Render Smoke Color

This page provides information on the Smoke color rollout the Rendering Volumetric Options dialog.

Overview

This rollout controls the Diffuse color of the Volumetric Shader. The Smoke Color can be a uniform color, a mixture of colors dragged by the simulation (RGB), a function of any physical channel, or sampled from a texture. It is also affected by lights and shadows cast by objects in the scene.

In the volume shader, there are two types of content — **diffuse** and **emissive**. For simplicity's sake, we call the diffuse content **Smoke** and the emissive content **Fire**.

Diffuse (Smoke) needs external light in order to become visible, and also casts shadows. On the other hand, emissive (Fire) is visible even without being lit by lights. It also ignores their illumination, and does not cast shadows on itself.

Phoenix has different tools that help you to control how the emissive fire illuminates the smoke and the scene - see the Create Fire Lights section in the Render Fire rollout for more details.

Fire and smoke also have their own Color and Alpha. Alpha is a synonym for opacity, and transparency is the opposite of opacity. The **Smoke Color**, **Smoke Opacity**, as well as the **Fire's Color** and **Opacity**, can be mapped to a physical **Grid Channel** from the simulation, using the Based on parameter.
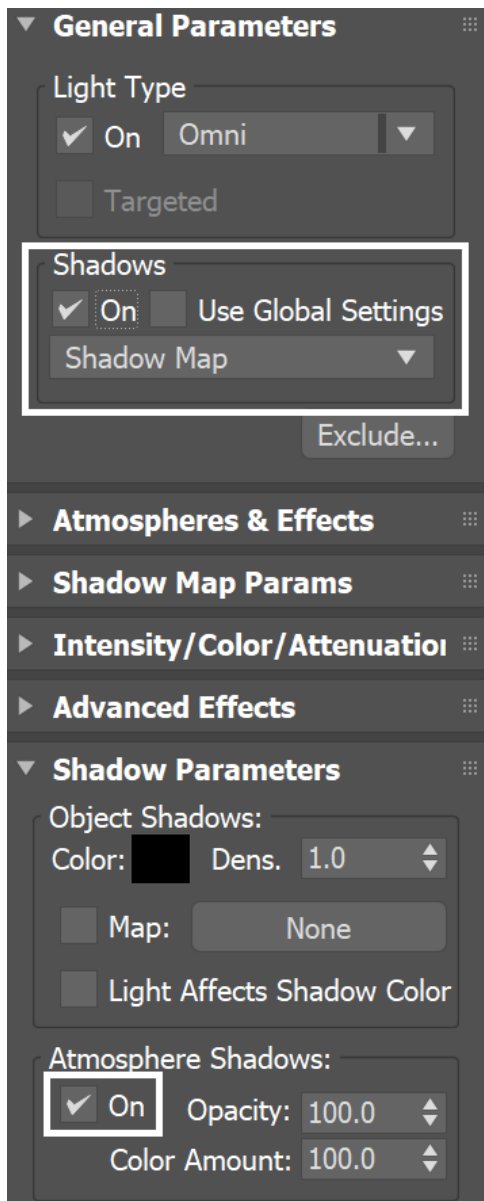
The **Based on** parameter specifies the source channel that will be rendered, and is set independently for the Fire, Smoke Color, and Smoke Opacity respectively. This makes shading very flexible, so that you can even render a simulation's fire as smoke, and smoke as fire, depending on the channel you set the Based on parameter to for each.

Fire, Smoke Color, and Smoke Opacity, can also be **mapped to a texture**, or to a Grid Channel multiplied by a texture, to achieve a wide variety of different results.

Textures have infinite resolution, and so they can increase the detail above the resolution of the grid, meaning that even with a low resolution simulation, you can get a detailed result when using a texture.

By default, textures are static in space. However, if you export the **Grid Texture UVW Channel** and use the TexUVW feature, textures can move together with the fluid to create even more possibilities, such as melting textured icing on a cake.

The diffuse color needs external light to become visible. When using a 3ds Max omni light, it should have Shadows and Atmosphere Shadows enabled, as well as Ray Traced Shadows (if rendering with the default Scanline renderer) or VRayShadow (if rendering with V-Ray). V-Ray lights will work with their default settings.

General Parameters

Light Type

☑ On   Omni ▾

☐ Targeted

Shadows

☑ On   ☐ Use Global Settings

Shadow Map ▾

Exclude...

▸ **Atmospheres & Effects**

▸ **Shadow Map Params**

▸ **Intensity/Color/Attenuatio**

▸ **Advanced Effects**

▾ **Shadow Parameters**

Object Shadows:

Color: ⬛   Dens. 1.0 ⬍

☐ Map: None

☐ Light Affects Shadow Color

Atmosphere Shadows:

☑ On   Opacity: 100.0 ⬍

Color Amount: 100.0 ⬍

UI Path: ||Modify panel|| > Rendering Rollout > Volumetric Options... button > Volumetric Render settings dialog > Smoke color rollout

▾ **Rendering**

Render Presets...

Mode   Volumetric ▾

Volumetric Options...

Step %   90 ⬍
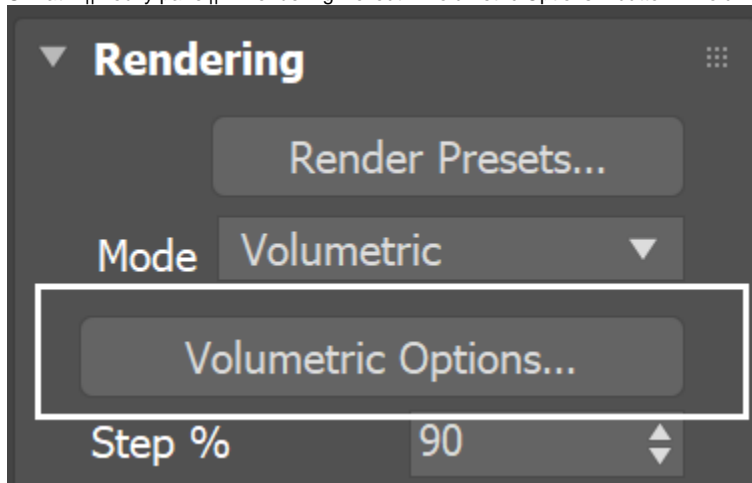
Parameters

When there is both **Smoke** and **Fire** (diffuse and emissive) in the same voxel, you can use the **Fire Opacity Mode** option in the **Render Fire** rollout, to determine the resulting color and alpha.

The Fire can simply use the opacity that is set for the smoke in the Smoke Opacity rollout, or you can separate their opacities for additional control, using either the **Fully Visible** or **Use Own Opacity** modes.

Using the **Based on** parameter below, you can achieve a wide variety of results when shading **Fire** & **Smoke**.

For example, the Fire (emissive content) is by default based on the **Temperature Grid Channel**. If instead it were based on **Smoke Grid Channel** data, then voxels containing Smoke values would be shaded as emissive (Fire). In other words, the Smoke would be rendered as Fire.

On the other hand, if the **Smoke Color** (diffuse) & Smoke Opacity were based on Temperature, then voxels with Temperature values would be shaded as diffuse volumetrics (Smoke Color & Opacity). In other words, the Fire would be rendered as Smoke.

Based on | *darg* – Specifies the source channel that will be used to determine the smoke color. By default, a uniform simple color set by the **Constant Color** option is used. This could also be a texture map, or read from the cache files - in that case the corresponding grid channel must be enabled from the Output rollout before the simulation is run.

- **Disabled** - the smoke color will not be rendered.
- **Temperature** - the smoke color will be based on the values of the Temperature cache channel and can be remapped using the color gradient.
- **Smoke** - the smoke color will be based on the values of the Smoke cache channel and can be remapped using the color gradient.
- **Speed** - the smoke color will be based on the values of the Speed cache channel and can be remapped using the color gradient. See the Smoke Color by Speed example below.
- **Texture** - the smoke color will be based on the values of a texture map plugged into the **Texture** slot.
- **Constant Color** - the smoke color will be the same for the entire volume and will be taken from the **Constant Color** box.
- **RGB** - the smoke will be based on the values of the RGB channel.
- **Fuel** - the smoke will be based on the values of the Fuel channel and can be remapped using the color gradient.

? – Opens the help documents.

Texture | *dtxt* – If **Based on** is set to **Texture**, this slot specifies which texture to use. For more information on texture mapping in Phoenix, please check the Texture mapping, moving textures with fire/smoke/liquid, and TexUVW page.

Modulate | *mod_d* – When **Based on** is not set to **Texture**, the color is multiplied by the texture in the **Texture** slot.

Reset to Defaults – Resets the settings to their default values.

Constant color | *simple_color* – If **Based on** is set to **Constant color**, this color is used to uniformly color the smoke.

Scattering | *noscatter* – Controls how the light rays are scattered inside the volume.

Ray-traced (GI only) – Enables physically accurate scattering of light rays. This mode produces the most realistic results but it's the slowest to render. It requires V-Ray with enabled Global Illumination, otherwise the rendered result would be the same as if the option is **Disabled.** The **Diffuse Multiplier** does not affect the rendering in this mode. Use this mode when you want to render bright clouds.

Disabled – Disables scattering. The **Diffuse Multiplier** value can be used to correct the brightness because without light scattering the volume would generally render darker.

Approximate – Uses an approximate formula which is faster than **Ray-Traced** scattering and produces good-looking results. Brighter areas of the volume would transfer light farther than dark areas. This option is not supported by the Volumetric Geometry render mode. **Approximate+Shadows** – Same as **Approximate**, but also affects the strength of shadows over the scene geometry. This option is not supported by the Volumetric Geometry render mode.

Phase Function | *phase_function* – Controls the direction in which the light will scatter inside the volume. Negative values correspond to backward scattering, which mimics a volume made up of solid particles and will produce denser and more detailed looks. Negative values are more suitable for smoke or dust effects. Positive values correspond to forward scattering, which mimics a volume made up of water droplets where light will scatter more. Positive values are suitable for highly scattering volumes such as clouds. The default value of 0 will scatter the light in all directions and create an even and diffuse look. *For more information, see the Phase Function example below.*

Note that values very close to 1.0 or -1.0 will produce very directional scattering that will be invisible from most angles, so such values are not recommended. When rendering with a Phase Function different than 0 and **Volume Light Cache** is enabled the rendered result will be brighter. The **Phase function** is ignored when the scattering is set to **Approximate** or **Approximate+Shadows.**

Optimize Anisotropic Scattering | *anisotropic_scattering_op timization* **– [V-Ray specific] When enabled it can greatly reduce the render times for highly scattering volumes such as clouds. Due to certain assumptions made for the optimizations, the optimized render result might produce different results compared to the exact brute-force anisotropic scattering. The option is ignored when the scattering is set to Approximate or Approximate+Shadows .**

Volume Light Cache | *lightcache* – Enables light caching for each voxel of the grid, which speeds up bucket rendering considerably when the voxel size of the grid is significantly larger than the rendering pixel. If the grid resolution is large enough compared to the rendering resolution, disabling this option will speed up rendering.

- This option refers to the internal **Phoenix Light Cache**, which is not related to the V-Ray Light Cache.
- This option is ignored when using V-Ray GPU or Corona. It works with V-Ray CPU and Defscanline.
- When using **V-Ray Progressive Rendering**, the Volume Light Cache option might **slow down** rendering startup or the overall render speed.
- This option might produce **artifacts** when the smoke is very dense.
- This option might produce **artifacts** when objects cast shadows through the smoke.
- Rendering **multiple copies or instances** of the Simulator node with this option turned on might lead to slow-downs.
- This option will consume additional memory, so beware of high RAM usage when rendering very large cache files or a large number of copied or instanced volumes.
- When rendering in Volumetric Geometry mode, for any of these Render Elements: [Shadows], [Raw Shadows], [Global Illumination], [Raw Global Illumination], the **Volume Light Cache** will require additional memory to produce the correct result.
- Some V-Ray render elements might not work when Volume Light Cache is enabled. Please consult this table.

Light Cache Speedup | *subred* – [V-Ray specific] [Valid only when **Volume Light Cache** is enabled] The higher this option is set, the faster the rendering will be, but the lower the quality of the **Volume Light Cache**. You can increase this parameter and gain render speed as long as you don't start getting dark cubic grid artifacts on the smoke.

Diffuse Multiplier | *difmul* – A general multiplier for the diffuse color. It is ignored when the **Scattering** mode is set to **Ray-traced (GI only)**.

Own Light Scatter Mult | *transfown* – [Valid only when **Scattering** is set to **Approximate** or **Approximate+Shadows**] This multiplier determines how far the light travels through the volumetric, based on the distance from the emissive light itself.

External Light Scatter Mult | *transfext* – [Valid only when **Scattering** is set to **Approximate** or **Approximate+Shadows**] This multiplier determines how far the light from all external light sources travels through the volumetric. Both Scatter Boost parameters don't alter the visual density of the smoke to the camera, and only apply to the illumination.

Shadow Strength | *shad_strength* – A multiplier for the opacity of the shadow that the volumetric casts on other objects.

Color gradient | *dcolorx_t, dcolory_t, dcolorx_s, dcolory_s, dcolorx_v, dcolory_v, dcolorx_f, dcolory_f* – When the **Based on** option is set to a grid channel, this color gradient remaps grid simulation data to render smoke color. The sim data channel is laid out horizontally along the X axis in the color gradient.

The channel data range is highlighted in the curve with a light-blue range. You can find out more about Phoenix Grid Channel Ranges here.

You can use the following controls in the color gradient:

Double click – Creates a new point or changes an existing one.
Left button drag over a point – Moves the point. If several points were selected beforehand, they will move the same amount.
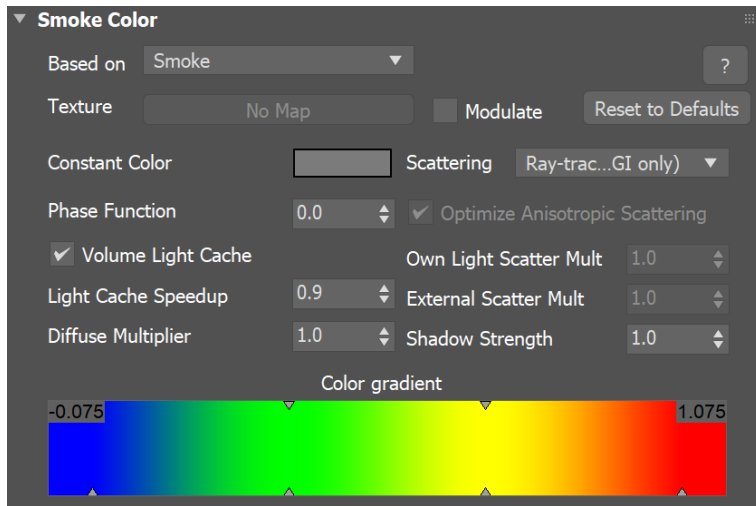**Left button drag over several points** – Selects several points.
Middle button drag over the background – Drags the visible area.
Mouse wheel – Zooms in/out.
Right click – Displays a drop-down menu where you can add a point, edit or delete a selected point, and fit the entire gradient into the view. If multiple points are selected, they can be edited simultaneously.
Backspace – Deletes the selected points.

Smoke Color

Based on	Smoke ▼	?
Texture	No Map	☐ Modulate	Reset to Defaults

Constant Color	☐	Scattering	Ray-trac...GI only) ▼
Phase Function	0.0 ↕	✔ Optimize Anisotropic Scattering
✔ Volume Light Cache		Own Light Scatter Mult	1.0 ↕
Light Cache Speedup	0.9 ↕	External Scatter Mult	1.0 ↕
Diffuse Multiplier	1.0 ↕	Shadow Strength	1.0 ↕

Color gradient

-0.075 .... 1.075

Example: Phase Function with front lighting

Phase Function = -0.5

Phase Function = 0
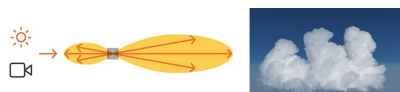
Phase Function = 0.5

Move the slider to see the example renders

Phase Function = -0.5 (Backward Scattering)
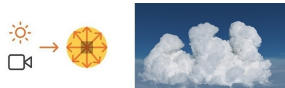
Phase Function = 0 (Isotropic Scattering)

Phase Function = 0.5 (Forward Scattering)

The orange arrow represents a ray of light going through the volume and the possible scattering directions for the ray.

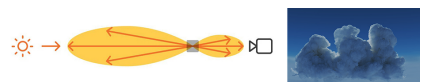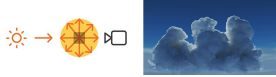## Example: Phase Function with back lighting

Phase Function = -0.5

Phase Function = 0

Phase Function = 0.5

Move the slider to see the example renders

Phase Function = -0.5 (Backward Scattering)

Phase Function = 0 (Isotropic Scattering)

Phase Function = 0.5 (Forward Scattering)

The orange arrow represents a ray of light going through the volume and the possible scattering directions for the ray.

Example: Smoke Color Based on Speed

The Smoke Color can be based on any channel present in the cache files. The Speed channel can be used to drive the color of the smoke. It is automatically generated by Phoenix based on the Velocity Grid Channel. In this example, the slowly moving smoke is blue while the fast moving smoke is red to yellow.