11

Creating Shaders

Why Create a Shader?

Shaders are a powerful way of bringing your 3D objects to life. You can use Ray Dream Studio's extensive libraries of shaders to add life to your objects, but since there is no limit to the types of objects you can create, you may need to build a unique shader to suit each object you create.

Ray Dream Studio's shader structure makes it possible to create your own custom shaders. Much like mixing your own colors for painting on an artist's palette, you'll use the Shader Tree to create an infinite number of different textures, colors and surfaces.

This chapters explains the Shader Tree and guides you through the process of creating your own shaders.

Shader Structure and Content

This section and the two that follow—"Shader Components" on page 201 and "Shader Channels" on page 213—explain the central concepts of defining shaders. After reading these three sections, you should understand how to define a shader by building a shader tree from Ray Dream Studio's various shader components.

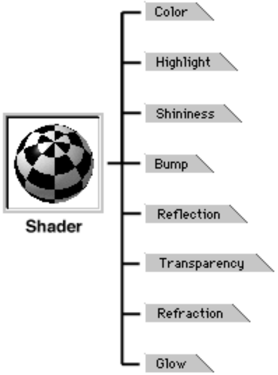
When you're ready to start designing your own shaders, read the step-by-step instructions for using the Shader Editor in "Building and Editing Shaders" on page 217.

What is a Shader Tree

A *shader* is a set of surface properties that you can assign to an object or to a paint shape on the surface of an object. Ray Dream Studio features a modular structure for defining shaders: the *shader tree*. The shader tree's modular nature allows for great flexibility in shader content.

Understanding Shader Channels

A shader tree contains all of the settings for a single shader. The shader itself is at the root of the tree. Immediately beneath the root, the tree has eight branches, one for each of the shader channels: **Color**, **Highlight**, **Shininess**, **Bump**, **Reflection**, **Transparency**, **Refraction**, and **Glow**.



The shader tree contains the eight surface properties, called shader channels, that define an object's surface in Ray Dream Studio.

To specify shader settings, you add components to the shader tree in one or more of the channels. The components beneath a particular channel represent that channel's settings.

Depending on the components you use, the settings in each channel may be simple or complex. The **Color** channel might specify either a plain color or a multi-color pattern. Likewise, the **Reflection** channel might specify uniform or varying levels of reflectiveness across the surface of an object.

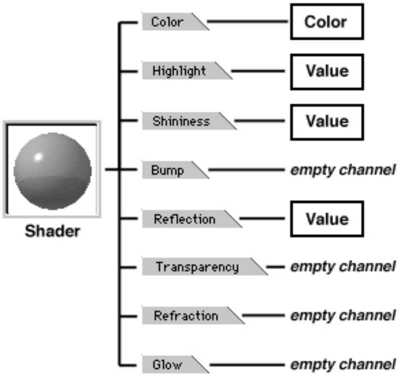
A shader need not contain settings in all eight channels. If you don't want to define a particular shading attribute, you can simply leave that channel empty.

Understanding Shader Components

There are three types of components: basic components, operators, and functions.

Basic Components

Basic components are the fundamental building blocks of shaders. Colors, values, and texture maps are examples of basic components. The diagram below shows a shader tree built entirely with basic components. Notice the structural simplicity of the tree—no branching occurs below the level of the channels.



You can adjust the contents of a shader's channels by placing different components in each channel.

For more information on specific basic components, refer to "Basic Components" on page 201.

Operator Components

To create more complex and varied shading effects, you can use operators to combine the effects of two basic components within a shader channel. The arithmetic operators (Add, Subtract, and Multiply) combine components mathematically. An additional operator, the versatile **Mix** operator, uses functions to combine components in a variety of ways.

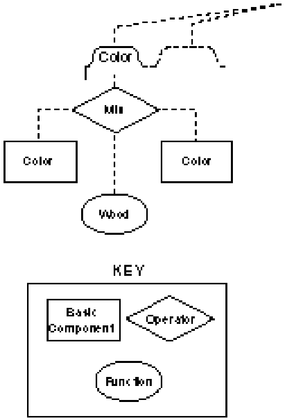
For more information on specific operators, refer to "Operators" on page 206.

Function Components

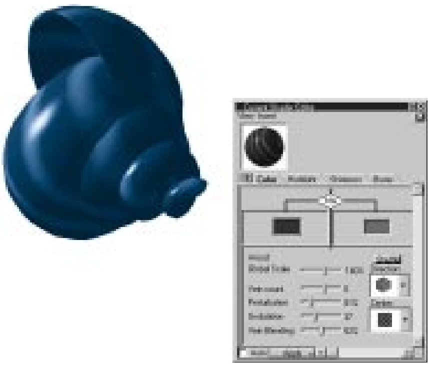
By choosing which function to use, you control how the Mix operator combines a pair of components. Ray Dream Studio ships with several functions, including **Cellular**, **Checkers**, **Wires**, **Wood**, **Spots**, **Gradient**, **Psychedelic** and **Marble**. You can also define your own function with a formula.

For more information on specific functions, refer to "Functions" on page 207.

The diagram below demonstrates the use of an operator. In this particular example, the Mix operator is used in conjunction with the Wood function to mix two colors in a wood grain pattern. Notice that placing an operator in the shader tree causes the tree to branch.



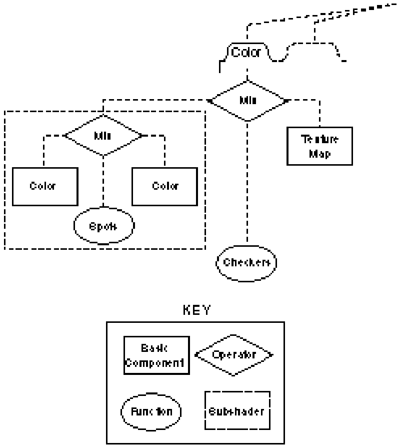
Placing the Mix operator in the shader tree creates three branches. One for the mixing function and one for each of the components you're mixing.



The wood grain applied to this object was created by using the Mix operator with two colors and a wood grain function.

When a component such as the Mix operator has branches extending below it, the component and all of the branches below are referred to as a *subshader*. Subshaders allow you to create truly complex effects—anywhere you can place a component on the shader tree, you can place a subshader instead.

The shader in the example below uses the Mix operator with the checkers function to create a complex checkerboard pattern in the Color channel. Half of the squares in the checkerboard are spotted—the spot pattern is produced by a second Mix operator, nested as a subshader within the first. The appearance of the remaining squares is determined by a texture map, loaded into Ray Dream Studio from a 2D image file. This image file might contain a multi-color gradient or fractal pattern, for example.



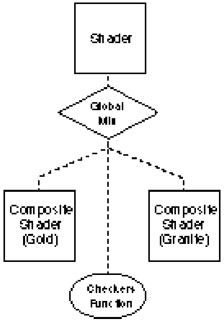
Using two mix operators in the shader tree lets you create a subshader.

Composite Shaders vs. Global Mix Shaders

The shaders in all of the preceding examples are called *Composite* shaders, because they are simply collections of individual channel settings. The settings within the different channels combine to determine the shader's overall appearance, but settings in one channel have no effect on the settings in the other channels.

In the last example, the Mix operator was used to create a complex pattern in a shader's Color channel. Suppose you wanted to create a shader which mixed attributes in all of the shader channels at once. For example, you might want to create a checkerboard pattern with alternating squares of shiny, reflective gold and rough granite. To accomplish this with a Composite shader, you would need to apply the same mix function to each channel. Fortunately, Ray Dream Studio provides another type of tree structure, the *Global Mix*, which makes it easier to achieve effects like this. The diagram below illustrates how a Global Mix shader works.

The checkerboard pattern applied to this object contains a second mix operator which produces spots in one half of the squares in the checker pattern.



Any component placed under the Global Mix affects all the shader channels.

The checker pattern applied to this object has alternating squares of gold and granite. It was created using the Global Mix shader.

The top level of a Global Mix shader tree has only one branch, which affects all eight shader channels. The Global Mix operator always appears on that branch. The Global Mix operator functions identically to the Mix operator, except that it mixes complete shaders rather than individual shader components. These complete shaders may be Composite shaders, as in the example above, or other Global Mix shaders. You can achieve some very complex shading effects by nesting Global Mix shaders.

Instructions for creating and editing both Composite and Global Mix shaders appear in "Building and Editing Shaders" on page 217.

Shader Components

This section describes Ray Dream Studio's various shader components. It explains what each component does, and how to set each component's unique parameters. Within this section, the components are divided by type into three groups: basic components, operators, and functions.

"Building and Editing Shaders" on page 217 explains how to use the Shader Editor to place shader components on the shader tree.

You may have received other shader components, in addition to those described here. Ray Dream Studio's open architecture allows developers to program their own shader components in the form of plug-ins. These third-party extensions join seamlessly with Ray Dream Studio's built-in shading tools.

Basic Components

Color

The *Color* component allows you to specify any color. Although you can place the Color component anywhere on the shader tree, it's best suited for use in the **Color**, **Highlight**, **Reflection**, **Transparency**, and **Glow** channels, which are designed for color input.

For a description of how colors affect the each channel, refer to "Shader Channels" on page 213.

- In the Bump channel, the Color component produces no effect because it gives a constant value across the surface of an object. To create the illusion of bumpiness, the Bump channel requires variation across an object's surface.

For a thorough discussion of the Bump channel, refer to "Bump" on page 214.

- In a non-color channel—Shininess or Refraction—colors are converted to values. Dark colors convert to low values, light colors to high values.

When you place the Color component on the shader tree, it appears as a color chip.

In the shader tree, a Color component appears as a Color chip.

To set a color:

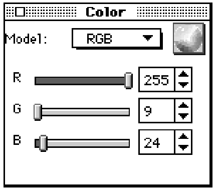
1 In the **Current Shader Editor**, click a channel tab.

2 Choose **Insert menu ▸Color**.

You can open the editor by clicking an object with the **Eyedropper** tool.

3 Double-click a color chip to edit the color. The color picker appears.

4 You can switch between the **RGB** and **CMYK** color models using the **Model** pop-up.



The Ray Dream Studio color editor.

5 Drag the **RGB** or **CMYK** sliders to set the color you want.

You can also enter specific values into the text boxes. The valid range for RGB values is from zero to 255. The valid range for CMYK values is from zero to 100%.

Your choice of a color model is for your convenience only—it has no effect on the content of your final rendered image. Like other rendering applications, Ray Dream Studio always renders RGB images. You can use an image processing application to convert rendered images to CMYK for separation.

To use the system color picker:

You might prefer to use the system color picker.

1 In the Ray Dream Studio color editor, click the color wheel in the upper right corner of the RBB/CMYK color picker.

Note: You can use the system color picker as the default. Choose **File menu▸ Preferences**. Choose **Shader Editor** from the pop-up. Enable the **Use system color picker** option.

Value

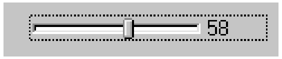
The *Value* component allows you to set a value between zero and 100%. You'll use the Value component frequently in every channel but the Color channel.

- If you use it by itself, it specifies a constant level for a particular attribute across the surface of an object or paint shape.

  For example, a 30% value in the Transparency channel means that the shader will make an object 30% transparent.

- If you place it beneath the Color channel, its value is converted to a shade of gray. Zero converts to black, 100 to white.

When you place the Value component on the shader tree, it appears as a slider.



In the shader tree, a Value component appears as a slider.

To set a value:

Drag the pointer along the **Value** slider. The far left side of the slider represents zero, the far right 100. The number to the right of the slider displays the current value.

Texture Map

The *Texture Map* component allows you to use a 2D image, such as a scanned photograph or paint-type illustration, in your shader. Used effectively, texture maps can lend your shaders unparalleled realism. Many complex real-world surfaces are nearly impossible to simulate through other means.

To assist you in creating texture maps, Fractal Design has developed Detailer—a dedicated program for preparing texture map images. Detailer and Ray Dream Studio work together and help you achieve superior results.

Using Color Images as Texture Maps

Texture maps using color images are extremely useful in the Color channel. You might import an actual product logo as a texture map and apply it like a decal to a 3D package model. Or to mimic an extremely detailed natural surface, you could import a small photographic sample of the surface and tile it—duplicate it a specified number of times—to cover your entire object.

The conch shell surface of this object was created by placing a texture map in the color channel.

Color texture maps are also appropriate in the **Highlight**, **Reflection**, **Transparency** and **Glow** channels.

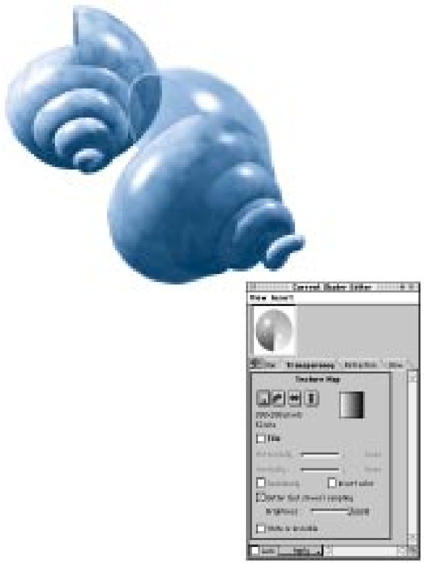
If you use a color image in a non-color channel, Ray Dream Studio internally converts it to grayscale.

You can use an image of any color depth as a texture map. In most cases, an 8-bit (256 color) image with a custom color palette provides excellent results, while requiring substantially less memory and disk space than a 32-bit image. Ray Dream Studio cannot use CMYK images as texture maps. If you want to use a CMYK image, you must convert it to RGB before importing it.

Black & White and Grayscale Images as Texture Maps

You can use a black and white or grayscale texture map in any channel (besides Color) to specify varying levels of a particular shading attribute. The shade of each pixel in the image determines the level of the attribute for the corresponding point on the object or paint shape.

If you use a black and white image, each black pixel turns the attribute on—sets it to 100%—while each white pixel turns the attribute off. An 8-bit grayscale image allows subtler effects, with 256 possible shades for each pixel. For example, a grayscale blend from white to black in the Transparency channel would make an object or paint shape fade smoothly from opaque to transparent.



The fade in on this object was created by using a texture map in the Transparency channel.

You can also use a black and white or grayscale texture map with the Mix operator, in place of a function. For more information, refer to "Using Other Components as Functions" on page 213.

Storing Maps Internally vs. Externally

By default, Ray Dream Studio saves copies of all texture maps internally. You can also have Ray Dream Studio save only a reference to an external file.

There are advantages and disadvantages to each option. Storing maps internally avoids organizational hassles, since you don't need to keep track of any external files. However, internally saved maps greatly increase the size of a file, which can result in slower loading and saving. With an external map, you can modify it with another program, and you don't need to reload the map.

In general, you can store maps internally unless your file contains particularly large texture maps (or many smaller maps).

If you move an externally referenced image on your hard drive, Ray Dream Studio prompts you to locate the image file the next time you open your scene file or shader Browser document.

Choosing an Image

When you place a Texture Map component on the shader tree, Ray Dream Studio displays the standard **Open** dialog, prompting you to choose an image.

To choose an image:

1 Select an image file from the file list in the **Open** dialog.

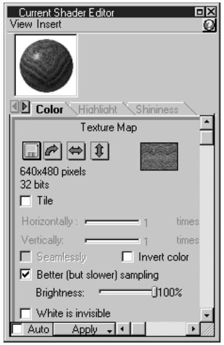
2 To specify whether you want a copy of the image saved within your Ray Dream Studio file, click the **Options** button in the **Open** dialog. The **Texture Map Options** dialog appears.

3 Select **Internal** or **External**.

4 Click **OK** to close the **Options** dialog.

5 Click the **Open** button in the **Open** dialog, or press **Return**/**Enter**. The texture map controls appear, displaying the image you have chosen.

Depending on the current display setting in the Shader Editor and the location of your Texture map component on the shader tree, you may not be prompted immediately to choose an image—in this case, a preview sphere appears in place of the texture map controls. To open the controls and choose an image, simply double-click on the preview sphere. A discussion of the Shader Editor's display options and instructions for navigating the shader tree appear in "Building and Editing Shaders" on page 217.



640x480 pixels
32 bits
□ Tile
Horizontally : 1 times
Vertically: 1 times
□ Seamlessly □ Invert color
☑ Better (but slower) sampling
Brightness: 100%
□ White is invisible

You can use an image as a component in shader tree using the Shader Editor's texture map controls.

A thumbnail preview of the image appears in the upper right corner of the Texture map controls. Immediately to the left of the preview, Ray Dream Studio displays the image's dimensions in pixels, and its color depth.

To flip or rotate the image:

Click the appropriate button to the left of the preview.

To tile the image:

1 Click the **Tile** check box to turn tiling on. Ray Dream Studio enables the tiling sliders.

2 Use the sliders to set the number of repetitions in each direction.

If you want Ray Dream Studio to rotate and flip neighboring tiles to maximize continuity, click the **Seamlessly** check box. If your image is specifically designed for seamless tiling, you don’t need to enable this option.

To filter the image:

1. Click the disk icon in the **Texture Map** controls.

2. Choose **Filter** from the pop-up which appears. A dialog listing available Adobe Photoshop™-compatible filters appears.

3. Select a filter and click **OK**.

If you have Adobe Photoshop-compatible filters on your hard drive, but none appear in the Filters dialog, you need to use the Preferences dialog to set the location of your third-party plug-ins. Step-by-step instructions for this procedure appears in “Advanced Filter Techniques” on page 376.

To replace the image:

Click the disk icon in the **Texture Map** controls, and choose **Open** from the pop-up. The **Open** dialog appears, allowing you to choose a different image.

or

Click the disk icon in the **Texture Map** controls, and choose **Import** from the pop-up. A dialog appears, allowing you to choose from any Adobe Photoshop-compatible plug-ins you may have available. If you have the appropriate plug-in, you can acquire an image from a scanner or a PhotoCD, for example.

The White is invisible check box allows you to achieve a type of masking effect by instructing Ray Dream Studio to ignore any purely white pixels in the image. For example, you could use this option to place a logo with an irregular shape on the surface of an object. Simply create a map with a white background and place it in the Color channel; then apply it to an object using the 3D Paint Rectangle tool. Wherever there is a white pixel in the map, the paint shape or primer below shows through.

Movie

The Movie component allows you to use any QuickTime (Macintosh) or AVI (Windows) movie within a shader. The movie behaves just as a texture map, except that the image differs in each frame of the rendered animation.

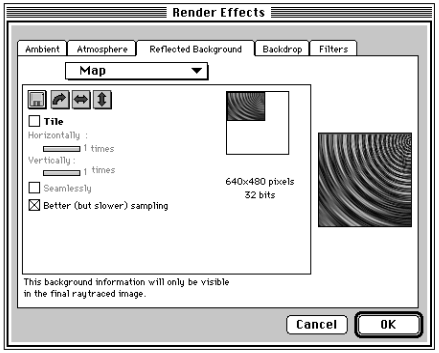
For more information on using movies as shader components, refer to “Rotoscoping” on page 334.

To select a movie in the shader component:

1. Click the disk icon. Studio displays the **Open** dialog.

2. Locate and open a movie file.

   When you’ve opened a movie, a preview player appears. This player lets you synchronize the movie with your animation. Refer to “Rotoscoping” on page 334.



□ Tile
Horizontally : 1 times
Vertically : 1 times
□ Seamlessly
☒ Better (but slower) sampling

640x480 pixels
32 bits

This background information will only be visible in the final raytraced image.

Movie Controls

3 You can click the directional buttons to change the movie's orientation.

4 You can reduce the image's brightness with the **Brightness** slider.

5 Enable the Better (but slower) sampling to view a more precise preview.

6 Enable the **Invert Color** option if you want to invert the movie's colors.

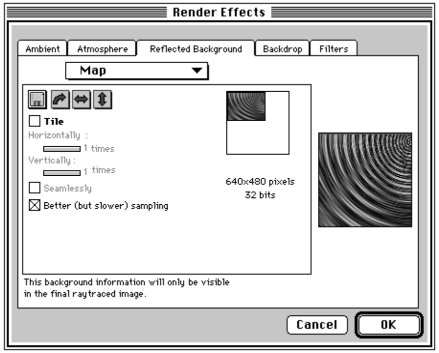
7 If you want the frame to repeat, enable the **Tile** option.

8 Use the **Horizontally** and **Vertically** sliders to set the number of tiles in each direction.

9 Enable the **Seamlessly** option to smooth the transitions between tiles.

10 You can get more information on this movie by clicking **More**.

Studio opens a window that provides technical information and a movie preview player.



Map

☐ Tile

Horizontally : 1 times

Vertically : 1 times

☐ Seamlessly

☒ Better (but slower) sampling

640x480 pixels
32 bits

This background information will only be visible in the final raytraced image.

Movie Time Selection dialog.

11 Enable the **White is Invisible** option if you want white regions of the movie to render transparent.

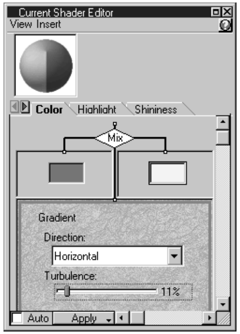
Operators

Operators allow you to create complex shading effects by combining two components or subshaders within a single shader channel.

Mix

The *Mix* operator is Ray Dream Studio's most versatile shading tool. It can produce a wide variety of results, depending on which function you choose to mix the two components.

When you place a Mix operator on the shader tree, it appears as a node with three branches below it. The left and right branches are placeholders for the two components you want to mix. The mixing function goes on the middle branch.



Gradient

Direction: Horizontal

Turbulence: 11%

The mix operator mixes the other components in the shader tree.

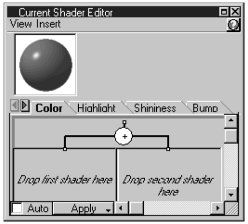
For each point on an object or paint shape, the function generates a value between zero and 100. The Mix operator uses this value to combine the components on the left and right branches. When the value is closer to zero, more of the left component is used. When the value is closer to 100, more of the right component is used.

Some functions, like Checkers and Wires, generate a value of zero or 100 for each point. These functions result in clear divisions between the two components. Other functions, like Wood and Spots, generate a range of values between zero and 100. These functions result in gradual blends between the components.

The various functions are described in "Functions" on page 207.

Add (+)

The *Add* operator sums the values of two components. When you place the Add operator on the shader tree, it appears as a node with two branches extending below.



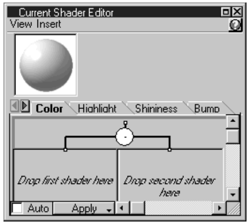
When you place the Add operator in the Shader Editor, it sums the values of the other components in the tree.

The two branches are placeholders for the components you want to combine. For each point on a paint shape or object, the Add operator sums the values of the two components.

Subtract (–)

The *Subtract* operator subtracts the value of the right component from the value of the left component.

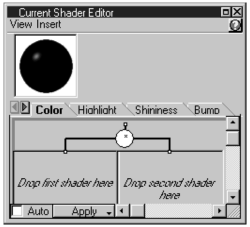
You can use the Subtract operator to invert the value of another component. Just place the component you want to invert in the right branch and a Value component set to 100 in the left branch. Try this technique on a complex shader if you want to invert its colors.



When you place the Subtract operator in the Shader Editor, it subtracts the value of the right component from the value of the left.

Multiply (*)

The *Multiply* operator multiplies the values of the left and right components.



When you place the Multiply operator in the Shader Editor, it multiplies the values of the other components in the tree.

The Multiply Operator can be used to tint a grayscale bitmap by multiplying the bitmap shader by a color shader.

Functions

Functions are used almost exclusively with the Mix operator. They can, however, be used by themselves. A function generates a value between zero and 100 for each point on an object or paint shape.

When you place a function on the middle branch of the Mix operator, the operator uses the values generated by the function to mix the components on the left and right branches. Where the value equals zero, the left component is used; where the value equals 100, the right component is used. Intermediate values produce a blending of the two components.

When you place a function in a channel by itself, it assigns a value directly to each point on the object or paint shape. In a non-color channel, each value is used “as is.” In any of the channels designed for color input (Color, Highlight, Reflection, Transparency or Glow), each value is converted to a shade of gray, with zero translating to white and 100 to black.

2D and 3D Functions

A 2D function creates a flat image that is ten wrapped around the shaded object. Checkers and Wires are 2D functions.

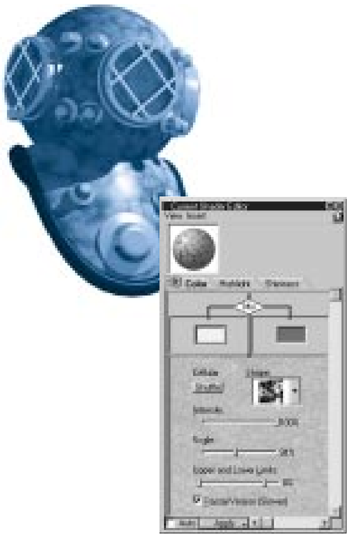
A 3D function creates a three-dimensional shader that “shares space” with the object. The object’s visible shading depends on how its surfaces coincide with the variations in the 3D shading volume. Cellular, Spots, Wood, Marble and

Psychedelic are 3D functions. These functions give the appearance that the object has been carved from a solid block of material.

You'll recognize the importance of 3D shading when you consider a block of wood. If the grain runs along one surface, the perpendicular surface shows *the ends of the grain*.

Cellular

The *Cellular* function creates a surface that looks like a network of cells. Cellular is a 3D function. It can be used effectively in the Color and Bump channels. You can experiment with it in the other channels for other effects.



You can create skin-like textures by using the Cellular function in the Shader Editor.

To set Cellular options:

1. Use the **Shape** menu to select the basic shape of the cells.

2. Drag the **Intensity** slider to adjust the contrast between the two colors.

   The slider ranges from -100% to 100%. Negative settings invert the image.

3. Drag the **Scale** slider to change the size of the cell shape. The slider ranges from 1% to 200%.

4. Drag the markers on the **Upper** and **Lower Limits** slider to controls the range of values created by the function.

   The range determines the balance between the two (light and dark) components. The left marker controls the lower limit. The right marker controls the upper limit.

5. Enable the **Fractal Version** option to use fractals instead of more regular shapes for the cells.

   Using the **Fractal Version** can significantly increase the time required to redraw and to render your image.

Spots

The Spots function produces a random pattern of spots. The spots are irregular, more like leopard spots than polka dots. Spots is a 3D function.

You can create a spot pattern using the Spots function in the Shader Editor.

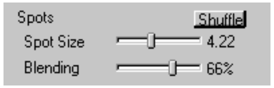
To set Spots options:

1. Adjust the **Size** slider to control the size of the spots.
2. Adjust the **Blending** slider to control how the spots blend together.

   When **Blending** is set to 0%, the function produces equal numbers of zero-value and 100-value spots, with gradual blending.

   When **Blending** is set to 50%, the function produces equal numbers of zero-value and100-value spots, with abrupt transitions.

3. Click the **Shuffle** button to randomize the spot pattern.



Spots — Shuffle
Spot Size 4.22
Blending 66%

Use the Shuffle button to randomize the pattern of spots.

Wood

The *Wood* function produces a wood grain pattern. It assigns a value of 100 to the veins and a value of zero to the spaces between the veins. Wood is a 3D function.

You can create a variety of wood grain textures by mixing different colors with the Wood function.

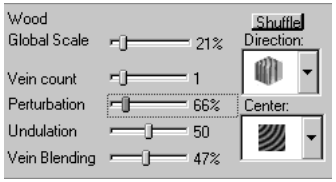
To set Wood options:

1. Adjust the **Global Scale** slider to set the size of the wood grain pattern in relation to the object.
2. Adjust the **Vein count** slider to set vein spacing.

   High values produce dense vein patterns, while low values produce sparse vein patterns.

3. Adjust the **Perturbation** slider to control the size of the “waves” in the veins.
4. Adjust the **Undulation** slider to control the frequency of the waves in the grain.
5. Adjust the **Vein Blending** slider to determine whether the transitions between veins and spaces are abrupt or gradual.
6. Use the **Direction** pop-up to set the orientation of the wood grain pattern in relation to the object.
7. Use the **Center** pop-up to set whether the wood grain pattern is taken from the center of the tree, where the curvature of the veins is quite pronounced or from a point further from the center, where the curvature is more gradual. There are three settings to choose from.

8. Click the **Shuffle** button to randomize the wood grain pattern.



Wood

Global Scale 21%

Vein count 1

Perturbation 66%

Undulation 50

Vein Blending 47%

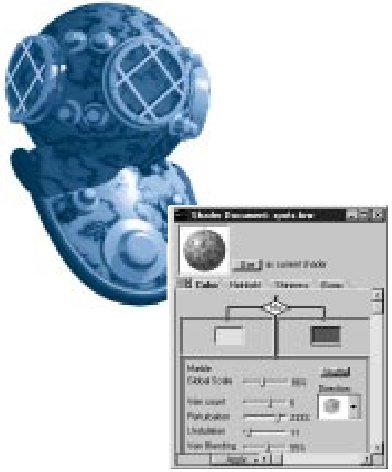
Direction:

Center:

Use the Shuffle button to randomize the pattern of the wood grain.

Marble

The *Marble* function produces a marble pattern. It assigns a value of 100 to the veins and a value of zero to the spaces between the veins. Like Spots and Wood, Marble is a 3D function.



You can create marble surface by using the Marble function in the shader tree.

To set Marble options:

1. Adjust the **Global Scale** slider to set the size of the marble pattern in relation to the object.

2. Adjust the **Vein count** slider to set vein spacing.

   High values produce dense vein patterns, while low values produce sparse vein patterns.

3. Adjust the **Perturbation** slider to control the size of the “waves” in the veins.

4. Adjust the **Undulation** slider to control the frequency of the waves in the pattern.

5. Adjust the **Vein Blending** slider to determine whether the transitions between veins and spaces are abrupt or gradual.

6. Use the **Direction** pop-up to set the orientation of the marble pattern in relation to the object.

7. Click the **Shuffle** button to randomize the marble pattern.

Objects shaded with 3D functions like Spots, Wood, and Marble appear to be carved or sculpted from solid blocks of material, but this appearance is only skin-deep—like all of the objects you create in Ray Dream Studio, these objects are still “hollow.” To illustrate this point, if you make a marble object partially transparent, you won’t see veins running through the inside of the object.

Checkers (and Stripes)

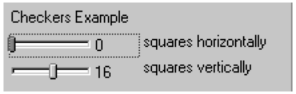
The *Checkers* function produces a checkerboard pattern. Every point is assigned a value of exactly zero or 100, so no blending occurs. Checkers is a 2D function.

The Checkers function can be used to create a checkerboard pattern on an object.

To set Checkers options:

1. Drag the **Squares horizontally** slider to set the number of horizontal squares.

2. Drag the **Squares vertically** slider to set the number of vertical squares.



Checkers Example

0 squares horizontally

16 squares vertically

Use the sliders to set the number of horizontal and vertical squares.

By setting one of the sliders to zero, you get stripes—either horizontal or vertical, depending on which slider is at zero.

Wires

The *Wires* function produces a grid of lines. Like the Checkers function, Wires assigns a value of either zero or 100 to each point on an object or paint shape. Specifically, it assigns a value of 100 to the lines of the grid, and a value of zero to the spaces. Wires is a 2D function.

You can create a lined pattern on the surface of an object using the Wires function.

To set Wires options:

1. Drag the **Horizontal count** slider to set the number of lateral wires.

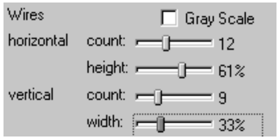
2. Drag the **Height** slider to set the size of the lateral wires.

The thickness of the horizontal/vertical wires is expressed as a percentage of the height/width of the entire object or paint shape. You can determine the thickness of a single wire by dividing the percentage by the number of wires. For example, if ten horizontal wires represent 50% of a paint shape's height, the thickness of each wire is 5% of the height.

3. Drag the **Vertical count** slider to set the number of longitudinal wires.

4. Drag the **Width** slider to set the size of the longitudinal wires.

5. Enable the **Gray Scale** option if you want to smooth the transitions between the spaces and the wires.



Wires □ Gray Scale

horizontal count: 12

height: 61%

vertical count: 9

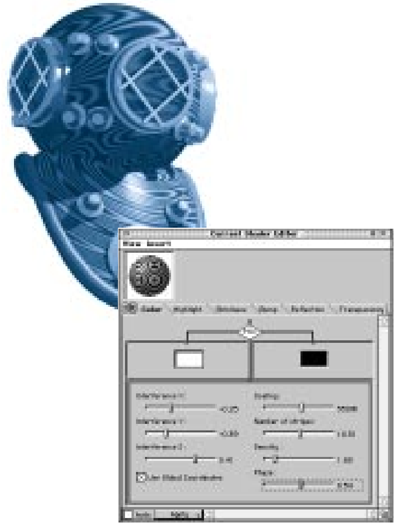
width: 33%

The grayscale option smooths the transition between wires and spaces.

This option is especially useful in the Bump channel, where gradual transitions produce more striking results.

Psychedelic

The *Psychedelic* shader function creates unusual textures, including swirls of color reminiscent of the psychedelic Pop Art of the 1960s. Psychedelic is a 3D function.



Use the Psychedelic function to create colorful swirls on an object.

To set Psychedelic options:

1. Adjust the three **Interference** sliders to control the amount of interference applied to each plane.

   The sliders range from -1.00 to 1.00. Each slider controls the stripes applied on the specific plane. A setting of 0 results in no interference on that plane. Using all three planes you can create more interesting design elements.

2. Enable the **Use Global Coordinates** check box to use the Global Coordinates for the Psychedelic function.

   By default the function uses the individual Object's coordinate system. The shading is constant as you move and rotate the object.

   When **Use Global Coordinates** is enabled, the shading remains static in global space when the object moves. The object behaves as a “window” on the shading. This is like moving your hand through the dappled shadows under a tree. Instead of shadows, though, the psychedelic shading passes across the moving object.

3. Drag the **Scaling** slider to control the size of the design elements.

4. Adjust the **Number of Stripes** slider to control the number of stripes in each element.

5. Adjust the **Density** slider to control the density of the pattern.

6. Adjust the **Phase** slider to set the current position in the psychedelic cycle.

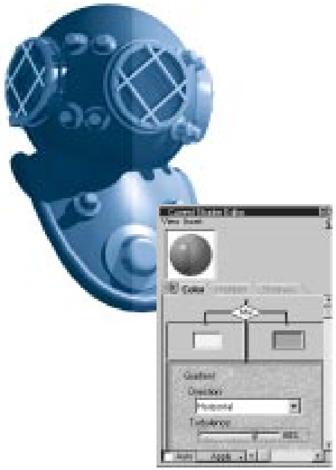
   The **Phase** slider ranges from 0.00 to 1.00. The two extreme settings display the cycle at the same point.

   Use the flat preview to see how the shader changes when you adjust the **Phase**.

Note: Phase settings are particularly useful in animations. Set key events to animate the **Phase** from zero to one using the **Oscillate: Saw/Loop** tweener.

Gradient

The *Gradient* function produces a gradual blend from one color to another. Gradient is useful in the Color and Transparency channels. Gradient is a 2D function.



Use the Gradient function to blend two colors in the shader.

To set Gradient options:

1. Choose **Direction menu▸ Horizontal** or **Vertical** to set the direction of the blend.

2. Adjust the **Turbulence** slider to mix up the colors as they change.

   A low setting results in a uniform blend. A high setting increases irregularity in the blend.

Formula

The *Formula* function lets you design a mixing function with a mathematical formula.

For more information, refer to the PDF file "Using Formulas" contained on the Ray Dream Studio 5 CD.

Using Other Components as Functions

You can actually use any component as a function in the Mix operator. This allows you to mix components in ways not supported by Ray Dream Studio's basic functions.

When you use a component as a function, Ray Dream Studio automatically converts its output to values so that the Mix operator can use it to mix the components on the left and right branches.

Value and *Texture* map components are especially useful in place of functions.

- You can use a *Value* component to blend the components on the left and right branches of the Mix operator in a given ratio. A value of 50 blends the two components evenly. Lower values favor the left component, while higher values favor the right component.

- You can use a black and white or grayscale texture map to mix the left and right components in a custom pattern. For each white pixel in the image, the Mix operator uses the left component. For each black pixel, the Mix operator uses the right component. Intermediate shades produce a blending of the two components.

Shader Channels

Color

No shader channel has a more obvious effect on the appearance of an object or paint shape than the *Color* channel. Depending on the components you place beneath the Color channel, you can specify anything from a plain color to a complex, multi-color pattern.

- To specify a plain color, simply place a Color component.

- To specify a multi-color pattern, use a Texture Map component or combine colors with the Mix operator.

If you find that your single-color objects look unrealistic, try using the Mix operator with the Spots function to create minute color variations. Simply mix two slightly different hues in a pattern of tiny spots, with Blending set to zero for smooth blends. When viewed from a distance, these subtle color variations can give an object a more realistic appearance.

The default content of the Color channel is a Color component, set to a shade of red. Ray Dream Studio assigns the default content of each channel, as part of the default primer, to each new object you create. The default content is also used to calculate the preview of a shader in the Shader Editor, when a particular channel is empty.

Highlight and Shininess

Most objects in the real world show highlights when illuminated. These bright spots or streaks are direct reflections of light sources, like the sun glinting off a chrome bumper.

Metallic objects have small, bright highlights. Plastics have dim, but large highlights. And stones usually have no highlights, unless polished.

You can control the color, intensity and size of an object's highlights by placing shader components beneath the Highlight and Shininess channels.

The *Highlight* channel controls highlight intensity. A high value produces bright highlights, while a low value produces dim highlights.

The *Shininess* channel controls highlight size. A high value produces small highlights, while a low value produces large highlights.

Note: Typically, highlights that are extremely bright are very small, while those that are softer spread larger.

By default, highlights are white. You can specify color highlights by placing a Color component (or any component that produces color information) in the Highlight channel. Color has no effect on the Shininess channel; any color information in the Shininess channel is internally converted to values.

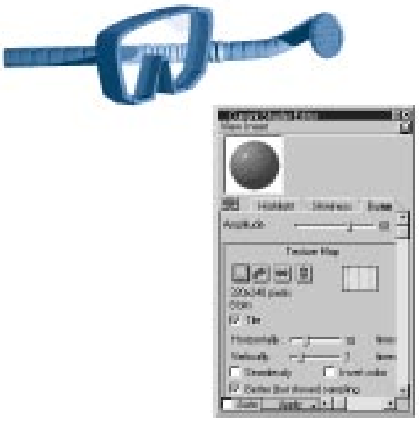
The default content of both the Highlight channel and the Shininess channel is a Value component, set to a value of 50.

If you want varying levels of highlight or shininess across the surface of an object or paint shape, you can use a texture map or a function.

Bump

Bumpy and rough surfaces are difficult to model geometrically. Imagine modeling the irregularities of the skin of an orange, point after point—the task would quickly become a nightmare. Fortunately, Ray Dream Studio allows you to use a shader to simulate variations on the surface of an object.

By placing shader components in the Bump channel, you can “perturb” a smooth surface to produce irregularities. This technique is known as bump mapping. Bump mapping can produce subtle effects like the bumps and pits on the surface of an orange, or more pronounced effects like the seams on a baseball.



The indents on the snorkel were created by placing a texture map in the Bump channel

Bump mapping does not change the actual topography of an object—no points are displaced. Rather, it changes the angle at which light rays reflect off the surface at certain points. This technique tricks the eye into perceiving texture. Because the bumpiness is just an illusion created by the renderer, the edge of the object remains smooth. In most cases, this is not a problem. However, if you require true surface variations to appear on the silhouette of your object, you’ll need to create them in the modeler.

To simulate bumps, the components you place beneath the Bump channel must specify some variation in the colors or values of adjacent pixels. Consequently, placing a simple color or value in the Bump channel does not produce bumping.

Texture maps and functions produce the best results in the bump channel. Like the other non-color channels, the Bump channel always converts color information to grayscale.

When you use a grayscale image in the Bump channel, lighter areas appear to be raised, while darker areas appear to be lowered. When you use a function that produces values, areas with higher values appear to be raised, while areas with lower values appear to be lowered.

- For the most convincing results, you should create a smooth blend between extreme values like black and white or zero and 100.

If a grayscale image with hard edges between black and white does not produce a satisfactory bump effect, try applying a blur effect to the image in an image-editing application.

- The relative height or “steepness” of the bumps depends on how rapidly you blend between extreme values. A blend that occurs over the range of many pixels produces shallow, gradual bumping. A blend that occurs over the range of a few pixels produces steeper, more sudden bumping.

The default content of the Bump channel is a Value component, set to a value of zero. This produces no bumping, since there is no variation from pixel to pixel.

To change the amount of bump:

1. In the Bump channel tab, adjust the Amount slider.

   The Amount slider lets you strengthen or weaken the Bump effect.

2. Apply your changes to the object.

3. Use the Render Preview tool to check the results on the object.

Reflection

Many real-world surfaces are at least somewhat reflective. Most types of metal and glass are partially reflective. A mirror is so reflective that it takes nearly all of its color from the environment around it.

Ray Dream Studio allows you to specify reflectiveness by placing shader components beneath the Reflection channel.

Although a Value component works fine in the Reflection channel, both subtler and more fantastic effects can be achieved by using a Color component instead. To make a highly reflective object appear more vivid (less washed out), use a variation of the same color you have in the Color channel (if the object’s color is particularly bright, you may want to use a darker tint of the same color). To give reflections an unusual tint, try using a color that is markedly different from the one in the Color channel.

- When you use color data (a Color component, a Texture Map component, or a complex subshader) in the Reflection channel, its hue affects the tint of the reflection and its brightness determines the amount of reflection. Darker colors produce less reflection; brighter colors produce more.
- Using values in the Reflection channel produces the same effect as using shades of gray. Only the amount of reflection is affected.

Of course, if you want varying levels and tints of reflection across the surface of an object or paint shape, you can use a texture map or a function.

The default content of the Reflection channel is a Value component, set to a value of zero.

Transparency

When light strikes an opaque surface, it simply bounces off. When it strikes a semitransparent surface, some light bounces off, but some passes through. As a result, you can see through a semitransparent object. Glass, water and clear plastic are examples of semitransparent materials.

You can specify transparency by placing shader components beneath the Transparency channel.

While a Value component is perfectly appropriate in the Transparency channel, a Color component provides a much broader range of possible effects. Objects made of colored glass tend to look more realistic if you use a similar color in both the Color and Transparency channels (if the object’s color is bright, you may want to use a darker tint in the Transparency channel).

Try using a function or a black and white texture map in the Transparency channel to create a shader which makes transparent "holes" in an object. For example, placing the Wires function in the Transparency channel creates a wireframe effect.

- When you use color data (a Color component, a Texture Map component, or a complex subshader) in the Transparency channel, its hue affects the appearance of colors viewed through the transparent shader. Its brightness determines how transparent the shader is. Darker colors produce less transparency; brighter colors produce more.
- Using values in the Transparency channel produces the same effect as using shades of gray. Only the amount of transparency is affected.
- A 100% transparent surface may not be completely invisible—depending on the settings in the other channels, it may still refract and show highlights.

The default content of the Transparency channel is a Value component, set to a value of zero.

Refraction

When light rays pass through a semi-transparent object, their trajectories are deflected. This phenomenon is known as refraction. Glass, fluids, and other translucent materials refract light to some degree. This results in a distorted view of objects behind the refractive surface.

The distorted view of the snorkel hose was created by using refraction on the mask.

A simple refraction experiment: Stand a pencil in a glass of water. Refraction "bends" the pencil where it enters the water.

A single Value component is generally all you'll need to place in the Refraction channel.

Note: The setting in the Refraction channel only affects a shader with some degree of transparency. Light must pass through in order to be refracted.

Shaders simulating glass should have Refraction values around 20. Shaders simulating water or ice should have values that are somewhat lower. Try experimenting with different values until you achieve the effect you want.

- Color has no effect on the Refraction channel. Any color information in the Refraction channel is internally converted to values.

The default content of the Refraction channel is a Value component, set to a value of zero.

Glow

The settings in a shader's Glow channel determine the shader's luminance. An object whose shader has a high degree of luminance will appear bright, even if it is not lit by ambient light or external light sources.

Glowing shaders can be used to simulate things like neon tubing, lit windows in a cityscape at night, or the LED display of a digital clock. They can also be used for the subtler purpose of brightening a particular object in a dimly lit scene. This allows you to emphasize an object without affecting the lighting and mood of your entire scene.

Objects with glowing shaders do not cast light on other objects in the scene.

Although a Value component can be used in the Glow channel, a Color or Texture Map component is more appropriate. For realistic results, try using similar colors in the Color and Glow channels.

To create unusual effects, you can place markedly different colors in these two channels. For example, you might create a shader which appears green when well lit, but glows red when covered in shadow.

- When you use color data (a Color component, a Texture Map, or a complex subshader) in the Glow channel, its hue determines the color of the glow, and its brightness determines the intensity.

For a related glow technique, refer to "Aura" on page 359.

Building and Editing Shaders

Most of the work in building and modifying a shader takes place in the *Current Shader Editor*. The **Browser palette: Shaders tab** and shader document windows are also useful.

The Current Shader Editor

The **Current Shader Editor** palette provides all of the tools for customizing a shader.

You can use the **Current Shader Editor** to design a new shader, modify a shader you've stored in the Browser, or edit a shader you've applied to an object or paint shape.

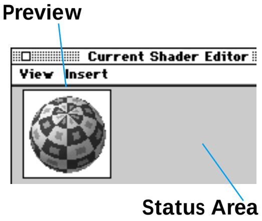
To display the Current Shader Editor:

Choose **Windows menu▸ Current Shader Editor**.

The contents of the Shader Editor depend on what type of shader you're editing: a Composite shader or a Global Mix shader.

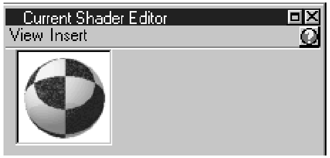
Because a Composite shader's tree can be quite complex, the Shader Editor does not display the entire tree at once. At any given time, you'll work with only the portion of the tree beneath a single shader channel. You can use the eight channel tabs to move from one channel to another.

The preview in the upper left corner of the Shader Editor shows you what the shader you are editing looks like. Each time you change the shader tree, Ray Dream Studio updates the preview to show how your change affects the shader.

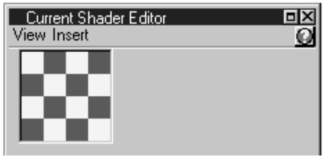


Ray Dream Studio's Shader Editor palette provides a preview of your shader.

You can choose a spherical preview, which shows you how your shader will look on the surface of an object, or a flat preview, which gives you an undistorted view of 2D image data like texture maps.



You can choose between a spherical shader preview...



... or a flat shader preview.

To switch between spherical and flat preview:

Choose **Current Shader Editor palette: View menu▸ Sphere Preview** or **Flat Preview**.

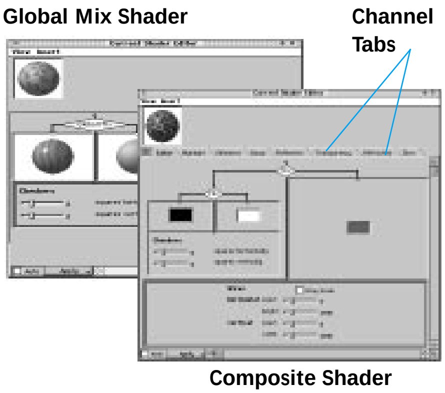
Note: This change affects only the preview of the shader you are currently editing.

To switch between shader channels:

Click the channel tab for the channel you want to edit.

The Shader Editor displays the branch of the tree beneath the channel you have chosen.

Note: The top level of a Global Mix shader's tree is fixed—it always contains a Global Mix component with three branches. The middle branch contains the mixing function, and the left and right branches contain the subshaders being mixed. Because all eight channels of the subshaders are mixed with the same function, no channel tabs appear in the Shader Editor when you are editing the top level of a Global Mix shader.



The Shader Editor.

To design a new shader:

1 Choose **Windows menu▸ Current Shader Editor**.

2 Use the **Shader Editor** tools and features to design the shader.

- Choose a tab, select a place holder, and add a component from the **Insert** menu.

Note: Refer to the earlier portions of this chapter for information on the shader components. Instructions for navigating and editing the shader tree appear below.

- You may also drag a shader from the browser into an appropriate place holder in the Shader Editor, where it becomes a subshader.
- You may also open a shader document from the Browser and drag its components or subshaders into the Shader Editor.

Note: If you drop a shader or subshader onto a place holder where its type is inappropriate, Ray Dream Studio alerts you. For example, you can't put a Global Mix shader where a simple component is required.

3 When you're finished designing, either apply your new shader to an object or add it to the Browser.

To add the current shader to the Browser:

1 Drag the shader preview from the **Current Shader Editor** to the **Browser** palette.

Note: You must drop the shader under a directory column.

Ray Dream Studio prompts you to name the new shader.

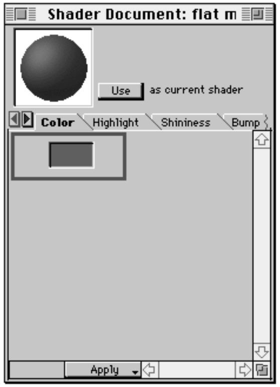
2 Enter a name and click **OK**.

To open/edit a shader from the Browser:

1. Double-click on the shader you want to edit.

Note: In the Browser, shaders are saved as documents.

Ray Dream Studio opens it in a shader document window.



Double-click on the shader to display the shader document window.

Note: If you don't see the shader document window right away, it might be hidden under one of the floating palettes—the **Current Shader Editor** or **Browser** palette.

2. Edit the shader by adjusting sliders and other settings.

You may also drag and drop components or subshaders from the Current Shader Editor or other open shader documents.

You may add or change components using the Insert menu on the main menu bar. This menu holds the same items as the Insert menu on the Current Shader Editor palette.

3. Click **Apply** to apply the modified shader to a selected object.

4. Click **Use** to move the contents of the document to the Current Shader editor.

5. You can save the modifications to the shader:

- Make sure the shader document is the active window.
- Choose **File menu▸ Save**.

Ray Dream Studio saves the changes into the Browser document file.

6. If you want to save the shader under a different name:

- Drag the preview into the Browser.
- Enter a new name.

7. Close the shader document window when you're done.

To edit an object's primer or a paint shape's shader:

1. Choose the **Paint Shape Selection** tool.

2. Display the **Current Shader Editor** palette.

3. Click an object to edit its primer—make sure you click an area where there are no paint shapes.

4. Click on a paint shape to edit its shader.

Note: If the Current Shader Editor isn't already open, double-click an object or paint shape to open the shader in the editor.

5. Edit the shader tree by adding, removing, and modifying shader components.

Instructions for navigating and editing the shader tree appear below.

6. When you are satisfied with the changes you've made, click the **Apply** button in the Shader Editor.

Navigating the Shader Tree

A complex shader tree has many levels. Each time the tree branches, a new level is added. When you are working with a

Composite shader, you can choose to view just one level at a time, or you can expand the tree to show all of the levels beneath a particular channel.

Viewing all levels requires more space on your screen, but eliminates the need to jump from one level to another to edit components.

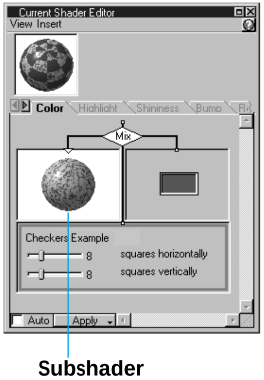
To choose a display mode:

Choose **Current Shader Editor palette: View menu▸ One** level or **All** levels.

The All levels display option does not apply to Global Mix shaders. If you want to edit the subshaders on the left and right branches, you need to “jump into” them—see below.

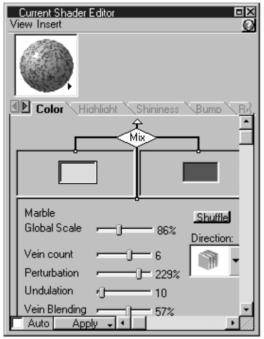
In the following examples, the Shader Editor is set to display only one level at a time. The screen shot on the left shows the highest level of a Composite shader’s Color channel. The screen shot on the right shows a subshader on the level immediately below.

In the screen shot on the left, notice that one branch of the Mix operator contains a preview sphere, rather than a set of controls. This indicates that the branch contains a subshader and that the tree continues to branch below the current level. To edit the subshader, you would need to “jump into” it—that is, jump down to the next level of the shader tree. After you finished viewing and editing the components in the subshader, you could then “jump out,” back to the level above.



Checkers Example
8 squares horizontally
8 squares vertically

Jumping into this subshader...



Marble
Global Scale 86%
Vein count 6
Perturbation 229%
Undulation 10
Vein Blending 57%

...allows you to edit its components.

When you jump into the left or right subshader of a Global Mix shader, the channel tabs appear, giving you access to the settings in each of the subshader’s channels.

To jump into a subshader:

There are two ways to jump into a subshader.

- Double-click the preview of the subshader.
- Click on the preview to select it, then choose **Current Shader Editor palette: View menu▸ Next Level**.

To jump out of a subshader:

There are two ways to jump out of a subshader.

- Choose **Current Shader Editor palette: View menu▸ Previous Level**.
- The tiny arrow in the lower right corner of the Shader Editor’s main preview indicates a pop-up. Press on it and select any higher level.

Editing the Shader Tree

You'll build a shader by placing components and subshaders on the branches of the shader tree. To edit a shader tree, you simply add, remove, and replace shader components.

Ray Dream Studio's drag and drop interface pertains not only to applying shaders, but also to building them. You can drag a component or subshader from one branch of the shader tree to another. You can also drop a shader from the Shaders Browser onto a branch of the shader tree, or drag a subshader from the tree into the Browser for storage.

This section describes how to construct a shader tree from a collection of components. For detailed descriptions of the various shader components and a discussion of how they interact, refer to "Shader Structure and Content" on page 198.

To specify a shader type—Composite or Global Mix:

Choose **Shader Editor palette: Insert menu▸ Composite** or **Global Mix**.

This replaces the current contents of the Editor with an empty Composite or Global Mix shader tree.

To place a shader component or subshader on the shader tree:

1. Display the shader channel tab where you want to work.

2. Click on the branch (component or place holder) you want to select. Ray Dream Studio outlines the selected branch in gray.

Note: The root of the channel is automatically selected.

If the component you select is the root of a subshader, the entire subshader is selected.

3. Choose the component you want to add from the **Current Shader Editor palette: Insert menu**.

Note: If you're working in a shader document, not the Editor, the Insert menu is on the main menu bar.

For complete information on the components, refer to "Shader Components" on page 201.

Note: When you add a simple component to the left or right subshader branch of a Global Mix shader, it is placed in the Color channel of the subshader. To edit its settings, you need to jump into the subshader.

To move components by dragging:

Drag a subshaders or component onto the branch of the shader tree where you want it.

- You can drag a shader from the Browser.
- You can drag a component or subshader from an open shader Browser document. This has the advantage of bringing all of its parameters with it.

  Ray Dream Studio copies the component or subshader you drag onto the branch where you drop it, replacing any contents that may have been there before.

When you drop a shader onto any branch of a Composite shader, only the contents of the applicable channel are placed on the branch. If the shader you're dropping has no components in the applicable channel, Ray Dream Studio notifies you, and nothing is added to the branch.

Note: When you drop a shader onto the left or right branch of a Global Mix shader, the entire shader you're dropping (all eight channels) is placed on the tree.

To remove the contents of a branch:

1. Select a component or subshader.

2. Choose **Edit menu▸ Clear**, or press the **Delete/Backspace** key.

   If the component you remove is the root of a subshader, Ray Dream Studio removes the entire subshader.

To copy a component or subshader to another channel:

Drag a component or subshader from its branch and drop it on a different channel tab.

Ray Dream Studio replaces the entire contents of the channel with the component or subshader you drop.

or

Drag a component or subshader from its branch and hold it over any channel tab. After a moment, that tab comes forward and you can drop the shader onto any branch in that channel.

Ray Dream Studio replaces that component or subshader with what you drop.