

## 10.3.4.1 Render - Cycles Render Engine - Nodes - Introduction

Nodes.....	1
Introduction.....	3
Shaders.....	3
Textures.....	3
More.....	3
Open Shading Language.....	4

### Nodes

- Introduction
  - Shaders
  - Textures
  - More
  - Open Shading Language
- Input Nodes
  - Camera Data
  - Value
  - RGB
  - Attribute
  - Wireframe
  - Geometry
  - Light Path
  - Object Info
  - Fresnel
  - Layer Weight
  - Texture Coordinate
  - UV Map
  - Particle Info
  - Hair Info
  - Tangent
- Output Nodes
  - Material Output
  - Lamp Output
  - World Output
- Shader Nodes
  - Diffuse
  - Translucent
  - Glossy
  - Anisotropic
  - Toon
  - Transparent

- Glass
- Refraction
- Velvet
- Subsurface Scattering
- Emission
- Background
- Holdout
- Ambient Occlusion
- Mix and Add
- Texture Nodes
  - Image Texture
  - Environment Texture
  - Sky Texture
  - Noise Texture
  - Wave Texture
  - Voronoi Texture
  - Musgrave Texture
  - Gradient Texture
  - Magic Texture
  - Checker Texture
  - Brick Texture
  - Point Density
- More Nodes
  - Value
  - RGB
  - Geometry
  - Wireframe
  - Wavelength
  - Blackbody
  - Texture Coordinates
  - Bump
  - Vector Transform
  - Tangent
  - Normal Map
  - Object Info
  - Particle Info
  - Hair Info
  - Attribute
  - Mapping
  - Layer Weight
  - Fresnel
  - Light Path
  - Light Falloff
  - Nodes shared with the Compositor

## Introduction

Materials, lights and backgrounds are all defined using a network of shading nodes. These nodes output values, vectors, colors and shaders.

### *Shaders*

An important concept to understand when building node setups is that of the **shader socket**. The output of all surface and volume shaders is a shader, describing lighting interaction at the surface or of the volume, rather than the color of the surface.

There are a few types of shaders available as nodes:

- BSDF shader describing light reflection, refraction and absorption at an object surface.
- **Emission** shader describing light emission at an object surface or in a volume.
- **Volume** shader describing light scattering inside a volume.
- **Background** shader describing light emission from the environment.

Each shader node has a color input, and outputs a shader. These can then be mixed and added together using Mix and Add Shader nodes. No other operations are permitted. The resulting output can then be used by the render engine to compute all light interactions, for direct lighting or global illumination.

### *Textures*

Each texture type in Cycles corresponds to a node, with a texture coordinate and various parameters as input, and a color or value as output. No texture data-blocks are needed; instead node groups can be used for reusing texture setups.

For UV mapping and texture painting in the viewport, the Image texture node must be used. When setting such a node as active, it will be drawn in Textured draw mode, and can be painted on in texture paint mode.

The default texture coordinates for all nodes are Generated coordinates, with the exception of Image textures that use UV coordinates by default. Each node includes some options to modify the texture mapping and resulting color, and these can be edited in the texture properties.

### *More*

Nodes for geometric data, texture coordinates, layering shaders and non-physically based tricks.

### *Open Shading Language*

Custom nodes can be written using the Open Shading Language.