






## Overview

The **3DText v3.1** is a high quality ActiveX component, which allows you to create interactive 3D texts.

This component uses standard Nevron LLC automation objects including:

-  [NFillEffect](#) – the fill effect of the text and the background is consistently specified with the help of this automation object. Currently the fill effect can be either a solid color, 22 types of Excel like gradients, image texture, animated texture (AVI file used as a texture – even some professional 3D packages do not have this feature) or 48 types of Excel like patterns.
-  [NMaterial](#) – control the material properties of the text with this automation object. They are very important when the scene is rendered with lightning. They describe the diffuse, ambient and specular reflection of the object surface as well as its self – illuminance and shininess.
-  [N3DView](#) – easily control your viewpoint on the scene and the used projection type. The orthographic projection is very useful in the creation of 2D texts.
-  [NLightModel](#).

 [NLightSource](#).

 [NLightSourceCollectoin](#) – with them you can create an amazing lighting of the scene with up to 8 light sources.

Enjoy using this wonderful ActiveX control and don't forget to check out [www.nevron.com](http://www.nevron.com) for more new and exciting ActiveX components.

## **General Property Page**

From this property page you can specify the displayed text and the font which shall be used for the creation of the 3d text. Only True Type fonts are enumerated because these are the only font types supported by OpenGL. Strikeout and Underline effects are also not supported.

## Advanced Property Page

When you press the **FillEffect** button the  [Select fill effect dialog](#) appears. You can specify the fill effect of the text.

The **Material** button shows the  [Material properties Dialog](#). You can specify the material applied on the text when lightning is enabled.

The **Extrusion** edit controls the depth of extrusion of the basic text contour.

The **LineSpacing** edit controls the distance between the lines in a multi line text.

## N3DText

The main automation object. It has the following properties and methods :

**Name:** *Text*

**Type:** BSTR

**Description:** - Specifies the text to be displayed

**Name:** *TextFont*

**Type:** IFontDisp\*

**Description:** - Specifies the font of the displayed text

**Name:** *TextExtrusion*

**Type:** float

**Description:** - The extrusion of the basic text contour.

**Name:** *LineSpacing*

**Type:** float

**Description:** - Controls the spacing between the lines in a multilined text.

**Name:** *Redraw*

**Type:** VARIANT\_BOOL

**Description:** When false method and property calls do not redraw the control. It is highly recommended that you turn this flag off when doing multiple property changes.

## Methods

**Name:** *Update*

**Parameters:** None

**Return Value:** None

**Description:** This method allows you to redraw the client area of the control immediately without waiting for a paint message from Windows. This can be useful if you want to animate the text (for example to rotate it). In this case you should call Update after modifying the rotation parameters. You can see an example of this technique in the VC and VB examples supplied with the control.

**Name:** *FillEffect*

**Parameters:** None


**Return Value:** IDispatch\*

**Description:** Returns an IDispatch\* to the  [IFillEffect](#) interface that controls the fill effect applied to the text.

**Name:** *Material*

**Parameters:** None

**Return Value:** IDispatch\*

**Description:** Returns an IDispatch\* to the  [NMaterial](#) interface that controls the material applied to the text. Note that material properties take effect only when lightning is enabled.

**Name:** *Get3DView*

**Parameters:** None

**Return Value:** IDispatch\*

**Description:** Returns an IDispatch\* to the  [I3DView](#) interface that controls the viewpoint.

**Name:** *BackgroundFillEffect*

**Parameters:** None

**Return Value:** IDispatch\* - the  [NFillEffect](#) of the background

**Description:** Gives you access to the fill effect of the background.

**Name:** *LightSourceCollection*


**Parameters:** None

**Return Value:** IDispatch\* - the  [NLightSourceCollection](#) of the scene.

**Description:** Gives you access to the object that holds the lights applied on the scene. Note that light properties are discarded if lightning is disabled.

**Name:** *LightModel*

**Parameters:** None

**Return Value:** IDispatch\* - the  [NLightModel](#) of the scene.

**Description:** Gives you access to the NLightModel object that controls the general lightning conditions applied on the scene .


**Name:** *AboutBox*

**Parameters:** None

**Return Value:** None

**Description:** Shows the About Box

## Advanced light settings Dialog

From this dialog you can modify some advanced parameters of the light source. When you check the uniform light distribution the light source distributes light in all directions. Otherwise it is emitting light in the direction specified by the direction vector. When the light source is a directional one (e.g. it is distributing light in a certain direction) you can also specify the spot cutoff angle parameter. The spot cutoff angle is defined as the maximum angle between the direction of the light and the direction from the light to the vertex being lighted. If it is greater than the spot cutoff angle, then the light is completely masked. Otherwise the spot exponent and the attenuation factors control its intensity. For more information see the  [OpenGL lightning model](#) topic.

## Gradient Property Page

From this property page you can choose the gradient associated with the fill effect. Currently there are 6 shading styles. Each of them has 4 variants (except the **From Center** shading style). Use the radio buttons to choose the desired shading style and the variant selection window to specify a variant.

### **Color Property Page**

From this property page you can choose the solid color associated with the fill effect. The page shows the Windows basic colors.

Select the **Automatic** radio button if you want to synchronize the color with the current Windows settings for a window background. The color will automatically change when you change the Windows Appearance settings.

Press the **Choose Color...** button to select a color more precisely using the standard windows color selection dialog.



## Texture Property Page

From this property page you can choose the texture associated with the fill effect. The supported file formats are - JPEG, GIF, TARGA, TIFF, BMP, DIB and WMF (Windows Metafile). You can also control the mapping of the texture. In case you are applying the fill effect to a background the options are – **Tile**, **Center** and **Stretch**. In case you are applying the fill effect to the text the options are – **Object Linear** (mapping is relative to the object), **Eye Linear** (mapping is relative to the viewer) and **Sphere** map.

## Background Property Page

The **Fill Effect** button allows you to specify a fill effect for the background of the control.

### 3d View Property Page

Allows you to specify the polar coordinates of the camera as well as the zooming factor and the projection type. The camera always points towards the beginning of the coordinate system.

The **Rotation** and **Elevation** edit controls specify the polar coordinates.

The **Zoom** edit specifies the zooming factor. Possible values vary from 1 to 1000.

The **Projection type** combo allows you to choose between a Perspective and Ortogonal projections. In the case of a perspective projection you can also modify the perspective angle. The default projection type is perspective with angle 60 degrees. You can use Ortogonal projection in cases where you do not want perspective – for example when you want to display a 2D Chart, PieChart or Text. To achieve this select Ortogonal as the projection type and 90 as the Elevation degree – this will ensure that you look the scene perpendicular to the XY plane.

The **Enable virtual track ball** check allows you to turn on and off the **Virtual Track Ball**. The idea of the VTB is that you use the mouse to control your viewpoint on the scene in the same way as if you had sphere and the 3d image was inside of it. You do that by pressing the left mouse button and dragging the mouse in the desired direction to adjust your viewpoint.

## NFillEffect

One of the basic objects in the control. This object allows you to specify the visual effect applied on a particular object in the 3D scene. The object itself sometimes threatens the different properties of the NFillEffect object in a different way – for example when you are applying the fill effect to the background or to a plane or text.

### Properties:

**Name:** *Type*

**Type:** FEType

**Description:** Identifies the type of the fill effect. The possible values are **fetSolidColor**, **fetGradient**, **fetImage**, **fetAnimation** and **fetPattern**.

**Name:** *Automatic*

**Type:** VARIANT\_BOOL

**Description:** In the case of a solid color fill effect specifies that the color should be synchronized with the Windows system settings for window background.


**Name:** *Color*

**Type:** OLE\_COLOR

**Description:** In the case of a solid color fill effect specifies the fill effect's color. This property takes effect only if *Automatic* is false.


**Name:** *GradientStyle*

**Type:** FEGradientStyle

**Description:** In the case of a gradient fill effect specifies the shading style. Possible values are **gsHorizontal**, **gsVertical**, **gsDiagonalUp**, **gsDiagonalDown**, **gsFromCorner**, **gsFromCenter**. You can see the different shading styles and variants in the  [Gradient Property Page](#)

**Name:** *GradientVariant*

**Type:** FEGradientVariant

**Description:** In the case of a gradient fill effect specifies the variant of the shading style. Possible values are - **gvVariant**, **gvVariant2**, **gvVariant3**, **gvVariant4**. (Note – for the **gsFromCenter** shading style the possible values are only **gvVariant**, **gvVariant2**). You can see the different shading styles and variants in the  [Gradient Property Page](#)

**Name:** *GradientColor1*

**Type:** OLE\_COLOR

**Description:** In the case of a gradient fill effect specifies the starting gradient color.

**Name:** *GradientColor2*

**Type:** OLE\_COLOR

**Description:** In the case of a gradient fill effect specifies the ending gradient color.

**Name:** *ImageFileName*

**Type:** BSTR

**Description:** In the case of a texture fill effect specifies the file name of the texture. Currently the supported file formats are - JPEG, GIF, TARGA, TIFF, BMP, DIB and WMF (Windows Metafile).

**Name:** *ImageMapType*

**Type:** FEImageMapType

**Description:** In the case of a texture fill effect specifies the texture mapping type. Possible values are - mtMapType0, mtMapType1, mtMapType2. If you are applying the fill effect to an object the mapping types are treated as follows:

- mtMapType0 - Object linear (mapping is relative to the object).
- mtMapType1 - Eye linear (mapping is relative to the viewer).
- mtMapType2 – Sphere map.

If you are applying the fill effect to the background the mapping types are treated as follows:

- mtMapType0 – Tile
- mtMapType1 – Center
- mtMapType2 – Stretch

**Name:** *AnimationFileName*

**Type:** BSTR

**Description:** In the case of an animation fill effect specifies the file name of the animation. Currently the supported file format for animations is AVI.


**Name:** *AnimationMapType*

**Type:** FEImageMapType

**Description:** In the case of an animation fill effect specifies the animation mapping type. See ImageMapType for the possible values and the way they are treated depending on the object you are applying the animation to.


**Name:** *AnimationSpeed*

**Type:** long

**Description:** In the case of an animation fill effect specifies the speed used to play the animation in milliseconds. All values are automatically clamped to the range [50, 10000]. The default value is 50. Note that the actual speed of the animation may be slower if your computer is unable to display the animation with the requested speed. If you want to improve performance try to map smaller AVI files, or increase the *AnimationSpeed*. See also the  [Animated Texture Property Page](#) topic.

**Name:** *PatternType*

**Type:** FEPatternType

**Description:** In the case of a pattern fill effect specifies the pattern type. **NEVRON** controls support 48 pattern types as follows: **ptPercent5**, **ptPercent10**, **ptPercent20**, **ptPercent25**, **ptPercent30**, **ptPercent40**, **ptPercent50**, **ptPercent60**, **ptPercent70**, **ptPercent75**, **ptPercent80**, **ptPercent90**, **ptLightDownwardDiagonal**, **ptLightUpwardDiagonal**, **ptDarkDownwardDiagonal**, **ptDarkUpwardDiagonal**, **ptWideDownwardDiagonal**, **ptWideUpwardDiagonal**, **ptLightVertical**, **ptLightHorizontal**, **ptNarrowVertical**, **ptNarrowHorizontal**, **ptDarkVertical**, **ptDarkHorizontal**, **ptDashedDownwardDiagonal**, **ptDashedUpwardDiagonal**, **ptDashedHorizontal**, **ptDashedVertical**, **ptSmallConfetti**, **ptLargeConfetti**, **ptZigZag**, **ptWave**, **ptDiagonalBrick**, **ptHorizontalBrick**, **ptWeave**, **ptPlaid**, **ptDivot**, **ptDottedGrid**, **ptDottedDiamond**, **ptShingle**, **ptTrellis**, **ptSphere**, **ptSmallGrid**, **ptLargeGrid**, **ptSmallCheckerBoard**, **ptLargeCheckerBoard**, **ptOutlinedDiamond** and **ptSolidDiamond**. Use can see a preview of the different pattern types in the  [Pattern Property Page](#).

**Name:** *PatternSize*

**Type:** FEPatternSize

**Description:** In the case of a pattern fill effect specifies the file size of the pattern. The supported

sizes are :

- psSmall – use a 8x8 matrix for the pattern.
- psMedium – use a 16x16 matrix for the pattern (this is the default).
- psLarge – use a 32x32 matrix for the pattern.

**Name:** *PatternForegroundColor*

**Type:** OLE\_COLOR

**Description:** In the case of a pattern fill effect specifies the foreground color of the pattern.

**Name:** *PatternBackgroundColor*

**Type:** OLE\_COLOR

**Description:** In the case of a pattern fill effect specifies the background color of the pattern.

**Methods:**


**Name:** *ShowDialog*

**Parameters:** None

**Return Value:** VARIANT\_BOOL – when false indicates that no changes were made to the fill effect.

**Description:** Shows the Fill Effect Dialog through which you can visually edit the fill effect.

## N3DView

Controls the polar coordinates of the camera, the zooming factor and the projection type. The camera always points towards the beginning of the coordinate system. See the  [3d View Property Page](#) topic for more information.

**Name:** *ProjectionType*

**Type:** PTProjectionType

**Description:** Specifies the projection type of the camera. Possible values are ptPerspective and ptOrthogonal.

**Name:** *Rotation*

**Type:** float

**Description:** The rotation of the viewpoint around the Z-axis in degrees.

**Name:** *Elevation*

**Type:** float

**Description:** The elevation of the viewpoint in degrees.

**Name:** *Zoom*

**Type:** float

**Description:** The zooming factor applied to the 3D scene.

**Name:** *PerspectiveAngle*

**Type:** float

**Description:** The perspective angle in the case of a perspective projection. The default value is 60 degrees.

**Name:** *EnableMouse*

**Type:** VARIANT\_BOOL

**Description:** Enables or disables the virtual track ball.

## NTextEx

The NTextEx automation object is the primary means through which you orient and specify font, size and fill effect of a text. The orientation of the text begins from a point called **origin point**. The origin point is initially determined by the component. The positioning of the text follows the sequence:

1. On the text is applied the selected vertical and horizontal alignment (HorzAlign and VertAlign properties).
2. The text is translated to the polar coordinates specified by CoordRotateXY, CoordRotateZ and CoordRadius.
3. The text is rotated along the X, Y and Z - axes with the angles specified by OrientRotateXAxis, OrientRotateYAxis and OrientRotateZAxis.

## Properties

**Name:** *Font*

**Type:** IDispatch\* - the font automation object

**Description:** Gives you access to the text's font.

**Name:** *CoordRotateXY*

**Type:** float

**Description:** the rotation of the translation point in the XY plane

**Name:** *CoordRotateZ*

**Type:** float

**Description:** the rotation of the translation point around the Z-axis

**Name:** *CoordRadius*

**Type:** float

**Description:** distance of the translation point from the origin point.

**Name:** *OrientRotateXAxis*

**Type:** float

**Description:** rotation around the X – Axis.

**Name:** *OrientRotateYAxis*

**Type:** float

**Description:** rotation around the Y – Axis.

**Name:** *OrientRotateZAxis*

**Type:** float

**Description:** rotation around the Z – Axis.

**Name:** *Extrusion*

**Type:** float

**Description:** the extrusion of the text.

**Name:** *HorzAlign*

**Type:** HorizontalTextAlign (haCenter = 0, haLeft = 1, haRight = 2)



**Description:** the horizontal alignment of the text around the origin point.

**Name:** *VertAlign*

**Type:** VerticalTextAlign (vaCenter = 0, vaTop = 1, vaBottom = 2)

**Description:** the vertical alignment of the text around the origin point.

**Methods:**

**Name:** *FillEffect*


**Parameters:** None

**Return Value:** IDispatch\* - the  [NFillEffect](#) applied to the text.

**Description:** Gives you access the fill effect applied on the text.

**Name:** *Material*

**Parameters:** None

**Return Value:** IDispatch\* - the  [NMaterial](#) of the text

**Description:** Gives you access to the material applied to the text

**Name:** *ShowDialog*

**Parameters:** None

**Return Value:** VARIANT\_BOOL - when FALSE indicates that no changes were made to the TextEx object.

**Description:** Shows the TextEx dialog through which you can visually edit the NTextEx object.

## Select fill effect Dialog

This dialog allows you to control the fill effect applied to the background or the text. The dialog has three property pages as follows:

-  [Color](#)
-  [Gradient](#)
-  [Texture](#)
-  [Animated texture](#)
-  [Pattern](#)


## Lights Property Page

From this page you can control the lightning applied to the scene. From the Light Model section you can turn on and off the lightning, the local viewpoint lightning and select a global ambient light.

When you check the **Enable lightning** check the light properties are applied to the scene. Otherwise all the properties regarding the light are discarded (e.g. materials, individual lights and global ambient light).

The **Enable local viewpoint lightning** check allows you to control the way OpenGL performs the lightning calculations or more specifically the specular angle of the lights. When the local viewpoint lightning check is off specular reflection angles take the view direction to be parallel to and in the direction of the  $-z$  axis, regardless of the location of the vertex in eye coordinates. Otherwise specular reflections are computed from the origin of the eye coordinate system. This check is off by default. You should use local viewpoint lightning only when you want to generate a more realistic scene, because it is more computationally expensive.

The **Global ambient light** color combo allows you to specify the global ambient light applied on the scene.

From the **Light properties section** you can control the parameters of each individual light. You can control the position, the direction (note that the direction vector is not applied if you use a uniform light distribution light) and the Ambient, Diffuse and Specular light components. The **Advanced...** button displays the  [Advanced light settings Dialog](#) that allows you modify some advanced features of the light. For a complete discussion of lightning see the

 [OpenGL lightning model](#) topic.

## NMaterial

The NMaterial automation object represents the material applied on the object. It has the following properties and methods. For more information about the meaning of the different properties and their effect on the object see the [OpenGL lighting model](#) topic

### Properties

**Name:** *AmbientColor*

**Type:** OLE\_COLOR

**Description:** Specifies the ambient color of the object.

**Name:** *SpecularColor*

**Type:** OLE\_COLOR

**Description:** Specifies the specular color of the object.

**Name:** *DiffuseColor*

**Type:** OLE\_COLOR

**Description:** Specifies the diffuse color of the object.

**Name:** *EmmiviveColor*

**Type:** OLE\_COLOR

**Description:** Specifies the emissive color of the object.

**Name:** *AmbientDiffuseColor*

**Type:** OLE\_COLOR

**Description:** Modifies the ambient and the diffuse color simultaneously. Note that this property is write only and is supplied only for programmer's convenience since it is very common that the ambient and diffuse color match.

**Name:** *AllColors*

**Type:** OLE\_COLOR

**Description:** Modifies all the colors of the material (e.g. the ambient, the diffuse, the specular and the emissive color) simultaneously. Note that this property is write only and is supplied only for convenience of the programmer.

**Name:** *Shininess*

**Type:** float

**Description:** Controls the shininess of the material.

### Methods:


**Name:** *ShowDialog*

**Parameters:** None

**Return Value:** VARIANT\_BOOL - when FALSE indicates that no changes were made to the NMaterial object.

**Description:** Shows the Material dialog that allows the user to modify the material properties.

## NLightModel

The NLightModel automation object represents the light model that is global for the whole scene. For more information see the  [OpenGL lightning model](#) topic. The interface has the following members:

### Properties

**Name:** *GlobalAmbientLight*

**Type:** OLE\_COLOR

**Description:** Specifies the ambient color that is global for the whole scene. This property takes effect only when lightning is enabled.


**Name:** *EnableLightning*

**Type:** VARIANT\_BOOL

**Description:** Turns on and off the lightning. When off all lightning properties (lights sources and materials) are discarded and have no effect on the rendered scene.

**Name:** *LocalViewpointLighting*

**Type:** VARIANT\_BOOL

**Description:** Controls the way OpenGL calculates specular reflection angles. See  [Lights Property Page](#) for more information

## NLightSource

This automation object represents a single light source. It has the following properties and methods. For more information see the [OpenGL lightning model](#) and [Lights Property Page](#) topics

### Properties

**Name:** *Name*

**Type:** BSTR

**Description:** Specifies the name of the light source. This name is supplied only for programmers convenience and does not have any effect on the light generated by this light source.

**Name:** *AmbientColor*

**Type:** OLE\_COLOR

**Description:** Specifies the ambient light that this light source is emitting.

**Name:** *SpecularColor*

**Type:** OLE\_COLOR

**Description:** Specifies the specular light that this light source is emitting.

**Name:** *DiffuseColor*

**Type:** OLE\_COLOR

**Description:** Specifies the diffuse light that this light source is emitting.

**Name:** *SpotCutoff*

**Type:** float

**Description:** Specifies the spot cutoff angle of the light source. When this property is set to 180.0 the light source is considered non-directional and emitting in all directions. If you set the SpotCutoff angle in the range of [0, 90] you specify that it is emitting light in the direction specified by the direction properties. All values not equal to 180 and not in the range of [0, 90] are automatically clamped to the range [0, 90]. For more information on this property see the [Lights Property Page topic](#).

**Name:** *SpotExponent*

**Type:** float

**Description:** Specifies the intensity distribution of the light.. All values are automatically clamped in the range [0,128]. Effective light intensity is attenuated by the cosine of the angle between the direction of the light and the direction from the light to the vertex being lighted, raised to the power of the spot exponent. Thus, higher spot exponents result in a more focused light source. The default spot exponent is 0, resulting in uniform light distribution

**Name:** *ConstantAttenuation*

**Type:** float

**Description:** Specifies the constant attenuation factor for this light.

**Name:** *LinearAttenuation*

**Type:** float

**Description:** Specifies the linear attenuation factor for this light.

**Name:** *QuadraticAttenuation*

**Type:** float

**Description:** Specifies the quadratic attenuation factor for this light.

**Name:** *PositionX*

**Type:** float

**Description:** Specifies X value of the light position vector.

**Name:** *PositionY*

**Type:** float

**Description:** Specifies Y value of the light position vector.

**Name:** *PositionZ*

**Type:** float

**Description:** Specifies Z value of the light position vector.

**Name:** *DirectionX*

**Type:** float

**Description:** Specifies X value of the light direction vector.

**Name:** *DirectionY*

**Type:** float

**Description:** Specifies Y value of the light direction vector.

**Name:** *DirectionZ*

**Type:** float

**Description:** Specifies Z value of the light direction vector.

## NLightSourceCollection

### Methods:

**Name:** *Add*

**Parameters:** BSTR Label

**Return Value:** IDispatch to the new NLightSourceObject.

**Description:** Add a new light to the NLightSourceCollection object.

**Name:** *GetAt*

**Parameters:** long nIndex

**Return Value:** IDispatch to the NLightSourceObject.

**Description:** Gets an IDispatch to the NLightSourceObject at nIndex. Note that the indexes are zero based.

**Name:** *RemoveAll*

**Parameters:** None

**Return Value:** None

**Description:** Removes all the objects in the NLightSourceCollection.

**Name:** *RemoveAt*

**Parameters:** long nIndex

**Return Value:** None

**Description:** Removes the light source at the specified index. Note that the indexes are zero based.

**Name:** *GetCount*

**Parameters:** None



**Return Value:** int

**Description:** Returns the number of light sources in the NLightSourceCollection.



## OpenGL lightning model


OpenGL approximates light and lightning as if light can be broken into red, green and blue components. Thus, the color of light sources is characterized by the amount of red, green and blue light they emit, and the material of surfaces is characterized by the percentage of the incoming red, green and blue components that is reflected in various directions.


In the OpenGL lightning model the light in a scene comes from several light sources. Some light comes from a particular position and some light is generally scattered about the scene. For example when you turn on a light bulb in a room, most of the light comes from the bulb, but some light comes after bouncing off one, two, three or more walls. This bounced light is called ambient light that seems to come from no particular direction, but it disappears if the light source is turned off. You can modify this light by setting the **AmbientColor** property of the  [NLightSource](#) object. There might be a general ambient light in the scene that comes from no particular source as if it had been scattered so many times that its original source is impossible to determine. You can control this global ambient light by modifying the **GlobalAmbientLight** property of the  [NLightModel](#) object.


In the OpenGL lightning model the light sources have an effect only when there are surfaces that absorb and reflect light. Each surface is assumed to be composed of a material with various properties. A material might emit its own light, scatter some light, or reflect light in a desired direction like a mirror.

So in general OpenGL divides the lightning in four independent components: emissive, ambient, diffuse and specular. All four of them are computed independently and then added together.

Ambient light as already mentioned is the light scattered so many times that its original light source is impossible to determine.


Diffuse light comes from a particular direction so it is brighter if it comes squarely down on a surface than if it barely glances off the surface, but once it hits the surface however it is scattered equally in all directions. You can modify the diffuse component of a particular light by modifying the **DiffuseColor** of the  [NLightSource](#) object.

Specular light comes from a particular direction and bounces off a surface in a preferred direction. You can modify the specular component of a particular light by modifying the **SpecularColor** of the  [NLightSource](#) object.


On the other hand materials are characterized by the percentages of light they reflect from the different light components. You can modify these percentages by changing the values of the **AmbientColor**, **DiffuseColor** and **SpecularColor** of the  [NMaterial](#) object. In addition to this materials can also emit light. You can modify the emissive light of a material by changing the value of the **EmmiviveColor** property of the NMaterial object. By default this value is RGB(0, 0, 0) which means that the object does not emit any light.

In addition to this you can also modify the attenuation factors for every light source. In the real world the intensity of the light decreases as distance from the light increases. You can control this process by modifying the **ConstantAttenuation**, **LinearAttenuation** and **QuadraticAttenuation** properties of the NLightSource object. OpenGL attenuates a light source by multiplying the contribution of that light source by an attenuation factor using the following formula:

$$\text{Attenuation factor} = 1 / (\text{KC} + \text{KL} * \text{D} + \text{KQ} * \text{D} * \text{D})$$

Where KC is the **ConstantAttenuation**, KL is the **LinearAttenuation** and KQ is the **QuadraticAttenuation** and D is the distance from the light source to the object. See also the  [Advanced light settings Dialog](#) topic.

## Material properties Dialog

From this dialog box you modify the material properties of the specified object. The properties include the ambient, diffuse, specular and the emissive colors of the material as well as the shininess. For more information on lightning see the  [OpenGL lightning model](#) topic.

## Animated Texture Property Page

From this property page you can choose the animated texture associated with the fill effect. Animations are AVI files. You can also control the mapping of the animation. In case you are applying the fill effect to a background the options are – **Tile**, **Center** and **Stretch**. In case you are applying the fill effect to the text the options are – **Object Linear** (mapping is relative to the object), **Eye Linear** (mapping is relative to the viewer) and **Sphere** map. You can also modify the speed (in milliseconds per frame) for displaying the animated texture. A speed of 50 is close to video quality. Use the spin control to modify the default speed.

**Warning!** Use the video speed with caution. Great speed may seriously impact the ability of the system to process messages and other user input. In general try to map smaller AVI files and use animated textures only if you have a fast processor.

## Pattern Property Page

From this property page you can choose the pattern associated with the fill effect.

Use the **Background** and **Foreground** color combo's to adjust these pattern parameters. The Default color **Background** color is black and the default **Foreground** color is white.

The **Pattern size** combo box allows you to specify the size of the pattern grid. Small pattern size corresponds to 8x8 pattern matrix, medium to 16x16 and large to a 32x32. The default size is medium.

