



# Material Definition Language

## Core definitions

9 January 2023  
Version 1.4



---

## Material Definition Language – Core definitions

### Copyright Information

© 2022 NVIDIA Corporation. All rights reserved.

Document build number 363600.1913

---

# Contents

1	Introduction	1
2	Materials and building blocks	2
2.1	Simple materials	2
2.1.1	Simple diffuse	2
2.1.2	Metal	3
2.1.3	Plastic	4
2.1.4	Retroreflective	5
2.1.5	Flexible material model	6
2.1.6	Thin glass	8
2.1.7	Thin translucent	9
2.1.8	Thick glass	10
2.1.9	Thick translucent	11
2.2	Modifier materials	12
2.2.1	Surface blender	12
2.2.2	Surface falloff	13
2.2.3	Apply clear coating	14
2.2.4	Apply thin film	15
2.2.5	Apply thin metal coating	16
2.2.6	Apply a cover of dust	17
2.2.7	Apply a color falloff	18
2.2.8	Apply flake coating	19
2.2.9	Flaky paint	20
2.2.10	Add cut-outs	21
2.2.11	Add simple sticker	22
2.2.12	Add global bumpmap	23
2.2.13	Add displacement	24
2.2.14	Add emission	25
2.2.15	Add thermal emission	26
2.3	Emissive materials	27
2.3.1	Diffuse emission	27
2.3.2	Spotlight emission	28
2.3.3	IES file based emission	29
3	Texturing functions	30
3.1	Blend colors	30
3.2	Bitmap texture	31

---

3.3	Perlin noise texture	32
3.4	Perlin noise texture – bump mapping	34
3.5	Cellular noise texture	35
3.6	Cellular noise texture – bump mapping	37
3.7	Flow noise texture	38
3.8	Flow noise texture – bump mapping	40
3.9	3D checker texture	42
3.10	3D checker texture – bump mapping	43
3.11	Bitmap texture, bump	44
3.12	Normal map texture	45
3.13	Blend normals	46
4	Enumerations	47
4.1	User interface group	47
4.2	Emission mode	47
4.3	Worley noise cell type	47
4.4	Worley noise cell shape	47

---

# 1 Introduction

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

The core definition materials are listed in “[Materials and building blocks](#)” (page 2). The materials are divided into three groups:

## *Simple materials (page 2)*

Simple materials are used either individually to model visual appearance or as components when creating more complex materials with material combiners and material modifiers.

## *Modifier materials (page 12)*

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

## *Emissive materials (page 27)*

Emissive materials create light sources from objects by defining how light is emitted from an object’s surface.

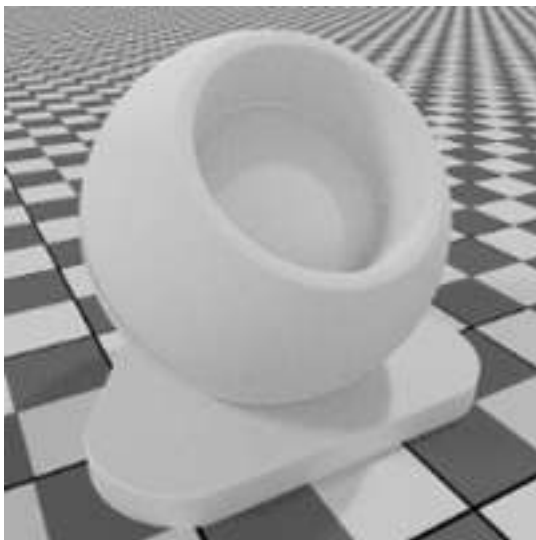
The functions and enumerations (“enums”) used by the core materials are described in “[Texturing functions](#)” (page 30) and “[Enumerations](#)” (page 47), respectively. For materials and functions, two tables describe their parameters. The first lists the “display names” used by applications for each parameter and a description of that parameter’s role in the material or function. The second table lists the display name along with that parameter’s data type, identifier, and default value. The tables in “[Enumerations](#)” (page 47) list the field names and their meaning.

---

## 2 Materials and building blocks

### 2.1 Simple materials

#### 2.1.1 Simple diffuse



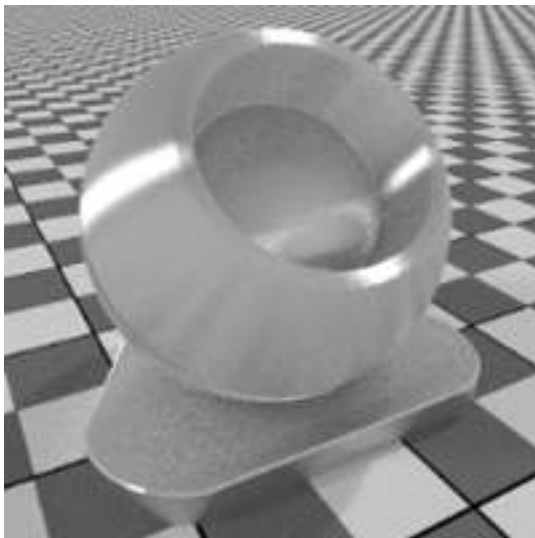
MDL identifier: `core_definitions::diffuse`

A basic diffuse material

<i>Display name</i>	<i>Description</i>
Color	The color of the material
Diffuse roughness	Higher roughness values lead a powdery appearance
Bumps	Attach bump or normal maps here

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Color	color	diffuse_color	<code>color(0.8)</code>
Diffuse roughness	float	roughness	<code>0.0</code>
Bumps	float3	normal	<code>state::normal()</code>

## 2.1.2 Metal



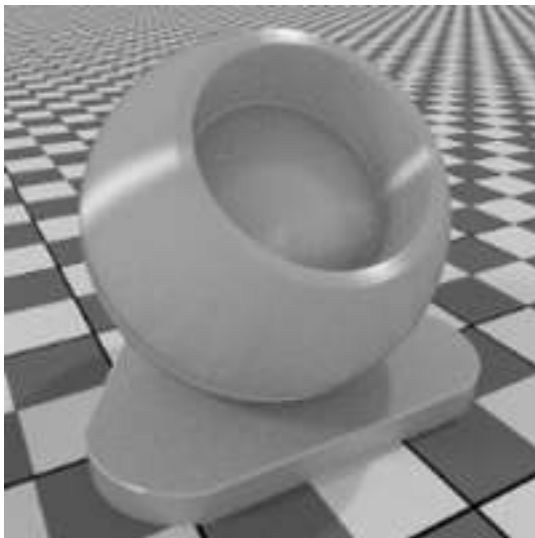
MDL identifier: `core_definitions::scratched_metal_v2`

A metallic material with stretched reflections

<i>Display name</i>	<i>Description</i>
Color	The color of the metal
Roughness	Higher roughness values lead to bigger highlights and blurry reflections
Reflection weight	Intensity of highlights and glossy reflections and highlights
Anisotropy	Higher values will stretch the highlight
Anisotropy rotation	Changes the orientation of the anisotropy. A value of 1 will rotate the orientation 360 degrees.
Bumps	Attach bump or normal maps here

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Color	color	metal_color	color(0.9)
Roughness	float	roughness	0.2
Reflection weight	float	glossy_weight	0.9
Anisotropy	float	anisotropy	0.0
Anisotropy rotation	float	anisotropy_rotation	0.0
Bumps	float3	normal	state::normal()

## 2.1.3 Plastic



MDL identifier: `core_definitions::scratched_plastic_v2`

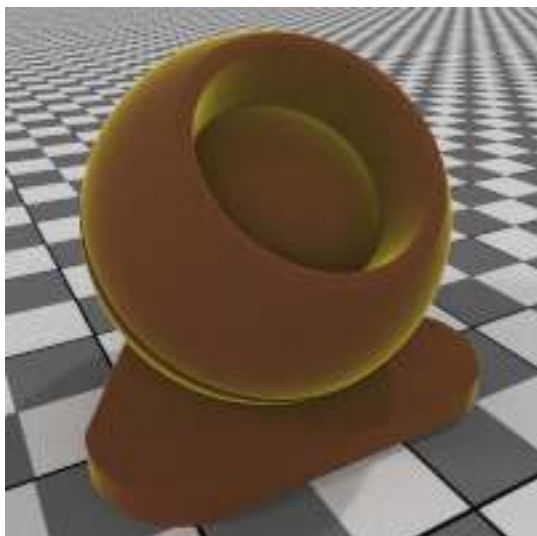
A basic dielectric, works for everything opaque that is not metallic. Supports stretched highlights.

<i>Display name</i>	<i>Description</i>
Color	The color of the material
Roughness	Higher roughness values lead to bigger highlights and blurry reflections
Anisotropy	Higher values will stretch the highlight
Anisotropy rotation	Changes the orientation of the anisotropy. A value of 1 will rotate the orientation 360 degrees.
IOR	Determines reflectivity
Reflection weight	Additional reflectivity control
Bumps	Attach bump or normal maps here

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Color	color	diffuse_color	color(0.5)
Roughness	float	roughness	0.2
Anisotropy	float	anisotropy	0.0
Anisotropy rotation	float	anisotropy_rotation	0.0
IOR	uniform float	ior	1.4
Reflection weight	float	reflectivity	1.0
Bumps	float3	normal	state::normal()



## 2.1.4 Retroreflective



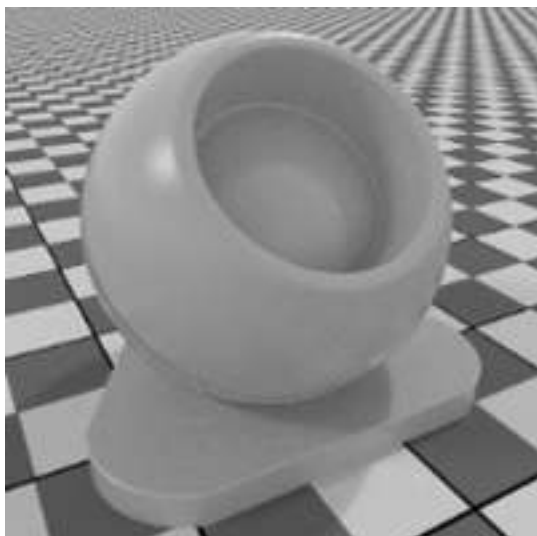
MDL identifier: `core_definitions::retroreflective`

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

<i>Display name</i>	<i>Description</i>
Color	The color of the material
Reflection color	The color of the retroreflection
Roughness	Higher roughness values lead to bigger highlights and blurry reflections
Reflection weight facing	Reflectivity control for geometry facing the viewer
Reflection weight edge	Reflectivity control for the reflectivity at geometry edges
Bumps	Attach bump or normal maps here

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Color	color	<code>base_color</code>	<code>color(0.2, 0.23, 0.23)</code>
Reflection color	color	<code>reflection_color</code>	<code>color(0.8, 0.8, 0.83)</code>
Roughness	float	<code>roughness</code>	<code>0.15</code>
Reflection weight facing	float	<code>normal_reflectivity</code>	<code>0.05</code>
Reflection weight edge	float	<code>grazing_reflectivity</code>	<code>1.0</code>
Bumps	float3	<code>normal</code>	<code>state::normal()</code>

## 2.1.5 Flexible material model



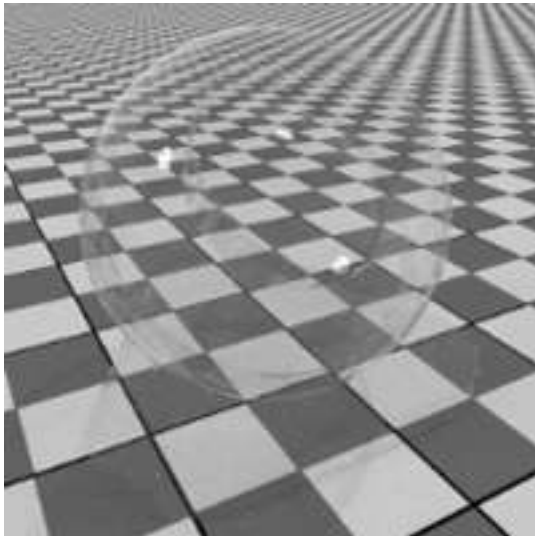
MDL identifier: `core_definitions::flex_material_v2`

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

<i>Display name</i>	<i>Description</i>
Base color	The color of the material
Diffuse roughness	Higher roughness values lead to a more “powdery” look
Metallic material	If 1.0, reflection will be colored and independent of view direction. If 0.0, reflection will be white and direction dependent. Directional dependence is in this case based on the IOR (Fresnel effect).
Reflection weight	Controls the amount of reflection
Reflection roughness	Higher roughness values lead to more blurry reflections
Reflection anisotropy	Higher values will stretch the highlight
Anisotropy rotation	Changes the orientation of the anisotropy. A value of 1 will rotate the orientation 360 degrees.
Transmission weight	Weights how much light passes through the object compared to its diffuse reflectivity.
Transmission color	Color effect for transmission independent of thickness of the object, similar to stained glass.
Volume color	If the material is not “Thin walled”, “Volume color” will be reached at “Volume reference distance” (m).
Transmission roughness	higher values lead to objects seen through the material to appear blurry
Volume reference distance	If the material is not “Thin walled”, “Volume color” will be reached at this distance (m). Enter a typical thickness of objects made of this material here.
IOR	Determines refraction in the volume. It also influences the reflectivity for materials that are not metallic.
Thin walled	Thin walled materials do not refract and do not have volume effects. Good for soap bubbles or window glass.
Bumps	Attach bump or normal maps here
Abbe number	Controls dispersion. A value of 0 switches dispersion off. Dispersive materials have Abbe numbers between 25 and 85

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Base color	color	base_color	color(0.5)
Diffuse roughness	float	diffuse_roughness	0.0
Metallic material	float	is_metal	0.0
Reflection weight	float	reflectivity	1.0
Reflection roughness	float	reflection_roughness	0.3
Reflection anisotropy	float	anisotropy	0.0
Anisotropy rotation	float	anisotropy_rotation	0.0
Transmission weight	float	transparency	0.0
Transmission color	color	transmission_color	color(1.0)
Volume color	uniform color	volume_color	color(1.0)
Transmission roughness	float	transmission_roughness	0.0
Volume reference distance	uniform float	base_thickness	0.1
IOR	uniform float	ior	1.5
Thin walled	uniform bool	thin_walled	false
Bumps	float3	normal	state::normal()
Abbe number	uniform float	abbe_number	0.0

## 2.1.6 Thin glass



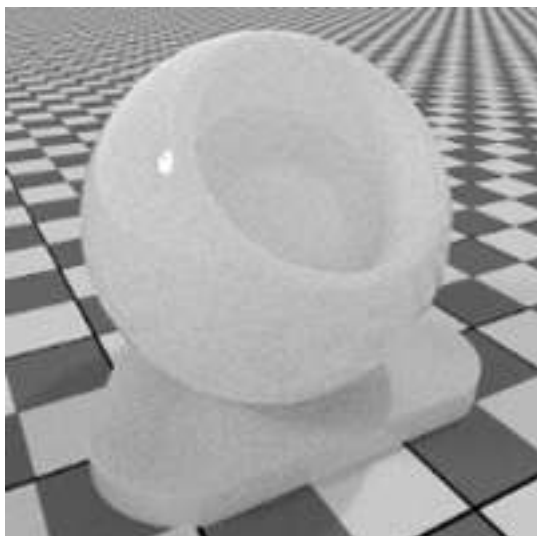
MDL identifier: `core_definitions::thin_glass_v2`

A basic transmissive dielectric without refraction or volume

<i>Display name</i>	<i>Description</i>
Transmission color	The color of the material
Roughness	Higher roughness values lead to bigger highlights and blurry reflections
IOR	Determines reflectivity
Bumps	Attach bump or normal maps here

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Transmission color	color	glass_color	color(0.95)
Roughness	float	roughness	0.0
IOR	uniform float	ior	1.4
Bumps	float3	normal	state::normal()

## 2.1.7 Thin translucent



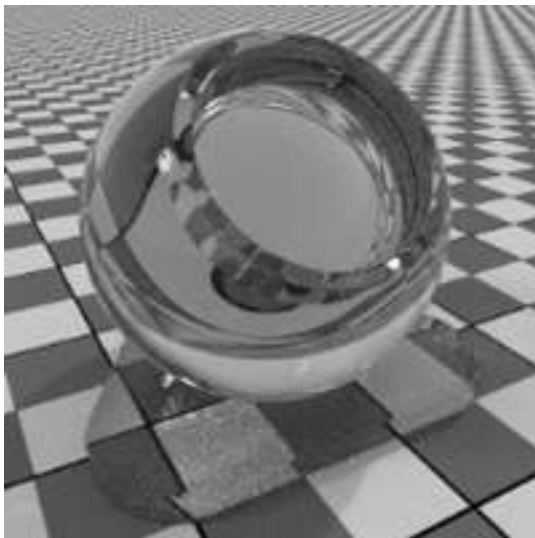
MDL identifier: `core_definitions::thin_translucent_v2`

A diffuse transmissive dielectric material

<i>Display name</i>	<i>Description</i>
Diffuse color	The color of the material
Translucence color	The color of the volume of the material
Translucence weight	Fraction of the incoming light that should be visible on the backside
Roughness	Higher roughness values lead to bigger highlights and blurry reflections
IOR	Determines reflectivity
Bumps	Attach bump or normal maps here

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Diffuse color	color	surface_color	color(0.95)
Translucence color	color	translucent_color	color(0.95)
Translucence weight	float	translucency	0.5
Roughness	float	roughness	0.0
IOR	uniform float	ior	1.4
Bumps	float3	normal	state::normal()

## 2.1.8 Thick glass



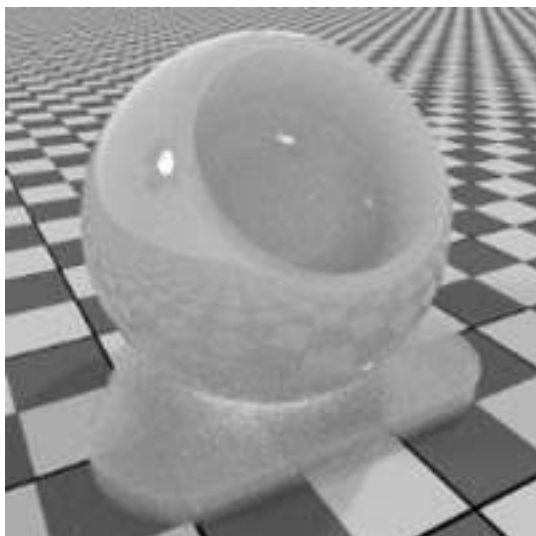
MDL identifier: `core_definitions::thick_glass_v2`

A basic transmissive dielectric with refraction and coloring in the volume

<i>Display name</i>	<i>Description</i>
Transmission color	Colors the light entering the volume, similar to stained glass
Volume color	The color of the glass body
Roughness	Higher roughness values lead to bigger highlights and blurry reflections
IOR	Determines reflectivity as well as amount of refraction
Volume reference distance	“Volume color” will be reached at this distance (m). Enter a typical thickness of objects made of this material here.
Bumps	Attach bump or normal maps here
Abbe number	Controls dispersion. 0 switches dispersion off. Dispersive materials have Abbe numbers between 25 and 85.

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Transmission color	color	<code>transmission_color</code>	<code>color(1.0)</code>
Volume color	uniform color	<code>glass_color</code>	<code>color(0.95)</code>
Roughness	float	<code>roughness</code>	<code>0.0</code>
IOR	uniform float	<code>ior</code>	<code>1.4</code>
Volume reference distance	uniform float	<code>base_thickness</code>	<code>0.1</code>
Bumps	float3	<code>normal</code>	<code>state::normal()</code>
Abbe number	uniform float	<code>abbe_number</code>	<code>0.0</code>

## 2.1.9 Thick translucent



MDL identifier: `core_definitions::thick_translucent_v2`

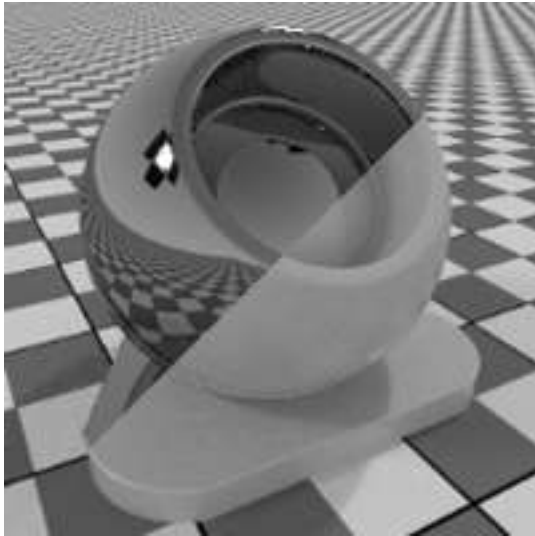
A subsurface scattering material

<i>Display name</i>	<i>Description</i>
Transmission color	The color of the material
Volume color	The color of the volume at "Volume reference distance"
Volume scattering	Amount of scattering for objects at "Volume reference distance"
Reflection roughness	Higher roughness values lead to bigger highlights and blurry reflections
Reflection weight	Overall reflectivity of the material
Volume reference distance	"Volume color" and "Volume scattering" will be reached at this distance (m). Enter a typical thickness of objects made of this material here.
Bumps	Attach bump or normal maps here
IOR	Determines reflectivity as well as amount of refraction
Abbe number	Controls dispersion. A value of 0 switches dispersion off, Dispersive materials have Abbe numbers between 25 and 85.

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Transmission color	color	<code>transmission_color</code>	<code>color(0.95)</code>
Volume color	uniform color	<code>volume_color</code>	<code>color(0.95)</code>
Volume scattering	uniform float	<code>volume_scattering</code>	<code>0.5</code>
Reflection roughness	float	<code>roughness</code>	<code>0.0</code>
Reflection weight	float	<code>reflectivity</code>	<code>1.0</code>
Volume reference distance	uniform float	<code>base_thickness</code>	<code>0.1</code>
Bumps	float3	<code>normal</code>	<code>state::normal()</code>
IOR	uniform float	<code>ior</code>	<code>1.4</code>
Abbe number	uniform float	<code>abbe_number</code>	<code>0.0</code>

## 2.2 Modifier materials

### 2.2.1 Surface blender



MDL identifier: `core_definitions::blend`

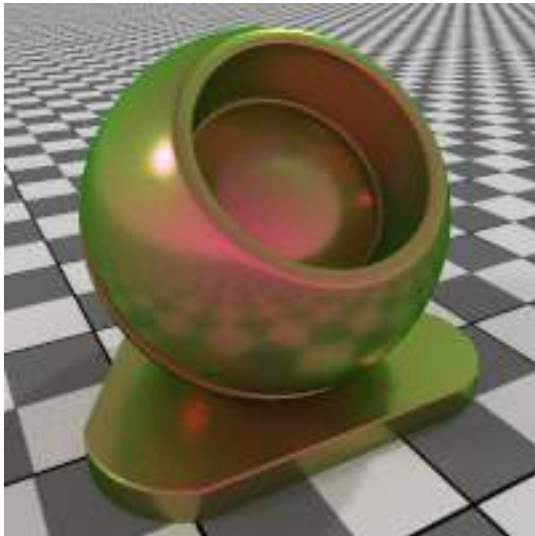
Blend surface characteristics of 2 materials or mask them using a texture

<i>Display name</i>	<i>Description</i>
Base material	The material the blend is based on
Blend Material	Surface properties to use for the blend
Blend weight	Blend weight or mask texture

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Base material	material	base	scratched_plastic_v2()
Blend Material	material	blend	scratched_metal_v2()
Blend weight	float	weight	0.0



## 2.2.2 Surface falloff



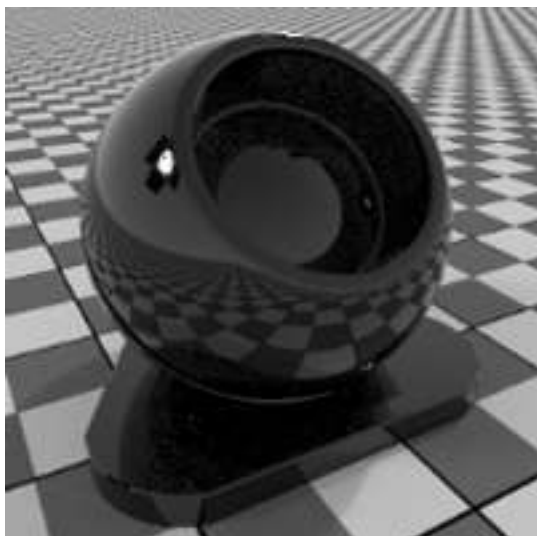
MDL identifier: `core_definitions::surface_falloff`

Blend surface characteristics of 2 materials or mask them using a texture

<i>Display name</i>	<i>Description</i>
Base material	The material the blend is based on
Blend Material	Surface properties to use for the blend
Blend weight facing	Blend weight or mask texture
Blend weight edge	Blend weight or mask texture
Blend bias	Controls how fast the transition should happen. 5.0 results in fresnel like transition.

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Base material	material	base	scratched_plastic_v2()
Blend Material	material	blend	scratched_metal_v2()
Blend weight facing	float	facing_weight	0.0
Blend weight edge	float	edge_weight	1.0
Blend bias	float	blend_bias	1.0

### 2.2.3 Apply clear coating



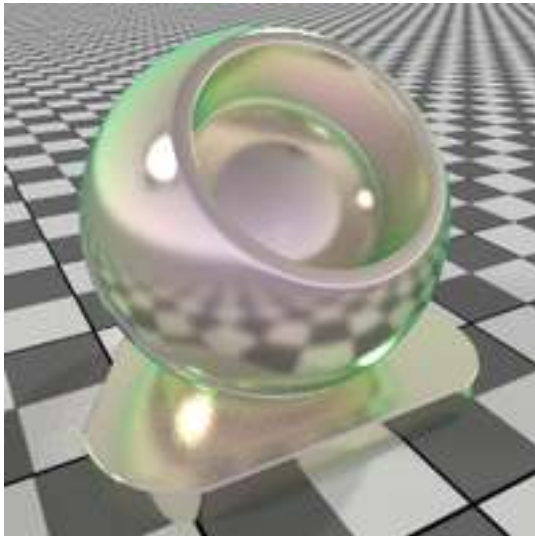
MDL identifier: `core_definitions::apply_clearcoat_v2`

Apply clear coat to an existing material

<i>Display name</i>	<i>Description</i>
Base material	The material that will get a clear coating applied
IOR	Determines reflectivity of the clear coat
Reflection roughness	Determines roughness of the clear coat
Coat visibility	Determines visibility of the clear coat
Bumps	Attach bump or normal maps here
Coat filter color	For simulating coatings with colored resins that modulate the color of underlying layers

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Base material	material	base	<code>diffuse(color(0.0))</code>
IOR	uniform float	ior	1.6
Reflection roughness	float	roughness	0.0
Coat visibility	float	visibility	1.0
Bumps	float3	normal	<code>state::normal()</code>
Coat filter color	color	coat_filter_color	<code>color(1.0)</code>

### 2.2.4 Apply thin film



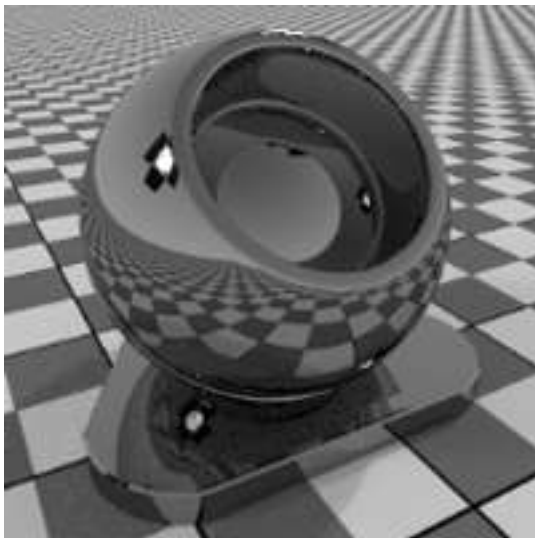
MDL identifier: `core_definitions::apply_thinfilm`

Apply thin film to an existing material

<i>Display name</i>	<i>Description</i>
Base material	The material that will get shows a thin film effect
IOR	The IOR of the thin film interface
Thickness	Thickness of the thin film in nm

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Base material	material	base	scratched_metal_v2()
IOR	uniform float	ior	1.6
Thickness	float	thickness	400.0

### 2.2.5 Apply thin metal coating



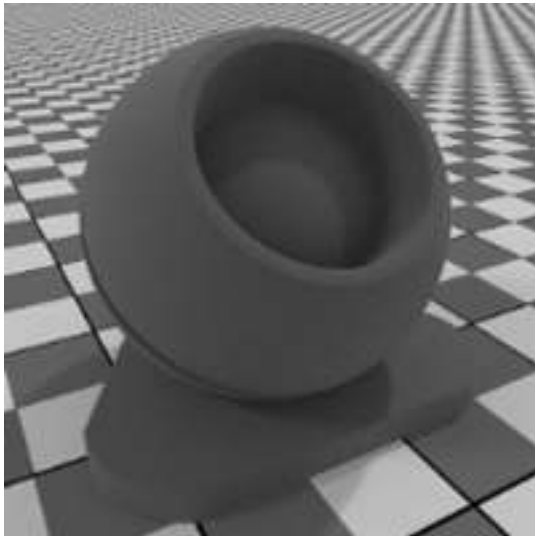
MDL identifier: `core_definitions::apply_metalcoat_v2`

Apply metal coat to an existing material

<i>Display name</i>	<i>Description</i>
Base material	The material that will get a metallic coating applied
Reflection color	The color of the metal
Reflection weight	The opacity of the coat
Reflection roughness	Determines roughness of the metal coat
Bumps	Attach bump or normal maps here

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Base material	material	base	<code>diffuse(color(0.0))</code>
Reflection color	color	metal_color	<code>color(0.95)</code>
Reflection weight	float	visibility	<code>0.3</code>
Reflection roughness	float	roughness	<code>0.0</code>
Bumps	float3	normal	<code>state::normal()</code>

## 2.2.6 Apply a cover of dust



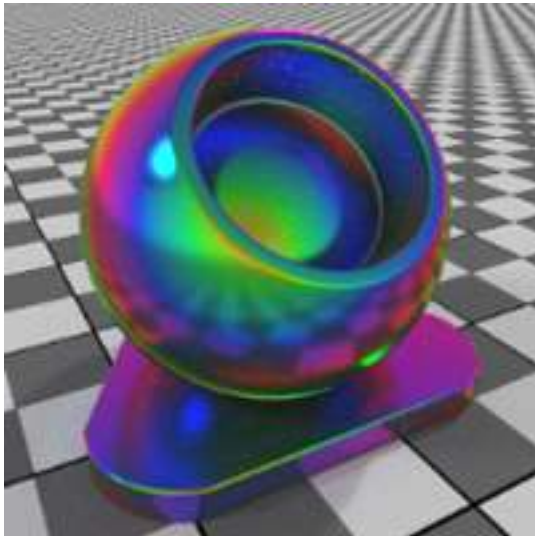
MDL identifier: `core_definitions::apply_dustcover`

Apply a diffuse cover of dust or dirt

<i>Display name</i>	<i>Description</i>
Base material	The material that will get a clear coating applied
Dust color	The color of the dust
Dust density	The opacity of the cover
Dust amount	How dusty the material is
Bumps	Attach bump or normal maps here

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Base material	material	base	<code>diffuse(color(0.0))</code>
Dust color	color	dust_color	<code>color(0.7)</code>
Dust density	float	visibility	1.0
Dust amount	uniform float	dust_density	0.5
Bumps	float3	normal	<code>state::normal()</code>

### 2.2.7 Apply a color falloff



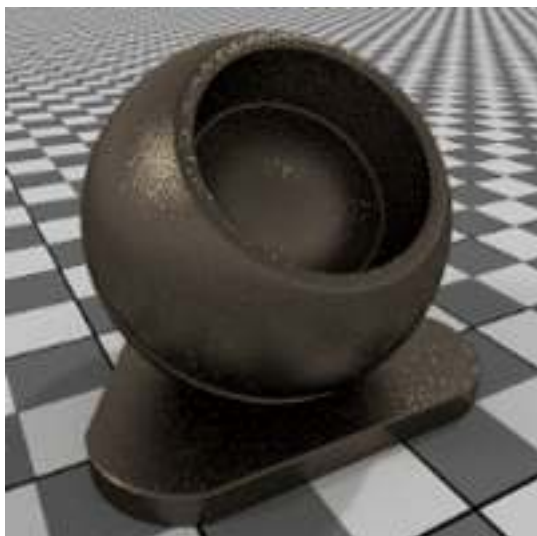
MDL identifier: `core_definitions::apply_colorfalloff_v2`

Makes the color view dependent

<i>Display name</i>	<i>Description</i>
Base material	The material that will get a clear coating applied
Color 1	Color 1 (facing direction)
Color 2	Color 2
Color 3	Color 3
Color 4	Color 4
Color 5	Color 5 (object edges)

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Base material	material	base	<code>scratched_metal_v2(metal_color: color(1.0))</code>
Color 1	uniform color	color_0	<code>color(1.0, 1.0, 1.0)</code>
Color 2	uniform color	color_1	<code>color(0.0, 0.0, 0.0)</code>
Color 3	uniform color	color_2	<code>color(0.0, 0.0, 0.0)</code>
Color 4	uniform color	color_3	<code>color(1.0, 1.0, 1.0)</code>
Color 5	uniform color	color_4	<code>color(0.0, 0.0, 0.0)</code>

### 2.2.8 Apply flake coating



MDL identifier: `core_definitions::apply_metallicflakes`

Apply layer of metallic flakes to an existing material

<i>Display name</i>	<i>Description</i>
Base material	The material that will get a flake layer applied
Color	The color of the flakes
Roughness	Determines roughness of the metallic flakes
Flake size	Determines size of the metallic flakes, in mm
Flake amount	Determines amount of visible metallic flakes
Flake opacity	Determines visibility of the metallic flakes
Flake orientation randomness	Larger numbers will increase the sparkle radius around highlights

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Base material	material	base	<code>diffuse(color(0.0))</code>
Color	color	flake_color	<code>color(0.9,0.9,0.9)</code>
Roughness	float	roughness	0.0
Flake size	uniform float	size	1.0
Flake amount	uniform float	amount	0.5
Flake opacity	uniform float	opacity	0.5
Flake orientation randomness	uniform float	bump	1.0

## 2.2.9 Flaky paint



MDL identifier: `core_definitions::flake_paint`

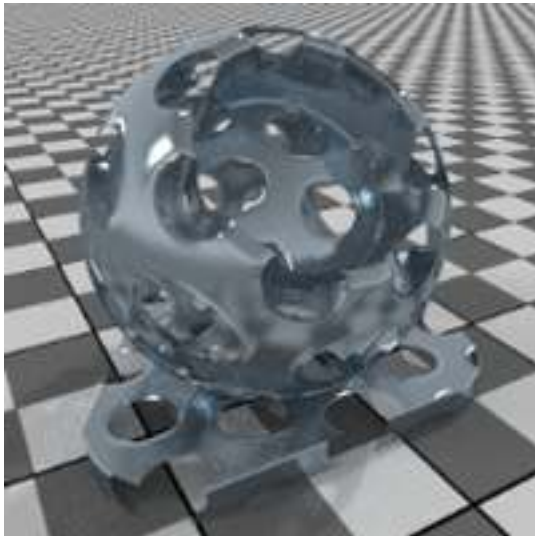
A multi layer paint material containing flakes

<i>Display name</i>	<i>Description</i>
Base color	The color of the base paint
Flake color	The color of the flakes
Flake roughness	Determines roughness of the metallic flakes
Flake size	Determines size of the metallic flakes, in mm
Flake amount	Determines amount of visible metallic flakes
Flake weight	Determines visibility of the metallic flakes
Flake orientation randomness	Larger numbers will increase the sparkle radius around highlights
IOR	Determines reflectivity of the clear coat
Coat roughness	Determines roughness of the clear coat
Coat bump	Attach bump or normal maps here

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Base color	color	base_color	color(0.3, 0.31, 0.31)
Flake color	color	flake_color	color(0.6, 1, 0.6)
Flake roughness	float	roughness	0.15
Flake size	uniform float	size	1.0
Flake amount	uniform float	amount	0.4
Flake weight	uniform float	opacity	0.8
Flake orientation randomness	uniform float	bump	1.0
IOR	uniform float	ior	1.6
Coat roughness	float	coat_roughness	0.0
Coat bump	float3	coat_bump	state::normal()



## 2.2.10 Add cut-outs



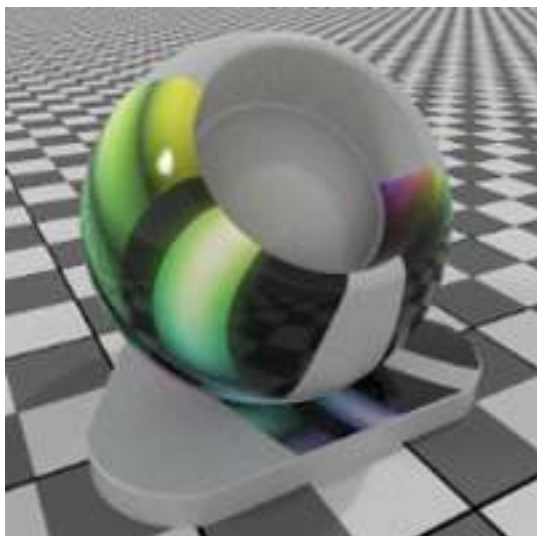
MDL identifier: `core_definitions::add_cutout`

Adds cut-outs to existing materials. Also forces material to be thin-walled. Good for modeling leaves, grass or fences.

<i>Display name</i>	<i>Description</i>
Base material	The material that will get a cut-out
Cutout	Determines where the object is visible

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Base material	material	base	<code>scratched_plastic_v2()</code>
Cutout	float	cutout	1.0

## 2.2.11 Add simple sticker



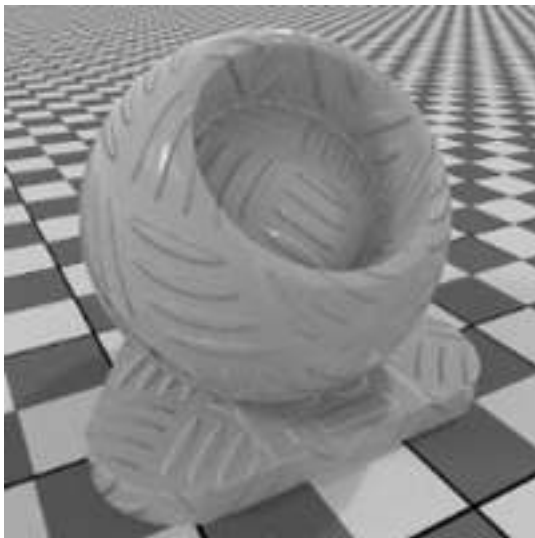
MDL identifier: `core_definitions::add_simple_sticker`

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

<i>Display name</i>	<i>Description</i>
Sticker color	The color of the material
Sticker roughness	Higher roughness values lead to bigger highlights and blurry reflections
Sticker IOR	Determines reflectivity
Sticker reflectivity	Additional Reflectivity control
Sticker bumps	Attach bump or normal maps here
Sticker mask	Determines extent of the sticker
Base material	The material that will get a sticker added

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Sticker color	color	<code>diffuse_color</code>	<code>color(0.5)</code>
Sticker roughness	float	<code>roughness</code>	<code>0.05</code>
Sticker IOR	uniform float	<code>ior</code>	<code>1.4</code>
Sticker reflectivity	float	<code>reflectivity</code>	<code>1.0</code>
Sticker bumps	float3	<code>sticker_normal</code>	<code>state::normal()</code>
Sticker mask	float	<code>sticker_mask</code>	<code>0.0</code>
Base material	material	<code>base</code>	<code>scratched_plastic_v2()</code>

## 2.2.12 Add global bumpmap



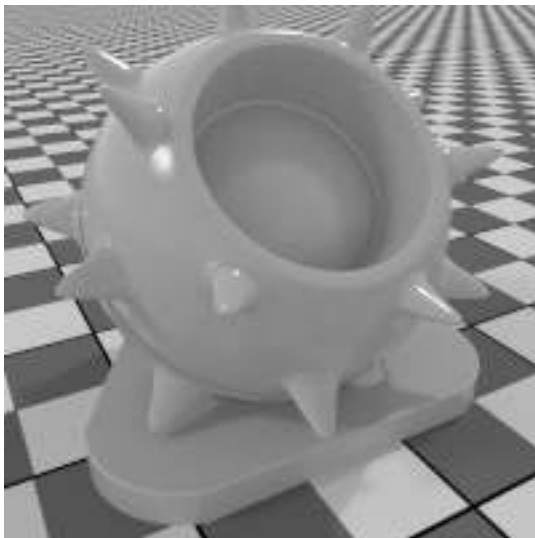
MDL identifier: `core_definitions::add_globalbump`

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

<i>Display name</i>	<i>Description</i>
Base material	The material that will get a bump map
Bump	An additional global bump map for the material. Local bump map of the base material is preserved.

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Base material	material	base	scratched_plastic_v2()
Bump	float3	normal	state::normal()

## 2.2.13 Add displacement



MDL identifier: `core_definitions::add_displacement`

Adds displacement to existing materials

<i>Display name</i>	<i>Description</i>
Base material	The material that will get a bump map
Displacement amount	Attach displacement texture here. Note that the object needs to be set up correctly to have good displacement results.
Displacement scale	A global scale factor for the displacement amount

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Base material	material	base	scratched_plastic_v2()
Displacement amount	float	displacement	0.0
Displacement scale	uniform float	displacement_scale	1.0

## 2.2.14 Add emission



MDL identifier: `core_definitions::add_emission`

Adds emission to a material

<i>Display name</i>	<i>Description</i>
Base material	The material that will get emission added
Color	The color of the light
Intensity	The brightness of the light source
Unit scale	Modeling unit to meter conversion factor
Unit for emission	The physical unit of "Intensity"

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Base material	material	base	<code>diffuse(color(0.0))</code>
Color	color	tint	<code>color(1.0)</code>
Intensity	uniform float	intensity	1000
Unit scale	uniform float	unit_scale	1.0
Unit for emission	uniform emission_type	unit	lumen_m2

## 2.2.15 Add thermal emission



MDL identifier: `core_definitions::add_thermal_emission`

Adds emission to a material, color is based on a “color temperature”

<i>Display name</i>	<i>Description</i>
Base material	The material that will get emission added
Temperature	The color temperature of the light in Kelvin
Intensity	The brightness of the light source
Unit for emission	The physical unit of “Intensity”

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Base material	material	base	<code>diffuse(color(0.0))</code>
Temperature	uniform float	temperature	6500.0
Intensity	uniform float	intensity	1000
Unit for emission	uniform emission_type	unit	lumen_m2

## 2.3 Emissive materials

### 2.3.1 Diffuse emission



MDL identifier: `core_definitions::light_omni`

Emissive material emitting in all directions

<i>Display name</i>	<i>Description</i>
Color	The color of the light
Intensity	The brightness of the light source
Unit scale	Modeling unit to meter conversion factor
Unit for emission	The physical unit of "Intensity"

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Color	color	tint	color(1.0)
Intensity	uniform float	intensity	1000
Unit scale	uniform float	unit_scale	1.0
Unit for emission	uniform emission_type	unit	lumen_m2

## 2.3.2 Spotlight emission



MDL identifier: `core_definitions::light_spot`

Emissive material emitting focused in one direction

<i>Display name</i>	<i>Description</i>
Color	The color of the light
Intensity	The brightness of the light source
Unit scale	Modeling unit to meter conversion factor
Spot focus	larger values lead to more focused spotlights
Unit for emission	The physical unit of "Intensity"

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Color	color	tint	color(1.0)
Intensity	uniform float	intensity	1000
Unit scale	uniform float	unit_scale	1.0
Spot focus	uniform float	spot_exponent	30
Unit for emission	uniform emission_type	unit	lumen_m2



## 2.3.3 IES file based emission



MDL identifier: `core_definitions::light_ies`

Emissive material emitting as described in an IES file

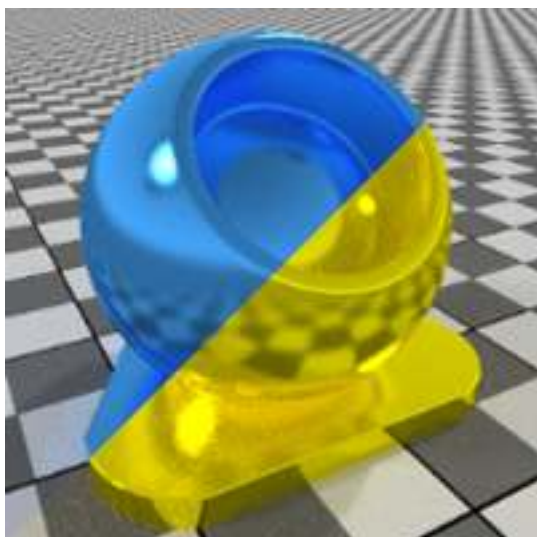
<i>Display name</i>	<i>Description</i>
IES light profile data	Data to describe the distribution of the light
Color	The color of the light
Intensity	The brightness of the light source
Unit scale	Modeling unit to meter conversion factor

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
IES light profile data	uniform light_profile	profile	
Color	color	tint	color(1.0)
Intensity	uniform float	intensity	1
Unit scale	uniform float	unit_scale	1.0

---

## 3 Texturing functions

### 3.1 Blend colors



MDL identifier: `core_definitions::blend_colors`

Allows combining textures and colors in varied ways

<i>Display name</i>	<i>Description</i>
Color 1	
Color 2	
Blend mode	Describes how Color 1 and Color 2 are combined
Blend weight	Defines strength of the effect. At weight 0, only color 1 will be visible. At weight 1, the blend function will have full effect.
Linear blend	The blend operation can either be applied in linear or gamma (2.2) color space

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Color 1	color	color_1	color(0.0)
Color 2	color	color_2	color(1.0)
Blend mode	uniform color_layer_mode	mode	color_layer_blend
Blend weight	float	weight	1.0
Linear blend	uniform bool	linear_blend	true

## 3.2 Bitmap texture



**FILE**

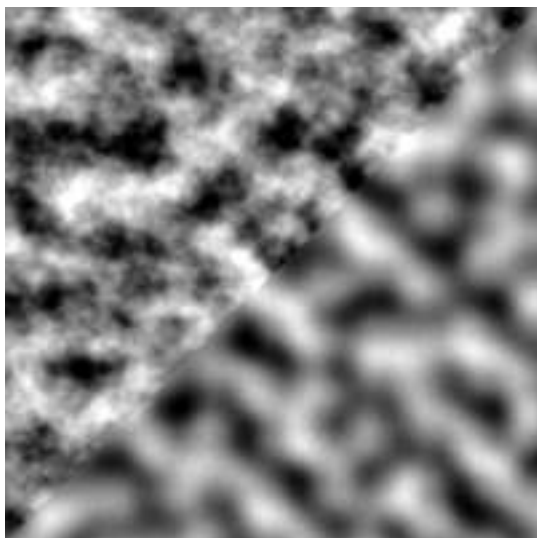
MDL identifier: `core_definitions::file_texture`

Allows texturing using image files of various file formats

<i>Display name</i>	<i>Description</i>
Bitmap file	
Scalar mode	Defines how the texture applies to scalar parameters
Brightness	
Contrast	
Tiling	Controls the scale of the texture on the object
Offset	Controls position of the texture on the object
Rotation	Rotation angle of the texture in degrees
Clip	If set to true, texture will not repeat. Outside of the texture, color will be black and the scalar value will be 0.
UV space index	Selects a specific UV space
Invert image	Invert image

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Bitmap file	uniform texture_2d	texture	
Scalar mode	uniform mono_mode	mono_source	mono_average
Brightness	uniform float	brightness	1.0
Contrast	uniform float	contrast	1.0
Tiling	uniform float2	scaling	float2(1.0)
Offset	uniform float2	translation	float2(0.0)
Rotation	uniform float	rotation	0.0
Clip	uniform bool	clip	false
UV space index	uniform int	texture_space	0
Invert image	uniform bool	invert	false

### 3.3 Perlin noise texture



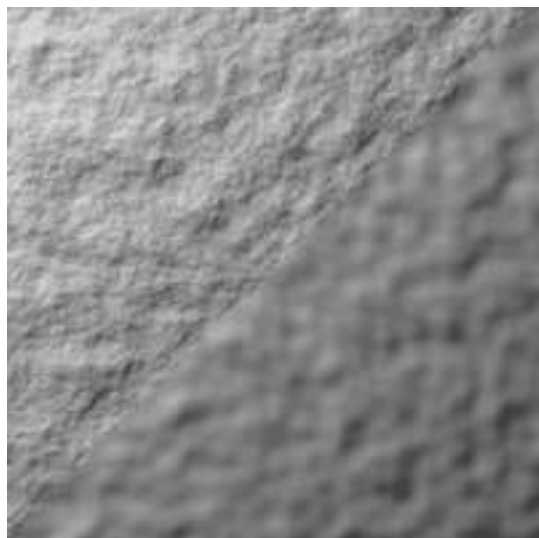
MDL identifier: `core_definitions::perlin_noise_texture`

Enable texturing with a random noise pattern

<i>Display name</i>	<i>Description</i>
Color 1	
Color 2	
Use object space	If off, UV 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.
Levels	Higher amounts will add detail to the noise
Billowing appearance	
Upper threshold	Lowering this value will create bigger areas uniformly colored with Color 1
Lower threshold	Increasing this value will create bigger areas uniformly colored with Color 2
Tiling	Controls the scale of the texture on the object
Offset	Controls position of the texture on the object
Rotation	Rotation angle of the texture in degrees
UV space index	Only applies if "Use object space" is off. Selects a specific UV space.

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Color 1	color	color1	color(1.0)
Color 2	color	color2	color(0.0)
Use object space	uniform bool	object_space	true
Levels	uniform int	noise_levels	3
Billowing appearance	uniform bool	absolute_noise	false
Upper threshold	uniform float	noise_threshold_high	1.0
Lower threshold	uniform float	noise_threshold_low	0.0
Tiling	uniform float3	scaling	float3(10.0)
Offset	uniform float3	translation	float3(0.0)
Rotation	uniform float3	rotation	float3(0.0)
UV space index	uniform int	texture_space	0

### 3.4 Perlin noise texture – bump mapping



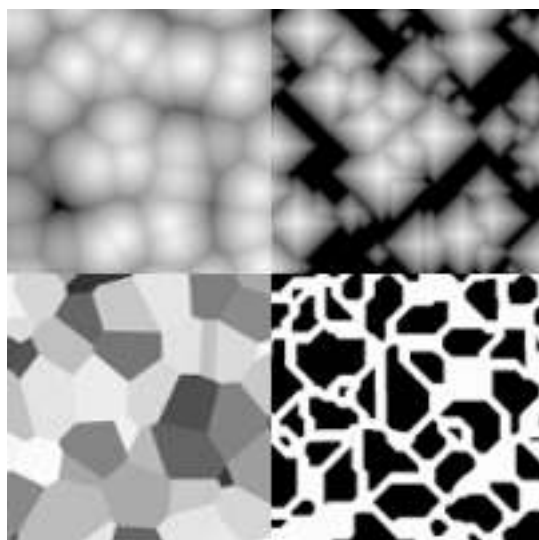
MDL identifier: `core_definitions::perlin_noise_bump_texture`

Enable bump-map texturing with a random noise pattern

<i>Display name</i>	<i>Description</i>
Bump strength	
Tiling	Controls the scale of the texture on the object
Levels	Higher amounts will add detail to the noise
Use object space	If off, UV 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.
Billowing appearance	
Upper threshold	Lowering this value will create bigger areas uniformly colored with Color 1
Lower threshold	Increasing this value will create bigger areas uniformly colored with Color 2
Offset	Controls position of the texture on the object
Rotation	Rotation angle of the texture in degrees
UV space index	Only applies if “Use object space” is off. Selects a specific UV space.

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Bump strength	uniform float	factor	1.0
Tiling	uniform float3	scaling	float3(10.0)
Levels	uniform int	noise_levels	1
Use object space	uniform bool	object_space	true
Billowing appearance	uniform bool	absolute_noise	false
Upper threshold	uniform float	noise_threshold_high	1.0
Lower threshold	uniform float	noise_threshold_low	0.0
Offset	uniform float3	translation	float3(0.0)
Rotation	uniform float3	rotation	float3(0.0)
UV space index	uniform int	texture_space	0

### 3.5 Cellular noise texture



MDL identifier: `core_definitions::worley_noise_texture`

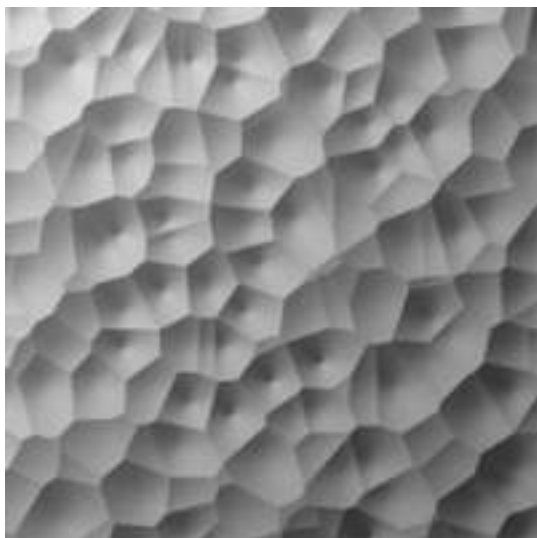
Allow texturing with a cell forming pattern

<i>Display name</i>	<i>Description</i>
Color 1	
Color 2	
Use object space	If off, UV 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	Only applies if "Use object space" is off. Selects a specific UV space.
Cell type	Cell pattern type
Cell shape	The shape of the cell form
Upper threshold	Lowering this value will create bigger areas uniformly colored with Color 1
Lower threshold	Increasing this value will create bigger areas uniformly colored with Color 2
Tiling	Controls the scale of the texture on the object
Offset	Controls position of the texture on the object
Rotation	Rotation angle of the texture in degrees

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Color 1	color	color1	color(1.0)
Color 2	color	color2	color(0.0)
Use object space	uniform bool	object_space	true
UV space index	uniform int	texture_space	0
Cell type	uniform cell_type	mode	simple_cells
Cell shape	uniform cell_base	metric	circular_cells
Upper threshold	uniform float	noise_threshold_high	1.0
Lower threshold	uniform float	noise_threshold_low	0.0
Tiling	uniform float3	scaling	float3(10.0)
Offset	uniform float3	translation	float3(0.0)
Rotation	uniform float3	rotation	float3(0.0)



### 3.6 Cellular noise texture – bump mapping



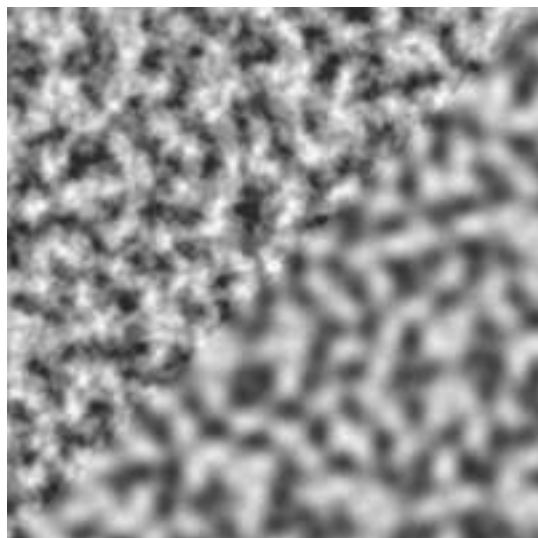
MDL identifier: `core_definitions::worley_noise_bump_texture`

Allow texturing with a cell forming pattern

<i>Display name</i>	<i>Description</i>
Bump strength	
Cell shape	The shape of the cell form
Use object space	If off, UV 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	Only applies if “Use object space” is off. Selects a specific UV space.
Upper threshold	Lowering this value will create bigger areas uniformly colored with Color 1
Lower threshold	Increasing this value will create bigger areas uniformly colored with Color 2
Tiling	Controls the scale of the texture on the object
Offset	Controls position of the texture on the object
Rotation	Rotation angle of the texture in degrees

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Bump strength	uniform float	factor	1.0
Cell shape	uniform cell_base	metric	circular_cells
Use object space	uniform bool	object_space	true
UV space index	uniform int	texture_space	0
Upper threshold	uniform float	noise_threshold_high	1.0
Lower threshold	uniform float	noise_threshold_low	0.0
Tiling	uniform float3	scaling	float3(10.0)
Offset	uniform float3	translation	float3(0.0)
Rotation	uniform float3	rotation	float3(0.0)

### 3.7 Flow noise texture



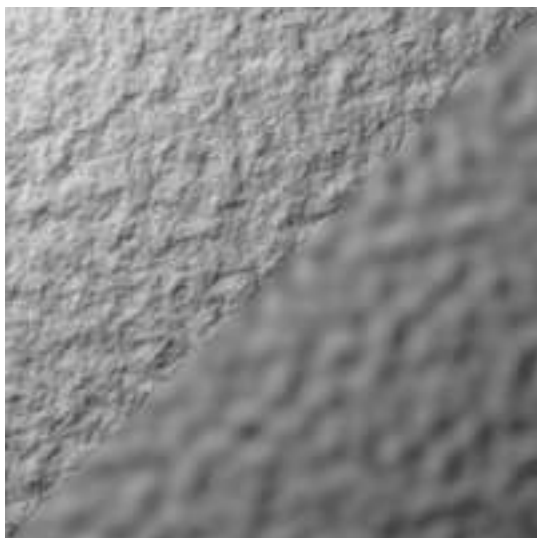
MDL identifier: `core_definitions::flow_noise_texture`

Allow texturing with a 2D noise pattern suitable for waves

<i>Display name</i>	<i>Description</i>
Color 1	
Color 2	
Use object space	If off, UV 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	Only applies if “Use object space” is off. Selects a specific UV space.
Levels	Higher amounts will add detail to the noise
Billowing appearance	
Phase offset	Controls the third dimension of the function
Level intensity gain	If multiple levels are used, this parameter specifies a weighting factor for subsequent levels
Level scaling	If multiple levels are used, this parameter specifies a global scaling factor for subsequent levels
Progressive u scale	If multiple levels are used, this parameter specifies an additional scaling factor in the “u” direction
Progressive v offset	If multiple levels are used, this parameter specifies an offset for subsequent levels in the “v” direction
Tiling	Controls the scale of the texture on the object
Offset	Controls position of the texture on the object
Rotation	Rotation angle of the texture in degrees

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Color 1	color	color1	color(1.0)
Color 2	color	color2	color(0.0)
Use object space	uniform bool	object_space	false
UV space index	uniform int	texture_space	0
Levels	uniform int	noise_levels	3
Billowing appearance	uniform bool	absolute_noise	false
Phase offset	uniform float	phase	0.0
Level intensity gain	uniform float	level_gain	0.5
Level scaling	uniform float	level_scale	2.0
Progressive u scale	uniform float	level_progressive_u_scale	1.0
Progressive v offset	uniform float	level_progressive_v_motion	0.0
Tiling	uniform float3	scaling	float3(10.0)
Offset	uniform float3	translation	float3(0.0)
Rotation	uniform float3	rotation	float3(0.0)

### 3.8 Flow noise texture – bump mapping



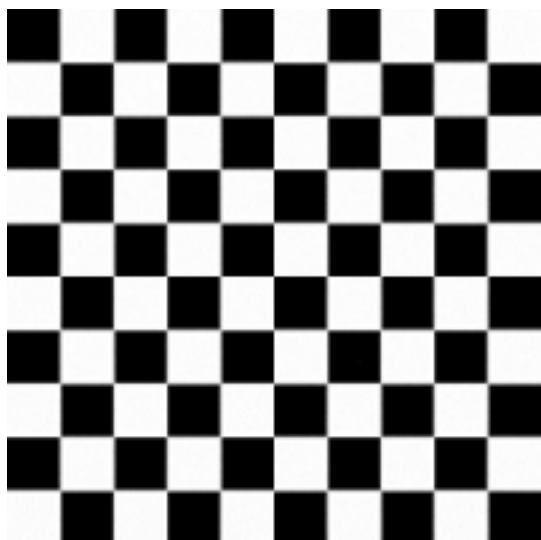
MDL identifier: `core_definitions::flow_noise_bump_texture`

Allow texturing with a 2D noise pattern suitable for waves

<i>Display name</i>	<i>Description</i>
Bump strength	
Tiling	Controls the scale of the texture on the object
Levels	Higher amounts will add detail to the noise
Use object space	If off, UV 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	Only applies if “Use object space” is off. Selects a specific UV space.
Billowing appearance	
Phase offset	Controls the 3rd dimension of the function
Level intensity gain	If multiple levels are used, this parameter specifies a weighting factor for subsequent levels
Level scaling	If multiple levels are used, this parameter specifies a global scaling factor for subsequent levels
Progressive u scale	If multiple levels are used, this parameter specifies an additional scaling factor in the “u” direction
Progressive v offset	If multiple levels are used, this parameter specifies an offset for subsequent levels in the “v” direction
Offset	Controls position of the texture on the object
Rotation	Rotation angle of the texture in degrees

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Bump strength	uniform float	factor	1.0
Tiling	uniform float3	scaling	float3(10.0)
Levels	uniform int	noise_levels	1
Use object space	uniform bool	object_space	false
UV space index	uniform int	texture_space	0
Billowing appearance	uniform bool	absolute_noise	false
Phase offset	uniform float	phase	0.0
Level intensity gain	uniform float	level_gain	0.5
Level scaling	uniform float	level_scale	2.0
Progressive u scale	uniform float	level_progressive_u_scale	1.0
Progressive v offset	uniform float	level_progressive_v_motion	0.0
Offset	uniform float3	translation	float3(0.0)
Rotation	uniform float3	rotation	float3(0.0)

### 3.9 3D checker texture



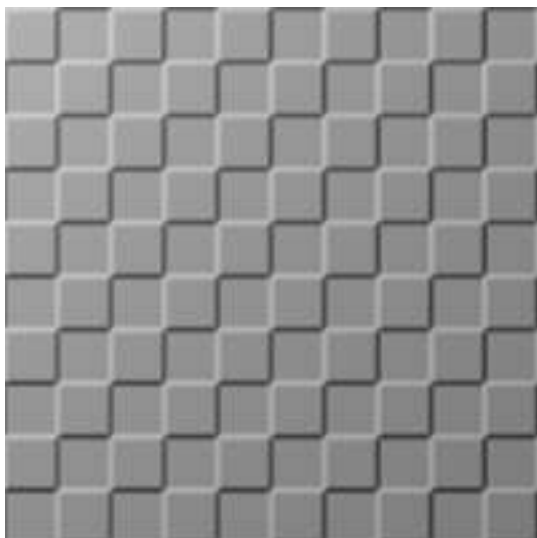
MDL identifier: `core_definitions::checker_texture`

Allows texturing a checkerboard pattern

<i>Display name</i>	<i>Description</i>
Color 1	
Color 2	
Tiling	Controls the scale of the texture on the object
Offset	Controls position of the texture on the object
Use object space	If off, UV 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.
Blur	
Rotation	Rotation angle of the texture in degrees
UV space index	Only applies if "Use object space" is off. Selects a specific UV space.

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Color 1	color	color1	color(1.0)
Color 2	color	color2	color(0.0)
Tiling	uniform float3	scaling	float3(10.0)
Offset	uniform float3	translation	float3(0.0)
Use object space	uniform bool	object_space	false
Blur	uniform float	blur	0
Rotation	uniform float3	rotation	float3(0.0)
UV space index	uniform int	texture_space	0

## 3.10 3D checker texture – bump mapping



MDL identifier: `core_definitions::checker_bump_texture`

Allows texturing a checkerboard pattern

<i>Display name</i>	<i>Description</i>
Bump strength	
Blur	
Use object space	If off, UV 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.
Tiling	Controls the scale of the texture on the object
Offset	Controls position of the texture on the object
Rotation	Rotation angle of the texture in degrees
UV space index	Only applies if "Use object space" is off. Selects a specific UV space.

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Bump strength	uniform float	factor	1.0
Blur	uniform float	blur	0
Use object space	uniform bool	object_space	false
Tiling	uniform float3	scaling	float3(10.0)
Offset	uniform float3	translation	float3(0.0)
Rotation	uniform float3	rotation	float3(0.0)
UV space index	uniform int	texture_space	0

### 3.11 Bitmap texture, bump



**FILE**

MDL identifier: `core_definitions::file_bump_texture`

Allows texturing using image files of various file formats

<i>Display name</i>	<i>Description</i>
Bitmap file	
Bump mode	Defines how the texture is evaluated to create the bumps
Tiling	Controls the scale of the texture on the object
Offset	Controls position of the texture on the object
Rotation	Rotation angle of the texture in degrees
Clip	If set to true, texture will not repeat. Outside of the texture the surface will be flat
Bump strength	
UV space index	Selects a specific UV space

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Bitmap file	uniform texture_2d	texture	
Bump mode	uniform mono_mode	bump_source	mono_average
Tiling	uniform float2	scaling	float2(1.0)
Offset	uniform float2	translation	float2(0.0)
Rotation	uniform float	rotation	0.0
Clip	uniform bool	clip	false
Bump strength	uniform float	factor	1
UV space index	uniform int	texture_space	0



## 3.12 Normal map texture



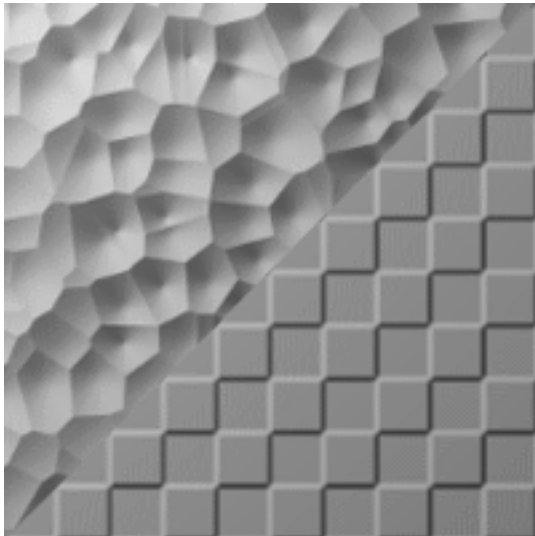
MDL identifier: `core_definitions::normalmap_texture`

Allows the use of tangent space normal maps

<i>Display name</i>	<i>Description</i>
Normalmap file	
Tiling	Controls the scale of the texture on the object
Offset	Controls position of the texture on the object
Rotation	Rotation angle of the texture in degrees
Clip	If set to true, texture will not repeat. Outside of the texture the surface will be flat.
Strength	
UV space index	Selects a specific UV space
Flip V	Flip handedness of the tangent space

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Normalmap file	uniform texture_2d	texture	
Tiling	uniform float2	scaling	float2(1.0)
Offset	uniform float2	translation	float2(0.0)
Rotation	uniform float	rotation	0.0
Clip	uniform bool	clip	false
Strength	uniform float	factor	1
UV space index	uniform int	texture_space	0
Flip V	uniform bool	flip	false

### 3.13 Blend normals



MDL identifier: `core_definitions::blend_normals`

Blend two normals or apply a detail normal map

<i>Display name</i>	<i>Description</i>
Base normal	The base normal
Base normal weight	Specifies the strength of the base normal using a linear blend between <code>state::normal</code> and “base_normal”
Detail normal	The detail normal
Detail normal weight	Specifies the strength of the detail normal using a linear blend between <code>state::normal</code> and “detail_normal”

<i>Display name</i>	<i>Type</i>	<i>Parameter</i>	<i>Default</i>
Base normal	float3	base_normal	<code>state::normal()</code>
Base normal weight	float	base_normal_weight	1.0f
Detail normal	float3	detail_normal	<code>state::normal()</code>
Detail normal weight	float	detail_normal_weight	1.0f

---

## 4 Enumerations

### 4.1 User interface group

MDL identifier: `core_definitions::material_type`

User interface grouping hint for materials

<i>Field</i>	<i>Index</i>	<i>Description</i>
<code>simple_material</code>	<code>0</code>	Simple material
<code>complex_material</code>	<code>1</code>	Complex material
<code>combiner_material</code>	<code>2</code>	Combiner material
<code>modifier_material</code>	<code>3</code>	Material modifier

### 4.2 Emission mode

MDL identifier: `core_definitions::emission_type`

Emission mode definition for light sources

<i>Field</i>	<i>Index</i>	<i>Description</i>
<code>lumen_m2</code>	<code>0</code>	lumen/m2
<code>lumen</code>	<code>1</code>	lumen
<code>candela</code>	<code>2</code>	candela
<code>nit</code>	<code>3</code>	nit (candela/m2)

### 4.3 Worley noise cell type

MDL identifier: `core_definitions::cell_type`

Behavior of the Worley noise cell

<i>Field</i>	<i>Index</i>	<i>Description</i>
<code>simple_cells</code>	<code>0</code>	Simple cells
<code>crystal_cells</code>	<code>1</code>	Crystal cells
<code>bordered_cells</code>	<code>2</code>	Bordered cells

### 4.4 Worley noise cell shape

MDL identifier: `core_definitions::cell_base`

## Shape of the Worley noise cell

<i>Field</i>	<i>Index</i>	<i>Description</i>
<code>circular_cells</code>	<code>0</code>	Circle base
<code>diamond_cells</code>	<code>1</code>	Diamond base