters:

name	type	default	description
Color 1	color	100% white	1st noise color.
Color 2	color	0% black	2nd noise color.
Cell type	enum	"Simple Cells"	Can be "Simple Cells", "Crystal cells" or "Borders". Describes fill pattern for the cells.
Cell shape	enum	"Circle base"	Can be "Circle base" or "Diamond base". The basic shape of the cells
Lower threshold	float	0.0	Increasing this value will create bigger areas uniformly colored with Color 2.
Upper threshold	float	1.0	Lowering this value will create bigger areas uniformly colored with Color 1.
Use Object Space	bool	false	If off, UVW space will be used. If on, 3d texturing in object space will apply. For applications that do not support object space, world space will be used.
UV space index	int	0	Allows the use of a specific UV channel if the object has multiple UV sets defined. Only applies if "Use Object Space" is off.
Rotation	float3	0.0, 0.0, 0.0	Rotation angle of the texture in degrees. To just rotate in the "uv" plane provide the the rotation value as the 3rd component.
Offset	float3	0.0, 0.0, 0.0	Controls position of the texture on the object.
Tiling	float3	1.0, 1.0, 1.0	Controls the scale of the texture on the object. Higher values result on higher repetition of the image.

3.9 Cellular noise texture (bump variant)

This function creates patterns of randomly positioned cells. Also known as "Worley noise"

Parameters:

name	type	default	description
Bump strength	float	1	The strength of the bump mapping effect.
Cell shape	enum	"Circle base"	Can be "Circle base" or "Diamond base". The basic shape of the cells
Lower threshold	float	0.0	Increasing this value will create bigger uniformly flat low areas.
Upper threshold	float	1.0	Lowering this value will create bigger uniformly flat high areas.

Use Object Space	`bool`	false	If off, UVW space will be used. If on, 3d texturing in object space will apply. For applications that do not support object space, world space will be used.
UV space index	`int`	0	Allows the use of a specific UV channel if the object has multiple UV sets defined. Only applies if "Use Object Space" is off.
Rotation	`float3`	0.0, 0.0, 0.0	Rotation angle of the texture in degrees. To just rotate in the "uv" plane provide the the rotation value as the 3rd component.
Offset	`float3`	0.0, 0.0, 0.0	Controls position of the texture on the object.
Tiling	`float3`	1.0, 1.0, 1.0	Controls the scale of the texture on the object. Higher values result on higher repetition of the image.

3.10 Flow noise texture (color/float variant)

Allow texturing with a 2D noise pattern suitable for waves. Note that even though the positioning is happening in 3D, the actual pattern is only 2D.

Parameters:

name	*type*	*default*	*description*
Color 1	`color`	100% white	1st noise color.
Color 2	`color`	0% black	2nd noise color.
Levels	`int`	3	Higher amounts will add detail to the noise. Value must be between 1 and 6.
Billowing appearance	`bool`	false	The noise will take on a more cloud like shape if set to true.
Phase offset	`float`	0.0	Controls the 3rd dimension of the function and can be used to animate the waves.
Level intensity gain	`float`	0.5	If multiple levels are used, "Level intensity gain" specifies a weighting factor for subsequent levels.
Level scaling	`float`	2.0	If multiple levels are used, "Level scaling" specifies a global scaling factor for subsequent levels
Progressive u scale	`float`	1.0	If multiple levels are used, "Progressive u scale" specifies an additional stretching factor in the "u" direction
Progressive v offset	`float`	0.0	If multiple levels are used, "Progressive v offset" specifies an offset for subsequent levels in the "v" direction

Use Object Space	`bool`	false	If off, UVW space will be used. If on, 3d texturing in object space will apply. For applications that do not support object space, world space will be used.
UV space index	`int`	0	Allows the use of a specific UV channel if the object has multiple UV sets defined. Only applies if "Use Object Space" is off.
Rotation	`float3`	0.0, 0.0, 0.0	Rotation angle of the texture in degrees. To just rotate in the "uv" plane provide the the rotation value as the 3rd component.
Offset	`float3`	0.0, 0.0, 0.0	Controls position of the texture on the object.
Tiling	`float3`	1.0, 1.0, 1.0	Controls the scale of the texture on the object. Higher values result on higher repetition of the image.

3.11   Flow noise texture (bump variant)

Allow texturing with a 2D noise pattern suitable for waves. Note that even though the positioning is happening in 3D, the actual pattern is only 2D.

Parameters:

name	*type*	*default*	*description*
Bump strength	`float`	1	The strength of the bump mapping effect.
Levels	`int`	1	Higher amounts will add detail to the noise. Value must be between 1 and 6.
Billowing appearance	`bool`	false	The noise will take on a more cloud like shape if set to true.
Phase offset	`float`	0.0	Controls the 3rd dimension of the function and can be used to animate the waves.
Level intensity gain	`float`	0.5	If multiple levels are used, "Level intensity gain" specifies a weighting factor for subsequent levels.
Level scaling	`float`	2.0	If multiple levels are used, "Level scaling" specifies a global scaling factor for subsequent levels
Progressive u scale	`float`	1.0	If multiple levels are used, "Progressive u scale" specifies an additional stretching factor in the "u" direction
Progressive v offset	`float`	0.0	If multiple levels are used, "Progressive v offset" specifies an offset for subsequent levels in the "v" direction

Use Object Space	`bool`	false	If off, UVW space will be used. If on, 3d texturing in object space will apply. For applications that do not support object space, world space will be used.
UV space index	`int`	0	Allows the use of a specific UV channel if the object has multiple UV sets defined. Only applies if "Use Object Space" is off.
Rotation	`float3`	0.0, 0.0, 0.0	Rotation angle of the texture in degrees. To just rotate in the "uv" plane provide the the rotation value as the 3rd component.
Offset	`float3`	0.0, 0.0, 0.0	Controls position of the texture on the object.
Tiling	`float3`	1.0, 1.0, 1.0	Controls the scale of the texture on the object. Higher values result on higher repetition of the image.

3.12   Blend colors

Blend colors is a helper function that can be used to combine multiple textures together or to modify the result of a texture function. It implements layer blend operations commonly known from painting applications.

Parameters:

name	*type*	*default*	*description*
Color 1	`color`	0% black	The base color/texture of the blend.
Color 2	`color`	100% white	The layer color/texture for the blend
Blend mode	`enum`	`color_ layer_blend`	Describes how Color 1 and Color 2 are combined.
Blend weight	`float`	1.0	Defines strength of the effect. At weight of 0.0, only "Color 1" will be visible. At weight 1.0, the blend function will have full effect.

The return value of the "Blend color" function therefore is:

```
"Blend function" * "Blend weight" + "Color1" * ( 1- "Blend weight")
```

For colors A and B, the following is a list of all blend modes as well as a short description of the matching blend function.

Blend mode	Blend function
`color_layer_blend`	B
`color_layer_add`	B + A
`color_layer_multiply`	B * A
`color_layer_screen`	1 - ((1 - B) * (1 - A))
`color_layer_overlay`	For each channel individualy: if A < 0.5: B * A * 2, else: 2 * (B + A - B * A - 0.5)
`color_layer_brightness`	Hue of the A layer combined with the intensity of the B
`color_layer_color`	Intensity of the A layer combined with the hue of the B
`color_layer_exclusion`	A + B - A * B * 2
`color_layer_average`	Average of B and A layer
`color_layer_lighten`	Maximum of B and A layer
`color_layer_darken`	Minimum of B and A layer
`color_layer_sub`	A + B - 1
`color_layer_negation`	1 - math::abs(1 - (A + B))
`color_layer_difference`	Absolute difference of B and A layer
`color_layer_softlight`	(B < 0.5) ? 2 * (B * A + A * A * (0.5 - B)) : 2 * (math::sqrt(A) * (B - 0.5) + A - B * A)
`color_layer_colordodge`	A / (1 - B)
`color_layer_reflect`	A * A/(1 - B)
`color_layer_colorburn`	1 - (1 - A)/B
`color_layer_phoenix`	Minimum of both layers minus the maximum of both layers (plus 1.0)
`color_layer_hardlight`	For each channel individualy: if B > 0.5: B * A * 2, else: 2 * (B + A - B * A - 0.5)
`color_layer_hardmix`	For each channel individualy: (B + A <= 1) ? 0 : 1
`color_layer_lineardodge`	B + A (clamped)
`color_layer_linearburn`	A + B - 1 (clamped)
`color_layer_spotlight`	2 * B * A
`color_layer_spotlightblend`	B * A + A
`color_layer_hue`	Uses hue from B layer, saturation and brightness from A
`color_layer_saturation`	Uses saturation from B layer, hue and brightness from A