

## 11.1.8 Editors - Shader Editor - Header - Add Menu - Shader

Detailed table of content.....	1
Add menu - Shader.....	7
Add Shader.....	7
Anisotropic BSDF.....	8
Background.....	9
Diffuse BSDF.....	9
Emission.....	10
Glass BSDF.....	10
Glossy BSDF.....	11
Hair BSDF.....	12
Holdout.....	13
Mix Shader.....	13
Inputs.....	13
Outputs.....	14
Principled BSDF.....	14
Principled Hair BSDF.....	17
Principled Volume.....	20
Refraction BSDF.....	21
Specular.....	22
Subsurface Scattering.....	23
Toon BSDF.....	25
Translucent BSDF.....	26
Transparent BSDF.....	26
Velvet BSDF.....	27
Volume Absorption.....	27
Volume Scatter.....	28

### Detailed table of content

#### Detailed table of content

Detailed table of content.....	1
Add menu - Shader.....	7
Add Shader.....	7
Inputs.....	7
Shaders.....	7
Outputs.....	8
Shader.....	8
Anisotropic BSDF.....	8
Inputs.....	8
Color.....	8
Roughness.....	8
Anisotropy.....	8
Rotation.....	8
Normal.....	8
Tangent.....	8
Properties.....	8

Distribution.....	8
Outputs.....	9
BSDF.....	9
Background.....	9
Inputs.....	9
Color.....	9
Strength.....	9
Outputs.....	9
Background.....	9
Diffuse BSDF.....	9
Inputs.....	9
Color.....	9
Roughness Cycles Only.....	9
Normal.....	9
Outputs.....	9
BSDF.....	9
Emission.....	10
Inputs.....	10
Color.....	10
Strength.....	10
Outputs.....	10
Emission.....	10
Glass BSDF.....	10
Inputs.....	10
Color.....	10
Roughness.....	10
IOR.....	10
Normal.....	10
Properties.....	10
Distribution.....	10
Sharp.....	10
GGX.....	10
Multiple-scattering GGX.....	11
Beckmann.....	11
Ashikhmin-Shirley.....	11
Outputs.....	11
BSDF.....	11
Glossy BSDF.....	11
Inputs.....	11
Color.....	11
Roughness.....	11
Normal.....	11
Properties.....	11
Distribution.....	11
Sharp.....	11
Beckmann.....	11
GGX.....	12
Ashikhmin-Shirley.....	12
Multiple-scattering GGX.....	12
Outputs.....	12
BSDF.....	12
Hair BSDF.....	12
Inputs.....	12

Color.....	12
Offset.....	12
Roughness U/V.....	12
Tangent.....	12
Properties.....	12
Component.....	12
Reflection.....	13
Transmission.....	13
Outputs.....	13
BSDF.....	13
Holdout.....	13
Outputs.....	13
Holdout.....	13
Mix Shader.....	13
Inputs.....	13
Shader.....	13
Factor.....	13
Outputs.....	14
Shader.....	14
Principled BSDF.....	14
Inputs.....	14
Base Color.....	14
Subsurface.....	14
Subsurface Radius.....	14
Subsurface Color.....	14
Metallic.....	14
Specular.....	15
Specular Tint.....	15
Roughness.....	15
Anisotropic.....	15
Anisotropic Rotation.....	15
Sheen.....	15
Sheen Tint.....	15
Clear coat.....	16
Clear coat Roughness.....	16
IOR.....	16
Transmission.....	16
Transmission Roughness.....	16
Emission.....	16
Emission Strength.....	16
Alpha.....	16
Normal.....	16
Clear coat Normal.....	16
Tangent.....	16
Properties.....	16
GGX.....	16
Multiple-scattering GGX.....	16
Subsurface Method.....	17
Christensen-Burley.....	17
Random Walk.....	17
Outputs.....	17
BSDF.....	17
Principled Hair BSDF.....	17

Coloring Hair.....	17
Direct coloring.....	17
Melanin concentration.....	18
Absorption coefficient.....	18
Randomizing Properties.....	18
Inputs.....	18
Color.....	18
Melanin.....	18
Melanin Redness.....	18
Tint.....	19
Absorption coefficient.....	19
Roughness.....	19
Radial Roughness.....	19
Coat.....	19
IOR.....	19
Offset.....	19
Random Color.....	19
Random Roughness.....	19
Random.....	20
Properties.....	20
Parametrization.....	20
Outputs.....	20
BSDF.....	20
Principled Volume.....	20
Inputs.....	20
Color.....	20
Color Attribute.....	20
Density.....	20
Density Attribute.....	20
Anisotropy.....	20
Absorption Color.....	20
Emission Strength.....	21
Emission Color.....	21
Blackbody Intensity.....	21
Blackbody Tint.....	21
Temperature.....	21
Temperature Attribute.....	21
Outputs.....	21
Volume.....	21
Refraction BSDF.....	21
Inputs.....	21
Color.....	21
Roughness.....	21
Normal.....	21
Properties.....	22
Distribution.....	22
Outputs.....	22
BSDF.....	22
Specular.....	22
Inputs.....	22
Base Color.....	22
Specular Color.....	22
Roughness.....	22

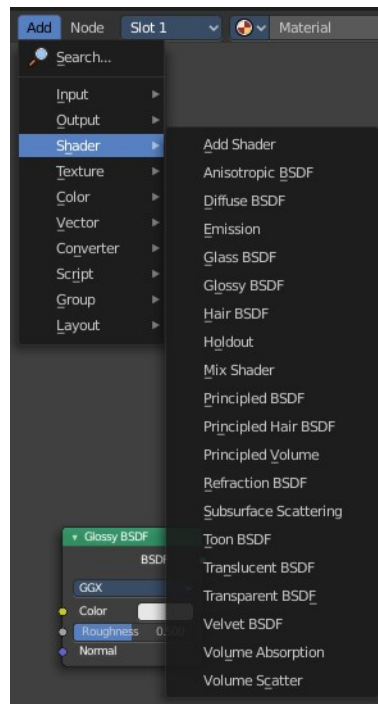
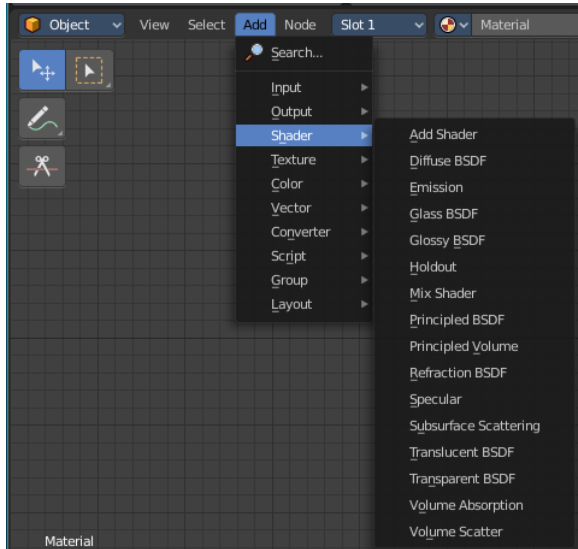
Emissive Color.....	23
Transparency.....	23
Normal.....	23
Clear coat.....	23
Clear coat Roughness.....	23
Clear coat Normal.....	23
Ambient Occlusion.....	23
Outputs.....	23
BSDF.....	23
Subsurface Scattering.....	23
Inputs.....	24
Color.....	24
Scale.....	24
Radius.....	24
Sharpness.....	24
Normal.....	24
Texture Blur.....	24
Properties.....	24
Falloff Method.....	24
Christensen-Burley.....	24
Random Walk.....	24
Cubic.....	25
Gaussian.....	25
Outputs.....	25
BSSRDF.....	25
Toon BSDF.....	25
Inputs.....	25
Color.....	25
Size.....	25
Smooth.....	25
Normal.....	25
Properties.....	25
Component.....	25
Diffuse.....	25
Glossy.....	26
Outputs.....	26
BSDF.....	26
Translucent BSDF.....	26
Inputs.....	26
Color.....	26
Normal.....	26
Outputs.....	26
BSDF output.....	26
Transparent BSDF.....	26
Inputs.....	26
Color.....	26
Outputs.....	26
BSDF.....	26
Velvet BSDF.....	27
Inputs.....	27
Color.....	27
Sigma.....	27
Normal.....	27

Outputs.....	27
BSDF.....	27
Volume Absorption.....	27
Inputs.....	27
Color.....	27
Density.....	27
Outputs.....	27
Volume.....	27
Volume Scatter.....	28
Inputs.....	28
Color.....	28
Density.....	28
Anisotropy.....	28
Output.....	28
Volume.....	28

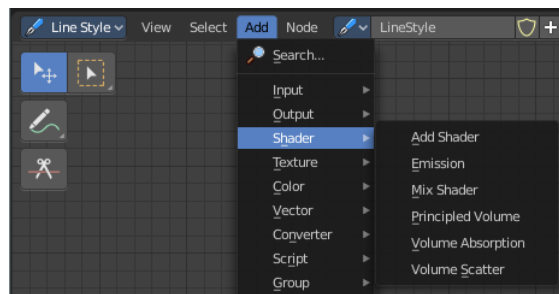
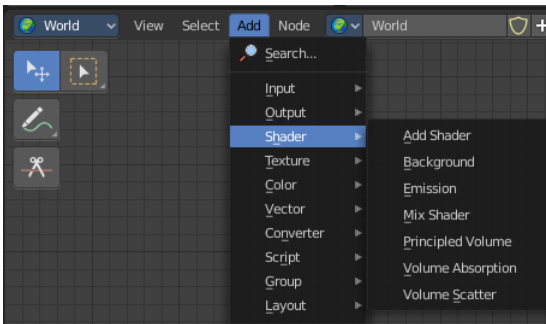
## Add menu - Shader

The shader section contains the different shader nodes. The content is different for the sub modes Object, World and Line Style. And it is dependant of the chosen render engine.

Left Eevee, right Cycles:



The modes ...



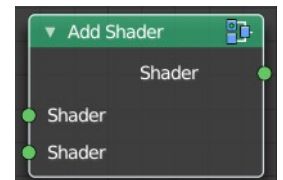
## Add Shader

The Add node is used to add two Shaders together.

### Inputs

### Shaders

Standard shader inputs.



## Outputs

### *Shader*

Standard shader output.

## Anisotropic BSDF

### Cycles Only

Adds a glossy reflection. The U and V direction roughness can be controlled separately. The tangents used for shading are derived from the active UV map. If no UV map is available, they are automatically generated using a sphere mapping based on the mesh bounding box.

### Inputs

#### *Color*

Color of the surface, or physically speaking, the probability that light is reflected for each wavelength.

#### *Roughness*

Sharpness of the reflection; perfectly sharp at 0.0 and smoother with higher values.

#### *Anisotropy*

Amount of anisotropy in the reflection; 0.0 gives a round highlight. Higher values give elongated highlights orthogonal to the tangent direction; negative values give highlights shaped along the tangent direction.

#### *Rotation*

Rotation of the anisotropic tangent direction. Value 0.0 equals 0° rotation, 0.25 equals 90° and 1.0 equals 360° = 0°. This can be used to texture the tangent direction.

#### *Normal*

Normal used for shading; if nothing is connected the default shading normal is used.

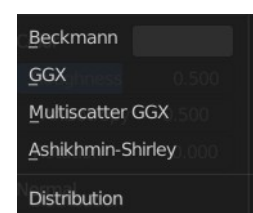
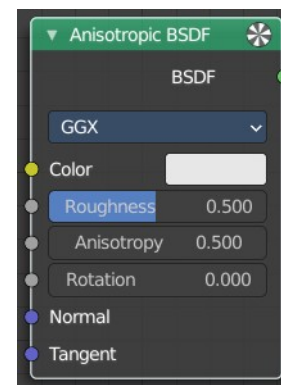
#### *Tangent*

Tangent used for shading; if nothing is connected the default shading tangent is used.

### Properties

#### *Distribution*

The available Microfacet distribution methods. Beckmann, GGX and Ashikhmin-Shirley can use the Roughness input for blurry reflections.





## Outputs

### ***BSDF***

Standard shader output.

---

## Background

### Shader Type World

The Background shader node is used to add background light emission.

## Inputs

### ***Color***

Color of the emitted light.

### ***Strength***

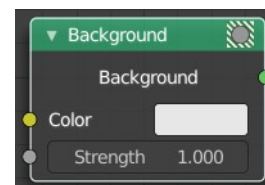
Strength of the emitted light.

## Outputs

### ***Background***

Standard shader output.

---



## Diffuse BSDF

### Shader Type Object

The Diffuse BSDF node is used to add Lambertian and Oren-Nayar diffuse reflection.

## Inputs

### ***Color***

Color of the surface, or physically speaking, the probability that light is reflected or transmitted for each wavelength.

### ***Roughness Cycles Only***

Surface roughness; 0.0 gives standard Lambertian reflection, higher values activate the Oren-Nayar BSDF.

### ***Normal***

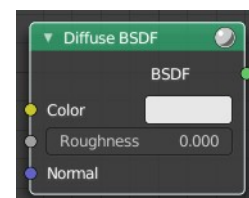
Normal used for shading; if nothing is connected the default shading normal is used.

## Outputs

### ***BSDF***

Standard shader output.

---



## Emission

The Emission node emits light.

### Inputs

#### *Color*

Color of the emitted light.

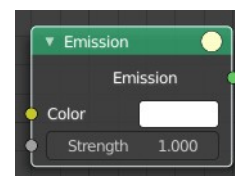
#### *Strength*

Strength of the emitted light. For point and area lights, the unit is Watts. For materials, a value of 1.0 will ensure that the object in the image has the exact same color as the Color input, i.e. make it 'shadeless'.

### Outputs

#### *Emission*

The Emission shader output can both be plugged into the Surface Input as well as the Volume Input of the Material Output node.



## Glass BSDF

A glass shader.

### Inputs

#### *Color*

Color of the surface, or physically speaking, the probability that light is transmitted for each wavelength.

#### *Roughness*

Influences sharpness of the refraction; perfectly sharp at 0.0 and smoother with higher values.

#### *IOR*

Index of refraction (IOR) defining how much the ray changes direction. At 1.0 rays pass straight through like transparent; higher values give more refraction.

#### *Normal*

Normal used for shading.

### Properties

#### *Distribution*

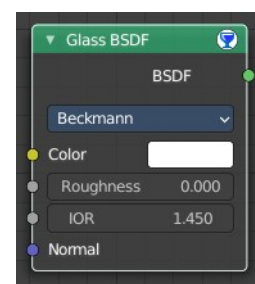
The micro facet distribution method to use.

#### **Sharp**

Results in perfectly sharp reflections like a mirror. The Roughness value is not used.

#### **GGX**

GGX micro facet distribution.



## Multiple-scattering GGX

### Cycles Only

Takes multiple bounce (scattering) events between micro facets into account. This gives a more energy conserving results, which would otherwise be visible as excessive darkening.

## Beckmann

### Cycles Only

Beckmann micro facet distribution.

## Ashikhmin-Shirley

### Cycles Only

Ashikhmin-Shirley micro facet distribution.

## Outputs

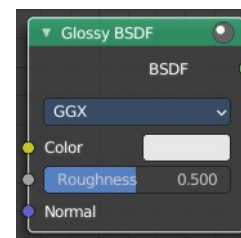
### ***BSDF***

Standard shader output.

## Glossy BSDF

### Object sub mode only

The Glossy BSDF node is used to add reflection with micro facet distribution, used for materials such as metal or mirrors.



## Inputs

### ***Color***

Color of the surface, or physically speaking, the probability that light is reflected for each wavelength.

### ***Roughness***

Input for the surface roughness resulting in sharp to blurry reflections.

### ***Normal***

Normal used for shading.

## Properties

### ***Distribution***

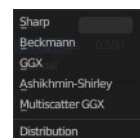
Micro facet distribution to use.

### **Sharp**

Results in perfectly sharp reflections like a mirror. The Roughness value is not used.

### **Beckmann**

### **Cycles Only**



Beckmann micro facet distribution.

## **GGX**

GGX micro facet distribution.

## **Ashikhmin-Shirley**

Cycles Only

Ashikhmin-Shirley micro facet distribution.

## **Multiple-scattering GGX**

Cycles Only

Takes multiple bounce (scattering) events between micro facets into account. This gives a more energy conserving results, which would otherwise be visible as excessive darkening.

## **Outputs**

### ***BSDF***

Standard shader output.

---

## **Hair BSDF**

Cycles Only

The Hair BSDF node is used to add shading for Hair.

## **Inputs**

### ***Color***

Color of the hair.

### ***Offset***

Controls the way the light is rotated (angular shift) for the reflection/transmission.

### ***Roughness U/V***

Controls the roughness in the direction light is skewed, and perpendicular to it.

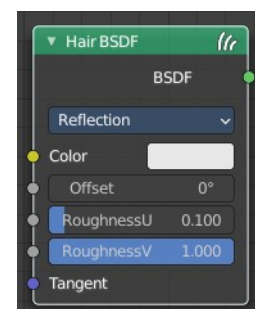
### ***Tangent***

Input tangent.

## **Properties**

### ***Component***

There are two components that can be used to control the look of the hair. Usually you are going to want each of these and use a Mix Node.



## Reflection

The light that bounces off the surface of the hair.

## Transmission

The light that passes through the hair and comes out the other side.

## Outputs

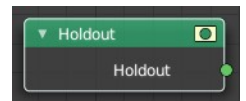
### BSDF

Standard shader output.

## Holdout

The Holdout shader node is used to create a “hole” in the image with zero alpha transparency, which is useful for compositing (see alpha channel).

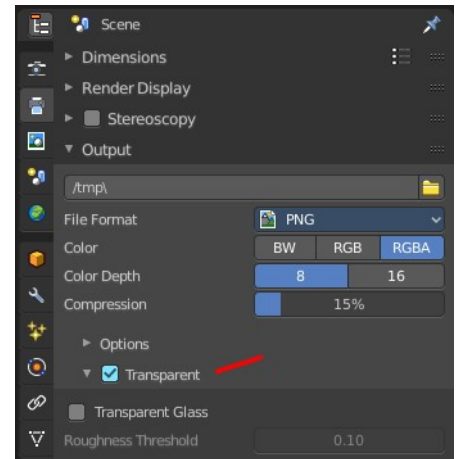
Note that the holdout shader can only create alpha when Transparent is enabled. This can be found in the render settings. Properties > Render > Film > Transparent is enabled. If it is disabled, the holdout shader will be black.



## Outputs

### Holdout

Standard shader output.



## Mix Shader

The Mix node is used to mix two shaders together. Mixing can be used for material layering, where the Factor input may, for example, be connected to a Blend Weight node.



## Inputs

### Shader

Shaders to mix, such that incoming rays hit either with the specified probability in the Factor socket.

### Factor

Blend weight to use for mixing two shaders; at zero it uses the first shader entirely and at one the second shader.

## Outputs

### Shader

Standard shader output.

## Principled BSDF

The Principled BSDF that combines multiple layers into a single easy to use node. It is based on the Disney principled model also known as the “PBR” shader, making it compatible with other software such as Pixar’s Renderman® and Unreal Engine®. Image textures painted or baked from software like Substance Painter® may be directly linked to the corresponding parameters in this shader.

This “Uber” shader includes multiple layers to create a wide variety of materials. The base layer is a user controlled mix between diffuse, metal, subsurface scattering and transmission. On top of that there is a specular layer, sheen layer and clear coat layer.

Note! The emphasis on compatibility with other software means that it interprets certain input parameters differently from older Blender nodes.

### Inputs

#### **Base Color**

Diffuse or metal surface color.

#### **Subsurface**

Mix between diffuse and subsurface scattering. Rather than being a simple mix between Diffuse and Subsurface Scattering, it acts as a multiplier for the Subsurface Radius.

#### **Subsurface Radius**

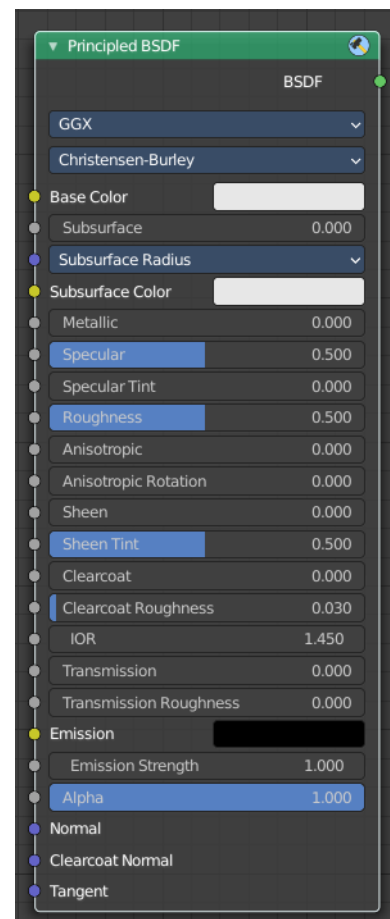
Average distance that light scatters below the surface. Higher radius gives a softer appearance, as light bleeds into shadows and through the object. The scattering distance is specified separately for the RGB channels, to render materials such as skin where red light scatters deeper. The X, Y and Z values are mapped to the R, G and B values, respectively.

#### **Subsurface Color**

Subsurface scattering base color.

#### **Metallic**

Blends between a non-metallic and metallic material model. A value of 1.0 gives a fully specular reflection



tinted with the base color, without diffuse reflection or transmission. At 0.0 the material consists of a diffuse or transmissive base layer, with a specular reflection layer on top.

### ***Specular***

Amount of dielectric specular reflection. Specifies facing (along normal) reflectivity in the most common 0 - 8% range.

Hint. To compute this value for a realistic material with a known index of refraction, you may use this special case of the Fresnel formula:  $\text{specular} = ((\text{ior} - 1) / (\text{ior} + 1))^2 / 0.08$

For example:

water: ior = 1.33, specular = 0.25

glass: ior = 1.5, specular = 0.5

diamond: ior = 2.417, specular = 2.15

Since materials with reflectivity above 8% do exist, the field allows values above 1.

### ***Specular Tint***

Tints the facing specular reflection using the base color, while glancing reflection remains white.

Normal dielectrics have colorless reflection, so this parameter is not technically physically correct and is provided for faking the appearance of materials with complex surface structure.

### ***Roughness***

Specifies micro facet roughness of the surface for diffuse and specular reflection.

Hint. When converting from the older Glossy BSDF node, use the square root of the original value.

### ***Anisotropic***

#### **Cycles Only**

Amount of anisotropy for specular reflection. Higher values give elongated highlights along the tangent direction; negative values give highlights shaped perpendicular to the tangent direction.

### ***Anisotropic Rotation***

#### **Cycles Only**

Rotates the direction of anisotropy, with 1.0 going full circle.

Hint. Compared to the Anisotropic BSDF node, the direction of highlight elongation is rotated by 90°. Add 0.25 to the value to correct.

### ***Sheen***

Amount of soft velvet like reflection near edges, for simulating materials such as cloth.

### ***Sheen Tint***

Mix between white and using base color for sheen reflection.

## ***Clear coat***

Extra white specular layer on top of others. This is useful for materials like car paint and the like.

## ***Clear coat Roughness***

Roughness of clear coat specular.

## ***IOR***

Index of refraction for transmission.

## ***Transmission***

Mix between fully opaque surface at zero and fully glass like transmission at one.

## ***Transmission Roughness***

### ***Cycles Only***

With GGX distribution controls roughness used for transmitted light.

## ***Emission***

Light emission from the surface, like the Emission shader.

## ***Emission Strength***

The strength of the light emission.

## ***Alpha***

Controls the transparency of the surface, with 1.0 fully opaque. Usually linked to the Alpha output of an Image Texture node.

## ***Normal***

Controls the normals of the base layers.

## ***Clear coat Normal***

Controls the normals of the Clear coat layer.

## ***Tangent***

Controls the tangent for the Anisotropic layer.

## **Properties**

Distribution

Micro facet distribution to use.

## **GGX**

A method that is faster than Multiple-scattering GGX but is less physically accurate. Selecting it enables the Transmission Roughness input.

## **Multiple-scattering GGX**

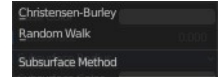
Takes multiple bounce (scattering) events between micro facets into account. This gives a more energy conserving results, which would otherwise be visible as excessive darkening.





## Subsurface Method

Rendering method to simulate subsurface scattering.



### Christensen-Burley

Is an approximation to physically-based volume scattering. Gives less blurry results than Cubic and Gaussian functions.

### Random Walk

Provides the most accurate results for thin and curved objects. This comes at the cost of increased render time or noise for more dense media like skin, but also better geometry detail preservation. Random Walk uses true volumetric scattering inside the mesh, which means that it works best for closed meshes. Overlapping faces and holes in the mesh can cause problems.

## Outputs

### BSDF

Standard shader output.

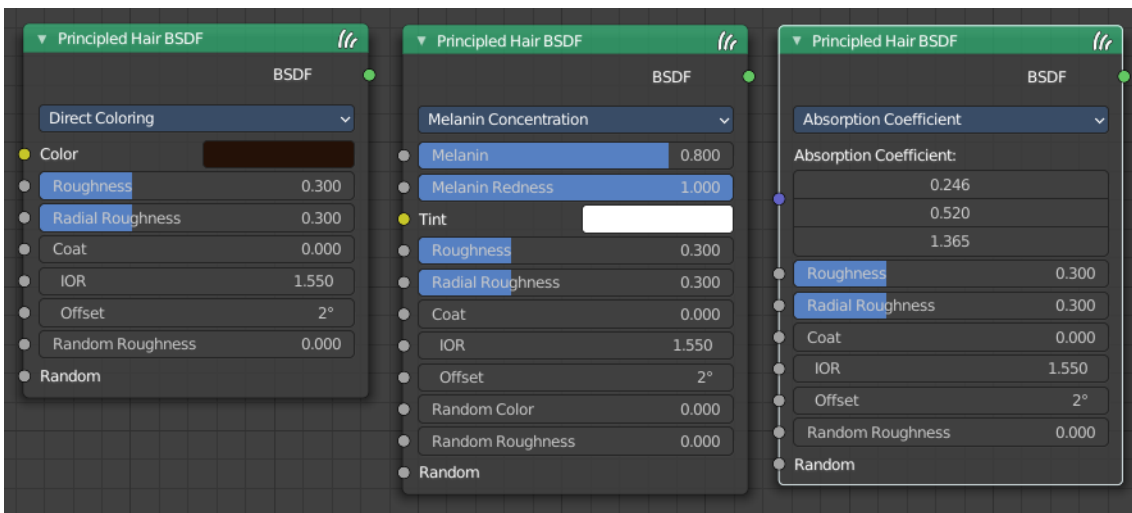
## Principled Hair BSDF

### Cycles Only

The Principled Hair BSDF is a physically-based, easy-to-use shader for rendering hair and fur.

### Coloring Hair

The shader provides three different ways, or parametrizations, to color the hair strands. Direct coloring, Melanin concentration and Absorption coefficient. They have some different settings and inputs.



### Direct coloring

Choose the desired RGB color and the shader will approximate the necessary absorption coefficient (below).

## Melanin concentration

This mode defines the color as the quantity and ratio of the pigments which are commonly found in hair and fur, eumelanin (prevalent in brown-black hair) and pheomelanin (red hair). The quantity is specified in the Melanin input, and the ratio between them in Melanin Redness. Increasing concentrations darken the hair (the following are with Melanin Redness 1):

White (Melanin 0), Blonde (Melanin 0.25), Reddish (Melanin 0.5), Brown (Melanin 0.75), Black (Melanin 1)

Additionally, the Tint inputs allows to dye the hair with the desired color.

## Absorption coefficient

Specifies the attenuation coefficient  $s_a$ , as applied by the Beer-Lambert law. This mode is intended mainly for technical users who want to use coefficients from the literature without any sort of conversion.

## Randomizing Properties

Realistic hair should have a minimum of variance between each strand. The shader allows for this by specifying two values, Random Color and Random Roughness, which remap the specified Melanin/Roughness values to the range  $\text{Color/Roughness} \pm \text{Randomization}\%$ .

## Inputs

### Color

The RGB color of the strand. Only used in Direct coloring.

Hint. The chosen color is converted to an absorption coefficient with the following formula (section 4.2 of [CBTB16]):

$$s_a = \ln(\text{Color}) (5.969 - 0.215\beta_N + 2.532\beta_N^2 - 10.73\beta_N^3 + 5.574\beta_N^4 + 0.245\beta_N^5)^2$$

where  $\beta_N$  is the radial roughness of the hair after applying randomization (if specified).

### Melanin

Absolute quantity of pigment. Range [0,1] equivalent to [0%,100%].

Hint. This is a linear mapping to the underlying exponential function:  $\text{melanin\_qty} = -\ln(\max(1.0 - \text{Melanin}, 0.0001))$

### Melanin Redness

Ratio of pheomelanin to eumelanin. Range [0,1] equivalent to [0%,100%].

Hint. The ratio formula is:  $\text{eumelanin} = \text{Melanin} * (1.0 - \text{MelaninRedness})$ ,  
 $\text{pheomelanin} = \text{Melanin} * \text{MelaninRedness}$ .

The resulting quantities are converted (after randomization, if specified) to absorption concentration via the following formula (section 6.1 of [EFHLA11], adjusted for the range [0,1]):  $s_a = \text{eumelanin} * 0.5060.8411.653 + \text{pheomelanin} * 0.3430.7331.924$

## ***Tint***

Color used for dyeing the hair after applying the melanin pigment. It is not subject to randomization. It can be disabled by setting the color to white.

Hint. This is converted via the Color mapping above and added to the absorption coefficient of the melanin concentration.

## ***Absorption coefficient***

Attenuation coefficients.

## ***Roughness***

Specify how much the glints are smoothed in the direction of the hair shaft. Too low values will smoothen the hair to the point of looking almost metallic, making glints look like fireflies; while setting it too high will result in a Lambertian look.

## ***Radial Roughness***

Specify how much the glints are smoothed in the direction of the hair tangent. Too low values will concentrate the glint; while setting it too high will spread the light across the width of the strand.

Hint. Mathematically, this parameter is mapped to the logistic distribution's scale factor  $s$  (section 4.1 of [CBTB16]).

## ***Coat***

Simulate a shiny coat of fur, by reducing the Roughness to the given factor only for the first light bounce (diffuse). Range [0,1] equivalent to a reduction of [0%,100%] of the original Roughness.

## ***IOR***

Index of refraction (IOR) defining how much the ray changes direction. At 1.0 rays pass straight through like in a transparent material; higher values give more refraction. Default value is 1.55.

## ***Offset***

Tilts the glint of the hair by increasing the angle of the scales of the hair's cuticle with respect to the hair shaft. Human hair usually has low values.

## ***Random Color***

For each strand, vary the melanin concentration by RandomFactor. Range [0,1] equivalent to [0%,100%] of the initial melanin concentration.

Hint. The melanin concentration is multiplied by randomFactor, where  $\text{randomFactor} = 1.0 + 2.0 * (\text{Random} - 0.5) * \text{RandomColor}$ .

## ***Random Roughness***

For each strand, vary both Roughness values by RandomFactor. Range [0,1] equivalent to [0%,100%] of the initial roughness values.

Hint. The applied formula is the same one as for Random Color.

## **Random**

Random number source. If no node is connected here, it is automatically instanced with the value obtained from Hair Info > Random.

## **Properties**

### **Parametrization**

Choose one of the three coloring modes described above.

## **Outputs**

### **BSDF**

Standard shader output.

## **Principled Volume**

The Principled Volume shader combines all volume shading components into a single easy to use node. Volumes like smoke and fire can be rendered with a single shader node, which includes scattering, absorption and blackbody emission.

## **Inputs**

### **Color**

Volume scattering color.

### **Color Attribute**

Volume grid for coloring the volume. Use “color” for smoke simulations.

### **Density**

Density of the volume.

### **Density Attribute**

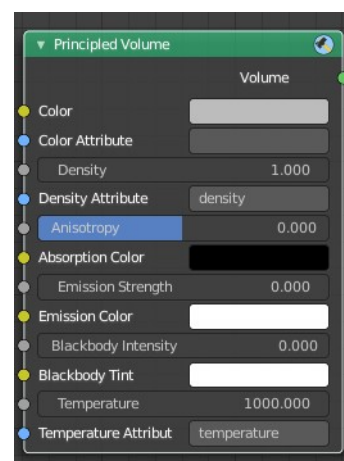
Volume grid to define the density, typically “density”.

### **Anisotropy**

Backward or forward scattering direction.

### **Absorption Color**

Volume shadow color tint.



## ***Emission Strength***

Amount of light to emit.

## ***Emission Color***

Emission color tint.

## ***Blackbody Intensity***

Blackbody emission for fire. Set to 1 for physically accurate intensity.

## ***Blackbody Tint***

Color tint for blackbody emission.

## ***Temperature***

Temperature in kelvin for blackbody emission, higher values emit more.

## ***Temperature Attribute***

Volume grid to define the temperature, typically “temperature”.

## **Outputs**

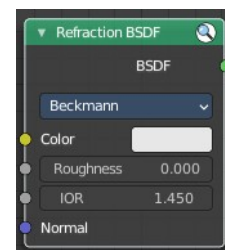
### ***Volume***

Standard shader output.

---

## **Refraction BSDF**

The Refraction BSDF is used to add glossy refraction with sharp or micro facet distribution, used for materials that transmit light. For best results this node should be considered as a building block and not be used on its own, but rather mixed with a glossy node using a Fresnel factor. Otherwise it will give quite dark results at the edges for glossy refraction.



## **Inputs**

### ***Color***

Color of the surface, or physically speaking, the probability that light is refracted for each wavelength.

### ***Roughness***

Influences sharpness of the refraction; perfectly sharp at 0.0 and smoother with higher values.

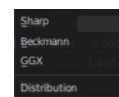
### ***Normal***

Normal used for shading; if nothing is connected the default shading normal is used.

## Properties

### **Distribution**

Micro facet distribution to use. Sharp results in perfectly sharp refraction, while Beckmann and GGX can use the Roughness input for blurry refraction.



## Outputs

### **BSDF**

Standard shader output.

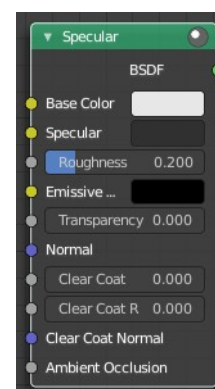
## Specular

### **Eevee Only**

The Specular BSDF combines multiple layers into a single easy to use node.

It is similar to the Principled BSDF node but uses the specular workflow instead of the metallic. It has far fewer parameters and supports less features. Both might be merged into one node in the future.

The specular workflow functions by specifying the facing (along normal) reflection color. The result may not be physically plausible because there is no energy conservation.



## Inputs

### **Base Color**

Diffuse surface color. For conductor materials (metals) it should be black.

### **Specular Color**

Amount of specular reflection. Specifies facing (along normal) reflectivity. Conductor materials (metals) can have colored specular reflection.

Hint. To compute this value for a realistic material with a known index of refraction, you may use this special case of the Fresnel formula:  $\text{specular} = ((\text{ior} - 1) / (\text{ior} + 1))^2$

For example:

water:  $\text{ior} = 1.33$ ,  $\text{specular} = 0.25$

glass:  $\text{ior} = 1.5$ ,  $\text{specular} = 0.5$

diamond:  $\text{ior} = 2.417$ ,  $\text{specular} = 2.15$

### **Roughness**

Specifies micro facet roughness of the surface for diffuse and specular reflection.

Hint. When converting from the older Glossy BSDF node, use the square root of the original value.

### ***Emissive Color***

Color of the emitted light. This light is added to the BSDF result.

### ***Transparency***

Transparency factor. This is the inverse of the alpha channel (1 - alpha) you find in an image. Use an Invert node to convert alpha to transparency. This will only have an effect if the material uses a blend mode other than opaque.

### ***Normal***

Controls the normals of the base layers.

### ***Clear coat***

Extra white specular layer on top of others. This is useful for materials like car paint and the like.

### ***Clear coat Roughness***

Roughness of clear coat specular.

### ***Clear coat Normal***

Controls the normals of the Clear coat layer.

### ***Ambient Occlusion***

Amount of occlusion to apply to indirect lighting. Usually a bake ambient occlusion map. The final occlusion factor is the minimum of this input and the runtime ambient occlusion effect.

## **Outputs**

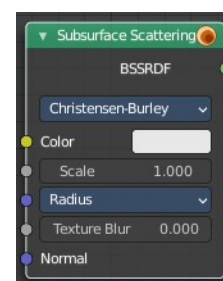
### ***BSDF***

Standard shader output.

## **Subsurface Scattering**

The Subsurface Scattering node is used to add simple subsurface multiple scattering, for materials such as skin, wax, marble, milk and others. For these materials, rather than light being reflect directly off the surface, it will penetrate the surface and bounce around internally before getting absorbed or leaving the surface at a nearby point.

How far the color scatters on average can be configured per RGB color channel. For example, for skin, red colors scatter further, which gives distinctive red-colored shadows, and a soft appearance.



## Inputs

### **Color**

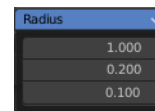
Color of the surface, or physically speaking, the probability that light is reflected for each wavelength.

### **Scale**

Global scale factor for the scattering radius.

### **Radius**

Average distance that light scatters below the surface. Higher radius gives a softer appearance, as light bleeds into shadows and through the object. The scattering distance is specified separately for the RGB channels, to render materials such as skin where red light scatters deeper. The X, Y and Z values are mapped to the R, G and B values, respectively.



### **Sharpness**

Used only with Cubic falloff. Values increasing from 0 to 1 prevents softening of sharp edges and reduces unwanted darkening.

### **Normal**

Normal used for shading; if nothing is connected the default shading normal is used.

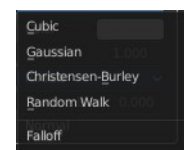
### **Texture Blur**

How much of the texture will be blurred along with the lighting, mixing the texture at the incoming and outgoing points on the surface. Note that the right choice depends on the texture. Consider for example a texture created from a photograph of skin, in this case the colors will already be pre-blurred and texture blur could be set to 0. Even for hand-painted textures, no blurring or minimal blurring might be appropriate, as a texture artist would likely paint in softening already. One would usually not even know what an unblurred skin texture looks like; we always see it blurred. For a procedural texture on the other hand this option would likely have a higher value.

## Properties

### **Falloff Method**

Rendering method to simulate subsurface scattering.



### **Christensen-Burley**

Is an approximation to physically-based volume scattering. Gives less blurry results than Cubic and Gaussian functions.

### **Random Walk**

#### **Cycles Only**

Provides the most accurate results for thin and curved objects. This comes at the cost of increased render time or noise for more dense media like skin, but also better geometry detail preservation. Random Walk uses true volumetric scattering inside the mesh, which means that it works best for closed meshes. Overlapping faces and



holes in the mesh can cause problems.

## **Cubic**

Is a sharp falloff useful for many simple materials. The function is  $(\text{radius}-x)^3$ .

## **Gaussian**

Gives a smoother falloff following a normal distribution, which is particularly useful for more advanced materials that use measured data that was fitted to one or more such Gaussian functions. The function is  $e^{-8x^2/\text{radius}^2}$ , such that the radius roughly matches the maximum falloff distance. To match a given measured variance  $v$ , set  $\text{radius}=\sqrt{16\times v}$ .

## **Outputs**

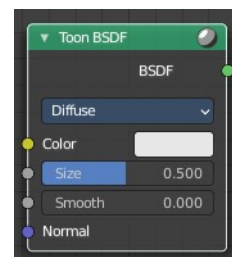
### **BSSRDF**

BSSRDF shader output.

## **Toon BSDF**

### **Cycles Only**

The Toon BSDF is used to create Diffuse and Glossy materials with cartoon light effects.



## **Inputs**

### **Color**

Color of the surface, or physically speaking, the probability that light is reflected for each wavelength.

### **Size**

Parameter between 0.0 and 1.0 that gives an angle of reflection between  $0^\circ$  and  $90^\circ$ .

### **Smooth**

This value specifies an angle over which a smooth transition from full to no reflection happens.

### **Normal**

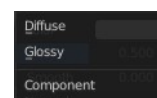
Normal used for shading; if nothing is connected the default shading normal is used.

## **Properties**

### **Component**

#### **Diffuse**

Use shading based on the Diffuse BSDF.



## Glossy

Use shading based on the Glossy BSDF for specular reflection.

## Outputs

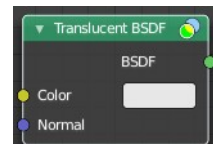
### **BSDF**

Standard shader output.

---

## Translucent BSDF

The Translucent BSDF is used to add Lambertian diffuse transmission.



## Inputs

### **Color**

Color of the surface, or physically speaking, the probability that light is transmitted for each wavelength.

### **Normal**

Normal used for shading; if nothing is connected the default shading normal is used.

## Outputs

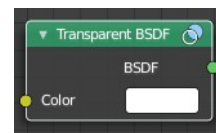
### **BSDF output**

Standard shader output.

---

## Transparent BSDF

The Transparent BSDF is used to add transparency without refraction, passing straight through the surface, as if there were no geometry there. Useful with alpha maps, for example. This shader affects light paths somewhat differently than other BSDFs. Note that only pure white transparent shaders are completely transparent.



## Inputs

### **Color**

Color of the surface, or physically speaking, the probability for each wavelength that light is blocked or passes straight through the surface.

## Outputs

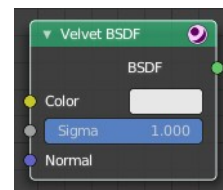
### **BSDF**

Standard shader output.

## Velvet BSDF

### Cycles Only

The Velvet BSDF is used to add reflection to materials such as cloth. It is meant to be used together with other shaders (such as a Diffuse Shader) and is not particularly useful on its own.



### Inputs

#### *Color*

Color of the surface, or physically speaking, the probability that light is reflected for each wavelength.

#### *Sigma*

Variance of the normal distribution, controlling the sharpness of the peak. It can be thought of as a kind of roughness.

#### *Normal*

Normal used for shading; if nothing is connected the default shading normal is used.

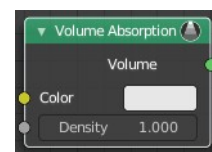
### Outputs

#### *BSDF*

Standard shader output.

## Volume Absorption

The Volume Absorption node allows light to be absorbed as it passes through the volume. Typical usage for this node would be water and colored glass.



### Inputs

#### *Color*

Color of the volume.

#### *Density*

The density of the absorption effect.

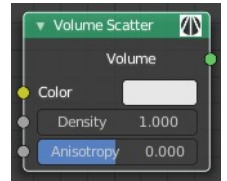
### Outputs

#### *Volume*

The Volume Shader.

## Volume Scatter

The Volume Scatter node allows light to be scattered as it passes through the volume. Typical usage would be to add fog to a scene. It can also be used with the Volume Absorption node to create smoke.



### Inputs

#### ***Color***

Color of the volume.

#### ***Density***

The density of the scatter effect.

#### ***Anisotropy***

Controls the look of the scatter effect depending on the direction of the light passing through it.

### Output

#### ***Volume***

The Volume Shader output must be plugged into the Volume Input of the Material or World Output node.