Particle Texture

This page provides information on the Phoenix Particle Texture (PhoenixFDParticleTex).

Overview

The **Particle Texture** is a **3D texture** that reads particles and colors the positions of each particle in a particle system. It can be created from the **Material Editor** in **3ds Max**.

The Particle Texture can also shade a **Particle's color** based on **Particle Channels**, such as its **Age** or **Speed**. As a result, you can change the Particle's color over time, based on the behavior of those Particle Channels.

Note that the **Particle Texture** is generated so that each particle is white by default, with the area around each particle having soft edges, that fade to black with further distance from each particle's position. This area can also use color from another texture, or even use different pieces of a texture for each particle.

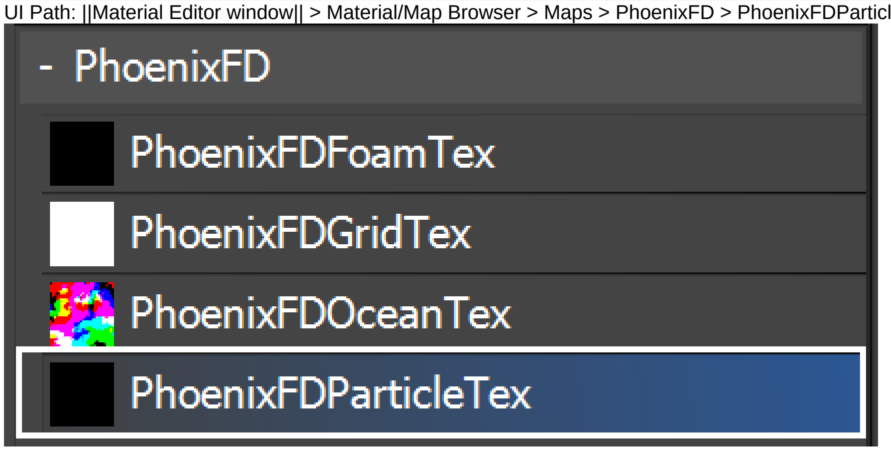
If a constant color is used, it can either be the same for all particle areas, or it could come from a certain Particle Channel - Age, Velocity, RGB, etc.

You can also plug the Particle Texture into a Phoenix Particle Shader's **color map slot**, which would enable you to shade particles as different colors, based on their Particle Channels.

The Particle Texture can also be plugged into a material, and used to shade the surfaces of geometry objects. If you are simulating liquid that creates a **WetMap** particle system over geometries, you can use the Particle Texture as a grayscale mask to blend between two materials, for example, a wet material and a dry surface material. This way, the geometry that is covered by the WetMap particles will appear wet, and the rest of the geometry will appear dry.

You can also turn particles into a **3D mesh**, by using the Particle Texture as a Surface Channel texture in the Rendering rollout. This gives you additional flexibility for shading particles, to achieve more advanced effects. For example, you could use the Particle Texture to mesh specific particles based on their RGB color, which would make it possible to create effects like mixing together a solid and transparent liquid, using two meshes.

UI Path: ||Material Editor window|| > Material/Map Browser > Maps > PhoenixFD > PhoenixFDParticleTex



Parameters

Source Particle System | *system* – Specifies the particle system that will be used. It can be any standard 3ds Max particle system, and you can pick the Phoenix Simulator or the Phoenix PRT Reader.

Render-time Only | *render_time* – The particle texture will be evaluated only during rendering. This can help speed up simulations that use this texture. Otherwise, the texture will be evaluated during simulation and could cause loading and blending of frames, which is slow and in turn can produce warnings about missing particle ID or Age channels.

Particle Area Radius | *radius* – Specifies the size of the particle area. This option is critical to the performance. The larger the **Particle Area Radius**, the longer it would take for the Particle Texture to render. In case you are using the Particle Texture as a **Color Map** in a Phoenix Particle Shader, always start with a very small **Particle Area Radius** - this way the particles would render with black edges. Keep increasing the **Particle Area Radius** until the black edges are no longer visible, and this way you would get the best render speed for your setup.

PA | *radius_time* – You could use this option only when you have animated the **Particle Area Radius**. When **PA** is off, the animation of **Particle Area Radius** affects all particles simultaneously. When you enable **PA**, the animation of **Particle Area Radius** will be read using the Age of each particle. This way the animation curve will start from the birth time of each particle, instead of the first timeline frame. This way at one point in time different particles could use different moments of the animation, if they were born at different times. Note: The particle ages must be exported from the Output rollout of the Phoenix Simulator. If the particles are imported from another plugin or application, e.g. through the PRT Reader object, you must make sure to export the particles' ages from the other software.

Areas Blending Method | *blending* – Specifies the method for blending particles that have overlapping areas.

> **Equal** – All particles add the same contribution to the end result.
> **Proportional** – The contribution of each particle is determined by the distance to the particle.
> **Biggest** – The particle with the biggest contribution determines the end result.
> **Voronoi** – The nearest particle is used.

Blend Alpha | *blend_alpha* – When enabled, the alpha of the sampled color texture is blended. Otherwise, it is set to 1.

Color Intensity | *amplitude*, *amplitude_time* – The color strength in the particle area.

PA | *amplitude_time* – You could use this option only when you have animated the **Color Intensity**. When **PA** is off, the animation of **Color Intensity** affects all particles simultaneously. When you enable **PA**, the animation of **Color Intensity** will be read using the Age of each particle. This way the animation curve will start from the birth time of each particle, instead of the first timeline frame. This way at one point in time different particles could use different moments of the animation, if they were born at different times. Note: The particle ages must be exported from the Output rollout of the Phoenix Simulator. If the particles are imported from another plugin or application, e.g. through the PRT Reader object, you must make sure to export the particles' ages from the other software.

Mult. by Size | *mulbysz* – When enabled, the **Color Intensity** value is multiplied by the particle's Size channel. This option can be used when blending wet and dry materials when using WetMap particles (when **Wetting** is enabled in the Dynamics rollout) because the WetMap particles shrink with time.

Color Map | *map* – Specifies a texture map connected to the particles. The contribution of each particle is the color from the texture in the **Particle Area Radius**, fading to black with soft edges or blending with overlapping particle areas using the **Areas Blending Method**. If no texture is specified, the particle area is white.

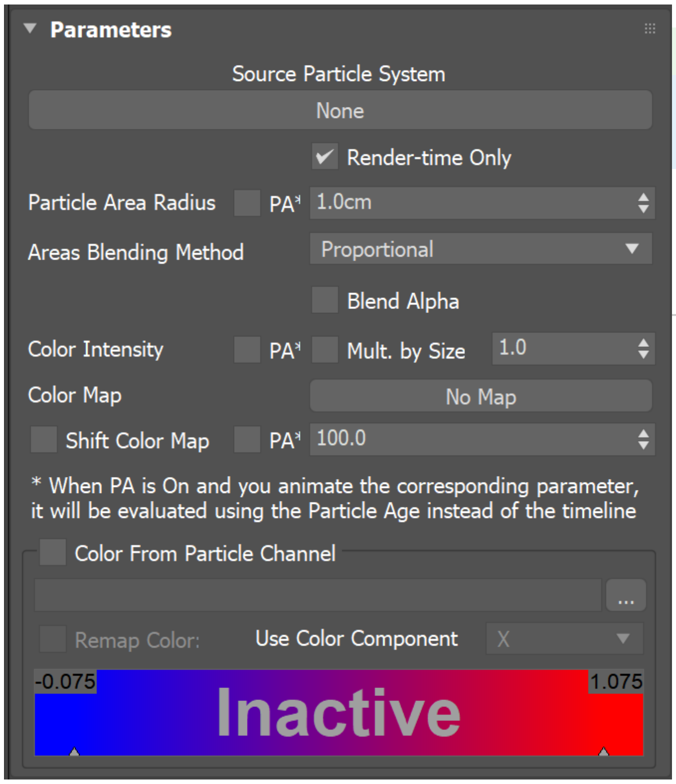
Shift Color Map | *shift*, *shift_mode* , *shift_time* – Enabling this option changes the calculation algorithm. Instead of contributing to the resulting colors, the particles instead produce shifted coordinates for the texture specified in the **Color Map** slot. When the simulation starts, each particle relates to a pixel on the texture, and also to a number of pixels around it using the radius specified by the **Shift Color Map** number field. As the particles move, they "pull along" the related part of the texture. If **PA** (particle age) option is enabled, the particle Age channel is used to shift the texture rather than the particle Position. This mode has no effect if no texture is specified in the **Color Map** field. *See the Shift mode example below for more information.*

Color From Particle Channel | *colfrompartchan* – When enabled, the particle areas are colored using a specified particle channel of the connected particle system, such as Age, Size, Velocity, Position, etc. Note that if the particle channel is a vector channel, such as RGB, Position and Velocity, it can directly be shown as color in each particle area, but if the channel is a scalar such as the Size, Age or ID of the particles, it will produce grayscale color, unless you use the **Remap Color** option.

Remap Color | *partcolremap* – This option remaps a different color for scalar particle channels such as the Size, Age or ID of the particles, or a component from vector particle channels such as RGB, Position, or Velocity. For example, you can remap the particle Age which ranges from 0 (black) for newborn particles to several tens or hundreds (seconds), to a gradient from blue color for newborn particles to red color for old ones - you can do this by placing the blue color at position zero in the color gradient, and placing the red color at a value that is the maximum particle Age (in seconds). You need to know the value ranges of the particle channels in order to remap them correctly - check the Particle Channel Ranges page for more information. When remapping vector channels, you have to set the **Use Color Component** option to one of the vector components of the channel.

Use Color Component | *remapcolcomp* – Specifies the vector component of channels such as RGB, Position, or Velocity to be used when remapping.

X – Uses the X axis of positions or velocities, or the Red channel from RGB color.
Y – Uses the Y axis of positions or velocities, or the Green channel from RGB color.
Z – Uses the Z axis of positions or velocities, or the Blue channel from RGB color.
Length – Uses the length of the particle channel. Can be used with the **Velocity** channel when you need to remap the Speed of the particles to colors. If X, Y or Z are used with Velocity, this will remap the Speed respectively in the X, Y or Z directions.



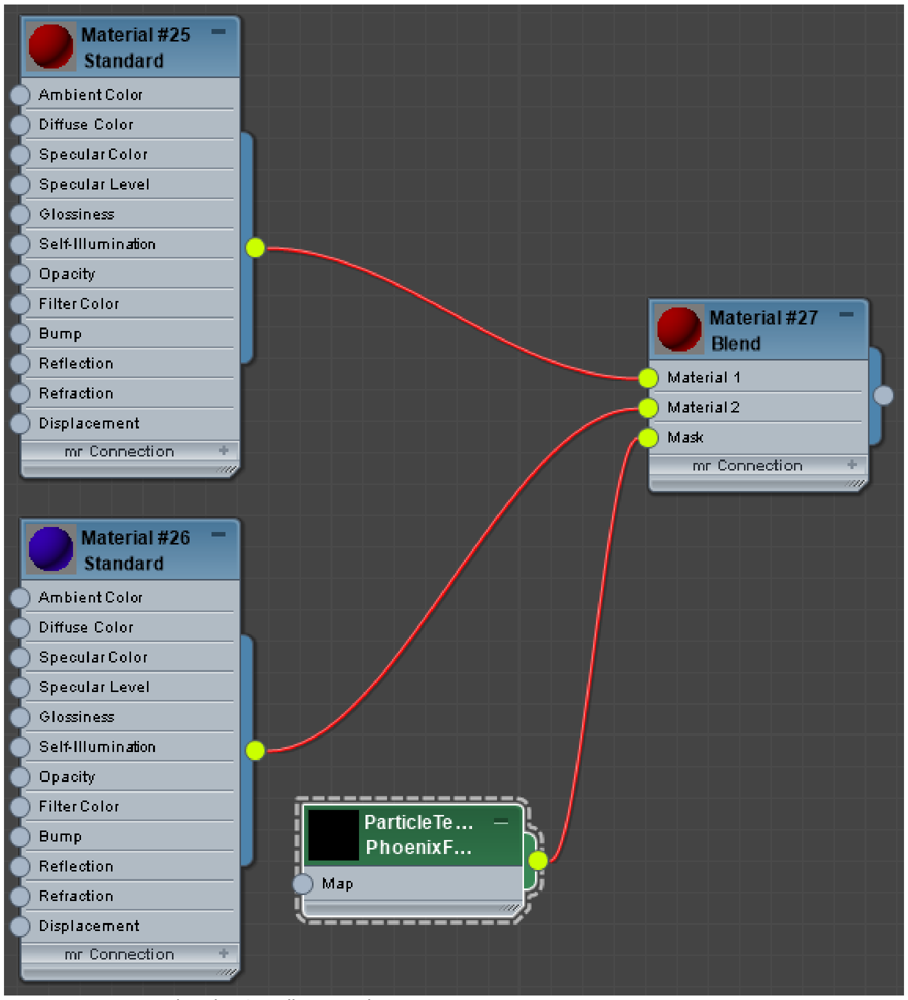
The **Remap Color** gradient stores its parameters with the *partcolor_p*, *partcolor_c* and *partcolor_i* script names.

You can render out the **Color From Particle Channel** as a separate render element by using the Phoenix Particle Texture as an input for the VRayExtraTex render element and turning on **Render as Geometry** in the Particle Shader's options.

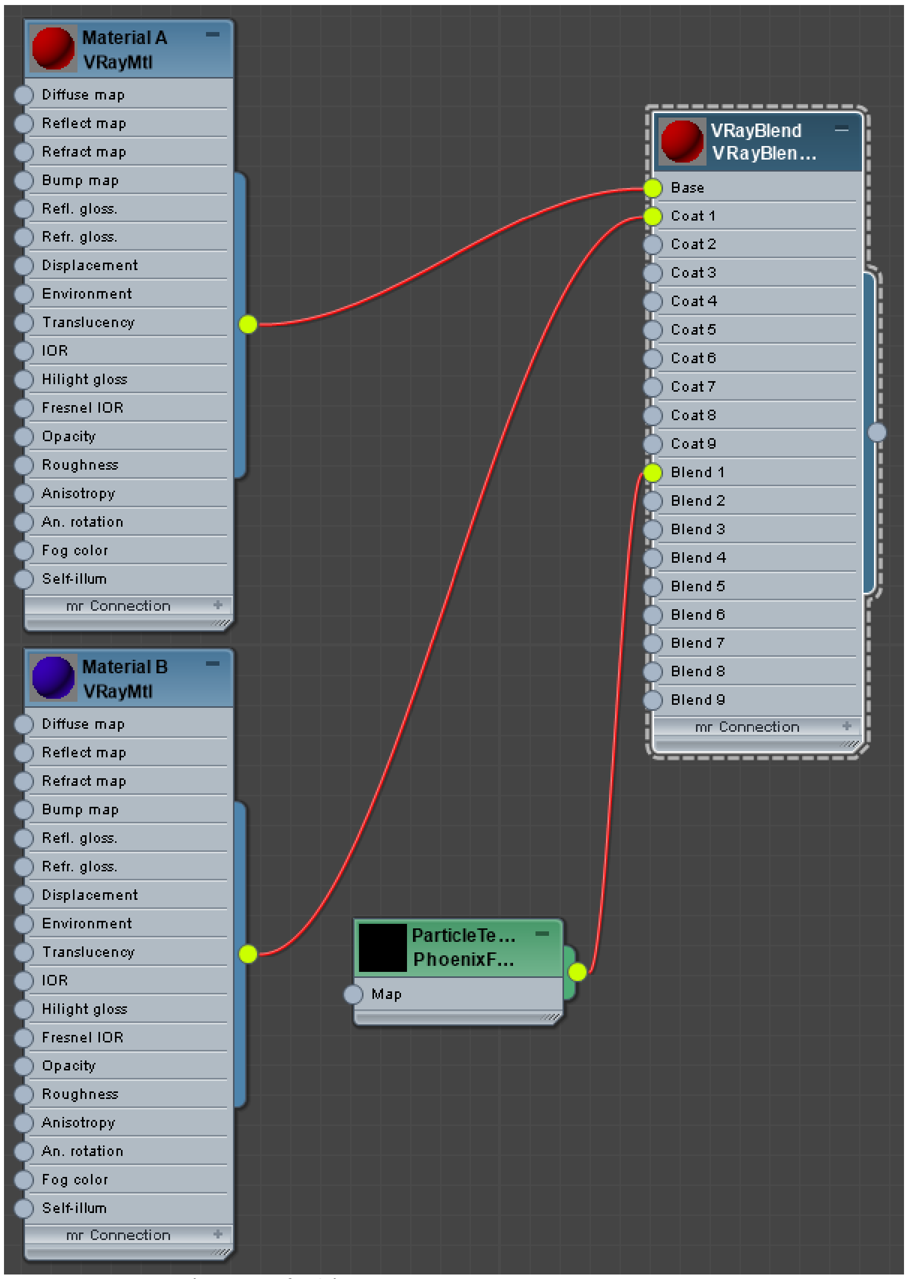
Setup for shading dry and wet materials

The Particle Texture can be used to blend between two materials. The examples below illustrate setups when using the default Scanline Renderer or V-Ray for 3ds Max. When using the Scaline Renderer, the Particle Texture is set as the Blend material's **Mask**. When using V-Ray, two different V-Ray Materials are set as the **Base Material** and **Coat Material** of a V-Ray Blend Material. The Particle Texture node is set as the **Blend** map.

This setup can be used when rendering the **Wetmap** particles of the Phoenix simulation. You can use a Particle Texture to read the Wetmap particle system and then blend between two materials - a dry and a wet one.



Using the Scanline Renderer



Using V-Ray for 3ds Max