# 10.2.1 Render - Blender Render Engine - Materials

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bforartists Render Engine</td>
<td>5</td>
</tr>
<tr>
<td>Introduction to Materials</td>
<td>5</td>
</tr>
<tr>
<td>How Materials Work</td>
<td>6</td>
</tr>
<tr>
<td>Using Materials</td>
<td>7</td>
</tr>
<tr>
<td>Assigning a Material</td>
<td>7</td>
</tr>
<tr>
<td>Creating a new Material</td>
<td>8</td>
</tr>
<tr>
<td>New Material Panel Buttons</td>
<td>9</td>
</tr>
<tr>
<td>Reusing Existing Materials</td>
<td>11</td>
</tr>
<tr>
<td>Deleting a Material</td>
<td>12</td>
</tr>
<tr>
<td>Multiple Materials</td>
<td>13</td>
</tr>
<tr>
<td>Introduction to Properties</td>
<td>14</td>
</tr>
<tr>
<td>Material Properties</td>
<td>14</td>
</tr>
<tr>
<td>Material Preview</td>
<td>15</td>
</tr>
<tr>
<td>Options</td>
<td>15</td>
</tr>
<tr>
<td>Examples</td>
<td>16</td>
</tr>
<tr>
<td>Diffuse Shaders</td>
<td>16</td>
</tr>
<tr>
<td>Common Options</td>
<td>16</td>
</tr>
<tr>
<td>Technical Details</td>
<td>17</td>
</tr>
<tr>
<td>Lambert</td>
<td>17</td>
</tr>
<tr>
<td>Oren-Nayar</td>
<td>18</td>
</tr>
<tr>
<td>Options</td>
<td>18</td>
</tr>
<tr>
<td>Minnaert</td>
<td>20</td>
</tr>
<tr>
<td>Options</td>
<td>20</td>
</tr>
<tr>
<td>Fresnel</td>
<td>21</td>
</tr>
<tr>
<td>Options</td>
<td>22</td>
</tr>
<tr>
<td>Specular Shaders</td>
<td>22</td>
</tr>
<tr>
<td>Common Options</td>
<td>22</td>
</tr>
<tr>
<td>Technical Details</td>
<td>23</td>
</tr>
<tr>
<td>CookTorr</td>
<td>23</td>
</tr>
<tr>
<td>Options</td>
<td>24</td>
</tr>
<tr>
<td>Phong</td>
<td>24</td>
</tr>
<tr>
<td>Options</td>
<td>25</td>
</tr>
<tr>
<td>Blinn</td>
<td>25</td>
</tr>
<tr>
<td>Options</td>
<td>26</td>
</tr>
<tr>
<td>Toon</td>
<td>26</td>
</tr>
<tr>
<td>Options</td>
<td>26</td>
</tr>
<tr>
<td>WardIso</td>
<td>27</td>
</tr>
<tr>
<td>Options</td>
<td>27</td>
</tr>
<tr>
<td>Color Ramps</td>
<td>27</td>
</tr>
<tr>
<td>Options</td>
<td>28</td>
</tr>
<tr>
<td>Colorbands</td>
<td>29</td>
</tr>
<tr>
<td>Options</td>
<td>30</td>
</tr>
<tr>
<td>Color Ramps</td>
<td>31</td>
</tr>
<tr>
<td>Options</td>
<td>31</td>
</tr>
<tr>
<td>Colorbands</td>
<td>33</td>
</tr>
<tr>
<td>Options</td>
<td>33</td>
</tr>
<tr>
<td>Shading</td>
<td>35</td>
</tr>
<tr>
<td>Transparency</td>
<td>36</td>
</tr>
<tr>
<td>Common Options</td>
<td>36</td>
</tr>
</tbody>
</table>

Bforartists Reference Manual - © Copyright - This page is under OCL license
Output Node................................................................................................................................. .88
Vector Nodes....................................................................................................................................... .89
  Normal Node...................................................................................................................................... .89
  Inputs................................................................................................................................................ .90
  Outputs............................................................................................................................................. .90
  Controls............................................................................................................................................ .90
Mapping Node....................................................................................................................................... .91
  Inputs................................................................................................................................................ .91
  Outputs............................................................................................................................................. .91
  Controls............................................................................................................................................ .91
Vector Curves....................................................................................................................................... .93
  Inputs................................................................................................................................................ .93
  Outputs............................................................................................................................................. .93
Options.................................................................................................................................................. .93
Materials............................................................................................................................................... .94
  Context.............................................................................................................................................. .94
    Material slots................................................................................................................................ .94
    Multiple materials........................................................................................................................ .94
    Material naming and linking........................................................................................................ .95
    Material type................................................................................................................................ .95
Material Properties Overview.............................................................................................................. .96
  Surface and Wire materials............................................................................................................ .96
    Preview......................................................................................................................................... .96
    Diffuse......................................................................................................................................... .96
    Specular....................................................................................................................................... .96
    Shading........................................................................................................................................ .96
    Transparency................................................................................................................................. .97
    Mirror.......................................................................................................................................... .97
    Subsurface Scattering.................................................................................................................. .97
    Strand......................................................................................................................................... .97
    Options........................................................................................................................................ .97
    Shadow....................................................................................................................................... .98
Volume Material................................................................................................................................... .98
Halo Material....................................................................................................................................... .98
Special Material Effects....................................................................................................................... .98
Halo Rendering..................................................................................................................................... .99
  Options............................................................................................................................................ .99
    Halo Panel.................................................................................................................................... 100
    Flare Panel................................................................................................................................... 102
      Lens Flares................................................................................................................................. 102
Halo Texturing..................................................................................................................................... 103
Examples............................................................................................................................................. 103
  Dotmatrix display........................................................................................................................... 103
Volume Rendering.............................................................................................................................. 105
  Options............................................................................................................................................ 106
    Density.......................................................................................................................................... 106
    Shading......................................................................................................................................... 107
    Asymmetry.................................................................................................................................... 108
    Transmission................................................................................................................................. 108
    Emission....................................................................................................................................... 109
    Reflection..................................................................................................................................... 109
    Hints.............................................................................................................................................. 110
    Lighting...................................................................................................................................... 110
Bforartists Render Engine

The Bforartists render engine is the first render engine that existed in Bforartists. It's the old render engine, present since pre 2.5 days. It still delivers some good results. But it is aged. And not longer in active development. The render engine under active development is Cycles.

Introduction to Materials

A material defines the artistic qualities of the substance that an object is made of. In its simplest form, you can use materials to show the substance an object is made of, or to “paint” the object with different colors. Usually, the substance is represented by its surface qualities (color, shininess, reflectance, etc.) but it can also exhibit more complicated effects such as transparency, diffraction and sub-surface scattering. Typical materials might be brass, skin, glass, or linen.

Various basic materials (single, multiple material, transparency, vertex paint).

The basic (un-textured) Bforartists material is uniform across each face of an object (although the various pixels of each face of the object may appear differently because of lighting effects). However, different faces of the object may use different materials (see Multiple Materials).

In Bforartists, materials can (optionally) have associated textures. Textures describe the substance: e.g. polished brass, dirty glass or embroidered linen. The Textures chapter describes how to add textures to materials.
How Materials Work

Before you can understand how to design effectively with materials, you must understand how simulated light and surfaces interact in Bforartists’s rendering engine and how material settings control those interactions. A deep understanding of the engine will help you to get the most from it.

The rendered image you create with Bforartists is a projection of the scene onto an imaginary surface called the viewing plane. The viewing plane is analogous to the film in a traditional camera, or the rods and cones in the human eye, except that it receives simulated light, not real light.

To render an image of a scene we must first determine what light from the scene is arriving at each point on the viewing plane. The best way to answer this question is to follow a straight line (the simulated light ray) backwards through that point on the viewing plane and the focal point (the location of the camera) until it hits a renderable surface in the scene, at which point we can determine what light would strike that point.

The surface properties and incident light angle tell us how much of that light would be reflected back along the incident viewing angle (Rendering engine basic principle).

Rendering engine basic principle.

Two basic types of phenomena take place at any point on a surface when a light ray strikes it: diffusion and specular reflection. Diffusion and specular reflection are distinguished from each other mainly by the relationship between the incident light angle and the reflected light angle.

The shading (or coloring) of the object during render will then take into account the base color (as modified by the diffusion and specular reflection phenomenon) and the light intensity.

Using the internal ray tracer, other (more advanced) phenomena could occur. In ray-traced reflections, the point of a surface struck by a light ray will return the color of its surrounding environment, according to the rate of reflection of the material (mixing the base color and the surrounding environment’s) and the viewing angle.

On the other hand, in ray-traced refractions, the point of a surface struck by a light ray will return the color of its background environment, according to the rate of transparency (mixing the base color and the background environment’s along with its optional filtering value) of the material and the optional index of refraction of the material, which will distort the viewing angle.

Of course, shading of the object hit by a light ray will be about mixing all these phenomena at the same time during the rendering. The appearance of the object, when rendered, depends on many inter-related settings:

- World (Ambient color, Radiosity, Ambient Occlusion)
- Lights
- Material settings (including ambient, emission, and every other setting on every panel in that context)
• Texture(s) and how they are mixed
• Material Nodes
• Camera
• Viewing angle
• Obstructions and transparent occlusions
• Shadows from other opaque/transparent objects
• Render settings
• Object dimensions (SS settings are relevant to dimensions)
• Object shape (refractions, fresnel effects)

**Using Materials**

**Tip**

Check your Render

When designing materials (and textures and lighting), frequently check the rendered appearance of your scene, using your chosen render engine/shader settings. The appearance might be quite different from that shown in the texture display in the 3D panel.

As stated above, the material settings usually determine the surface properties of the object. There are several ways in which materials can be set up in Bforartists. Generally speaking, these are not compatible - you must choose which method you are going to use for each particular object in your scene.

First, you can set the **Properties** in the various Material panels.

Second, you can use **Nodes**; a graphical nodes editor is available.

Last, you can directly set the color of object surfaces using various special effects. Strictly speaking, these are not materials at all, but they are included here because they affect the appearance of your objects. These include Vertex Painting, **Wire Rendering**, **Volume Rendering**, and **Halo Rendering**.

The exact effect of Material settings can be affected by a number of system settings. First and foremost is the Render Engine used - Cycles and the Bforartists Render Engine (aka Bforartists Internal or BI) require quite different illumination levels to achieve similar results, and even then the appearance of objects can be quite different. Also, the material properties settings can be affected by the texture method used (Single Texture, Multitexture or GLSL). So it is recommended to always select the appropriate system settings before starting the design of materials.

**Assigning a Material**

Materials available in the currently-open Bforartists file can be investigated by clicking on the Materials button in the Properties Window Header. In this section we look at how to assign or remove a material to/from the Active Object in Bforartists, either by:
- creating a new material,
- re-using an existing material, or
- deleting a material.

We also give hints about practical material usage.

**Creating a new Material**

Every time a new Object is created it has no material linked to it. You can create a new material for the object by

- Selecting the object
- In the Properties window, click on the object button
- Click on the Materials button in the Properties Panel Header (1)

The Shading context window then appears. This contains the following elements:

**Add new material**

- Context - The currently-selected scene and object
- Object Material Slots (3) - this window shows the “slots” for the material (or materials) that this object data contains.
- Active Material (2). Initially empty, asking for “New”.

To add a new material, click “+” in the Active Material box. This action has a series of effects:
Materials Panel with New Entry

- opens the new material in the Active Material box,
- brings up additional buttons in the immediate panel,
- adds the new material to the Available Materials list,
- adds the new material to the Object Material Slots list for the active object (or its object data - see below)
- brings up a preview of the new material,
- provides you with a range of panels allowing you to select the properties of the new material.

**New Material Panel Buttons**

Details of the additional buttons which appear in the Material panel for a new Active Material are as follows:

**Active Material**

Available Materials See Reusing Existing Materials below.

**Name**

Like other data-blocks, Bforartists will automatically set the name of the new material to `Material, Material.001` and so on. You can change this by over-typing with your own choice of name.

**Number of Users**

Specifies the number of meshes which use this material.

**F - Fake User**

When enabled, this material will always be saved within the Bforartists file, even if it has no meshes which use it (see Deleting a Material).

**X**

Delete this material (see Deleting a Material).

**Tip**

**Naming materials**

It’s a very good idea to give your materials clear names so you can keep track of them, especially when they’re linked to multiple objects. Try to make your names descriptive of the material, not its function (e.g. “Yellow Painted” rather than “Kitchen Table Color”)

**Nodes**

If dark, use the Shader Nodes to generate the material.

**Data**

Specifies whether the material is to be linked to the Object or to the Object Data.
The Link pop-up menu has two choices, Data and Object. These two menu choices determine whether the material is linked to the object or to the data, (in this case) the mesh (or curve, nurbs, etc.). The Data menu item determines that this material will be linked to the mesh’s data-block which is linked to the object’s data-block. The Object menu item determines that the material will be linked to the object’s data block directly. This has consequences of course. For example, different objects may share the same mesh data-block. Since this data-block defines the shape of the object, any change in edit mode will be reflected on all of those objects. Moreover, anything linked to that mesh data-block will be shared by every object that shares that mesh. So, if the material is linked to the mesh, every object will share it. On the other hand, if the material is linked directly to the object data-block, the objects can have different materials and still share the same mesh. Short explanation: If connected to the object, you can have several instances of the same obData using different materials. If linked to mesh data, you can’t. See Data System for more information.

Object Render Format (menu)

This menu has four options which define how the object is to be rendered:

Surface
Material applied to object planes.

Wire
Material applied to wires following the object edges

Volume
Material applied to the object volume.

Halos
Material applied to halos around each object vertex.
Reusing Existing Materials

Bforartists is built to allow you to reuse *anything*, including material settings, between many objects. Instead of creating duplicate materials, you can simply re-use an existing material. There are several ways to do this using the Available Materials menu:

*Single Object* - With the object selected, click the sphere located to the left of the Material name. A drop-down list appears showing all the materials available in the current Bforartists file. To use one, just click on it.
Select an existing material.

List of available materials

**Tip**

**Searching for Materials**

The search field at the bottom of the material list allows you to search the names in the list. For example, by entering “wood” all existent materials are filtered so that only materials containing “wood” are displayed in the list.

**Multiple Objects** - In the 3D View, with Ctrl-L you can quickly link all selected objects to the material (and other aspects) of the active object. Very useful if you need to set a large number of objects to the same material; just select all of them, then the object that has the desired material, and Ctrl-L link them to that “parent”. (See Tip on Linking Data in Creating about data linking.)

---

**Deleting a Material**

To delete a material, select the material and click X in the Available Materials List entry.

Although the material will seem to disappear immediately, the Delete action can depend on how the material is used elsewhere.

If the material is linked to the Object and there are other objects which use this material, then the material will be removed from that object (but remain on all its other objects).
If the “Fake User” button (F) has been lit in the Available Materials list, then the material will be retained when the file is saved, even if it has no users.

Only if it has 0 “real” users, and no “Fake” user, will the material be permanently deleted. Note that it will still remain in the Materials list until the Bforartists file is saved, but will have disappeared when the file is reloaded.

**Multiple Materials**

Normally, different colors or patterns on an object are achieved by adding textures to your materials. However, in some applications you can obtain multiple colors on an object by assigning different materials to the individual faces of the object.

1. **Add new material**

To apply several materials to different faces of the same object, you use the Material Slots options (3) in the Materials header panel.

2. **Material menu in edit mode**

The workflow for applying a second material to some faces of an object covered by a base material is as follows:
• In Object mode, apply the base material to the whole object (as shown in Assigning a material).
• Create/select the second material (the whole object will change to this new material).
• In the Active Material box (2), re-select the base material.
• Go to Edit Mode - Face Select (a new box appears above the Active Material box with Assign/Select/Deselect).
• Select the face/faces to be colored with the second material.
• In the Object Material Slots box (3), click the Plus to create a new slot, and while this is still active, click on the second material in the Available Materials list.
• Click the Assign button, and the second material will appear on the selected object faces.
• You can also make this new material a copy of an existing material by adding the data block:

Select object, get the material, (R Click) - Copy data to clipboard. When you have renamed the material, click “Data - Data” to link to the existing material. Proceed to assign faces as required. NB: If you change the material on the original object, the new object color changes too.

Introduction to Properties

Material Properties

Materials can have a wide array of properties. It is the combination of all of these things that define the way a material looks, and how objects using that material will appear when rendered. These properties are set using the various Properties panels.

Remember that the appearance of your materials are affected by the way that they are rendered (surface, wire, volume or halo), and by the rendering engine (Bforartists, Cycles, or Game) used. Most properties for images rendered using Cycles can only be controlled using the Node system.

The list below sets out the various Properties panels available in Bforartists Render and Game Engine, and brief details of their scope. Details of their controls and settings are given on the relevant pages.

**Preview**
- A preview of the current material mapped on to one of several basic objects.

**Diffuse Shaders**
- The basic color of the material, together with different models for dispersion.

**Specular Shaders**
- The reflected highlights: color, strength and different models for dispersion.

**Color Ramps**
- How to vary the base color over a surface in both Diffuse ans Specular shaders.

**Shading**
- Properties of various characteristics of the shading model for the material.

**Transparency**
- Can other objects be seen through the object, and if so, how?

**Mirror**
- (Only Bforartists Render): Reflective properties of the material.

**SubSurface Scattering**
- (Only Bforartists Render): Simulates semi-translucent objects in which light enters, bounces around, then exits in a different place.

**Strand**
(Only Bforartists Render): For use when surfaces are covered with hair, fur, etc.

**Options**
Various options for shading and coloring the object.

**Shadow:**
Controls how objects using this material cast and receive shadows.

**Game Settings**
(Only Bforartists Render): Controls settings for real-time rendering of Game Engine objects.

---

## Material Preview

### Reference

<table>
<thead>
<tr>
<th>Mode</th>
<th>All Modes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Panel</td>
<td>Shading/Material Context → Preview</td>
</tr>
</tbody>
</table>

The Preview panel gives a quick visualization of the active material and its properties, including its Shaders, Ramps, Mirror Transp properties and Textures. It provides several shapes that are very useful for designing new shaders: for some shaders (like those based on Ramp colors, or a Diffuse shader like Minnaert), one needs fairly complex or specific previewing shapes to decide if the shader being designed achieves its goal.

### Options

**Flat XY plane**
Useful for previewing textures and materials of flat objects, like walls, paper and such.

**Sphere**
Useful for previewing textures and materials of sphere-like objects, but also to design metals and other reflective/transparent materials, thanks to the checkered background.

**Cube**
Useful for previewing textures and materials of cube-like objects, but also to design procedural textures. Features a checkered background.

**Monkey**
Useful for previewing textures and materials of organic or complex non-primitive shapes. Features a checkered background.

**Hair strands**
Useful for previewing textures and materials of strand-like objects, like grass, fur, feathers and hair. Features a checkered background.

**Large Sphere with Sky**
Useful for previewing textures and materials of sphere-like objects, but also to design metals and other reflective materials, thanks to the gradient Sky background.

Preview uses OSA (oversampling). Whatever the preview option, it will make use of OSA (oversampling) in order to provide better quality. Disable this option if your computer is already slow or old.
Examples

Plane preview.
Sphere preview.
Cube preview.

Monkey preview.
Hair Strands preview.
Sky Sphere preview.

Diffuse Shaders

Reference
Mode: All Modes
Panel: Shading/Material Context --> Diffuse

A diffuse shader determines, simply speaking, the general color of a material when light shines on it. Most shaders that are designed to mimic reality give a smooth falloff from bright to dark from the point of the strongest illumination to the shadowed areas, but Bforartists also has other shaders for various special effects.

Common Options

All diffuse shaders have the following options:

Color
Select the base diffuse color of the material.

Intensity
The shader’s brightness, or more accurately, the amount of incident light energy that is actually diffusely reflected towards the camera.

Ramp
Allows you to set a range of colors for the Material, and define how the range will vary over a surface. See Color Ramps for details.
Technical Details

Light striking a surface and then re-irradiated via a Diffusion phenomenon will be scattered, i.e., re-irradiated in all directions isotropically. This means that the camera will see the same amount of light from that surface point no matter what the incident viewing angle is. It is this quality that makes diffuse light viewpoint independent.

Of course, the amount of light that strikes the surface depends on the incident light angle. If most of the light striking a surface is reflected diffusely, the surface will have a matte appearance (Light re-irradiated in the diffusion phenomenon).

Light re-irradiated in the diffusion phenomenon.

Tip

Shader Names

Some shaders’ names may sound odd - they are traditionally named after the people who first introduced the models on which they are based.

Lambert

Reference

Mode: All Modes
Panel: Shading/Material Context —> Shaders
Lambert Shader

This is Bforartists’s default diffuse shader, and is a good general all-around workhorse for materials showing low levels of specular reflection.

**Johann Heinrich Lambert (1728-1777)**

was a Swiss mathematician, physicist and astronomer who published works on the reflection of light, most notably the Beer-Lambert Law which formulates the law of light absorption.

This shader has only the default option, determining how much of available light is reflected. Default is 0.8, to allow other objects to be brighter.

![Lambert Shader](image)

The Lambert diffuse shader settings.

### Oren-Nayar

**Reference**

Mode: All Modes  
Panel: Shading/Material Context –> Shaders

Oren-Nayar Shader

Oren-Nayar takes a somewhat more ‘physical’ approach to the diffusion phenomena as it takes into account the amount of microscopic roughness of the surface. Michael Oren and Shree K. Nayar Their reflectance model, developed in the early 1990s, is a generalization of Lambert’s law now widely used in computer graphics.

**Options**

**Roughness**

The roughness of the surface, and hence, the amount of diffuse scattering.
The Toon shader is a very ‘un-physical’ shader in that it is not meant to fake reality but to produce cartoon cel styled rendering, with clear boundaries between light and shadow and uniformly lit/shadowed regions.
**Options**

**Size**
- The size of the lit area

**Smooth**
- The softness of the boundary between lit and shadowed areas

---

The Toon diffuse shader settings.

---

**Minnaert**

**Reference**
- Mode: All Modes
- Panel: Shading/Material Context -> Shaders

---

Minnaert works by darkening parts of the standard Lambertian shader, so if *Dark* is 1 you get exactly the Lambertian result. Higher darkness values will darken the center of an object (where it points towards the viewer). Lower darkness values will lighten the edges of the object, making it look somewhat velvet. Marcel Minnaert (1893-1970) was a Belgian astronomer interested in the effects of the atmosphere on light and images who in 1954 published a book entitled *The Nature of Light and Color in the Open Air*.

---

**Options**

**Dark**
- The darkness of the ‘lit’ areas (higher) or the darkness of the edges pointing away from the light source (lower).
The Minnaert diffuse shader settings.

**Fresnel**

Reference

Mode: All Modes  
Panel: Shading/Material Context → Shaders

Various settings for the Fresnel shader, Cook-Torr Specular shader kept at Intensity 0.5, Hardness: 50

Fresnel Shader, Different Spec

With a Fresnel shader the amount of diffuse reflected light depends on the incidence angle, i.e. from the direction of the light source. Areas pointing directly towards the light source appear darker; areas perpendicular to the incoming light become brighter. Augustin-Jean Fresnel (1788-1827) was a French physicist who contributed significantly to the establishment of the theory of wave optics.
Options

Fresnel
   Power of the Fresnel effect, 5.0 is max.

Factor
   Blending factor of the Fresnel factor to blend in, 5.0 is max.

Specular Shaders

Reference
Mode: All Modes
Panel: Shading/Material Context -> Specular

Specular shaders create the bright highlights that one would see on a glossy surface, mimicking the reflection of light sources. Unlike diffuse shading, specular reflection is viewpoint dependent. According to Snell’s Law, light striking a specular surface will be reflected at an angle which mirrors the incident light angle (with regard to the surface’s normal), which makes the viewing angle very important.

Tip
Not a Mirror!

It is important to stress that the specular reflection phenomenon discussed here is not the reflection we would see in a mirror, but rather the light highlights we would see on a glossy surface. To obtain true mirror-like reflections you would need to use the internal raytracer. Please refer to section RENDERING of this manual.

Common Options

Each specular shader share the following common options:

Specular Color
   The color of the specular highlight

Intensity
   The intensity, or brightness of the specular highlight. This has a range of [0-1].

Ramp
   Allows you to set a range of specular colors for Material, and define how the range will vary over a surface. See Ramps for details.
As a result, a material has at least two different colors, a diffuse, and a specular one. The specular color is normally set to pure white (the same “pure white” as the reflected light source), but it can be set to different values for various effects (e.g. metals tend to have colored highlights).

Technical Details

Specular Reflection.

In reality, the quality of Diffuse and Specular reflection are generated during the same process of light scattering, but are not the same. Diffusion is actually subsurface scattering at a very small scale.

Imagine that a surface is made up of extremely microscopic semi-transparent, reflective facets. The sharpness of Specular reflection is determined by the distribution of the angle of these microfacets on the surface of an object. The more deep and jagged these facets are, the more the light spreads when it hits the surface. When these facets are flatter against the “macrosurface”, the surface will have a tighter reflection, closer to a mirror. This is a condensed explanation of the generally accepted microfacet theory of reflectance, which is the basis of all modern BRDFs (Bi-directional Reflectance Distribution Functions), or shading models.

Because these microfacets are transparent, some light that hits them travels into the surface and diffuses. The light that makes it back out is roughly Lambertian most of the time, meaning that it spreads evenly in all directions. It is also attenuated by the pigmentation in the surface, hence creating what we perceive as diffuse, and the color of an object.

Note that at glancing angles, the reflectivity of a surface will always go to 1.

If it is difficult for you to understand this relationship, try to imagine a ball (say, of centimeter scale): if you throw it against a wall of raw stones (with a scale of roughness of a decimeter), it will bounce in a different direction each time, and you will likely quickly lose it! On the other hand, if you throw it against a smooth concrete wall (with a roughness of, say, a millimeter scale), you can quite easily anticipate its bounce, which follow (more or less!) the same law as the light reflection.

CookTorr

<table>
<thead>
<tr>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode: All Modes</td>
</tr>
<tr>
<td>Panel: Shading/Material Context → Shaders</td>
</tr>
</tbody>
</table>
CookTorr Shader (Lambert 0.8)

CookTorr (Cook-Torrance) is a basic specular shader that is most useful for creating shiny plastic surfaces. It is a slightly optimized version of Phong. Robert L. Cook (LucasFilm) and Kenneth E. Torrance (Cornell University) in their 1982 paper *A Reflectance Model for Computer Graphics* (PDF), they described “a new reflectance model for rendering computer synthesized images” and applied it to the simulation of metal and plastic.

**Options**

**Hardness**

Size of the specular highlight

Phong Shader (Lambert 0.8)

Phong is a basic shader that’s very similar to CookTorr, but is better for skin and organic surfaces. Bui Tuong Phong (1942-1975) was a Vietnamese-born computer graphics pioneer that developed the first algorithm for...
simulating specular phenomenon. His model included components not only for specular lighting, but also diffuse and ambient lighting.

**Options**

**Hardness**
Size of the specular highlight.

<table>
<thead>
<tr>
<th>Tip</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Planet Atmosphere</strong></td>
</tr>
<tr>
<td>Because of its fuzziness, this shader is good for atmosphere around a planet. Add a sphere around the planet, slightly larger than the planet. For its material, use a phong specular shader. Set it to a very low alpha (.05), zero diffuse, low hardness (5) but high specularity (1).</td>
</tr>
</tbody>
</table>

**Blinn**

**Reference**

**Mode: All Modes**

**Panel: Shading/Material Context → Shaders**

Blinn Shader (Oren-Nayar Int 0.8, Rough 0.5)

Blinn is a more ‘physical’ specular shader, often used with the Oren-Nayar diffuse shader. It can be more controllable because it adds a fourth option, an index of refraction, to the aforementioned three. James F. Blinn worked at NASA’s Jet Propulsion Laboratory and became widely known for his work on Carl Sagan’s TV documentary *Cosmos*. The model he described in his 1977 paper *Models of Light Reflection for Computer Synthesized Pictures* (PDF) included changes in specular intensity with light direction and more accurately positioned highlights on a surface.
**Options**

**Hardness**
Size of the specular highlight. The Blinn shader is capable of much tighter specular highlights than Phong or CookTorr.

**IOR**
‘Index of Refraction’. This parameter is not actually used to compute refraction of light rays through the material (a ray tracer is needed for that), but to correctly compute specular reflection intensity and extension via Snell’s Law.

---

**Toon**

**Reference**

Mode: All Modes  
Panel: Shading/Material Context –> Shaders

---

Toon Specular Shader (Toon Diffuse, Int 0.8, Size & Smooth match)

The Toon specular shader matches the Toon diffuse shader. It is designed to produce the sharp, uniform highlights of cartoon cels.

**Options**

**Size**
Size of the specular highlight.

**Smooth**
Softness of the highlight’s edge.

---

**Tip**

Alternative Method

The Toon shader effect can also be accomplished in a more controllable way using ColorRamps.
WardIso

WardIso is a flexible specular shader that can be useful for metal or plastic.

**Gregory J. Ward**

developed a relatively simple model that obeyed the most basic laws of physics. In his 1992 paper, *Measuring and modeling anisotropic reaction*, Ward introduced a Bidirectional Reflectance Distribution Function (BRDF) since then widely used in computer graphics because the few parameters it uses are simple to control. His model could represent both isotropic surfaces (independent of light direction) and anisotropic surfaces (direction dependent). In Bforartists, the Ward specular shader is still called Ward Isotropic but is actually anisotropic. (PDF)

**Options**

**Slope**

Standard deviation for of surface slope. Previously known as the root-mean-square or rms value, this parameter in effect controls the size of the specular highlight, though using a different method to that of the other specular shaders. It is capable of extremely sharp highlights.

**Color Ramps**

In many real life situations - like skin or metals - the color of diffuse and specular reflections can differ slightly, based on the amount of energy a surface receives or on the light angle of incidence. The Ramp Shader options...
in Bforartists allow you to set a range of colors for a Material, and define how the range will vary over a surface, and how it blends with the ‘actual color’ (typically from a material or as output of a texture).

Ramps allow you to precisely control the color gradient across a material, rather than just a simple blend from a brightened color to a darkened color, from the most strongly lit area to the darkest lit area. As well as several options for controlling the gradient from lit to shadowed, ramps also provide ‘normal’ input, to define a gradient from surfaces facing the camera to surfaces facing away from the camera. This is often used for materials like some types of metallic car paint that change color based on viewing angle.

Since texture calculations in Bforartists happen before shading, the Ramp Shader can completely replace texture or material color. But by use of the mixing options and Alpha values it is possible to create an additional layer of shading in Bforartists materials.

**Options**

![Ramp Shader](image)

**Ramps Panel**

The separate Ramp panels for the Diffuse shader and the Specular shader respectively can be toggled on and off using the button.

By default the Ramp panel opens with two colors; the first stop (0) is black and transparent (Alpha=0) and the second stop (1) is white and opaque (Alpha=1).

The position of the color stop markers can be altered by either (1) dragging the stop marker in the colorband or (2) by changing the Pos value in the box.

Color and alpha values for each marker can be set by clicking the box.

**Input**

The input menu contains the following options for defining the gradient:

- **Shader**
The value as delivered by the material’s shader (Lambert, CookTorr) defines the color. Here the amount of light doesn’t matter for color, only the direction of the light.

**Energy**
As Shader, now also lamp energy, color, and distance are taken into account. This makes the material change color when more light shines on it.

**Normal**
The surface normal, relative to the camera, is used for the Ramp Shader. This is possible with a texture as well, but added for convenience.

**Result**
While all three previous options work per lamp, this option only works after shading calculations. This allows full control over the entire shading, including ‘Toon’ style results. Using alpha values here is most useful for tweaking a finishing touch to a material.

### Blend pop-up menu

**Blend**
A list of the various blending modes available for blending the ramp shader with the color from Input.

**Factor**
This slider denotes the overall factor of the ramp shader with the color from Input.

### Colorbands

<table>
<thead>
<tr>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode: All Modes</td>
</tr>
<tr>
<td>Panel: Context Shading -&gt; sub-context Material -&gt; Ramps</td>
</tr>
</tbody>
</table>
A colorband can contain a gradient through a sequence of many colors (with alpha), each color acting across a certain position in the spectrum. Colorbands are used in both materials and textures, as well in other places where a range of colors can be computed and displayed.

**Options**

**Add**

Add a new mark to the center of the colorband with the default color (neutral gray). New marks can also be added by Ctrl-LMB clicking in the colorband itself, which will add the mark at the position of the click with the same color that already exists underneath the mouse pointer.

**Delete**

Remove the currently selected mark from the colorband.

**F**

Flip the colorband.

**0**

The number of the active mark. The values for this mark are those being displayed, and in the colorband, the active mark is displayed as a dashed line. Another marker can be selected (1) using the arrows in the slider, (2) by clicking on the number being displayed and entering a number of a color mark, or (3) by LMB clicking a marker in the colorband.

**Pos**

The position of the active color mark in the colorband (range 0.0–1.0). The position of the color marks can also be changed by LMB dragging them in the colorband.

### Note

**Reordering colors**

If the position of the color marks are reordered, they will be automatically renumbered so that they always start with 0 from the left and increment to the right.

The **Colorswatch** right of the **Position** slider displays the color of the active mark. LMB click it to display a color picker in which values for color (RGB) and transparency (Alpha) can be set.

### Interpolation pop-up menu

**Interpolation**

Various modes of interpolation between marker’s values can be chosen in the Interpolation menu:
Ease
  Ease by quadratic equation.
Cardinal
  Cardinal.
Linear
  Linear (default). A smooth, consistent transition between colors.
B-Spline
  B-Spline.
Constant
  Constant.

## Color Ramps

<table>
<thead>
<tr>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode: All Modes</td>
</tr>
<tr>
<td>Panel: Context Shading → sub-context Material → Ramps</td>
</tr>
</tbody>
</table>

In many real life situations - like skin or metals - the color of diffuse and specular reflections can differ slightly, based on the amount of energy a surface receives or on the light angle of incidence. The Ramp Shader options in Bforartists allow you to set a range of colors for a Material, and define how the range will vary over a surface, and how it blends with the ‘actual color’ (typically from a material or as output of a texture).

Ramps allow you to precisely control the color gradient across a material, rather than just a simple blend from a brightened color to a darkened color, from the most strongly lit area to the darkest lit area. As well as several options for controlling the gradient from lit to shadowed, ramps also provide ‘normal’ input, to define a gradient from surfaces facing the camera to surfaces facing away from the camera. This is often used for materials like some types of metallic car paint that change color based on viewing angle.

Since texture calculations in Bforartists happen before shading, the Ramp Shader can completely replace texture or material color. But by use of the mixing options and Alpha values it is possible to create an additional layer of shading in Bforartists materials.

## Options

Ramps Panel

The separate Ramp panels for the Diffuse shader and the Specular shader respectively can be toggled on and off using the
By default the Ramp panel opens with two colors; the first stop (0) is black and transparent (Alpha=0) and the second stop (1) is white and opaque (Alpha=1).

The position of the color stop markers can be altered by either (1) dragging the stop marker in the colorband or (2) by changing the Pos value in the box.

Color and alpha values for each marker can be set by clicking the box.

**Input**

The input menu contains the following options for defining the gradient:

- **Shader**
  The value as delivered by the material’s shader (*Lambert, CookTorr*) defines the color. Here the amount of light doesn’t matter for color, only the direction of the light.

- **Energy**
  As Shader, now also lamp energy, color, and distance are taken into account. This makes the material change color when more light shines on it.

- **Normal**
  The surface normal, relative to the camera, is used for the *Ramp Shader*. This is possible with a texture as well, but added for convenience.

- **Result**
  While all three previous options work per lamp, this option only works after shading calculations. This allows full control over the entire shading, including ‘Toon’ style results. Using alpha values here is most useful for tweaking a finishing touch to a material.
Blend pop-up menu

**Blend**

A list of the various blending modes available for blending the ramp shader with the color from *Input*.

**Factor**

This slider denotes the overall factor of the ramp shader with the color from *Input*.

---

### Colorbands

**Reference**

<table>
<thead>
<tr>
<th>Mode: All Modes</th>
<th>Panel: Context <em>Shading</em> → sub-context <em>Material</em> → <em>Ramps</em></th>
</tr>
</thead>
</table>

A colorband can contain a gradient through a sequence of many colors (with alpha), each color acting across a certain position in the spectrum. Colorbands are used in both materials and textures, as well in other places where a range of colors can be computed and displayed.

**Options**

**Add**

Add a new mark to the center of the colorband with the default color (neutral gray). New marks can also be added by Ctrl-LMB clicking in the colorband itself, which will add the mark at the position of the click with the same color that already exists underneath the mouse pointer.

**Delete**

Remove the currently selected mark from the colorband.
Flip the colorband.

0

The number of the active mark. The values for this mark are those being displayed, and in the colorband, the active mark is displayed as a dashed line. Another marker can be selected (1) using the arrows in the slider, (2) by clicking on the number being displayed and entering a number of a color mark, or (3) by LMB clicking a marker in the colorband.

Pos

The position of the active color mark in the colorband (range 0.0–1.0). The position of the color marks can also be changed by LMB dragging them in the colorband.

Note

Reordering colors

If the position of the color marks are reordered, they will be automatically renumbered so that they always start with 0 from the left and increment to the right.

The Colorswatch right of the Position slider displays the color of the active mark. LMB click it to display a color picker in which values for color (RGB) and transparency (Alpha) can be set.

Interpolation pop-up menu

Interpolation

Various modes of interpolation between marker’s values can be chosen in the Interpolation menu:

Ease

Ease by quadratic equation.

Cardinal

Cardinal.

Linear

Linear (default). A smooth, consistent transition between colors.

B-Spline

B-Spline.

Constant

Constant.
Shading

In the separate Shading tab six more options are available:

- **Emit**
  - Amount of light to emit

- **Ambient**
  - Amount of global ambient color the material receives. Each material has an Ambient slider that lets you choose how much ambient light that object receives. Set to 1.0 by default.
  - You should set this slider depending on the amount of ambient light you think the object will receive. Something deep in the cave will not get any ambient light, whereas something close to the entrance will get more. Note that you can animate this effect, to change it as the object comes out of the shadows and into the light.

Settings for Ambient Occlusion and Environment Lighting can be found in the World menu, with parameters affecting both these lighting components found in the World Gather menu.

- **Translucency**
  - Amount of diffuse shading on the back side

- **Shadeless**
  - Make this material insensitive to light or shadow; gives a solid, uniform color to the whole object.

- **Tangent Shading**
  - Use the material’s tangent vector instead of the normal for shading - for anisotropic shading effects (e.g. soft hair and brushed metal). This shading was introduced in 2.42, see also settings for strand rendering in the menu further down and in the Particle System menu.

- **Cubic Interpolation**
  - Use cubic interpolation for diffuse values. Enhances the contrast between light areas and shadowed areas

![Non-Cubic Shadow](image1.png) ![Cubic Shadow](image2.png)

Without Cubic enabled. With Cubic enabled.
Materials in Bforartists can be set to be transparent, so that light can pass through any objects using the material. Transparency is controlled using an “alpha” channel, where each pixel has an additional value, range 0-1, in addition to its RGB color values. If alpha=0, then the pixel is transparent, and the RGB values for the surface contribute nothing to the pixel’s appearance; for alpha=1, the surface is fully opaque, and the color of the surface determines the final color of the pixel.

Transparency Panel

In Bforartists, there are three ways in which the transparency of a material can be set: Mask, Z-Buffer and Raytrace. Each of these is explained in more detail below. The Material Preview option with a sphere object gives a good demonstration of the capabilities of these three options.

Common Options

The following property controls are available for all transparency options:

**Alpha**
- Sets the transparency of the material by setting all pixels in the alpha channel to the given value.

**Fresnel**
- Sets the power of the Fresnel effect. The Fresnel effect controls how transparent the material is, depending on the angle between the surface normal and the viewing direction. Typically, the larger the angle, the more opaque a material becomes (this generally occurs on the outline of the object).

**Specular -**
- Controls the alpha/falloff for the specular color.

**Blend**
- Controls the blending between transparent and non-transparent areas. Only used if Fresnel is greater than 0.

**Mask**

This option simply masks the Background. It uses the alpha channel to mix the color of each pixel on the active object plane with the color of the corresponding background pixel, according to the alpha channel of the pixel.
Thus for alpha = 1, the object color is seen - the object is completely opaque; but if alpha = 0, only the background is seen - the object is transparent (but note that any other object behind the active object disappears).

This is useful for making textures of solid or semi-transparent objects from photographic reference material - a mask is made with alpha opaque for pixels within the object, and transparent for pixels outside the object.

See *Mask Transparency*.

### Z Buffer

This uses the alpha buffer for transparent faces. The alpha value of each pixel determines the mix of the basic color of the material, and the color of the pixel is determined from the objects/background behind it. Only basic settings are available with this option; it does not calculate refractions.

### Raytraced Transparency

Uses ray tracing to calculate refractions. Ray tracing allows for complex refractions, falloff, and blurring, and is used for simulating the refraction of light rays through a transparent material, like a lens.

Note that the RayTrace option is only available in the Bforartists Render and Cycles render engines, but not in the Game Engine.

A ray is sent from the camera and travels through the scene until it encounters an object. If the first object hit by the ray is non-transparent, then the ray takes the color of the object.

If the object is transparent, then the ray continues its path through it to the next object, and so on, until a non-transparent object is finally encountered which gives the whole chain of rays its color. Eventually, the first transparent object inherits the colors of its background, proportional to its *Alpha* value (and the Alpha value of each transparent Material hit in between).

But while the ray travels through the transparent object, it can be deflected from its course according to the Index of Refraction (IOR) of the material. When you actually look through a plain sphere of glass, you will notice that the background is upside-down and distorted: this is all because of the Index of Refraction of glass.

---

**Note**

**Enable Raytracing**

To get ray-traced transparency, you need to:

- enable ray tracing in your Render settings. This is done in the Render context → Shading Panel. Ray tracing is enabled by default.
- set your Alpha value to something other than 1.0.
- in order for the background material to receive light passing through your transparent object, *Receive Transparent* must be turned on for that material in the Material → Shadow panel.
Options

The Transparency Panel.

In addition to the common options given above, the following property controls are available:

**IOR**

Index of Refraction. Sets how much a ray traveling through the material will be refracted, hence producing a distorted image of its background. See IOR values for Common Materials below.

**Filter**

Amount of filtering for transparent ray trace. The higher this value, the more the base color of the material will show. The material will still be transparent but it will start to take on the color of the material. Disabled (0.0) by default.

**Falloff**

How fast light is absorbed as it passes through the material. Gives ‘depth’ and ‘thickness’ to glass.

**Limit**

Materials thicker than this are not transparent. This is used to control the threshold after which the filter color starts to come into play.

**Depth**

Sets the maximum number of transparent surfaces a single ray can travel through. There is no typical value. Transparent objects outside the Depth range will be rendered pitch black if viewed through the transparent object that the Depth is set for. In other words, if you notice black areas on the surface of a transparent object, the solution is probably to increase its Depth value (this is a common issue with ray tracing transparent objects). You may also need to turn on transparent shadows on the background object.

**Gloss**

Settings for the glossiness of the material.

**Amount**

The clarity of the refraction. Set this to something lower than zero to get a blurry refraction.

**Threshold**

Threshold for adaptive sampling. If a sample contributes less than this amount (as a percentage), sampling is stopped.

**Samples**

Number of cone samples averaged for blurry refraction.
Examples

Index of Refraction
Influence of the IOR of an Object on the distortion of the background: spheres of Water, Glass and Diamond (top to bottom). There are different values for typical materials: Air is 1.000 (no refraction), Alcohol is 1.329, Glass is 1.517, Plastic is 1.460, Water is 1.333 and Diamond is 2.417.

**Fresnel**

16 pieces of glass rotated in various directions demonstrate the angle-dependent Fresnel effect with ray-traced (left) and alpha buffered transparency (right). Note that the major difference is the lack of IOR effect in the latter case. (Download .blend.)

Settings for Fresnel using ray-traced (left) and Z transparency (right).

Note the specular highlight in the F4 glass tile (which is facing midway between the light and the camera); the Fresnel effect can be seen in row C and column 6 where the faces are turned away from the camera.

The amount of Fresnel effect can be controlled by either increasing the Blend value or decreasing the Alpha value.
Depth

A simple scene with three glasses on a surface, and three lamps. Depth was set to 4, 8, 12, and 14, resulting in render times of 24 sec, 34 sec, 6 min, and 11 min respectively. (Download .blend.)

Increasing Depth also considerably increases render time. Each time a light ray passes through a surface, the ray-tracing algorithm is called recursively. In the example above, each side of each glass has an exterior and an interior surface. Light rays thus have to pass through four surfaces for each glass.

But not only that, at every point on a surface, some of the light can be reflected, or mirrored off the surface in various directions. This results in multiple rays needing to be calculated for each point (often referred to as a tree of rays). In each of the rendered images above there are $640 \times 400 = 256\,000$ pixels. By increasing Depth, at least one tree of rays is added to each pixel.

Be kind to your computer. Carefully placing objects in a scene to avoid overlapping transparent objects is often an interesting alternative.

Hints

Transparent shadows
By default, the shadows of transparent objects are rendered solid black, as if the object was not transparent at all. But in reality, the more transparent an object is, the lighter its shadow will be.

In Bforartists, transparent shadows are set on the materials that receive the shadows from the transparent object. This is enabled and disabled with the Receive Transparent button, in the Material context —> Shadow panel. The shadow’s brightness is dependent on the Alpha value of the shadow casting material.

Alternatives to transparent ray-traced shadows can be found in the World context, namely the Ambient Occlusion, Environment Lighting, and Gather panels. Alternatively, a texture can be used to control the Intensity value of the shadow-receiving material.

### IOR values for Common Materials

The following list provides some index of refraction values to use when ray-traced transparency is used for various liquids, solids (gems), and gases:

#### Gasses
- Air 1.000
- Carbon Dioxide 1.000449
- Oxygen 1.000276

#### Common Liquids
- Alcohol 1.329
- Milk 1.35
- Oil, vegetable (50- C) 1.47
- Shampoo 1.362
- Water (0- C) 1.33346
- Water (100- C) 1.31766
- Water (20- C) 1.33283
- Water (gas) 1.000261
- Water (35- C, room temp) 1.33157
- Vodka 1.363
Common Transparent Materials

- Glass 1.51714
- Ice 1.309
- Rock Salt 1.544

Common Opaque Materials

- Asphalt 1.635
- Chalk 1.510
- Plastic 1.46
- Rubber, Natural 1.5191
- Silicon 4.24

Gemstones

- Diamond 2.417
- Jade, Nephrite 1.61
- Opal 1.45
- Ruby 1.757 - 1.779

Metals

- Aluminum 1.44
- Bronze 1.18
- Copper 1.10
- Gold 0.47
- Iron 1.51
- Lead 2.01
- Platinum 2.33
- Silver 0.18
- Steel 2.50
- Titanium 2.16

Mirror Reflections

Mirror reflections are computed in the Bforartists Render and Cycles render engines using ray tracing. (NB: Reflections are not available in the Game Engine.) Ray tracing can be used to make a material reflect its surroundings, like a mirror. The principle of ray-traced reflections is very simple: a ray is fired from the camera and travels through the scene until it encounters an object. If the first object hit by the ray is not reflective, then the ray takes the color of the object. If the object is reflective, then the ray bounces from its current location and travels up to another object, and so on, until a non-reflective object is finally met and gives the whole chain of rays its color.

Eventually, the first reflective object inherits the colors of its environment, proportional to its Reflectivity value. Obviously, if there are only reflective objects in the scene, then the render could last forever. This is why a mechanism for limiting the travel of a single ray is set through the Depth value: this parameter sets the maximum number of bounces allowed for a single ray.

Note

You need to enable ray tracing in your scene settings if you want to use ray-traced reflections. This is done in the Scene/Render context -> Render Panel. Ray tracing is enabled by default in Bforartists 2.37 and higher.

The Color Swatch in the mirror panel is the color of the light reflected back. Usually, for normal mirrors, use white. However, some mirrors color the reflection (e.g. metals), so you can change the color by clicking on the swatch. The amount of mirrored reflection is determined by the Reflectivity value. If set to something greater than 0, mirrored reflectivity will be activated and the reflection will be tinted the color set in the swatch.
The Mirror Panel

**Enable ray-traced reflections**
Enable or disable ray-traced reflections

**Reflectivity**
Sets the amount of reflectiveness of the object. Use a value of 1.0 if you need a perfect mirror, or set it to 0.0 if you don’t want any reflection.

**Color swatch**
Color of mirrored reflection By default, an almost perfectly reflective material like chrome, or a mirror object, will reflect the exact colors of its surrounding. But some other equally reflective materials tint the reflections with their own color. This is the case for well-polished copper and gold, for example. In order to replicate this within Bforartists, you have to set the Mirror Color accordingly. To set a mirror color, simply click the color swatch in the mirror panel and select a color.

**Fresnel**
Sets the power of the Fresnel effect. The Fresnel effect controls how reflective the material is, depending on the angle between the surface normal and the viewing direction. Typically, the larger the angle, the more reflective a material becomes (this generally occurs on the outline of objects).

**Blend**
A controlling factor to adjust how the blending happens between the reflective and non-reflective areas.

**Depth**
Maximum allowed number of light inter-reflections. If your scene contains many reflective objects and/or
if the camera zooms in on such a reflective object, you will need to increase this value if you want to see surrounding reflections in the reflection of the reflected object (!). In this case, a Depth of 4 or 5 is typically a good value.

**Max Dist**
Maximum distance of reflected rays away from camera (Z-Depth) in Bforartists units. Reflections further than this range fade out to reduce compute time.

**Fade to**
The color that rays with no intersection within the Max Distance take. Material color can be best for indoor scenes, Sky color (World settings) for outdoor scenes.

---

Suzanne in the Fun House (.blend)

**Gloss**
In paint, a high-gloss finish is very smooth and shiny. A flat, or low gloss, finish disperses the light and gives a very blurry reflection. Also, uneven or waxed-but-grainy surfaces (such as car paint) are not perfect and therefore slightly need a Gloss < 1.0. In the example to the right, the left mirror has a Gloss of 0.98, the middle is Gloss = 1.0, and the right one has Gloss of 0.90. Use this setting to make a realistic reflection, all the way up to a completely foggy mirror. You can also use this value to mimic depth of field in mirrors.

**Amount**
The shininess of the reflection. Values < 1.0 give diffuse, blurry reflections and activate the settings below.

**Threshold**
Threshold for adaptive sampling. If a sampling contributes less than this amount (as percentage), sampling is stopped. Raising the threshold will make the adaptive sampler skip more often, however the reflections could become noisier.

**Samples**
Number of cone samples averaged for blurry reflection. More samples will give a smoother result, but will also increase render time.
Anisotropic tangent reflecting spheres with anisotropic set to 0.0, 0.75, 1.0. (.blend)

**Anisotropic**

The shape of the reflection, from 0.0 (circular) to 1.0 (fully stretched along the tangent). If the *Tangent Shading* is on, Bforartists automatically renders blurry reflections as anisotropic reflections. When Tangent is switched on, the *Anisotropic* slider controls the strength of this anisotropic reflection, with a range of 1.0 (default) being fully anisotropic and 0.0 being fully circular, as is when tangent shading on the material is switched off. Anisotropic ray-traced reflection uses the same tangent vectors as for tangent shading, so you can modify the angle and layout the same way, with the auto-generated tangents, or based on the mesh’s UV co-ordinates.
Examples

Fresnel
Demonstration of Fresnel effect with values equal to (from top to bottom) 0.0, 2.5 and 5.0

Let’s undertake a small experiment in order to understand what Fresnel is really about. After a rainy day, go out and stand over a puddle of water. You can see the ground through the puddle. If you kneel just in front of the puddle, your face close to the ground, and look again at a distant point on the puddle of water, the liquid surface part which is closer to you lets you see the ground, but if you move your gaze towards the other end of the puddle, then the ground is gradually masked until all you see is the reflection of the sky. This is the Fresnel effect: having a surface sharing reflective and non-reflective properties according to the viewing angle and the surface normal.

In *Demonstration of Fresnel effect with values equal to (from top to bottom) 0.0, 2.5 and 5.0*, this behavior is demonstrated for a perfectly reflective Material (Mirror Reflectivity 1.0).

Fresnel 0.0 stands for a perfect mirror Material, while Fresnel 5.0 could stand for a glossy Material. It’s barely noticeable but in the lower picture, the Material is perfectly reflective around the edges.

The smoothness of the Fresnel limit can be further controlled using the *Blend* slider.

## Subsurface Scattering

Many organic and inorganic materials are not totally opaque right at the surface, so light does not just bounce off the top surface. Instead, some light also penetrates the skin surface deeply, and scatters around inside, taking on the color of the insides and emerging back out at a different location. Human/animal skin, the skin of grapes, tomatoes, fruits, wax, gels (like honey, or Jello) and so on all have subsurface scattering (SSS), and photo-realism really cannot be achieved without it.

It is important to understand that subsurface scattering and diffuse are one and the same. The difference is in how far light can diffuse beneath the surface before it is absorbed or transmitted back out.

### How it works

Actually calculating the light path beneath the surface of an object is not practical. But it has been shown that it is not necessary to do this, and that one can use a different approach.

Bforartists calculates SSS in two steps:

- At first the irradiance, or brightness, of the surface is calculated, from the front side of the object as well as from its back side. This is pretty much the same as in a normal render. Ambient Occlusion, Radiosity, the type of diffuse Shader, the light color, etc. are taken into account.
- In the second step, the final image is rendered, but now the SSS shader replaces the diffuse shader. Instead of the lamps, the calculated lightmap is used. The brightness of a surface point is the calculated “Average” of the brightness of its surrounding points. Depending on your settings the whole surface may be taken into account, and it’s a bit more complicated than simply calculating the average, but don’t bother too much with the math behind it.

Instead let’s see what SSS does to a distinct light point.
If you turn on SSS, the light is distributed over a larger area. The size of this area depends on the radius values. Instead of distributing all colors with the same amount, you may choose different radius values for each of the RGB colors.

If you use a very large radius value for a color, its light is evenly distributed over the whole object.

**Note**

Note about scatter radius

Because of the way this scattering is calculated, when using large radius values, you will notice fringing artifacts that appear as the complementary color to the predominant color of the scattering. Above, you see in the last image a bluish band in the illuminated area. This is an unfortunate limitation. A way to lessen this effect is use multiple passes with different scatter radii, and average them.
Enabling Subsurface Scattering

Image 4: The SSS Panel. SSS is already enabled.

- Enable SSS by clicking on the Subsurface Scattering button.
- Accessible at the top are various presets. Add new or remove old presets by clicking the + and - buttons. When you select a preset, the Radius values, the RGB Radius and the IOR are set for you. The remaining options are not set (because they are mostly dependent on the size of your object).

SubSurface Scattering doesn’t need ray tracing. But since it is dependent on the incident light and shadows, you need proper shadow calculation (which may need ray tracing).

Options
The numeric sliders control how the light is scattered:

IOR
The Index Of Refraction value determines the falloff of incident light. Higher values means that light falls off faster. The effect is quite subtle and changes the distribution function only a little bit. By the examination of many different materials, values of 1.3 to 1.5 have been found to work well for most materials. If you know the exact material you are trying to simulate, see IOR values for Common Materials.

Scale
The scale of your object, in Bforartists units, across which you want the scattering effect to take place. Scale 1.0 means 1 Bforartists unit equals 1 millimeter, scale 0.001 means 1 Bforartists unit equals 1 meter. If you want to work out what scale value to use in your scene, just use the formula: (size in Bforartists units)/(real world size in millimeters)=scale.
The SSS Color Swatch

Scattering Color (Albedo)

Albedo is the probability that light will survive a scattering event. If you think of scattering as a filter, this is the height of the filter. It is multiplied by the surface color. In practice, this is unintuitive. It should be the same as the surface color, however changing this value has unintuitive results on the scattering effect:

The darker the color the more light is scattered. A value of 1 will produce no scattering effect.

So if you set it to green, the lit areas of the object will appear as green, and green is scattered only a little. Therefore the darker areas will appear in red and blue. You can compensate the different scattering by setting a larger radius for the color.

RGB Radius

This is not in fact the radius of the subsurface scattering, but the average path length between scattering events. As the light travels through the object it bounces around then emerges from the surface at some other point. This value corresponds to the average length the light travels between each bounce. The longer the path length is, the further the light is allowed to scatter. This is the main source of a material’s perceived “scatter color.” A material like skin will have a higher red radius than green and blue.

Subsurface scattering is the diffusion of light beneath the surface. You control how far the light spreads to achieve a specific result.

Blend:

Color
This controls how much the R, G, B option modulates the diffuse color and textures. Note that even with this option set to 0.0, the R, G, B option still influences the scattering behavior.

Texture
How much the surface texture is blurred along with the shading.

Scattering Weight:

Front
Factor to increase or decrease the front scattering. When light enters through the front of the object, how much is absorbed or added? (Normally 1.0 or 100%).

Back
Factor to increase or decrease the back scattering. Light hitting an object from behind can go all the way through the object and come out on the front of the object. This happens mostly on thin objects,
like hands and ears.

Error

This parameter controls how precisely the algorithm samples the surrounding points. Leaving it at 0.05 should give images without artifacts. It can be set higher to speed up rendering, potentially with errors.

Setting it at 1.0 is a good way to quickly get a preview of the look, with errors.

Developing your own SSS material

The Traditional Approach

A more common but less intuitive approach is to use “layering”. This is a simplified version of the layering approach. See the external links for more information:

- Set the SSS color on a value of your choice, normally the predominant color of the object. If you want to use different radii for the colors, don’t make it too dark.
- Set the scale factor. If you want to see much translucency you need small objects or large scale values.
- Set the radius values.
- Adjust the brightness with the Front and Back values.

A more intuitive approach

- Set the Scattering color to .5
- Set the Front weight to 2.
- Set the scale factor based on the size of your object relative to the scene. If you want to see much translucency you need small objects or large scale values.
- Set the radius values appropriately.

Examples

Skin

Increasing SSS scale (.blend)

See also

- Development Release Log: Subsurface Scattering
- Ben Simonds: Three Layer SSS in Bforartists Demystified
Strands

The Strand section of the Material editor is specific to the rendering of Hair particles. There are two different strand methods available:

**Polygon strands**

This is the default (old) method. The strands are rendered as flat polygons. The number of polygons depend on the Steps settings in the Render panel of the Object context, Particles sub-context.

**Strand Primitive**

You activate Strand Primitive with the button Strand render in the Render panel of the particle system. The hair curves are not stored as polygons; only the key points are stored, which are then converted to polygons on the fly. A second difference is the way transparency works. Rather than rendering using the existing system, all strand segments in a part are sorted front to back and rendered in that order.

Strand Primitives:

- Are more memory efficient and faster, to make rendering of large amounts of fur and grass possible. For good performance, the render steps button should be lowered (e.g. 2 should be good enough fur), since the result will be a smoothed curve anyway. You need 1 to 2 render steps less than steps in the 3D window. Also, using more render parts helps to reduce memory usage.
- Have a distance of vision reduction (in the Render panel under Child Simplification) for children from faces.
- May be faded out towards the tip without an additional texture.
- Are not ray traced. So they are not visible through ray-transparent materials or in a ray mirror (you can use Environment Mapping for that).
- Have shape problems if they are rendered with a greater width.
- Cannot carry a UV-Texture along the strand.

Polygon strands:

- Work well with greater width, so you can use them as an alternative to billboards because the strands may have an animated shape.
- Can be textured with a UV-Texture along the strands.
- Are seen by ray tracing.

**Strands Shading**

Strands are rendered with the material of the underlying face/vertex, including shading with a UV-Texture.
Since you can assign more than one material to each face, each particle system may have its own material and the material of the underlying face can be different from the material of the strands.

Additionally strands can be shaded along the strand (from root to tip) with a mono-dimensional texture; only polygon strands can carry a two-dimensional UV-Texture.

The options for strand shading are in the Strands section of the Material context.

**Root**

Width of the hair at the root.

**Tip**

Width of the hair at the tip.

**Minimum**

This is the minimum thickness (in pixels) of the strands. Strands below that size are not rendered smaller, but are faded to alpha (well, the fading works only for strand primitives). This gives a much better rendering result for thin hair.

**Bforartists Units**

Normally strands are quite thin; the thickness is given in screenpixels. If you use Bforartists units (BU) you may set the root value up to 2 BU, and the tip value up to 1 BU. You have to consider the overall object size, because the smallest possible size is 0.001 BU. So if you use 1 BU for 1 meter the smallest possible size would be 1 mm (too thick for thin hair).

**Use Tangent Shading**

Calculates the light as if the strands were very thin and round. This makes the hair appear brighter and shinier. Disabling the “Tangent Shading” option will still render nicely, but resembles more solid strands, as though made of metal or wood.

**Shape**

This slider allows you to control the interpolation. Default (0.0) is a linear interpolation between Root and Tip. A negative value will make the strand narrower (spiky), a positive value will make it fatter.

**Width Fade**

To fade out along the width of the strand. This works only for Strand Primitives. 0.0 is no fading at all, 1.0 linear fading out.

**UV Layer**

You can texture polygon strands with a UV-Texture. Fill in the name of the UV-Set (not the texture) here.
You also have to load the texture in the *Shading* context, *Texture* and *Material* sub-contexts (*Mapping: UV*; you may use every *Influence* setting you like - especially the alpha value; see *Image 3*).

**Surface Diffuse**

Computes the strand normal, taking the normal at the surface into account. This eases the coloring and lighting of hair a lot, especially for Strand Primitives. Essentially hair reacts similar to ordinary surfaces and don’t show exaggerated strong and large specular highlights.

**Distance**

The distance in Bforartists units over which to blend in the normal at the surface (if you want to use *Surface Diffuse* only for Grass/Fur at greater distances).

**Texturing along the Strand**

[Image 4: Fading a strand to alpha...]

Image 5: ...And the render result.

Strands can be textured along the strand, i.e. from root to tip. To do that you have to select *Strand/Particle* in the *Coordinates* drop-down in the *Mapping* panel of the *Material* sub-context.

Pretty much the most important setting is shown in (*Image 4*), how to fade the tip of a strand to alpha to make nice, fuzzy-looking hair. Normally you would use a linear blend texture for this.

You may of course set any attribute you like, especially color. Be careful with specularity; hairs tend to get too shiny.
Strand render Simplification

If you use Strand Primitives (Strand render button) and have activated Interpolated Children, the Child Simplification option appears. The strand render has options to remove child strands as the object’s faces become smaller.

Reference Size
This is the approximate size of the object on screen (in pixels), after which simplification starts.

Rate
How fast strands are removed.

Transition
The transition period for fading out strands as they are removed.

Viewport
This removes strands on faces that are outside of the viewport.

Rate
Controls how fast these are removed.

Options

Material Options Panel
This panel provides a series of control options concerning how objects using this material will appear in the rendered image. All controls are default “Off” unless otherwise stated.

Traceable (default On)
Include this material and the geometry that uses it in ray-tracing calculations. See Transparency for details of ray-tracing.

Full Oversampling
Force this material to render full shading/textures for all anti-aliasing samples.
Sky
  Render this material with zero alpha, but with sky background in place (scanline only)

Use Mist
  Use mist on this material (see “World Settings” for more details)

Invert Z depth
  Render material’s faces with an inverted Z buffer (scanline only)

Z Offset
  Give faces an artificial Z offset for Z transparency.

Light Group
  Limit lighting to lamps in this light group.

Exclusive
  Uses the light group exclusively - these lamps are excluded from other scene lighting

Local
  When linked in, uses local light group with the same name.

Face Textures
  Replace object’s base color with color from UV map image textures.

Face Textures Alpha
  Replace object’s base alpha with alpha from UV map image textures.

Vertex Color Paint
  Replace object’s base color with vertex paint colors (multiply with ‘texture face’ face assigned textures)

Vertex Color Light
  Add vertex paint colors as additional lighting. (This can be used to produce good incandescence effects).

Object Color
  Modulate the result with a per object color

UV Project (default On)
  Use to ensure UV interpolation is correct for camera projections (use with UV project modifier).

Pass Index
  Index number for the IndexMA render pass.

Shadows

The shadows that appear in a scene are affected by a combination of the layout of objects, the shape of the objects, the materials of the objects, and the lighting. In Bforartists, the Display Mode (Single Texture, Multitexture, or GLSL) also affects the appearance of shadows. See Shadows for a more complete description of this subject.

Tip

Shadows in 3D mode

To see shadows in 3D (textured) mode, you must have switched to GLSL mode before making any materials. In MultiTexture mode, shadows only appear in the rendered image: none of these can be seen in the preview image.
The Shadow panel in the Materials Properties window (Fig. 1) controls the effects that the material can have on the shadows that appear in the scene. The various properties are described in the sections below.

![Fig. 1: Shadow Panel.](image)

**Options**

The following properties can be set for each individual material with which objects in the scene are shaded. The effects are illustrated with rendered images for a simple scene (Fig. 2) consisting of two “posts”, one with a red (totally non-transparent) material; one green (partially transparent) material, set up on a light blue plane to receive the shadows. The illustrations were all taken in Bforartists Render engine, with Multitexture mode.

**Shadow Receiving Object Material**

The following options affect the material that receives shadows:

- **Receive**
  - Allows this material to receive full-intensity shadows (Fig. 3).

- **Receive Transparent**
  - Allows this material to receive shadows whose intensity is modified by the transparency and color of the shadow-casting object (Fig. 4).
**Shadow Casting Object Material**

The following options affect the material that casts shadows:

**Cast Only**
- Material appears transparent, but it still casts shadows (Fig. 5).

**Casting Alpha**
- ??

**Shadows Only**
- Material appears transparent except for where it receives shadows from other objects, and also it retains its own transparency (Fig. 6). Note the faint image of the partly-transparent post.

**Shadow and Distance**
- ??

---

Fig. 3: Plane - Receive.

Fig. 4: Plane - Receive + Receive Transparency.

Fig. 5: Posts - Cast Only.
Buffered Shadow Options

In addition to the shadow options described above, there are further material properties which control buffered shadow features. See section on Spot Buffered Shadows for further discussion of these techniques.

**Cast Buffer Shadow**
- Casts shadows from shadow buffer lamps.

**Buffer Bias**
- Multiplication factor for Buffer shadows (0 = ignore)

**Auto Ray Bias**
- Prevent raytraced shadow errors on surfaces with smooth shaded normals.

**Ray Bias**
- Bias value to be used.

**Cast Approximate**
- Allow this material to cast shadows when using approximate ambient occlusion.

---

**Introduction to Material Nodes**

In addition to creating materials as just described using all the settings on all the materials panels, Bforartists allows you to create a material by routing basic materials through a set of nodes. Each node performs some operation on the material, changing how it will appear when applied to the mesh, and passes it on to the next node. In this way, very complex material appearances can be achieved.

You should already be familiar with general material concepts and how to create materials/textures using the material menu. You should also have a general understanding of the texture coordinate systems available in Bforartists (e.g. Generated, UV, etc.). Also, many aspects of a node will be skipped here because in later sections you will see the function expanded upon. Each section builds off the previous.

To start, the node system does not make the material menu obsolete. Many features and material settings are still only accessible through the material panel (e.g. Ray Mirror). However with the advent of nodes, more complex and fantastic materials can be created since we now have greater control.

Just in case you’re not (yet) familiar with the concepts: when you create a system of nodes (otherwise known as a “noodle”), you’re describing a data-processing pipeline of sorts, where data “flows from” nodes which describe various sources, “flows through” nodes which represent various processing and filtering stages, and finally “flows into” nodes which represent outputs or destinations. You can connect the nodes to one another in many different ways, and you can adjust “knobs,” or parameters, that control the behavior of each node. This
gives you a tremendous amount of creative control. And, it will very quickly become intuitive.

Having said all that, let’s begin with a normal material.

Here we have the standard material we have added to a cube mesh. We could, as we have in the past, add color and other settings to this material and it would certainly look nice. But let’s say we are just not getting what we are looking for? What if we want to control the creation more tightly or add more complexity? Here is where nodes come in.

Making this node map is accomplished by working in a Node Editor window. This section covers:

- Enabling Material Nodes.
- The Node Editor window, its basic controls, and working with nodes.
- The specific types of nodes available for materials.

### Accessing The Node Editor

First lets enter the node editor and make sure that the node editor has the material node button (the sphere icon) pressed, not the composite or texture node buttons.

### Enabling Node Materials in the Material Buttons

Let’s take the base material and hit the Nodes button next to the material name in the material panel or the node editor. You will see a change in the material panel.

What you have just done is told Bforartists to make the material you were on to become the node tree. Most of
the panels we normally find in the material menu are now gone.

Accessing the Compositing screen

If you switch to the *Compositing* screen (**Ctrl-Left** if you are on the default screen) you’ll find a *Node Editor* on the top half of the screen. When you enabled material nodes, a material node and an output node were automatically added to the node editor.

You can also split the 3D view in the default screen in two and change one into a *Node Editor*.

It is important to note that you can add a new material (which you can edit and change like any other material in
the material panel), add an already created material or append a material from another Bforartists file, and also use the material that you used to create the node tree.

Here, we added a new material in the Node editor (Material.001), and as we did, we can access the properties of this material in the material’s menu.

Types of Material Nodes

This section is organized by type of node, which are grouped based on similar functions:

**Input**
- Introduces a material or component to the node map.

**Output**
- Displays the result in progress as a small image.

**Color**
- Manipulates the colors of the material.

**Vector**
- Change the way light is reflected off the material.

**Convertors**
- Convert colors to other material colors.

Node Types

- Color Nodes
  - MixRGB
  - RGB Curves
  - Invert
  - Hue Saturation Value
- Convertor Nodes
  - ColorRamp Node
  - RGB to BW Node
  - Math Node
  - Vector Math Node
  - Squeeze Value Node
  - Separate RGB Node
  - Combine RGB Node
  - Separate HSV Node
  - Combine HSV Node
- Input Nodes
  - Material Node
  - Extended Material Node
  - Camera Data Node
  - Lamp Data Node
  - Value Node
  - RGB Node
  - Texture Node
  - Geometry Node
• Output Node
• Vector Nodes
  • Normal Node
  • Mapping Node
  • Vector Curves

## Color Nodes

### MixRGB

![MixRGB Node](image)

MixRGB node

This node mixes a base color or image (threaded to the top socket) together with a second color or image (bottom socket) by working on the individual and corresponding pixels in the two images or surfaces. The way the output image is produced is selected in the drop-down menu. The size (output resolution) of the image produced by the mix node is the size of the base image. The alpha and Z channels (for compositing nodes) are mixed as well.

Not one, not two, but count ‘em, sixteen mixing choices include:

See also

Color Blend Modes for details on each blending mode.

### Inputs

**Fac**

The amount of mixing of the bottom socket is selected by the Factor input field (Fac:). A factor of zero does not use the bottom socket, whereas a value of 1.0 makes full use. In Mix mode, 0.5 is an even mix between the two, but in Add mode, 0.5 means that only half of the second socket’s influence will be applied.

**Color 1**

Input color value. The value can be provided by another node or set manually. Includes a color swatch, allowing you to select the color directly on the node.

**Color 2**
Input color value. The value can be provided by another node or set manually. Includes a color swatch, allowing you to select the color directly on the node.

**Outputs**

**Color**

Value of the color, combined by the node.

**Controls**

**Clamp**

Clamp result of the node to 0...1 range.

**RGB Curves**

For each color component channel (RGB) or the composite (C), this node allows you to define a bezier curve that varies the input (x-axis) to produce an output value (y-axis). Clicking on one of the C R G B components displays the curve for that channel.

**Note**

- Read more about using the Curve Widget.
- *RGB Curves node in the compositor* (includes examples)
Invert

This node simply inverts the input values and colors.

**Inputs**

**Fac:**
Factor. The degree of node’s influence in node tree. The value can be provided by another node or set manually.

**Color**
Input color value. The value can be provided by another node or set manually. Includes a color swatch, allowing you to select the color directly on the node.

**Outputs**

**Color**
Value of the color, combined by the node.

Hue Saturation Value

Use this node to adjust the Hue, Saturation, and Value of an input.

**Inputs**

**Fac**
Factor. The degree of node’s influence in node tree. The value can be provided by another node or set manually.

**Hue**
Input hue value of color. The value can be provided by another node or set manually.

**Saturation**
Input saturation value of color. The value can be provided by another node or set manually.

**Value**
Input HSV-Value of color. The value can be provided by another node or set manually.

**Fac**
Factor. The degree of node’s influence in node tree. The value can be provided by another node or set manually.

**Color**
Input color value. The value can be provided by another node or set manually. Includes a color swatch, allowing you to select the color directly on the node.

**Outputs**

**Color**
Value of the color, combined by the node.

---

## Convertor Nodes

As the name implies, these nodes convert the colors in the material in some way.

### ColorRamp Node

![ColorRamp node](image)

The ColorRamp Node is used for mapping values to colors with the use of a gradient. It works exactly the same way as a Colorband for textures and materials, using the Factor value as a slider or index to the color ramp shown, and outputting a color value and an alpha value from the output sockets.

By default, the ColorRamp is added to the node map with two colors at opposite ends of the spectrum. A completely black black is on the left (Black as shown in the swatch with an Alpha value of 1.00) and a whitewash white is on the right.
To select a color, LMB click on the thin vertical line/band within the colorband. The example picture shows the black color selected, as it is highlighted white. The settings for the color are shown above the colorband as (left to right): color swatch, Alpha setting, and interpolation type.

**Inputs**

**Fac:**
Factor. The degree of node’s influence in node tree. The value can be provided by another node or set manually.

**Outputs**

**Color**
Value of the color, combined by the node.

**Alpha**
Value of the alpha, combined by the node.

**Controls**

Add a new mark to the center of the colorband with the default color (neutral gray).

Remove the currently selected mark from the colorband.

Flip the colorband.

Modes of interpolation between marker’s values color ramp

**Interpolation**

Various modes of interpolation between marker’s values can be chosen in the Interpolation menu:

**Ease**
Ease by quadratic equation.
Cardinal
Cardinal.

Linear
Linear (default). A smooth, consistent transition between colors.

B-Spline
B-Spline.

Constant
Constant.

Colorband

**Colorband**
Contain a gradient through a sequence of many colors (with alpha), each color acting across a certain position in the spectrum.

The number of the active mark.

Pos. The position of the active color mark in the colorband (range 0.0–1.0). The position of the color marks can also be changed by LMB dragging them in the colorband.

Color swatch to color selection for a mark

**Color Selector**
Allows set color and alpha values for each marker.

See more details about node controls’ functions [here](#).

**RGB to BW Node**

RGB to BW node
This node converts a color image to black-and-white.
**Inputs**

**Color:**
Input color value. Includes a color swatch, allowing you to select the color directly on the node.

**Outputs**

**Value**
Black-and-white value of the input color, converted by the node.

**Math Node**

![Math node](image)

Math node
This node performs the selected math operation on an image or buffer. All common math functions are supported. If only an image is fed to one Value socket, the math function will apply the other Value consistently to every pixel in producing the output Value. Select the math function by clicking the up-down selector where the “Add” selection is shown.

**Inputs**

**Value**
Input value 1 (upper). The value can be provided by another node or set manually.

**Value**
Input value 2 (lower). The value can be provided by another node or set manually.

**Outputs**

**Value**
Output value, converted by the node.

**Controls**

**Clamp**
Clamps the result between 0 and 1.

**Operation**
Selector the math function for conversion.

**Add**
Add the two inputs

**Subtract**
Subtract input 2 from input 1

**Multiply**
Multiply the two inputs

**Divide**
Divide input 1 by input 2

**Sine**
The sine of input 1 (degrees)

**Cosine**
The cosine of input 1 (degrees)

**Tangent**
The tangent of input 1 (degrees)

**Arcsine**
The arcsine (inverse sine) of input 1 (degrees)

**Arccosine**
The arccosine (inverse cosine) of input 1 (degrees)

**Arctangent**
The arctangent (inverse tangent) of input 1 (degrees)

**Power**
Input 1 to the power of input 2 (input1^input2)

**Logarithm**
Log base input 2 of input 1

**Minimum**
The minimum of input 1 and input 2

**Maximum**
The maximum of input 1 and input 2

**Round**
Rounds input 1 to the nearest integer

**Less Than**
Test if input 1 is less than input 2, returns 1 for true, 0 for false

**Greater Than**
Test if input 1 is greater than input 2, returns 1 for true, 0 for false

**Modulo**
Division of input 1 by input 2 with remainder.

**Absolute**
Always return non-negative value from any operation input 2 between input 1.

### Vector Math Node

![Vector Math Node](image)
Vector Math node

This node performs the selected math operation on vectors. Select the math function by clicking the up-down selector where the “Add” selection is shown.

**Inputs**

**Vector**
- Input vector 1 (upper). The value can be provided by another node or set manually.

**Vector**
- Input vector 2 (lower). The value can be provided by another node or set manually.

**Outputs**

**Vector**
- Output vector, converted by the node.

**Value**
- Output value, converted by the node.

**Controls**

**Operation**

Selector the math function for conversion.

**Add**
- Adding input 1 and 2.

**Subtract**
- Subtracting input 1 and 2.

**Average**
- Averaging input 1 and 2.

**Dot Product**
- Algebraic operation that takes two equal-length sequences of vectors 1 and 2 and returns a single number. Result - scalar.

**Cross Product**
- Geometric binary operation on two vectors 1 and 2 in three-dimensional space. It results in a vector which is perpendicular to both and therefore normal to the plane containing them. Result - vector.

**Normalize**
- Normalizing input 1 and 2.

**Squeeze Value Node**

![Squeeze Value Node](image)
Squeeze Value node

This node is used primarily in conjunction with the Camera Data node used. The camera data generate large output values, both in terms of the depth information as well as the extent in the width. With the squeeze Node high output values to an acceptable material for the node degree, ie to values between 0.0 - 1.0 scaled down.

Inputs

Value
Any numeric value. The value can be provided by another node or set manually.

Width
Determines the curve between sharp S-shaped (width = 1) and stretched (Width = 0.1). Negative values reverse the course. The value can be provided by another node or set manually.

Center
The center of the output value range. This input value is replaced by the output value of 0.5. The value can be provided by another node or set manually.

Outputs

Value
A value between 0 and 1, converted by the node.

Separate RGB Node

This node separates an image into its red, green and blue channels. The colors are then converted to intensity, which returns a greyscale to the output. For example, if you have an image with pure green, then the red and blue outputs will be black and the green output will be completely white. Mixed colors will return mixed values according to their RGB intensity.

Inputs

Image
Input color value. Includes a color swatch, allowing you to select the color directly on the node.

Outputs

R
Value of the red color channel, separated out by the node.

G
Value of the green color channel, separated out by the node.
Value of the blue color channel, separated out by the node.

**Combine RGB Node**

This node combines a color (image) from separated red, green, blue channels.

**Inputs**

R
  - Input value of red color channel. The value can be provided by another node or set manually.

G
  - Input value of green color channel. The value can be provided by another node or set manually.

B
  - Input value of blue color channel. The value can be provided by another node or set manually.

**Outputs**

Image
  - Output value of the color, combined by the node.

**Separate HSV Node**

This node separates an image into image maps for the hue, saturation, value channels. Three values, often considered as more intuitive than the RGB system (nearly only used on computers)

Use and manipulate the separated channels for different purposes; i.e. to achieve some compositing/color
adjustment result. For example, you could expand the Value channel (by using the multiply node) to make all
the colors brighter. You could make an image more relaxed by diminishing (via the divide or map value node)
the Saturation channel. You could isolate a specific range of colors (by clipping the Hue channel via the
Colorramp node) and change their color (by the Add/Subtract mix node).

**Inputs**

**Color**

Input color value. Includes a color swatch, allowing you to select the color directly on the node.

**Outputs**

**H**

Value of the hue color channel, separated out by the node (choose a color of the rainbow).

**S**

Value of the saturation color channel, separated out by the node (the *quantity* of hue in the color (from
desaturate - shade of gray - to saturate - brighter colors)).

**V**

Value of the value color channel, separated out by the node (the *luminosity* of the color (from ‘no light’ -
black - to ‘full light’ - ‘full’ color, or white if Saturation is 0.0)).

**Combine HSV Node**

This node combines a color from separated hue, saturation, value color channels.

**Inputs**

**H**

Input value of hue color channel. The value can be provided by another node or set manually.

**S**

Input value of saturation color channel. The value can be provided by another node or set manually.

**V**

Input value of value color channel. The value can be provided by another node or set manually.

**Outputs**

**Color**

Output value of the color, combined by the node.
Input Nodes

A starting material is created in the Materials Panel. The Nodes button is enabled to add that material to the list of noded materials shown in the Node Editor window header. Other inputs to the node map include:

- A value
- A color
- A texture
- Geometry
- Material
- Camera Data
- Lamp Data

Material Node

Reference

Panel: Node Editor -> Material Nodes
Menu: Input -> Material
Material node

The Material node is used to add a material to the node program. Materials can be anything from pure shading to fully layered with textures. It inputs the main attributes of a material (color, alpha and normal vector) into the map.

Output

Materials can output color (which includes shading and any textures assigned to it), alpha, and the final normal calculated from any textures it has.

Color

value of the color, combined by the node.

Alpha

value of the alpha, combined by the node.

Normal

direction of the normal, combined by the node.

Input

Materials can take inputs for colors, inputs for diffuse color and specularity color, a value for reflectivity, and a normal.

Color

The base color of the paint. Can be set

- manually by LMB clicking on the color swatch applet next to the socket, choosing a color using the control panel that pops up, and pressing Return
- based on an Active Material which is specified using the material panels, or
- plugged in from an RGB color generator.

Spec

The color that is reflected as you get perpendicular to the light source reflecting off the surface. The color can be

- plugged in from another node or...
- set manually by LMB clicking on and using the color swatch applet.

Refi:

The degree to which the material reflects light and gives off its color. The value can be provided by another node or set manually.

Normal

The lighting condition.

Controls

Material field

You can browse and select materials here.

Diffuse toggle

Turn on/off Diffuse Color.

Specular toggle

Turns on/off Specularity calculation.

Invert Normal toggle
Inverts the material input normal when activated (which, of course, is a combination of the 3D normal given to it by the 3D object plus the normal input point).

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Normal Override</strong></td>
</tr>
<tr>
<td>The normal input socket does not in any way blend the source normal with the underlying geometry. Any plugged in Geometry here overrides the Normal lighting conditions.</td>
</tr>
</tbody>
</table>

### Using the Material Node with Specularity

To make a material node actually generate a color, you have to specify at least a basic input color, and optionally a specularity color. The specularity color is the color that shines under intense light.

For example, consider the mini-map to the right. The base color, a dark blue, is connected from an RGB color generator node to the Color input socket. The specular color, yellow, is connected to the Spec input. Under Normal lighting conditions on a flat surface, this material will produce a deep blue color and, as you approach a spot perpendicular to the light, you will see the yellow specular color mix in.
<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable Spec</td>
</tr>
</tbody>
</table>

To see specularity, you have to enable it by clicking the blue Spec button located just below the material color swatch in the node.
Extended Material Node

Adds additional input and output channels to the material node.
**Input**

**Color**
- Includes a color swatch, allowing you to select the color directly on the node.

**Mirror Color**
- Color of mirrored reflection.

**Ambient**
- Amount of global ambient color the material receives.

**Emit**
- Amount of light to emit.

**SpecTra**
- Alpha for the specular color.

**Ray Mirror**
- Amount of reflectiveness of the object.

**Alpha**
- Transparency of the material by setting all pixels in the alpha channel to the given value.

**Translucency**
- Amount of diffuse shading on the back side.

**Output**

Materials can additionally output the followings:

**Diffuse**
- Value of the diffuse color, combined by the node.

**Spec**
- Value of the specular color, combined by the node.

**AO**
- Value of the Ambient Occlusion, combined by the node.

---

**Camera Data Node**

- **View Vector**
  - A Camera space vector from the camera to the shading point.

- **View Z Depth**
  - How far away each pixel is from the camera.

- **View Distance**
  - Distance from the camera to the shading point.
**Lamp Data Node**

Lamp Data node

The Lamp Data node is used to obtain information related to a specified lamp object. Select a lamp object listed in the Lamp field, then the following outputs will be available:

**Color**  
Lamp color multiplied by the lamp energy.

**Light Vector**  
An unit vector in the direction from the shading point to the lamp.

**Distance**  
Distance from the shading point to the lamp.

**Shadow**  
Shadow color that the other objects cast on the shading point.

**Visibility Factor**  
Light falloff ratio at the shading point.

The light textures and the shadow textures affect the Color and Shadow outputs, respectively.

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Portability to Various Scenes</strong></td>
</tr>
</tbody>
</table>

If multiple materials use a Lamp Data node linking to the same lamp object, including the Lamp Data node into a node group is recommended. Otherwise, when the mesh objects are imported to the other scene, all the materials may need to be modified.

**Value Node**

Value node

The Value node has no inputs; it just outputs a numerical value (floating point spanning 0.00 to 1.00) currently entered in the NumButton displayed in its controls selection.
Use this node to supply a constant, fixed value to other nodes’ value or factor input sockets.

**RGB Node**

RGB node

The RGB node has no inputs. It just outputs the value Color currently selected in its controls section.

**Texture Node**

Texture node

A texture, from the list of textures available in the current blend file, is selected and introduced through the value and/or color socket.
Example of the applying Texture node

**Input**

**Vector**
Uses for map the texture to a specific geometric space.

**Outputs**

**Value**
Straight black-and-white value of the texture, combined by the node.

**Color**
Texture color output, combined by the node.

**Normal**
Direction of normal texture, combined by the node.

In the example to the right, a cloud texture, as it would appear to a viewer, is added to a base purple material, giving a velvet effect.

Note that you can have multiple texture input nodes. With nodes, you simply add the textures to the map and plug them into the map.
Geometry Node

The geometry node is used to specify how light reflects off the surface. This node is used to change a material’s Normal response to lighting conditions.

Use this node to feed the Normal vector input on the Material node, to see how the material will look (i.e. shine, or reflect light) under different lighting conditions. Your choices are:

Global
- Global position of the surface.

Local
- Local position of the surface.

View
- Viewed position of the surface.

Orco
- Using the Original Coordinates of the mesh.

UV
- Using the UV coordinates of the mesh, selected in the field in bottom node.

Normal
- Surface Normal; On a flat plane with one light above and to the right reflecting off the surface.

Vertex Color
- Allows for output value of group vertex colors, selected in the field in bottom node.

Vertex Alpha
- Allows for output alpha value of vertex.

Front/Back
Allows for output to take into account front or back of surface is light relative the camera.

**Note**

These are exactly the same settings as in the *Mapping* panel for *Textures*, though a few settings - like *Stress* or *Tangent* - are missing here. Normally you would use this node as input for a Texture Node.

**Geometry Node Example using a UV image**

Setup to render an UV-Mapped Image Texture.

E.g.: To render an UV-mapped image, you would use the *UV* output and plug it into the *Vector* Input of a texture node. Then you plug the color output of the texture node into the color input of the material node - which corresponds to the setting on the *Map To* panel.

**Output Node**
Output material node

At any point, you may want to see the work in progress, especially right after some operation by a node. Simply create another thread from the output socket of the node to the picture input socket of an Output node to see a mini-picture.

Connect the alpha channel to set/see transparency.

**Note**

**Effective Output Node**

The only Output node that is used for the Material in the end (i.e the only non-Preview) has a little **red sphere** on the upper right.

---

**Vector Nodes**

Vector nodes manipulate information about how light interacts with the material, multiplying vector sets, and other wonderful things which can become quite advanced. Even if you don’t have experience with vector maths, you’ll find these nodes to be very useful.

Vectors, in general, are two or three element values, for example, surface normals are vectors. Vectors are also important for calculating shading.

**Normal Node**

The Normal node generates a normal vector and a dot product. Click and Drag on the sphere to set the direction of the normal.

This node can be used to input a new normal vector into the mix. For example, use this node as an input to a Color Mix node. Use an Image input as the other input to the Mixer. The resulting colorized output can be easily varied by moving the light source (click and dragging the sphere).

The (face) normal is the direction of the face in relation to the camera. You can use it to do the following:
• Use this node to create a fixed direction –> output Normal.
• Calculate the Dot -Product with the Normal -Input. The Dot -Product is a scalar value (a number).
  • If two normals are pointing in the same direction the Dot -Product is 1.
  • If they are perpendicular the Dot -Product is zero (0).
  • If they are antiparallel (facing directly away from each other) the Dot -Product is -1. And you never thought you would use the Vector Calculus class you took in college - shame on you!

So now we can do all sorts of things that depends on the viewing angle (like electron scanning microscope effect). And the best thing about it is that you can manipulate the direction interactively.

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>One caveat</td>
</tr>
<tr>
<td>The normal is evaluated per face, not per pixel. So you need enough faces, or else you don’t get a smooth result</td>
</tr>
</tbody>
</table>

**Inputs**

**Normal**
3D-direction of the face in relation to the camera. The value can be provided by another node or set manually.

**Outputs**

**Normal**
Fixed 3D-direction, combined by the node.

**Dot**
Scalar value (a number), combined by the node.

**Controls**

**Interactive Normal node preview**
Allows click and drag on the sphere in node center to set the direction of the normal.
## Mapping Node

Essentially mapping node allows the user to modify a mapping of system of 3D-coordinates. Typically used for modifying texture coordinates.

Mapping can be rotated and clamped if desired.

### Inputs

**Vector**

The input vector (3D-direction in relation to the camera) of some the coordinates’ mapping. The value can be provided by another node or set manually.

### Outputs

**Vector**

The output vector, combined by the node.

### Controls

The controls of the node have been ordered in X, Y, Z order. If you want to use the clamping options, try enabling Min and Max.
**Vector type selector**

Type of vector that the mapping transforms.

**Texture**
- Transform a texture by inverse mapping the texture coordinates.

**Point**
- Transform a point.

**Vector**
- Transform a direction vector.

**Normal**
- Transform a normal vector with unit length.

Mapping Node Transforms controls

**Location**
- Transform position vector.

**Rotation**
- Transform rotation vector.

**Scale**
- Transform scale vector.

Mapping Node Clipping controls

**Min**
- Minimum clipping value.

**Max**
- Maximum clipping value.
**Vector Curves**

The Vector Curves node maps an input vector X, Y, and Z components to a diagonal curve. Use this node to remap a vector value using curve controls.

**Inputs**

**Fac:**
Factor. The degree of node’s influence in node tree. The value can be provided by another node or set manually.

**Vector**
The input vector (3D-direction in relation to the camera). The value can be provided by another node or set manually.

**Outputs**

**Vector**
The output vector, combined by the node.

**See also**

- Read more about using the Curve Widget.
Materials
Materials can be linked to objects and Object’s data in the materials panel, of the Shading/Material context. Here is where you can manage how materials are linked to objects, meshes, etc. and activate a material for editing in the rest of the panels.

Context

At the top of the material menu a list of icons explains the context in which the material is being edited. In the example above, the material Material is linked to the object Cube which is linked to the scene Scene.

By toggling the pin symbol on the left side on and off, Bforartists can be told to display only the selected material or to follow context.

Material slots

With a material linked or created, one or several material slots can be created and further options become available:

Plus sign
Add a new material slot or copy the one selected

Minus sign
Remove selected material slot

Down arrow
Copy and paste the selected material slot

Multiple materials

Meshes can handle having more than one material. Materials can be mapped on a per-face basis, as detailed on the Multiple Materials page. In edit mode, the following tools appear:

Assign
Assign the material in the selected material slot to selected vertices

Select
Select vertices assigned to the selected material slot

Deselect
Deselect vertices assigned to the selected material slot
Material naming and linking

Link material to object or to object’s data

Material’s name field
 click into this field to rename your material

Number of users (number field)
 The number of objects or object’s data that use the material. This material is linked between the various objects, and will update across all of them when edited. Clicking this number will make a ‘single user copy’, duplicating the material, with it linked only to the active object/object’s data.

F (Fake user)
 Gives the material a ‘fake user’, to keep the material data-block saved in the .blend file, even if it has no real users.

Plus sign
 Add a new material.

X sign
 Remove link to this material.

Nodes
 Designates this material to be a material node noodle, and not from the Material/Ramps/Shaders settings.

Data-block links

The Link pop-up menu has two choices, Data and Object. These two menu choices determine whether the material is linked to the object or to the data, (in this case, the mesh). The Data menu item determines that this material will be linked to the mesh’s data-block which is linked to the object’s data-block. The Object menu item determines that the material will be linked to the object’s data block directly.

This has consequences of course. For example, different objects may share the same mesh data-block. Since this data-block defines the shape of the object any change in edit mode will be reflected on all of those objects. Moreover, anything linked to that mesh data-block will be shared by every object that shares that mesh. So, if the material is linked to the mesh, every object will share it.

On the other hand, if the material is linked directly to the object data-block, the objects can have different materials and still share the same mesh.

Short explanation: If connected to the object, you can have several instances of the same obData using different materials. If linked to mesh data, you can’t.

Material type
Material added in edit mode These toggles tell Bforartists where this material fits into the Render Pipeline, and what aspects of the material are to be rendered.

Surface
 Render object as a surface

Wire
 Render the edges of faces as wires (not supported in ray tracing)
**Volume**
Render object as a volume. See Volume Material

**Halo**
Render object as halo particles. See Halo Material

**Material Properties Overview**
The usage of each section of the material properties are detailed in the next section.

**Surface and Wire materials**
Surface material types are the most common materials. They represent objects with a defined surface.

Wire materials simply turn all of an object’s edges into rods, which then become renderable, but uses the same shading options as surface materials.

**Preview**
This is a preview of the current material mapped on to one of several objects.

- Flat Plane
- Sphere
- Cube
- Monkey
- Strands
- Large Sphere with Sky

See Preview

**Diffuse**
Diffuse shading simulates light hitting a surface and bouncing off in a very wide angle. You can set the color of the diffuse shading, and set what model is used for the diffuse calculation.

See Diffuse Shaders

**Specular**
Specularity simulates reflections of light sources, that are often sharp, bright spots. You can set the color of the specular shading, and set what model is used for the specular calculation.

See Specular Shaders

**Shading**

**Emit**
Adds extra illumination, as if the material is glowing.

**Ambient**
Sets the global ambient light the material receives

**Translucency**
Amount of shading on the back side that shows through. Use to simulate thin objects, like leaves or paper.

**Shadeless**
This disables the calculation of any shading, so only color information is visible. This is essentially makes
it a “surface shader”

**Tangent Shading**
Use the material’s tangent vector instead of the normal for shading - for anisotropic shading effects (e.g. soft hair and brushed metal). This shading was introduced in 2.42, see also settings for strand rendering in the menu further down and in the Particle System menu.

**Cubic Interpolation**
Use cubic interpolation for diffuse values, for smoother transitions between light areas and dark areas

**Transparency**
Set options for objects in which light can pass through

See *Transparency*

**Mirror**
Here you can set options for materials that are reflective

See *Mirror*

**Subsurface Scattering**
Subsurface scattering simulates semi translucent objects in which light enters, bounces around, then exits in a different place. Examples are candles, human skin, cheese, etc.

See *Subsurface Scattering*

**Strand**
These settings are used when rendering the material on fur or hair

See *Strands*

**Options**

**Traceable**
Allows material to calculated raytracing, for reflections and refractions.

**Full Oversampling**
Forces material to render full shading and textures for all Anti-Aliasing Samples.

**Sky**
Renders material with no alpha, replacing the background with the sky

**Use Mist**
Uses Mist with this material.

**Invert Z Depth**
Renders materials faces with an inverted Z buffer.

**Z Offset**
If using Invert Z Depth, this is an artificial offset to z values.

**Light Group**
Limit material’s lighting calculation to a specific light group

**Exclusive**
Material uses light group exclusively

**Face Textures**
Replaces object’s base color with color from face assigned image textures.

**Face Textures Alpha**
Replaces object’s base alpha value with alpha from face assigned image textures.
**Vertex Color Paint**  
Replaces object’s base color with vertex colors.

**Vertex Color Light**  
Adds vertex color as additional light.

**Object Color**  
Modulate the result with a per object color.

**Shadow**

**Receive**  
Allows the material to receive shadows cast by other objects

**Receive Transparent**  
Allows material to receive transparent shadows cast by other transparent objects.

**Cast Only**  
Causes objects with the material to only cast a shadow, and not appear in renders.

**Casting Alpha**  
Sets the Alpha of shadow casting. Used for irregular and deep shadow buffering.

**Shadows Only**  
Renders shadows as materials alpha value, making materials transparent, except for shadowed areas.

**Shadow Only Type**

Set the type of shadows used when Shadows Only is enabled

- Shadow and Distance
- Shadow Only
- Shadows and Shading

**Cast Buffer Shadow**  
Allows material to cast shadows from buffer lamps.

**Buffer Bias**  
Factor to multiply shadow buffer by.

**Auto Ray Bias**  
Prevents raytraced shadow errors on surfaces with smooth normals

**Ray Bias**  
Shadow raytracing bias value to prevent terminator artifacts on shadow boundary.

**Cast Approximate**  
Allow material to cast shadows when using Approximate Ambient Occlusion} }

**Volume Material**

Volume materials represent volumes of tiny particles, like clouds or smoke. They are very different from standard materials, but are detailed in the Volume Page.

**Halo Material**

Halo materials renders each of the objects points as glowing dots. This is a useful material for simulating particle effects or lens flares. They are detailed on the Halo Page.

**Special Material Effects**

- Halo Rendering
Halo Rendering

Activating halo rendering

Bforartists provides a set of materials which do not obey the face-shader paradigm and which are applied on a per-vertex rather than on a per-face basis. These are called Halos because you can see them, but they do not have any substance. They are like little clouds of light; although they are not really lights because they do not cast light into the scene like a lamp.

Halos come in very handy when creating certain special effects, when making an object glow, or when creating a viewable light or fog/atmospherics around an actual light.

Options
Halo panels

To enable Halos, press the Halo button in the Material menu’s top panel.

As you will see in the 3D View, the mesh faces are no longer rendered. Instead just the vertex is rendered, since that is where each halo will originate. Halos can be hard to find in a crowded scene, so name it well for easy location in the outliner.

In the properties window, where we normally find the Diffuse, Specular, and Shading panels, we now see panels relative to the Halo characteristics:

**Halo Panel**

**Alpha**
- The transparency

**Color Swatch**
- The color of the halo itself

**Seed**
- If non-zero, randomizes the ring dimension and line location. To use, give any (integer) number to start the random-number generator.

**Size**
- Sets the dimension of the halo

**Hardness**
- Sets the hardness of the halo. Similar to specular hardness

**Effect of Add**

**Add**
- The Add slider determine how much the halo colors are ‘added to’, rather than mixed with, the colors of the objects behind and together with other halos. By increasing Add, the Halo will appear to light up objects that move behind it or through the Halo field.

**Texture**
- Gives halo a texture. By default, textures are applied to objects with Object coordinates and reflects on the halos by affecting their color, as a whole, on the basis of the color of the vertex originating the halo. Enable this feature to have the texture take effect within the halo, and hence to have it with varying colors or transparencies; this will map the whole texture to every halo. This technique proves very useful when you want to create a realistic rain effect using particle systems, or similar.

**Vertex Normal**
- Use the vertex normal to specify the dimension of the halo

**Extreme Alpha**
Boosts alpha

Shaded

Lets halo receive light and shadows from external objects

When shaded is enabled, the Halo will be affected by local light; a lamp will make it brighter and affect its diffuse color and intensity.

Soft

Softens the edges of the halos at intersections with other geometry

In addition, several other special effects are available. To enable some or all of these effects, set the number of points/rings, or set the color of each effect individually:

Rings

Adds circular rings around to the halo.

Lines

Adds lines from the center of the halo.

Star tips

Gives the halo a star shape.

You can not use color ramps. Lines, Rings and an assortment of special effects are available with the relevant toggle buttons, which include Flare, Rings, Lines, Star, Texture, Extreme Alpha, and Shaded. Halo Variations shows the result of applying a halo material to a single vertex mesh.

Halo Variations

The halo size, hardness and alpha can be adjusted with the pertinent sliders. These are very similar to traditional material settings
The Add slider determine how much the halo colors are ‘added to’, rather than mixed with, the colors of the objects behind and together with other halos. By increasing Add, the Halo will appear to light up objects that move behind it or through the Halo field.

To set the number of rings, lines, and star points independently, once they are enabled with the relative Toggle Button, use the Num Buttons Rings:, Lines: and Star:. Rings and lines are randomly placed and oriented, to change their pattern you can change the Seed: Num Button which sets the random numbers generator seed.

Flare Panel
Enabling Flare Renders the halo as a lens flare

Size
Sets the factor by which the flare is larger than the halo

Boost
Give the flare extra strength.

Seed
Specifies an offset in the flare seed table

Subflares
Sets the number of subflares

Subsize
Sets the dimensions of the subflares, dots, and circles

Lens Flares
Our eyes have been trained to believe that an image is real if it shows artifacts that result from the mechanical process of photography. Motion blur, Depth of Field, and lens flares are just three examples of these artifacts. The first two are discussed in the chapter_rendering; the latter can be produced with special halos. A simulated lens flare tells the viewer that the image was created with a camera, which makes the viewer think that it is authentic.

We create lens flares in Bforartists from a mesh object using first the Halo button and then the Flare options in the Shaders Panel of the material settings. Try turning on Rings and Lines, but keep the colors for these settings fairly subtle. Play with the Flares: number and Fl. seed: settings until you arrive at something that is pleasing to the eye. You might need to play with Boost: for a stronger effect (Lens Flare settings).

Note that this tool does not simulate the physics of photons traveling through a glass lens; it’s just a eye candy.
Bforartists’s lens flare looks nice in motion, and disappears when another object occludes the flare mesh.

Lens Flare

Halo Texturing

By default, textures are applied to objects with Object coordinates and reflects on the halos by affecting their color, as a whole, on the basis of the color of the vertex originating the halo. To have the texture take effect within the halo, and hence to have it with varying colors or transparencies press the Texture button; this will map the whole texture to every halo. This technique proves very useful when you want to create a realistic rain effect using particle systems, or similar.

Another Option is Shaded. When shaded is enabled, the Halo will be affect by local light; a lamp will make it brighter and affect its diffuse color and intensity.

Examples

Dotmatrix display

Let’s use a halo material to create a dotmatrix display.

- To begin, add a grid with the dimensions 32x16. Then add a camera and adjust your scene so that you have a nice view of the billboard.
- Use a 2D image program to create some red text on a black background, using a simple and bold font (if you are a lazy lizard [I hope this not offensive, I just like how it sounds!], you can just save the picture
below on your hard drive...). *Dot matrix image texture.* shows an image 512 pixels wide by 64 pixels high, with some black space at both sides.

**Dot matrix image texture.**

- Add a material for the billboard, and set it to the type *Halo*. Set the *HaloSize* to 0.06 and when you render the scene you should see a grid of white spots.
- Add a Texture, then change to the Texture Buttons and make it an image texture. When you load your picture and render again you should see some red tinted dots in the grid.
- Return to the Material Buttons and adjust the *sizeX* parameter to about 0.5 then render again; the text should now be centered on the Billboard.
- To remove the white dots, adjust the material color to a dark red and render. You should now have only red dots, but the billboard is still too dark. To fix this enter EditMode for the board and copy all vertices using the Shift-D shortcut (take care not to move them!). Then adjust the brightness with the *Add* value in the MaterialButtons.

**Dot Matrix display.**

You can now animate the texture to move over the billboard, using the *ofsX* value in the *Texture* panel of the MaterialButtons. (You could use a higher resolution for the grid, but if you do you will have to adjust the size of the halos by shrinking them, or they will overlap. *(Dot Matrix display).*

---

**Note**

**Note about material indices**

Halo materials only work when applied using the first material index. Any material(s) in a subsequent material index will not be rendered.
Volume Rendering

Activation volume rendering

Volume rendering is a method for rendering light as it passes through participating media, within a 3d region. The implementation in Bforartists a physically based model, which represents the various interactions of light in a volume relatively realistically.

Volume rendering

Rendering a volume is different than Solid Render. For volume light enters a 3D region of space (defined as the volume) that may be filled with small particles, such as smoke, mist or clouds. The light bounces around off the various molecules, being scattered or absorbed, until some light passes through the volume and reaches the camera. In order for that volume to be visible, the renderer must figure out how much material the light has passed through and how it has acted and reacted within that volume, the volume object needs to contain a 3D region of space, for example a manifold closed mesh, such as a cube, not just a flat surface like a plane. To get an image, the renderer has to step through that region, and see how much ‘stuff’ is there (density) in order to see how light is absorbed or scattered or whatever. This can be a time consuming process since it has to check a lot of points in space and evaluate the density at each.
Options

Density

Constant density vs textured density

Many things can happen to the light as it passes through the volume, which will influence the final color that arrives at the camera. These represent physical interactions that happen in the real world, and most of these are dependent on the density of the volume, which can either be a constant density throughout, or varied, controlled by a texture. It is by controlling the density that one can get the typical ‘volumetric’ effects such as clouds or thick smoke.

**Density**

   The base density of the material - other density from textures is added on top

**Density Scale**

   A global multiplier to increase or decrease the apparent density. This can be useful for getting consistent results across different scene scales.
Shading

Spot lamp scattering in a constant volume

When light enters a volume from an external source, it doesn’t just pass straight through. Light gets scattered off tiny particles in the volume, and some proportion of that light reaches the camera. This property makes it possible to see light beams as they travel though a volume and are scattered towards the eye.

Shading options

**Scattering**

The amount of light that is scattered out of the volume. The more light that is scattered out of the volume, the less it will penetrate through the rest of the volume. Raising this parameter can have the effect of making the volume seem denser, as the light is scattered out quickly at the ‘surface’ of the volume, leaving the areas internal to the volume darker, as the light doesn’t reach it.

Note in the examples below, the less light that is scattered out of the volume, the more easily it penetrates throughout the volume and to the shadow.
Asymmetry

Isotropic and Anisotropic scattering

The default method for scattering light in a volume is for the light to be deflected evenly in all directions - known as Isotropic scattering. In real life different types of media can scatter light in different angular directions, known as Anisotropic scattering. Back-scattering means that light is scattered more towards the incoming light direction, and forward-scattering means it’s scattered along the same direction as the light is traveling.

Asymmetry

Asymmetry controls the range between back-scattering (-1.0) and forward-scattering (1.0). The default value of 0.0 gives Isotropic scattering (even in all directions).

Transmission

Transmission is a general term for light that is transmitted throughout a volume.

This transmitted light can be the result of various different interactions, for example:

- the left over result of incoming light after it has reflected/scattered out of the volume
- the left over result of light after being absorbed by the volume (and converted to heat)

Here, the transmission color is used to set the end result color that light becomes after it is transmitted through the volume.

Transmission Color

The resultant color of light that is transmitted through the volume.
Note in the examples below, as more light is scattered out of the volume, there is less available to be transmitted through.

| Transmission color: Yellow, Scattering: 0.5 | Transmission color: Yellow, Scattering: 1.0 | Transmission color: Yellow, Scattering: 2.0 | Transmission color: Yellow, Scattering: 5.0 |

**Emission**

Some volumes can emit light where there was none before, via chemical or thermal processes, such as fire. This light is generated from the volume itself and is independent of light coming from external sources.

Currently, this emitted light does not affect other volumes or surfaces (similar to surface material type, ‘Emit’ option).

**Emission Color**

The color of light that is emitted by the volume.

**Emission**

An intensity multiplier for the emitted color, for scaling up and down.

| Emission 0.25, Scattering: 0.5 | Emission 0.25, Scattering: 1.0 | Emission 0.25, Scattering: 2.0 | Emission 0.25, Scattering: 5.0 |

**Reflection**

The ‘reflection’ parameters can be used to tint or scale the light that’s scattered out of the volume. This only affects light that has come from lamps and been scattered out, it doesn’t affect the color of transmitted or emitted light and is.

These settings are not physically correct because they don’t conserve energy - the light scattering out doesn’t affect the remaining light that is transmitted throughout the rest of the volume. For example, physically speaking, if the orange components of the light are scattered out of the volume towards the camera, only the inverse of that (blue) will remain to continue penetrating through the volume, causing the volume to take on a multi-colored appearance, which can be difficult to use. To make it a bit easier to plainly set the color of the volume, you can use the reflection parameters to quickly set an overall tint.

**Reflection Color**
The color of light that is scattered out of the volume.

**Reflection**

An intensity multiplier for the reflection, for scaling up and down.

**Hints**

Ideally try to accomplish as much as you can with the other volume settings and lighting before using the reflection controls. If you stick to what’s physically plausible, the material will act correctly, and be more predictable and usable in a wider range of lighting scenarios. Of course you can always break the rules too!

<table>
<thead>
<tr>
<th>Reflection: Green, Scattering: 0.5</th>
<th>Reflection: Green, Scattering: 1.0</th>
<th>Reflection: Green, Scattering: 2.0</th>
<th>Reflection: Green, Scattering: 5.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reflection: Green, Transmission: Yellow, Scattering: 0.5</td>
<td>Reflection: Green, Transmission: Yellow, Scattering: 1.0</td>
<td>Reflection: Green, Transmission: Yellow, Scattering: 2.0</td>
<td>Reflection: Green, Transmission: Yellow, Scattering: 5.0</td>
</tr>
</tbody>
</table>

**Lighting**

Several shading modes are available, providing a range of options between fast to render and physically accurate.

**Lighting Mode**

- **Shadeless**
  Shadeless is the simplest, useful for thin, wispy mist or steam.

- **Shadowed**
  Shadowed is similar, but with shadows of external objects.
Shaded
Shaded uses a volumetric single-scattering method, for self-shading the volume as light penetrates through.

Multiple Scattering
Allows multiple scatter calculations.

Shaded+Multiple Scattering
Combines Shaded and Multiple Scattering functionality.

Shaded Options:
- **External Shadows**
  Receive shadows from sources outside the volume (temporary).
- **Light Cache**
  Pre-calculate the shading information into a voxel grid, speeds up shading at slightly less accuracy.
- **Resolution**
  Resolution of the voxel grid, low resolutions are faster, high resolutions use more memory.

Multiple Scattering Options:
- **Diffusion**
  Diffusion factor, the strength of the blurring effect.
- **Spread**
  Proportional distance over which the light is diffused.
- **Intensity**
  Multiplier for multiple scattered light energy.

Transparency
The transparency settings are the same as *Solid Render* except you have less settings. For volume rendering you only have:
- Mask
- Z Transparency
- Raytrace

Integration

**Step Calculation Method**
Method of calculating the step through the volume.

- **Randomized**
  Randomized method of calculating the step.
- **Constant**
  Constant method of calculating the step.

**Step Size**
Distance between subsequent volume depth samples. Step Sizes determine how noisy the volume is. Higher values result in lower render times and higher noise.

**Depth Cutoff**
Stop ray marching early if transmission drops below this luminance - higher values give speedups in
dense volumes at the expense of accuracy.

Options

Material volume options

**Traceable**
Allow this material to calculate raytracing.

**Full Oversample**
Force this material to render full shading/textures for all anti-aliasing samples.

**Use Mist**
Use mist with this material (in world settings).

**Light Group**
Limit lighting of this material to lamps in this group.

**Exclusive**
Material uses this group exclusively. Lamps are excluded from other scene lighting.

**Smoke and Fire**

**Create the Material**
The material must be a volumetric material with a Density of 0, and a high Density Scale.
The Material Settings

Smoke requires a complex material to render correctly. Select the big cube and go to the material tab. There change the material to ‘Volume’ and set the density to 0. If you set the density to values bigger than 0 the domain cube will be filled with the volume material. The other settings will affect the smoke, though. We’ll cover those later.

Add the Texture

In addition, Smoke requires its own texture, you can use a volumetric texture known as Voxel Data. You must remember to set the domain object and change the influence.
The texture settings.

Go to the texture tab and change the type to *Voxel Data*. Under the Voxel Data-Settings set the domain object to our domain cube (it should be listed just as ‘Cube’ since we are using Bforartists’s default cube. Under Influence check ‘Density’ and leave it at 1.000 (Emission should be automatically checked, too). Now you should be able to render single frames. You can choose to color your smoke as well, by turning *Emission Color* back on.

![Finished Result](image)

---

**Tip**

To see the smoke more clearly

Under the world tab, chose a very dark color for the horizon.

---

**Extending the Smoke Simulator: Fire!**

You can also turn your smoke into fire with another texture! To make fire, turn up the Emission Value in the Materials panel.
The Fire material.

Then, add another texture (Keep the old texture or the smoke won’t show). Give it a fiery color ramp- which colors based on the alpha, and change the influence to emission and emission color. Change the blend to Multiply.
The fire texture settings.

The fire render.

### Wire Render

Wire Render

The Wire Render option in the Materials section provides a way of showing a rendered image of the edges in an object. Each edge is rendered as a single-pixel image of the edges which make up the mesh. The colors, alpha and other relevant properties of the lines are selected with the same control panels as provided by the Surface rendered image.
Wire Render