Beginners overview of RedSDK rendering
Including materials and lighting.

Version 1.1
Updated August 2023

Some information gleamed from TC, the Redway RedSDK document site.
Along with some experimentation.

Text in italics are from the original TC help files. With the odd few from checking the Redway documentation. Text not in italics are my thoughts.

IMPORTANT

I am not a rendering expert. Some things I have no idea what difference they make to a render scene. Some things I never use. And some I rarely use.
I cannot guarantee error free. Tested on my computer only.

Some items may not be applicable to older TC versions. As RedSDK 'evolved' with different features added over the years. Items tested in TC v2020 / 2021.

This document is provided as a guide only.

NOTE. This document may contain spelling errors and not good grammar.

Compiled by Andy H

Contents

Materials

Options 3
Double sided 3
Refraction **caustics** 4
Reflection **caustics** 4
Fresnel 7
IOR 8
Transmission Scattering 11
Reflectance fog 11

Pattern 13
Colour 13
Texture 13

Reflectance 17
Colour 17
Texture 17

Tranparency 20
Colour 20
Texture 21

BDRF Reflectance 24

BDRF Orientation 26

BDRF Transmission 27

Bump map 28

Displacement map 31

Texture space (wrapping) 33

Luminance

General standard properties 39
General physical properties 45
Ambient 47
Beam 48
Directional 51
Point 53
Rectangular area 55
Spot 57
Physical 60
Rectangular physical 63
Spherical physical 66
Sky 69
Sun 73

Environment

Environment 75
Plain 75
Environment map 75
Sky 79

Portal

Skyligh Portals 80

Volume

Volumetric effect 82

Global illumination

Global illumination. 87

GI - render options 87
GI - raytrace options 90
GI - GI options 96
Gi - Tone 99
GI - Sketch 101
GI - Baking 101

Redway
GI Suggestions 102

Vectors 103

Texture scaling 109
UV wrap 116

Seeding 118

Render manager dialog's 120

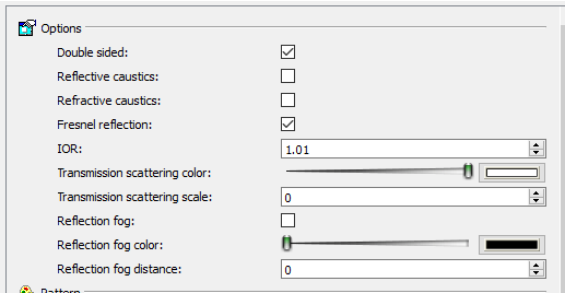
Drawing setup RedSDK 124

GI baking 106

Native draw options 127

Material Editing.

Fig 1



In the material editing section there are seven tabs, the first of which is

Options tab.

Double sided sheets. - One thing to ensure regarding this setting is that, it is almost essential to be turned on for ACIS surface 'sheet ' objects. By that it is Meant, a zero thickness objects. Unless you have a specific reason for using single sided.

In the Fig's below a single 2D line was drawn. Which is extruded and given a material. The object was copied and rotated 180 degrees. This gives a single facet object, with a front and back face.

In Fig 2. Double sided option was selected, and shows the texture was rendered on both sides. i.e. front and back of the facet

In Fig 3. Double sided was turned off. And because a sheet object is a single facet, it renders one of the sides invisible. This is because, when rotated we are now looking at the back side of the second object. And because the material is only rendered on one side of the facet, the other side is simply is not there.

Fig 2.

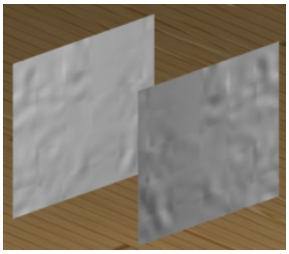
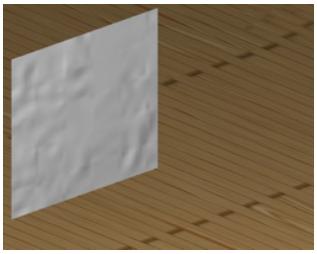
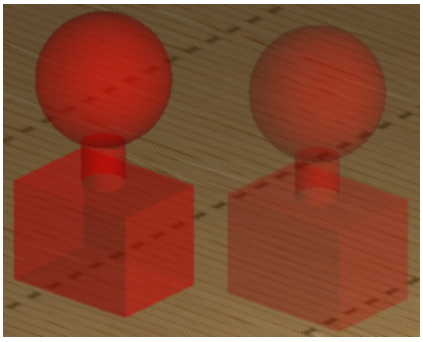


Fig 3.



Double sided and transparency, Double sided can also affect how transparency objects look. When light travels through the object, some of the light gets reflected. This is because the inner facets are also rendered.

Fig 4.



In Fig 4. the left hand object has double sided turned on. This gives clearly defined inner surfaces. However turning off double sided as in the object on the right, the inner facets are not rendered. Thus, as far as the render is concerned, that side is not there. And as there is no 'bounce back' or refraction, etc. It appears washed out.

With no 'sheet' objects or no transparency, the double sided option can safely be left turned on or off. Personal preference. However certain circumstances may require double sided on when used with 3d solid objects.

In this last example, we have a box inside a sphere. When using camera perspective, we can move the camera to see inside the sphere. This allows us to see the box surrounded by the spherical shape.

Fig 5. Shows the setup. A box inside a sphere with extra boxes positioned outside the sphere.

Fig 6. Shows the setup, with the material applied to the sphere. The camera is positioned inside the sphere (perspective turned on). Having double sided ticked allows the material to show on the outside the sphere, and the inside of the sphere. The extra boxes are not visible.

Fig 7. Has double sided turned off. This shows the white background being displayed, Essentially the back face (Inside of the sphere), is not rendered. This allows the boxes outside the sphere to show up.

Fig 5.

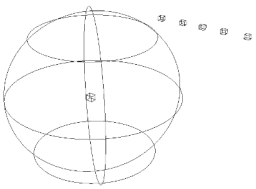


Fig 6.

Fig 7.

Would you want double sided turned off. There are situations where turning off is desirable.

In Fig 8 and Fig 9. We have a magnifying glass, (slightly domed on both sides). Fig 8 has double sided turned on, and because the glass is domed, we are getting distortion. Which although this may be realistic, it may not be desirable in some circumstances.

Fig 8.

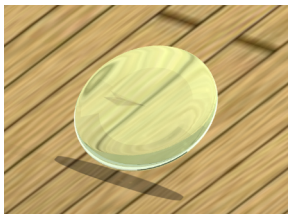
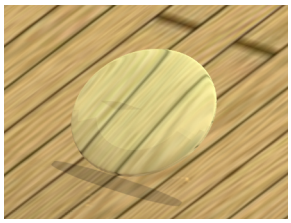


Fig 9.



Whilst this can be compensated for in other ways, a simple approach would be to turn off double sided. As the back (internal) faces of the objects are not rendered, there is much less distortion, allowing a cleaner effect.

Double sided and light.

Double sided also affects how light and shadows are created. As light hits a face normal (the side with the material applied) , behind the object, a shadow is created. However if the light is shining at the back (transparent) face, light goes straight through the object with no shadow created.

Below, camera / render properties, has 'Renderable objects' turned on, so then an object outline is visible even if the object is invisible.

Fig 9a, Is the normal setup, the the light is shining at the wrapped face and casting a shadow behind the objects.

In Fig 9b, the objects are rotated 180 degrees, so we are looking at the back faces. With double sided turned off, the back side it transparent, and light passes straight trough, with no shadow.

In Fig 9c, the camera view was changed, so we are now looking at the front textured faces of Fig 9b. Because the backs are transparent, the light goes straight through the backs, with no shadows.

In Fig 9d, the objects were again, so the we the same as in Fig 9a. The backs are transparent as we look at the objects, but with the light is now shining on the wrapped face. Shadows are created.

Fig 9a

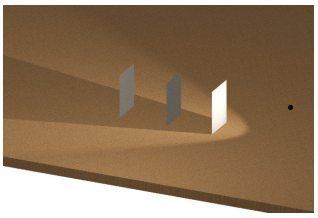


Fig 9b

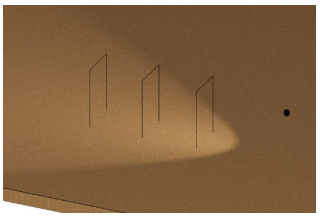


Fig 9c

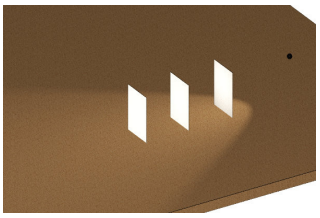
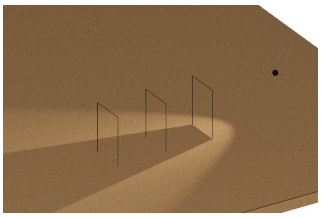


Fig 9d



Reflective Caustics. -
Refractive Caustics. -

Caustics are focused particles of light, due to reflections and refraction's. RedSDK has two tick boxes for caustics in the materials options tab. However caustics are more of a lighting setup than simply ticking some boxes.

Fig 10. Is rendered with caustics turned off. Fig 11. Has Caustics turned on. And shows the caustic effect on the base surface.

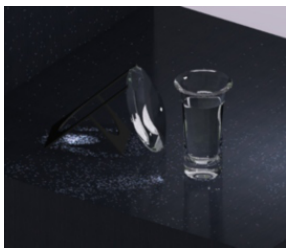
Caustics are really more for reflective / refractive media like glass and water, and therefore it is advisable to have both options turned on.

To use caustics one needs a light with a caustic option. The three physical Luminance fit the bill, along with Sun and Global luminance. All have a caustic tick box, which will tell the program to use photon mapping to track the light.

To use on a glass for example, one could have a glass material with caustics ticked. High transmission. Medium to low reflectance, Fresnel optional, IOR to suit. The images on the right, have an IOR 1.4.

Fig 10.

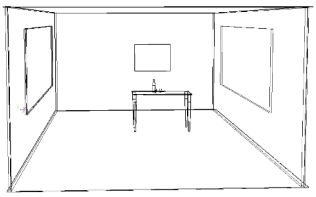
Fig 11.



In addition, a base object is needed, both for the object to sit on, and as a ‘collector’ for the caustics. The base could be set up with low reflectance. Caustic not ticked, and a low transmission. Ray-trace or GI render, cross fingers.

Caustic example.

Fig 507



Caustics require a number of things to come together, to achieve the desired result. For example from glass on a table. Or a swimming pool. Caustic light sources. And although raytrace Gi
The first of which is a material assigned to give the caustic effect.

Fig 507 shows a basic setup. A sealed room with windows (four walls / ceiling / floor. Table, bottle, glasses, and a picture (just for fun)

We need to assign materials. Some are just standard, floor, wall, table and picture. The glass is the material for applying the caustic effect.

Fig 508

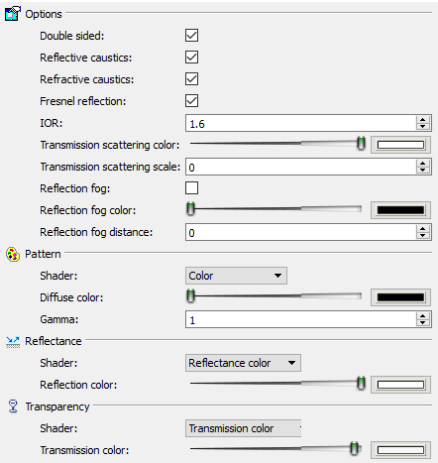


Fig 508 shows the setup of a glass material for this example.

Different situations will require different settings, For example. Diamond like objects require a higher IOR. Also although one would normally have Fresnel checked, there are occasions when it is better to turn Fresnel off.

And obviously colour and transmission will vary. The key items for caustics are ensuring Reflection and Refraction caustic boxes are checked.

We also add a second glass, coloured green for the bottle. Liquid inside the glass and bottle is obviously optional.

Fig 509

The next setup is lighting. As previously stated, the three Physical luminance can be used. Also Sunlight. A quick way is to simply point a spherical physical light at the objects. Giving a spotlight approach, This is simpler than sun, being easier to set the light and direction.

In this instance we will use a Sky / Sun combination. Sun direction is obviously important, so it shines the right direction through the window. And has been applied as a render scene environment, rather than using an object are the source.

Fig 510

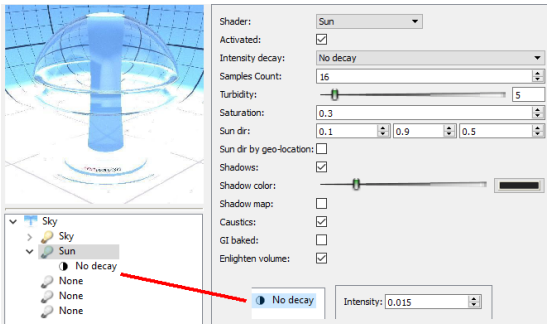


Fig 510 shows the sun setup. A low intensity decay was set. And Caustics was checked. Enlighten volume does not need checking. I just forgot to un-check it.

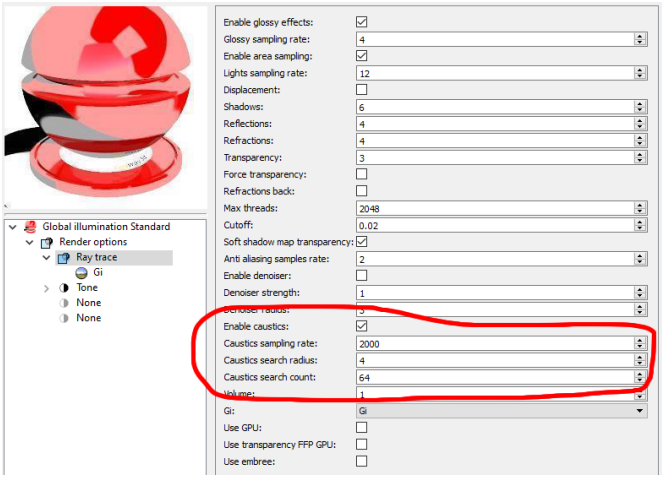
Geo location could be used, but at the time of testing, it would not have been in the correct direction.

The last setup is GI render. Standard Raytrace render could be used, but GI allows extra control over the caustic effect.

As well as checking caustics in material and lighting. There is also a caustic checkbox in GI, to tell the engine to include caustics in its canlculation.

Fig 511 Shows the four caustic settings contained in the GI advance render properties dialog.

Fig 511



The next job is to render, using GI advanced render, and then fine tune the settings.

Fig 512 shows the render and the caustic effect on the table.

Leaving aside the fact that the render isn't photo realistic.

We area concerned about the caustics, which as can be seen are scattered white dots about the table. Making it not look good.

Fig 512

We can fine tune the caustics using the GI Render Raytrace dialog.

In Fig 513. Caustic samples were increased to 3000, and Radius increased to 8.

A few other alterations were done to the render.

Gi shadow increased from 6 to 8
Gi reflection increased to 8
Gi refraction increased to 8
Gi transparency increased to 8

As can be seen (zoom in) the caustics are more blurred and less scattered than previously.

Fig 513

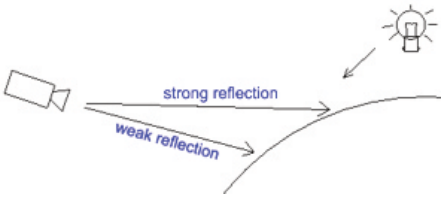
Fig 514

The full render

Fresnel Reflection.

Redway state - *"The fresnel option is a very powerful term that is used o modulate the amount of reflection emitted by the material based on the angle of the viewing direction with the surface. This must be turned on for all glasses materials."*

Fig 12.



When looking at objects or the ocean for example, parts appear more reflective at shallower angles, than looking straight on. Under natural light conditions, (i.e. your not shining a torch at something).

Fresnel alters the way the reflections are seen by the camera, to try and mimic this real life scenario,

In the example below.
Fig 13. Has Fresnel turned off for the wood surface, as well as the glass.
Fig 14. Has Fresnel applied to just the wood, and leaves a strong glare facing the camera,
Fig 15. Has Fresnel applied to just the glass, the glare is removed and reflects more at the edges.
Fig 16. Fresnel is applied to the glass and the wood.

Fig 13

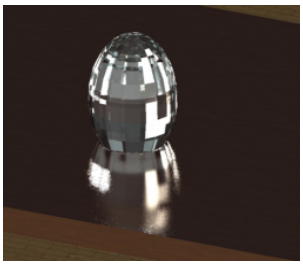


Fig 14

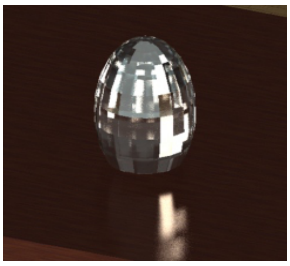


Fig 15

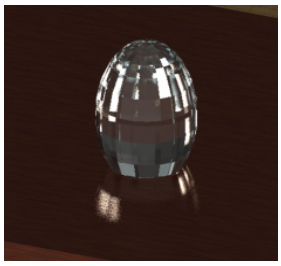
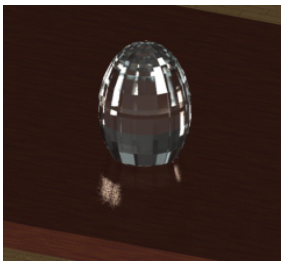


Fig 16



In the figures to right show a large difference in reflectance towards the camera on the circular disk.

Fig 17 has no Fresnel
Fig 18 Fresnel was ticked.

Fig 17

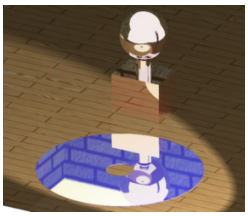
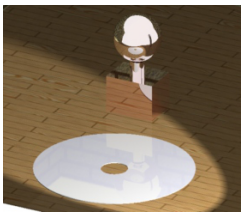


Fig 18



Many materials will have a Fresnel effect, for example wood (assuming it has a coating) glass, plastic, paint etc.

Metals are one substance where Fresnel may not be required.

If high reflectance is required whilst applying Fresnel. It may be necessary to increase the reflectance slider more towards the right hand side (toward white), than one may do with no Fresnel.

This last example simply shows the reflection, at the shallow angle of the camera view with the edges of a sphere, and he steeper angles with respect to the camera and centre portion of the sphere.

Fig 18a shows the sphere with Fesnel turned off. The high reflectance clearly shows the background reflected on the sphere.

Fig 18b has Fresnel turned on. One may need to zoom in, but one can see that the reflection of the leaves is still visible at the edges, but totally diffused towards the centre of the sphere. The tree trunks are also more defined towards the edges of the sphere.

Fig 18a

Fig 18b

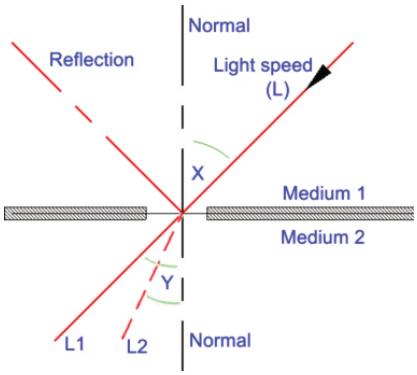
IOR.

The help file states "What makes the transmission effect acting like a magnifying / minifying glass is the Index of Refraction (IOR) of the material. Users can enter the value they need in the IOR field

Important note - I am not a scientist, information about IOR is obtained from the internet and experiments. The information is included here as a guide only.

Refraction index is the difference in velocity between light passing in a vacuum, and passing in any other substance. This difference in index, can cause light to bend when it strikes at an angle to the substance.

Fig 19.



Referring to the diagram Fig 19. Imagine a single beam of light (L) passing through a vacuum (Medium 1). It strikes Medium 2 at an angle. If the two mediums are the same refractive index, the light would pass though at angle (X = Y) and exit at (L1). Thus there would be no bending of the light, i.e. Medium would appear not to be there.

If Medium 1 was Air and Medium 2 was Water and the light entered on the normals. There would be a slowing down of the light in terms of velocity, but no bending. As the angle of entry and exit would both be 90 degrees

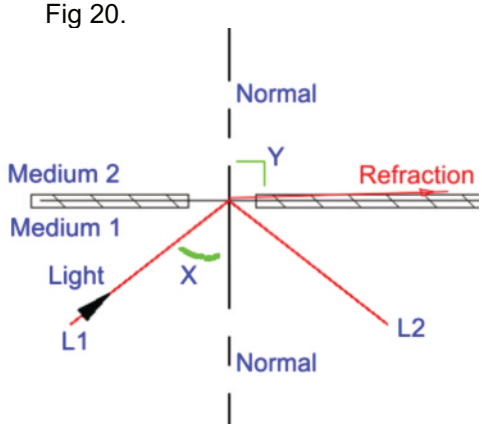
If the light enters at an angle. As it is slowed down, there would be bending of the light (L2). This is due to water having a higher refractive index of 1.333, compared with Air 1.0003. Thus altering the angle (Y).

Substances can have exactly the same refractive index, where the light will not be bent. There is an anomaly in the form of man-made materials, which have a negative refractive (not possible in nature). These can be made to deliberately bend the light, in a predetermined way around an object.

If anyone wants to do some maths, to work out the refraction angle. The formulae is n (index) = Sine(Ø1) / Sine(Ø2). Or from the diagram n = Sine(x) / Sine(y). If x = 35 degrees. Sine(y) = sine(35°) / 1.33 (Index of water). Which translates to, Sine(y) = 0.57357644 radian / 1.33. Which leads to, Inverse Sine(0.38755165). And thus, Y = 22.205 degrees (after conversion). This is quite a bit less than the 35 degree light entry angle.

There is also something called the **critical angle.** This is where the light is passing through a denser medium, and exits into a less dense medium, (Example - going from water into air). There is a certain angle, which causes the light to become trapped inside its current medium. This is called total internal reflection, as it and cannot pass into Medium 2. Think of optical fibres.

Fig 20.



This does not apply going from a less dense one into a dense one. Each material has its own critical angle, for light passes through a dense material, into a less dense material

Taking the picture on the right, Fig 20. Light passing from Medium 1 into the less dense Medium 2. Under normal circumstances, the light is refracted (it can exit).

However when the light angle X, is such that the refracted angle, becomes greater than 90 degrees (Y). No more refraction can take place, and the light is reflected to L2. This angle varies with materials but can be calculated.

The calculation is Sine(Øc) = Medium 2 / Medium 1. Or from the diagram Sine(Øc) = Y / X, (Y and X are the index of refraction for each Medium). As an example. Water to air (taken as 1.0003 and 1.333 index respectively). Transposing the calculation using Inverse Sine, becomes Øc = Inverse Sine(1.0003 / 1.333) = 0.851251 radians. Giving us 48.62613 degrees. Therefore if angle X is greater the 48.6 degrees, the light will reflect, instead of refract (assuming perfect conditions).

Of course light is never just a single beam, it is bouncing around all over the place. In the case of optical fibre it is specially manufactured to cause this total internal reflection.

A last note on objects like diamonds. The reflective index is so great (around 2.4), that it produces a critical angle of just 24 .43 degrees, when compared against air. Therefore as light enters, very little will escape through, and most will be internally reflected. Until such time as it finds the angle to escape, (this can depend on how the diamond has been cut). It is the reason why diamonds appear differently, than other gems in the same light conditions. (*All the above information obtained from the internet).*

OK what has all this got to do with TC.

RedSdk has an IOR setting, so one has to presume its been programmed to take refraction into Account, when calculating the render and indeed in practice it does.

Fig 21. Is a short list from the internet of common IOR values. However this is just a guide, as different internet sites, list different values.

One important point, The IOR setting is not limited to to these values. In TC the setting. According to the help file its max number is 1e+038 (a big number).

Fig 21.

Vacuum	1.0000
Air	1.0003
Bronze	1.180
Copper	1.100 2.430
Diamond	2.418
Glass	1.500
Gold	0.470
Ice	1.309
Iron	2.950
Milk	1.350
Oil, vegetable (50deg C)	1.470
Plastic	1.460
Water (35deg C)	1.325

The IOR setting is tied to other material settings like reflection, Transmission and Fresnel. Indeed altering IOR whilst Fresnel is selected can dramatically alter the material.

Fig 22. Below has medium reflectance, and 1.5 IOR. (Fresnel is turned on in all three images)
Fig 23. The middle image has same reflection but IOR set to 32.
Fig 24. The right image is same IOR but with reflectance turned down a bit. As one can see the IOR can play an important part in Fresnel reflection.

Fig 22.

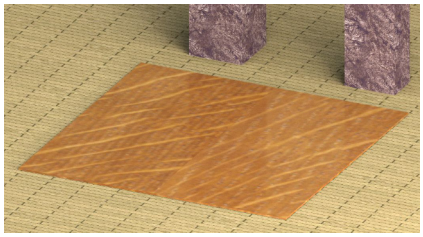


Fig 23.

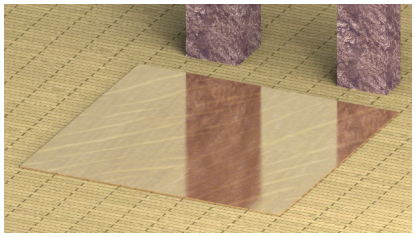
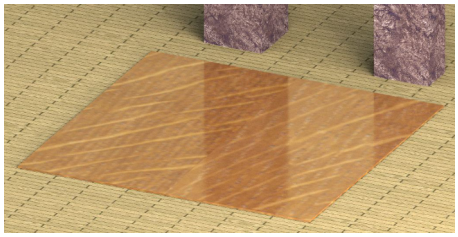


Fig 24.



The last example uses a sphere, and as can be seen, increasing the IOR only, (with all other settings the same). Tones down the Fresnel effect, so more of the reflection is revealed.

Sample IOR values. Note. Figures obtained from various site on the internet.

Aluminium	1.39 - 1.44
Brass	
Bronze	1.18
Chromium	2.97
Cobalt	1.71 - 1.97
Copper	1.10 - 2.43
Gold	0.16 - 0.47
Iron	1.51
Lead	2.01
Mercury (liq)	1.62
Nickel	1.08
Platinum	2.33
Silver	0.18 - 1.35
Steel	2.5
Titanium	2.16
Tungsten	
Agate	1.544
Agate, Moss	1.540
Asphalt	1.635
Calcite	1.486
Chalk	1.510
Coral	1.486 - 1.658
Feldspar	1.525 - 1.539
Feldspar, Labradorite	1.565
Formica	1.47
Ivory	1.540
Mylar	1.65
Nylon	1.53
PET (g)	1.57 - 1.575
Plastic	1.460 - 1.55
Polycarbonate	1.584 - 1.586
Polystyrene	1.55
Rubber, Natural	1.5191
Silicon	4.01 - 4.24

Acrylic glass	1.490 - 1.492
Amber	1.539 - 1.546
Amethyst	1.532 - 1.554
Aquamarine	1.567-1.590
Crystal	2.00
Diamond	2.419
Diamond	2.417
Emerald	1.560 - 1.605
Garnet, Almandine	1.760
Glass	1.51714
Glass, Arsenic Trisulfide	2.04
Glass, Albite	1.4890
Glass Crown (impure)	1.485 - 1.755
Glass Crown (pure)	1.50 - 1.54
Glass Flint (impure)	1.523 - 1.925
Glass Flint (pure)	1.60 - 1.62
Glass, Fused Silica	1.459
Glass, Pyrex	1.474
Jade, Jadeite	1.64 - 1.667
Jade, Nephrite	1.600 - 1.641
Moonstone	1.518 - 1.526
Onyx	1.486
Opal	1.430 - 1.460
Pearl	1.530
Plexiglas	1.488 - 1.51
Quartz	1.544 - 1.553
Quartz, Fused	1.45843
Ruby	1.757 - 1.779
Sapphire	1.757 - 1.779
Sapphire, Star	1.760 - 1.773
Topaz	1.620 - 1.630
Turquoise	1.610
Zirconia, Cubic	2.170 - 2.21

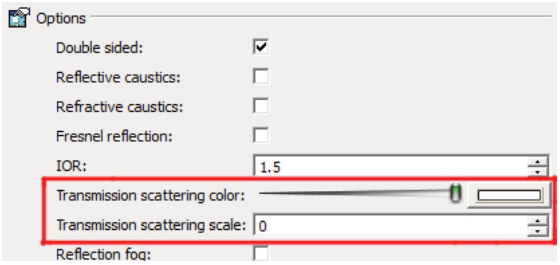
Air	1.0003
Argon	1.000281
Carbon Dioxide (gas)	1.000449
Helium	1.000036
Hydrogen (gas)	1.000140
Hydrogen (liq)	1.0974
Oxygen (gas)	1.000276
Oxygen (liq)	1.221
Nitrogen (gas)	1.000297
Nitrogen (liq)	1.2053
Vacuum	1.0000
Acetone	1.36
Alcohol	1.329
Beer	1.345
Carbonated Beverages	1.34 - 1.356
Citrine	1.532 - 1.554
Chlorine (liq)	1.385
Ethanol	1.36
Fruit Juice	
Glycerin	1,367 - 1.473
Glycerol	1.4729
Honey,	1..484 - 1.504
Ice	1.309 - 1.31
Methanol	1.329
Milk	1.35
Oil of Wintergreen	1.536
Oil, Clove	1.535
Oil, Lemon	1.481
Oil, Neroli	1.482
Oil, Orange	1.473
Oil, Safflower	1.466
Oil, vegetable (50° C)	1.47
Rum, White	1.361
Vodka	1.363
Water (0° C)	1.33346
Water (20° C)	1.33283
Water (100° C)	1.31766
Whisky	1.356

Transmission Scattering Color.

This is an extra absorption term that can be specified for the material transmission. It uses an out-scattering color and an out-scattering scale used to specify the amount of energy being absorbed along the path of a ray for each unit length crossed in the model media volume.

Transmission scattering scale - *Scales the effect of the transmission scattering color.*

Fig 25.



These two values are normally used together. When light passes through an object, it is often deflected, Which in effect means the light is bent within the object. With scattering an extra level of light bending occurs. The light beam is bent at multiple angles as it passes through the object.

The scattering colour changes the light colour within the object, whilst the scale increases the amount of scattering. The scale factor should be changed in small increments, as a large scale can 'black-out' the light. Fresnel also plays a part in how the light enters the object, thus affecting the scattering.

Examples of different settings with and without Fresnel are shown below.

Fig 26.

Fig 27.

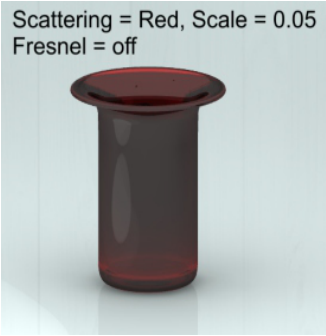


Fig 28.

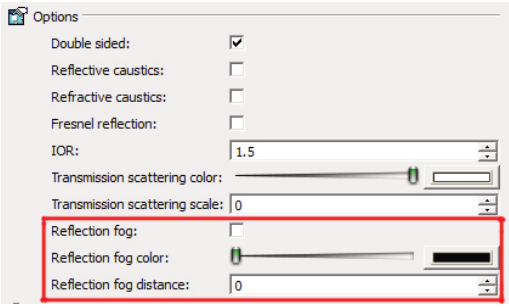
Fig 29.

Reflectance Fog.

This is fade-out term that can be specified or reflections. At a certain distance, the color of the reflection becomes equal to the specified reflection fog color, and any other visible reflection is lost.

Reflectance Fog Colour - *Controls the color of the reflection fog*
Reflectance Fog Distance - *Defines the distance at which the true reflection color gets replaced by the reflection fog color*

Fig 27.



Unlike environmental fog. Reflectance Fog is specific to an object. It only affects the fog applied to the object materials, not neighbouring objects. (if their material fog is turned off).

The fog effect is controlled by a colour. Which could be just a grey scale colour, or green for example. (in the example below). It is also controlled by the distance from the object. Therefore a low distance figure, can fog out the whole object, whereas a high figure can allow the original reflectance to show, before the fog takes affect.

Fig 28.

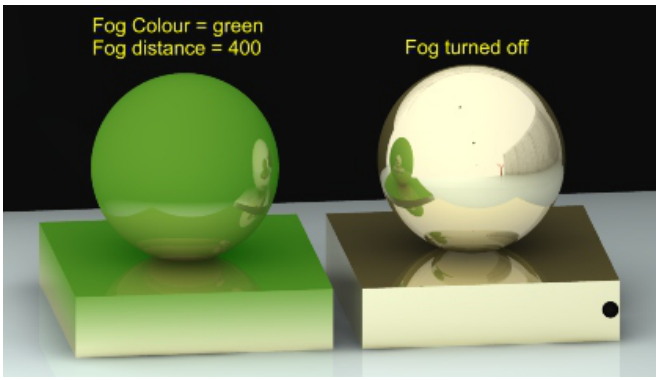
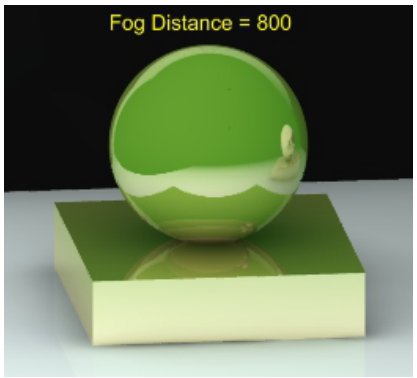


Fig 29.



In Fig 30. And Fig 31. The fog effect was applied to the mirror. It doesn't affect the two objects in front of the mirror. But shows the fog in the mirrors reflection.

Fig 30.

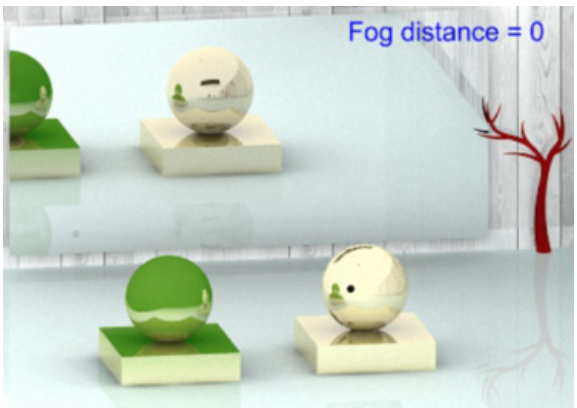
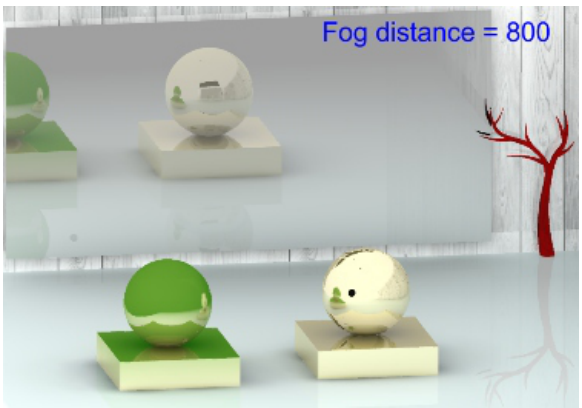


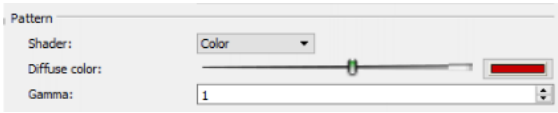
Fig 31.



Pattern.

The Pattern section is used for setting the base colour or texture. The controls differ depending on what is chosen using the shader drop down box. There are two options, Colour and Texture.

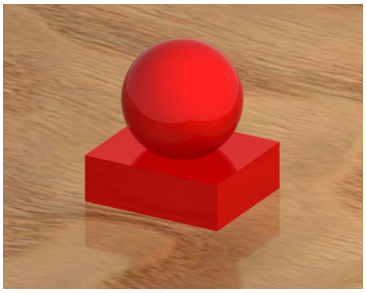
Fig 32.



Color (Colour).

Probably the simplest option. It is used for materials like Metal, Plastic and Glass. It is set by clicking the colour button to the right of the slider. This opens up a colour selector dialog.

Fig 33.



The slider is effectively a diffusion setting. When set to a pure colour, the slider is positioned in the centre. Moving it to the left darkens the colour, moving it to the right lightens the selected colour. The slider is ideal for altering the shade of material.

For realistic materials the colour should be combined with reflectance and / or transparency as required.

In 2019/2020, an option was added with a gamma setting. This allows changes to be made without altering the actual colour.

The images on the right, shows the effect of changing the gamma setting without altering the colour.

Fig 34 is with the Gamma set at 1.

Fig 35 has a gamma setting of 4 (way above normal)

Fig 34.

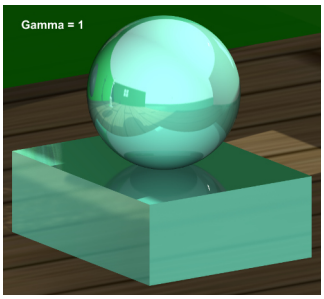
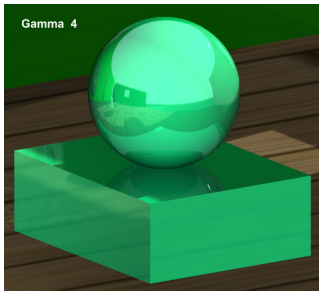


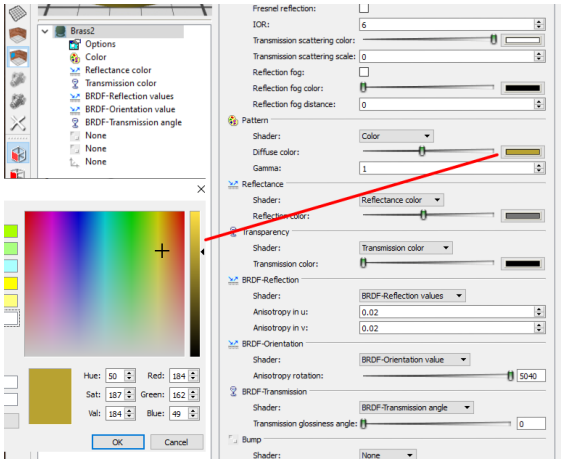
Fig 35.



Fig's 35a and 35b, are simply an example of using the colour, with some added reflection. The final setting is BDRF reflection to control if he object has some 'grain' or not.

Fig 35a.

Fig 35b.



The final example Fig 35c and 35d, controls the liquid inside the glass.

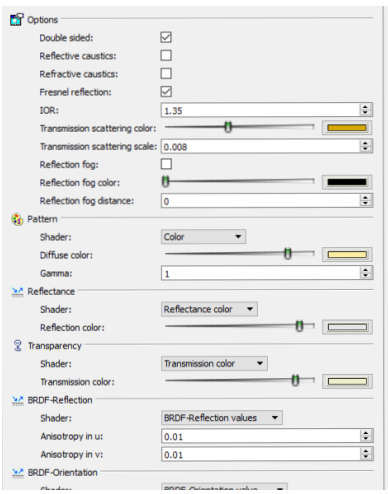
It uses many settings, but the base colour is set using the pattern colour shader.

The remaining settings, just change certain aspects of the way the colour is displayed.

These are discussed under their own heading.

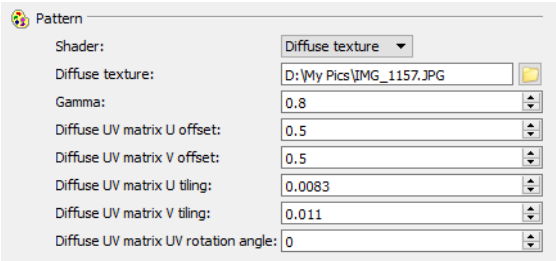
Fig 35c.

Fig 35d.



Pattern - Texture.

Fig 36.



Changing the shader in the drop down box to texture, brings up more options than a simple colour..

First a note regarding the textures. A Texture is simply a picture. In bmp, jpg, tif, or png format. One cannot alter the saturation, nor contrast of the original image within TC. Therefore if a picture requires altering, it would require changing in a paint program. It can occasionally be trial and error, to get the picture to match the scenes lighting.

In TC 2019 or TC 2020 an option was added to allow changing the gamma of an image.

The second thing to remember, is that if a small texture is to be used on a large area, the program will need to stretch the texture to fit, or tile the texture many times, causing repeated patterns. Therefore if the image resolution is too low, for example 256 pixels x 256 pixels, it may produce poor results if used on a large area.

Fig 36a.

Diffuse texture (File name). This is the full path to the image being used. The actual image is not saved with the file, only the file path is saved. Click the folder button at the right, to bring up the file location dialog box.

Gamma. Introduced in TC 2019 or TC 2020. This allows for some correction of the image. The default is 1. Fig 37 shows the default. Fig 38 increased to 2.2

By decreasing the gamma, the image will lighten. By increasing the gamma, the image darkens. Whilst not full control, like Saturation or contrast. It at least allows some control, if the image does not match a particular lighting setup.

Fig 37.

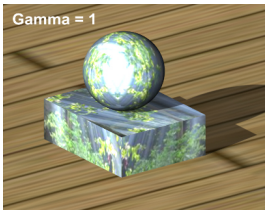
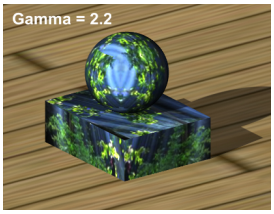


Fig 38.



Offset u. Offsets the texture horizontally by one drawing unit.
Offset v. Offsets the texture vertically by one drawing unit

Offset essentially dictates where the texture is placed on the object, the offset's move the texturet, and allows one to precisely position the texture.

Values are entered between 0 and 1. A value of 1, moves the texture once, in the x or y direction. Though in practice there is no point in using a value of 1. Often then these values will either 0 or 0.5. Though other values may be necessary, depending on the scale U / V settings. The value also depends on the image being used, For example, most 'built in' textures don't matter if the value is 0 or 0.5. Whereas, user supplied textures like a picture which needs to be in a frame, may need precise positioning.

In the examples below, the object size is one drawing unit by one drawing unit. Whether they be inches, mm etc. it does not matter. NOTE. The following assumes no 'Wrapping' selections have been made.

Fig 39.

The image in Fig 39. Is the original png image.

Fig 40.

Fig 41.

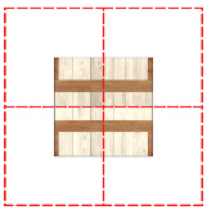


Fig 40. Shows the texture loaded with offset u and v set to '0'. As can be seen in Fig 41. (far right). The offset 0 value uses 4 images to produce the final render.

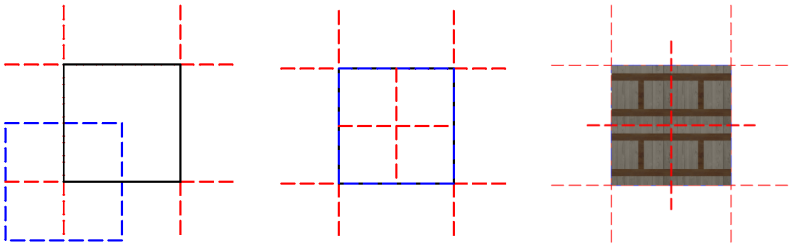
Therefore for a single texture on a single drawing unit, the offset needs to be a half (0.5) in both offset u and offset v. To display correctly.

An easy way to think of it is. If the number of tiles required is Odd (1,3,5 etc), the offset is 0.5, if the number is even the offset is 0.
But, obviously. If one wished something like one and a half tiles wide, the offset would have to altered accordingly.

An example of wanting two tiles wide and high, is shown in Fig 41a. Because of the wrapping it is essentially the same as 41b.

Therefore we do not need an offset value, so we leave offset at '0'. The result can be seen in 41c. which shows two tiles wide and high.

Fig 41a. Fig 41b. Fig 41c.



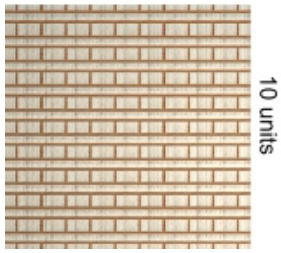
NOTE. These are drawn as a representation. I don't know the exact way it is programmed.

Scale u - scales the image (texture) in the horizontal direction by one drawing unit.
Scale v - Scales the image/Texture in the vertical direction by one drawing unit.

The scaling can seem counter intuitive. Normally when one thinks of scaling, the larger the number, the larger the image.

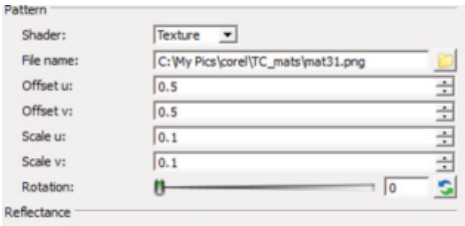
However lets say you have an object 10 mm x 10 mm. Because both scaling and offset are per drawing unit (example 1 wrap per mm). With the scale u and v set at 1, the texture would wrap 10 times in both directions. As in Fig 42.

Fig 42. 10 units



Therefore in order to get one texture at the correct size. We must take the number of tiles we wish, which in this case would be (1). We divide it by the number of scale units in the object (10). This makes 1 / 10 = 0.1. We also need to take the offset into account, as we want one tile, offset u and v = 0.5. The result is the same as the original image in Fig 39.

Fig 43.



Obviously in the real world things aren't exactly 1 unit or 10 units. For example, if the object was 42 mm x 31 mm, the scale would be scale 'u' 1 / 42 = 0.0238 and scale 'v' 1 / 31 = 0.032248. Again we with one texture, so the offset is 0.5. Shown in Fig 43b.

The result is that the texture is stretched to fit onto the face of the object. As in Fig 43a

Fig 43b.

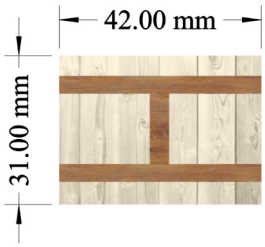
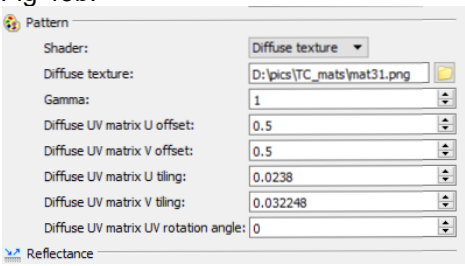


Fig 43b.



This brings up a point about aspect ratio, for both the image and object. For the texture to look correct, without undue stretching. Ideally the aspect ratio's of both the object and image, should match, or be pretty close. Unfortunately life isn't that simple. And many times a compromise will have to be made. Either by altering the image, or accepting that some stretching of the image will occur.

Textures should be created with care if they are to be wrapped multiple times. Too much difference between top / bottom. And left / right. Will show seams when rendered.

Fig 44.

Fig 44. On the right, was wrapped 3 times horizontally to prevent too much stretching. The repeat pattern and the seams are clearly visible.

Re-touching the image in a paint program may be necessary.

Some images are more awkward to re-touch, but correcting things like this, is well worth trying. Fig 45. was altered in a paint program, and wrapped onto the same box in TC as the last example. The alteration allowed for the image to wrap just once in TC, to avoid the seams.

However the stretching of the grain may be unacceptable.

Therefore an alternative it to tile the image, after it has been altered in a paint program,

Fig 46. is the same image, wrapped twice horizontally. Because the left and right sides of the image were altered to match up better. When the texture is wrapped multiple times. The seams are a less evident.

Fig 45.

Fig 46.

Rotation.

This setting rotates the applied texture.

The setting originally had two modes, with one having up/down arrows (Fig 47) to rotate 0 to 360. Though we can simply type in a figure, as it is simpler than messing with the arrows. The second was a slider mode, which is a percentage and goes from 1 to 100. (Fig 48). In later versions, the slider mode was removed. Leaving just Fig 48.

Fig 47.

Fig 48.

Mostly, one would be using 90 degree increments. To align things like wood grain, so the grain runs in the desired direction.

A note regarding rotation's link with scaling and offset. If we take the original object of one unit, and image tiled with scales set to 1 unit, offset 0, and apply 45 degree rotation. The image is not rotated around the centre of the object.

As can be seen by the Fig 50. This may not be what was required.

Fig 49.

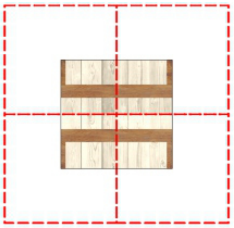


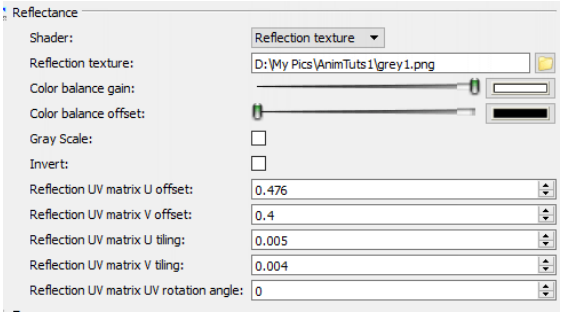
Fig 50.

When rotating at angles other than 90 degrees, trial and error may be required. As the actual calculation used is unknown to me.

Also. Care must be taken if using wrapping tab - texture space. As this also has the ability to rotate the image, Which can cause confusion if both Rotation, and the texture space settings are used at the same time.

Reflectance.

Fig 51

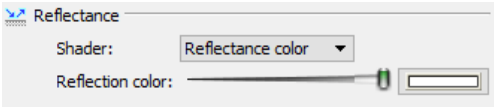


The reflectance tab houses various options with regards to how an object reflects light.

Shader - this has two options reflectance colour and reflectance map, so we will start with reflectance colour.

Reflectance colour is chosen by clicking the coloured button on the right. For a majority of circumstance this should be left as grey scale, (black to white). Moving the slider all the way to the right (white), gives mirror like finish. However, It is also linked to other settings, like Fresnel.

Fig 52



It is simple to change the reflectance by moving the slider. Note Fresnel was turned off for the three examples below.

Fig 53. is the plain object.

Fig 54. had the slider moved to the right. So the colour is nearly white. Giving a mirror like finish.

Fig 55. the reflectance slider is about midway, Allowing the original colour to show.

Fig 53

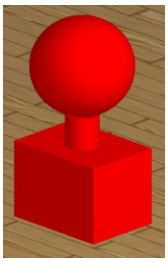


Fig 54.

Fig 55.

With the three figures below, the reflectance colour was changed. Fresnel was turned off for the first two images.

Fig 56. Was change to red. Giving a reflectance which is quite vibrant.

Fig 57. The reflectance colour changed to yellow which completely changes the colour. Some colour combinations will saturate the main object colour. In the case of a red pattern colour, yellow and magenta will saturate. Whilst green or blue will still show some red.

In Fig 58. The reflectance colour was change to green. Along with Fresnel turned on.

Fig 56.

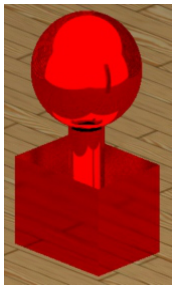


Fig 57.

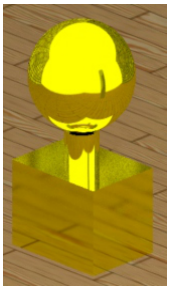
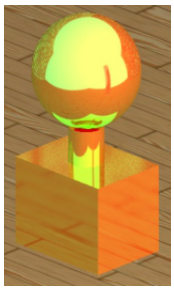


Fig 58.



As stated, normally you would leave this at black to white, so that it does not affect the pattern colour or pattern texture.

Reflectance Map

Reflectance maps use a texture (picture), as the basis for reflection, and use an image file from ones computer. Simply click the folder icon at the end of the file name box.

Like the colour shader. Reflectance maps use the RGB values to determine reflectance, with one important difference. Because they can be a pattern, different parts of the texture will reflect light at different rates.

Fig 59. uses a reflectance map, (Fig 60). The map is applied to the pole. This makes it quite easy easy to add decoration.

Fig 59.

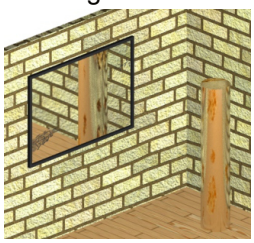
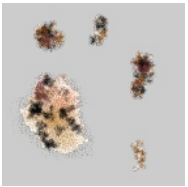


Fig 60.



An advantage of using reflectance map over applying the the image as a texture, is that one still has control over the diffusion (grey-scale colour) in the texture tab.

In this example, the base object was simply a grey colour (this method works best on a grey-scale object). Whilst we use different methods, we will add an image as a reflectance map. Because light will reflect differently depending on the colour, black being non reflective, white being fully reflective, adding the map dramatically affects the appearance.

Fig 61. is the original object. In Fig 62. a reflectance map was added instead of a pattern texture.

Fig 61.

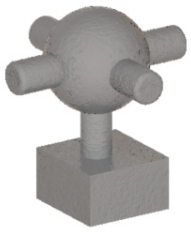


Fig 62

The next two examples, show the difference between using a coloured image for the reflectance texture, and using a grey scale image.

The first example uses a gradient image with colour as shown in Fig 63. One thing to note. When the texture is applies as a reflectance, the reflected colour is inverted. One can easily find the inverted colour in most paint programs. Or one can calculate the inverted RGB values.

What this means for our image, is that. Although the majority of the texture is purple, the inverted colour is mostly green. The texture gradient fades towards the edge, creating a more original shiny reflectance.

This effect can be clearly seen in Fig 64.

Fig 63

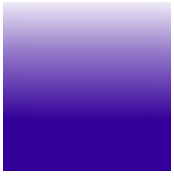
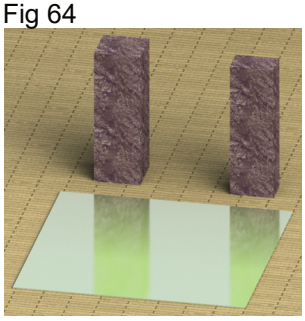


Fig 64



To show the difference, we use the same settings, but substitute a grey scale image. Which is shown in Fig 65.

Fig 65

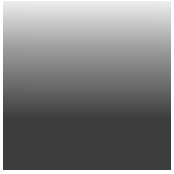
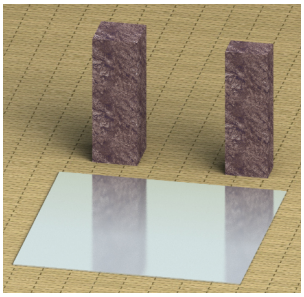


Fig 66



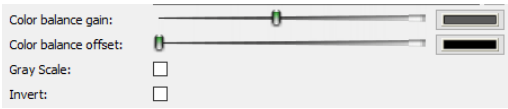
As can be seen in Fig 66. The reflectance is more like what one would expect. The texture causes a more faded reflection at one end, and shiny at the other.

The Offset, Scale and rotation settings, are the same as for Texture Map discussed earlier.

New in TC 2019 or TC 2020 (unknown which).

Color balance gain.
Color balance offset.

Fig 67



The Redway documentation states. *"Modifies the image pixels. This method performs a per pixel image operation: Each pixel's color is multiplied by iColorGain and added to iColorOffset. Saturation occurs for all clamped image formats (RGB, RGBA, BGRA, L8 and ALPHA)."*

In the examples right. It uses the same grey scale gradient image as above. For the reflectance texture.

The colour balance gain is by default set to white. Moving the slider to the left (towards black), effectively darkens the lighter pixels of the image.

Because the reflectance image gets darker. The reflectance of the object is reduced. Compare Fig 69 to the previous image in Fig 66 to see the difference.

Fig 68

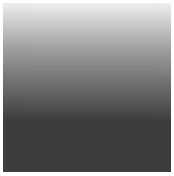
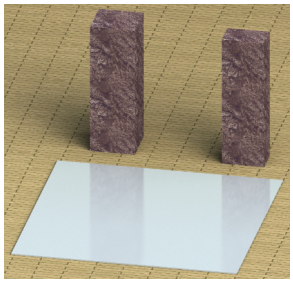


Fig 69



The colour balance offset. Add an extra control to the colour calculations. It has a default setting of black. When the slider is moved to the right (towards white). The reflectance is less diffused.

Fig 70

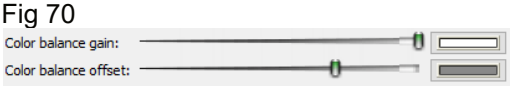
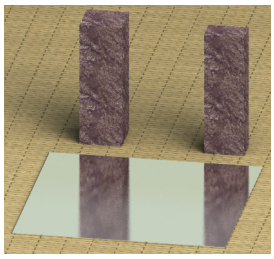


Fig 71



Generally one would use the two sliders above in unison to obtain the desired effect.

Gray Scale.

This is a simple switch which is either on / off. When ticked it takes any coloured radiance texture and creates a grey scale equivalent. The same and would be accomplished in a paint program.

The new grey scale image is then used to determine the reflection for the object.

Fig 72

Gray scale off

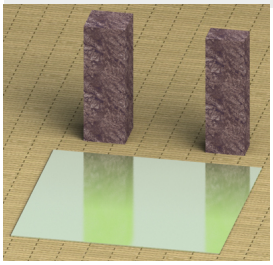
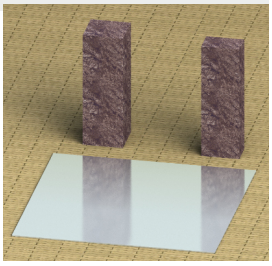


Fig 73

Gray scale on

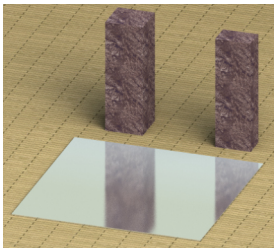


Invert.

This inverts the colours or grey scale of the texture. In the examples on the previous pages, Darker areas or colours, created a less reflective material.

The invert switch. Reverses this reflective parts of the mage. With invert turned on. A grey scale image, would have the opposite effect to Fig 69 (on the previous page). The darker areas now appear more reflective. With coloured images, Using Invert, will allow the original colour to show in the reflection..

Fig 74



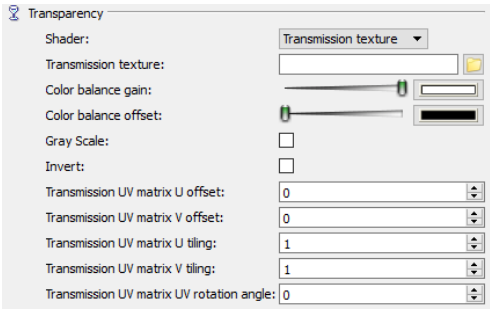
Reflectance UV matrix values

These are used in the same way as Pattern texture settings. They are used to set the amount of tiles required and the offset values, so the texture is applied as required.

Other things affecting reflection.
In older versions of RedSDK there was a simple glossiness setting, and an anisotropy setting. In later RedSDK versions there is a separate section called BRDF Reflection. This contains the control for Anisotropy.

Transparency

Fig 75 Dialog box from TC 2020



Like reflectance the first option is Shader (colour or map), and make either all or part of an object transparent.

Shader

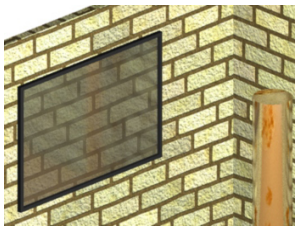
Transparency Color (Colour). this is a colour slider with black (0%) being non transparent, and white (100%) transparent,

Note - if one has a high reflectance, or using a reflectance map, this can overpower the transparency. If this is the case either turn down the reflectance or check the reflectance Fresnel option.

In the pictures below shows some tinted glass, Fig 76. on the left has reflectance Fresnel checked, and a medium transparency colour. Fig 77. On the right has Fresnel turned off, and a low reflectance set. This allows for a tinted look, but still have some reflectance, (Note - this actually goes against what Redway advise - but whatever works is the right way).

Fig 76

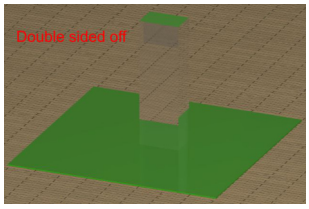
Fig 77



Another things needs mentioning and that is the double sided setting in the options tab. With transparent objects, double sided should ideally be turned on (ticked).

There are the occasions, when turned off may be preferable. But some transparent objects will not render correctly with double sided turned off.

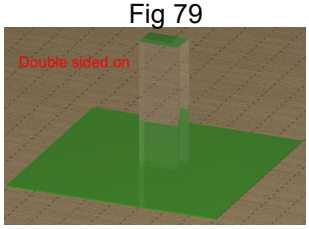
Fig 78



In Fig 78. The column has some reflectance and a lot of transparency. the glass 'double sided' is turned off, this results in the inside of the glass not being rendered, therefore a flat look and incorrect refraction.

With Fig 79. 'double sided' was turned on, therefore the column receives light through the glass, and also reflects the back side of the glass giving a more defined column.

Fig 79



One last thing to consider. Is IOR. As mentioned before. One is not limited to actual IOR values.

Whilst mostly, the IOR one sees is between 0.5 and 2.5. that's not to say you have to stick between these limits.

The colour slider, for most clear glass, should be left at grey scale (black to white). But ReDSDK does allow for the colour to be changed.

Fig 80

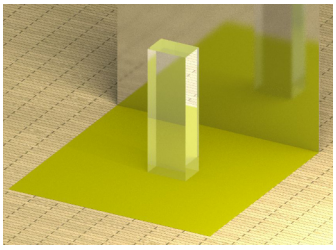


Fig 81

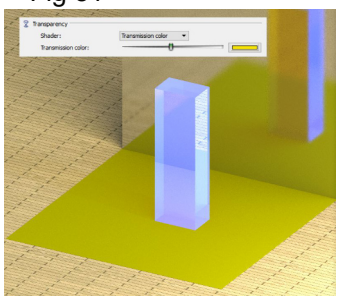


Fig 80. Is a grey scale colour and acts more like normal clear glass. The green tints are simply reflection.

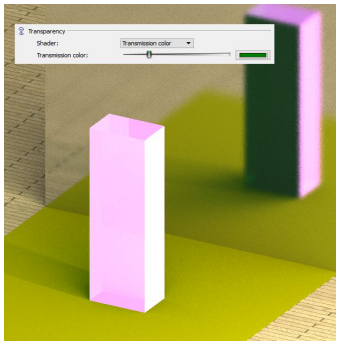
Fig 81. has had the colour changed to yellow.

Hmm. That's not yellow. When a colour is used on transparency, the colour inside is inverted. And the invert of yellow is blue.

Looking at the reflection, we can see the reflection of the back face is actually yellow.

If we wish a specific colour inside the object, we can either. Change the object colour in the pattern section of the material dialog.

Fig 82



Or we can choose the invert colour for the transmission colour. For example, if we wish magenta, we could use its invert, of dark green.

It is personal preference. But for simplicity it is easier, to just to keep the transmission colour as a grey scale colour.

Shader - Transparency map.

A transmission map allows for different parts of the object, to allow transmission at different rates. This allows a variety of effects to be achieved.

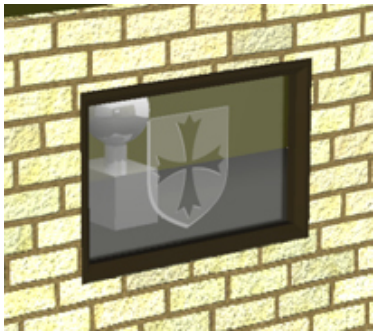
The map is simple an image, file. The settings are similar to the pattern. And the same as for reflectance. Therefore the info below does not go into every setting.

In the first drawing we will put some etching onto the glass.

Fig 83

Fig 83. Shows the image used for transparency - Note the blue background is not part of the image, its been put here to show the white part of the texture.

Fig 84



By using a texture map, one still has control over colour and reflectance to get the desired render, one thing to keep in mind is that RedSDK uses white as a transparent colour. Darker grey can produce a frosted appearance. As shown in Fig 84.

After selecting the required image, adjust the scale and offset to suit.

One thing to note is the IOR setting, because we don't want any distortion from the glass, it has been set to 1 (air - well a vacuum actually), adjust reflection and colour to suit.

Color balance gain. Color balance offset. Greyscale.

Operate similar to Reflectance. They affect the whole image / transmission, not just any highlighted parts. Used to adjust the image being used as the transmission texture. See reflection on page 17.

Invert.

Fig 85

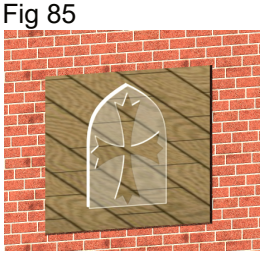
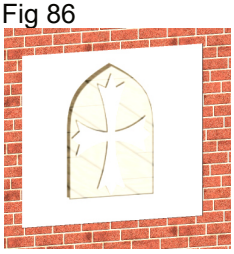


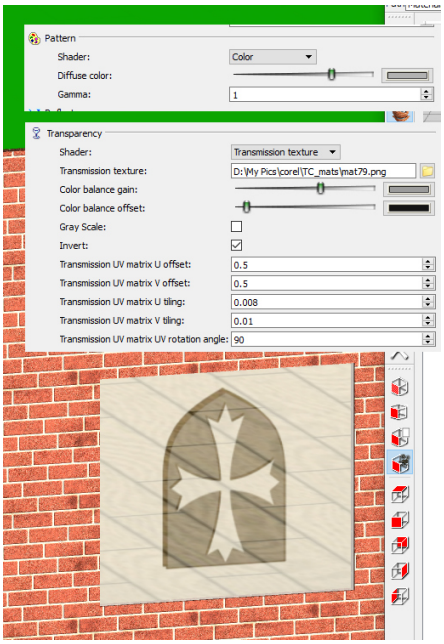
Fig 86



When using Invert, one needs to remember it affects the whole image, This can produce unexpected results. Fig 85. Is the original transmission image. Fig 86. Shows the transmission after invert is selected. As can be seen the image becomes a white-out.

One way to alleviate the problem, without altering the image in a paint program. Is to the alter the pattern colour, along with the colour balance. As in Fig 87.

Fig 87



Because we used invert, any frosting would occur on the cross and outside the shield. Whereas originally only the shield was frosted.

The colour balance is used to tone down then 'white-out' parts of the inverted image. After some tweaking, we should be able to get the shield to show clear,

Tree example.

The next example is a classic transparency used for trees and plants. The setup can be single sheet or multiple, in this example we will use two,at 90 degrees to each other.

It is a good idea to ensure that the aspect ration is equal for the image and object, though not essential. There are many ways to check this, one way is to divide the width of the image by the height, or vice versa, or use the object dimensions.

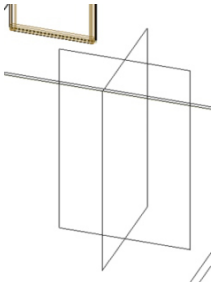
1) *Size the object to the size of the image aspect ratio.*
Image W314, pixels / H 500 pixels = 0.628.
Object W 15 units, Object H = 15 / 0.628 = 23.8854 units

2) *Size the object to the size of the image aspect ratio.*
Image H 500 pixels, / W 314, = 1.59236.
Object H 24 units, Object W = 24 / 1.59236 = 15.072 units

3) *Size the Image In a paint program, to match the object size.*
Object W 15 units / H24, = 0.625.
Image width = 314 pixels. Image height = 314 / 0.625 = 502.4 (502) pixels

4) *Size the Image in a paint program, to match the object size.*
Object H 24 units / W15, = 1.6.
Image Height - 500 pixels, image width = 500 / 1.6 = 312.5 (313) pixels

Fig 88

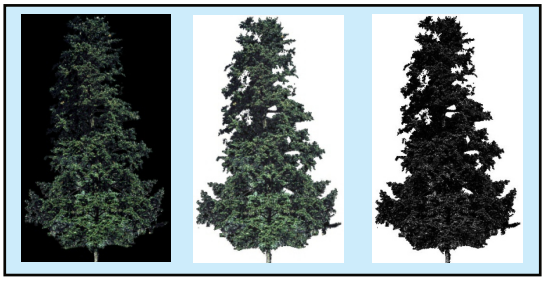


Whatever method used whether one of the above, or a free conversion tool. The point being, the more accurate the aspect matches, the better the results can be. Because RedSDK will not need to stretch the image to fit.

Method 1. Draw a single line, the correct width and extrude it to the height. Select it, activate make copy and rotate 90 degrees, as shown in the Fig 88 on the right.

Method 2. (Platinum). Draw a rectangle of the appropriate size. Select it, and then select the Surface from profile tool. This creates an ACIS surface object, perfect for this situation. Then rotate a copy as in method 1.

(A) Fig 89 (B) (C)

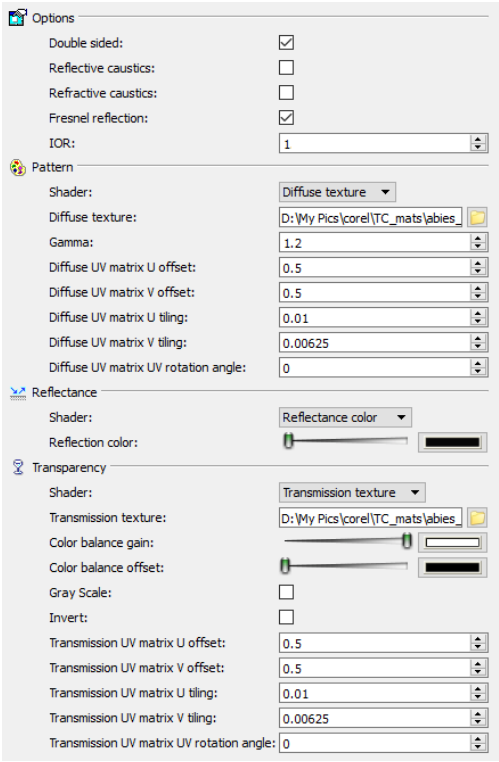


To use a normal Texture map, for this we need two images, one for the texture and a second black and white for the transparency to clear the background.

RedSdk uses white as the transparency colour. Of the three trees in Fig 89. . The third (C), which is a black(ish) tree against a white background, is used for the transparency. Be careful about jpg images, the compression of jpg can add stray coloured pixels which will affect the transparency. This is same as if one was using masks on jpg images in a paint program.

In Fig 89, in theory is doesn't matter whether A or B is the pattern texture, but in some circumstances (A) will give better results. This was due to the paint program causing mismatch, when turning the image black and white, for the transparency. This is easily possible when using jpg images.

Fig 90



The next step is to simply set up the texture map and transparency map in the material dialog.

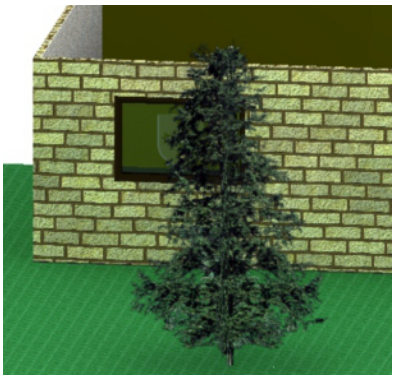
The offset and scale should be exactly the same in both pattern texture, and transparency.
Select the appropriate image file ensuring the black and white one is in transparency.

We don't really want any, (or at least very littler reflectance). Turn on Fresnel reflectance, and set IOR to 1 to prevent any distortion. Ensure double sided is checked in options.

Fig 90. A shows the setup used for the tree render.

Fig 91. Shows the finished render.

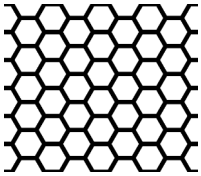
Fig 91



One could also try adding a bump map. But in practice, it doesn't add much to his type of image, and will just complicate things, so I would not bother. Feel free to experiment.

Transparency maps allows for a wide range of effects, the image below uses a transparency map, to add a link fence to the tree line. The tree image has been deliberately lightened to show off the fence better. And the fence darkened. So its definitely not going to look realistic.

Fig 91a



The setup is very simple, and consists of a pattern colour, which has been st as dark green.

The Transparency map is simply loaded as a transparency texture, and scaled to suit. There was no need to add any offset values.

Fig 91b

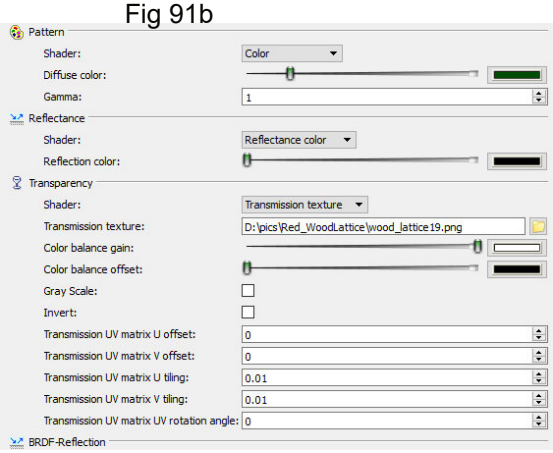


Fig 91c

Transmission and light.

Light passing through glass is generally handled by simply giving the glass a transparency colour setting. More often than not, it would simply be done using the grey scale, i.e. moving the slider, whilst the colour is set to black / white.

Fig 91d

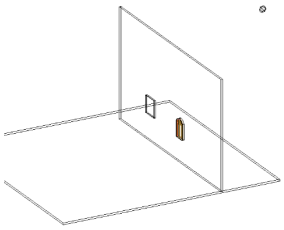
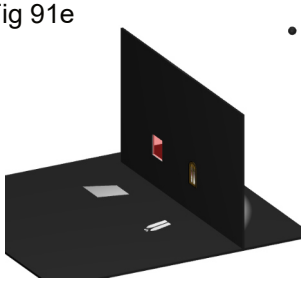


Fig 91e



To get the light trough a 'red' window, we can change the transmission colour to red.

As can be see far right, the red glass now shows red on the floor.

Fig 91f

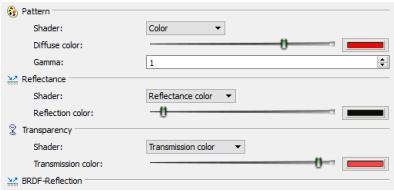
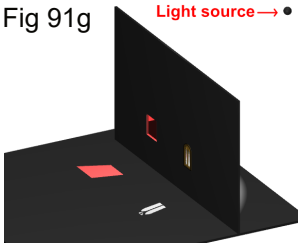


Fig 91g



We are not limited to plain colours though. We can do the same with a texture.

By applying the same texture for both the pattern and the transparency, we can get the same image to display for both the glass and the floor.

Fig 91h

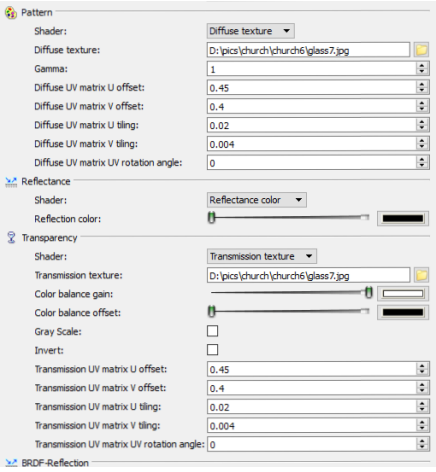


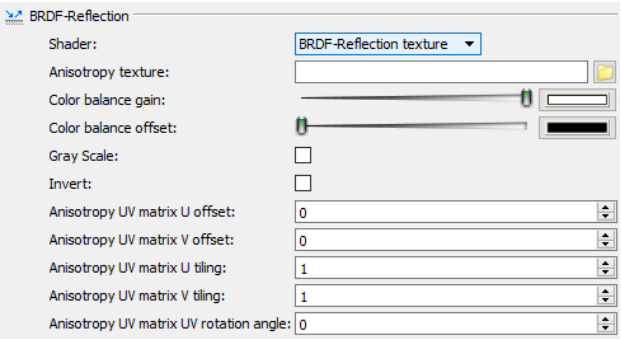
Fig 91i

As can be see on the left, we can simulate light entering a stain glass window, and being projected on the floor. The light source is the same as Fig 91g.

All that is missing is the light rays passing through the air. See volumetric info on page 82 for an example of creating light rays.

BRDF - Reflection.

Fig 92



This controls the Anisotropy values. This effectively alters the glossiness of the reflection applied to the object.

The Bidirectional Reflectance Distribution Function, is a function that defines how light is reflected at an opaque surface.

The Anisotropy / Glossiness settings, were part of the normal reflection section in Pre-2017 dialogs.

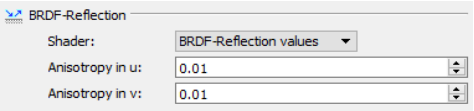
In 2017 Glossiness has been incorporated into Anisotropy. The glossiness parameters control how much the reflections/transmissions are blurry. Depending on the chosen glossiness values, the results may vary a lot:

Glossiness is achieved by increasing the "Anisotropy in U" and "Anisotropy in V" parameters. If both parameters are identical, then, the reflection is isotropic. If the parameters have different values, the reflection is anisotropic. Like on a CD-ROM surface or on hairs and furs.

Like the normal reflectance settings, there are two modes, Reflection Values, and Reflection Texture.

BDRF Reflection Values.

Fig 93



The first examples are using both figures Isotropic (both U, and V, the same). As can be seen, the higher the number, the more grain is present in the reflection. This may see odd, as the effect is called glossiness. But for a mirror finish, keep the values low.

Fig 94 U = 0.001 V = 0.001

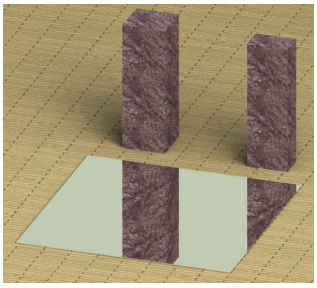


Fig 95 U = 0.01 V = 0.01

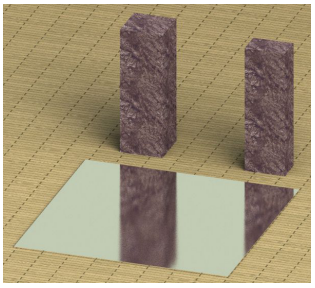
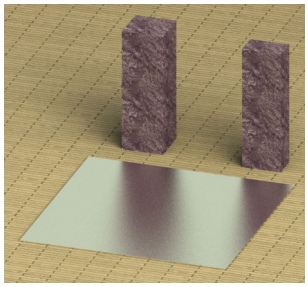


Fig 96 U = 0.1 V = 0.1



The next test uses anisotropic effect, by using different numbers for each U, V, value

Fig 97

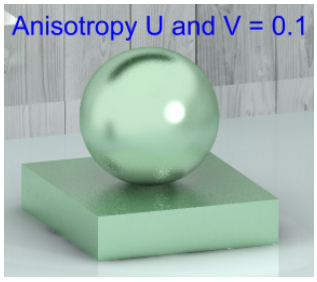


Fig 98

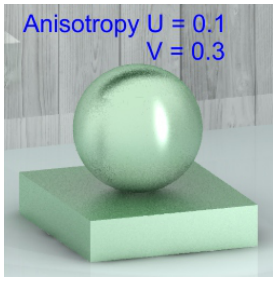


Fig 99

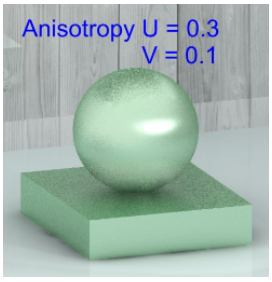
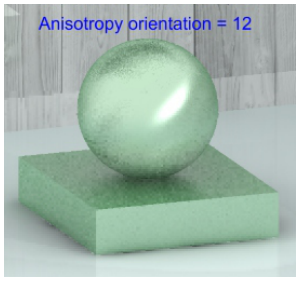


Fig 100



BDRF Reflection Texture.

Like the other sections, in later versions of TC / RedSDK. Colour balance, Grey scale, and Invert have been added, These work the same as previous sections. i.e. pattern texture, reflective texture etc. See reflectance on page 15 for further info on these settings.

Anisotropy UV matrix U offset: - Controls the translation of the texture along U

Anisotropy UV matrix V offset: - Controls the translation of the texture along V

Anisotropy UV matrix U tiling: - Controls the number of texture repetitions along U

Anisotropy UV matrix V tiling: - Controls the number of texture repetitions along V

Anisotropy UV matrix UV rotation angle: - Controls the rotation angle of the texture

This addition in 2017, applies a texture in the same way Reflectance texture works, however the texture is used to vary the glossiness settings, a variable glossiness is applied to the lighter areas (white = high glossiness = Grainy) and glossiness reduced or omitted from the darker / blacker areas, the tiling, offset etc. operate the same as standard reflection texture.

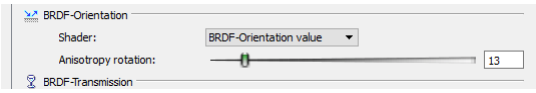
Using a texture, one can create a glossy / frosted look. With black showing as reflective and white as frosted effect. As shown in Fig 101.

Fig 101

This works pretty much the same as using a reflectance texture, except the anisotrophy effect produces a grainy finish.

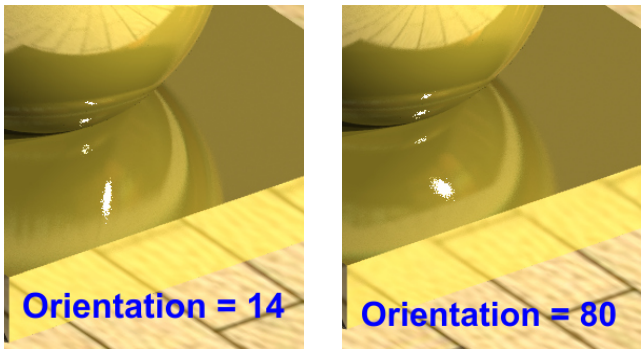
BDRF Orientation.

Fig 101a



As previously there are two shaders. Colour and Texture. Each shader reveals the same settings as previously.

Fig 101b



As to the operation. It does not appear to make a lot of difference to the finished render, for me.

Feel free to experiment, as it may require a specific set of circumstances.

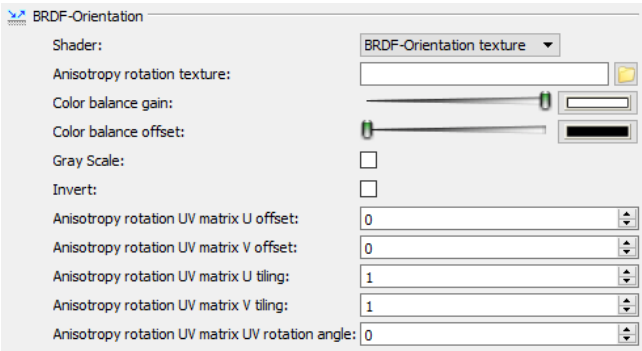
Rotation controls the anisotrophy orientation, shown up on objects.

Orientation texture does not seem logical.

It affects the direction of the grain or anisotrophy effect. But not the objects render as a whole.

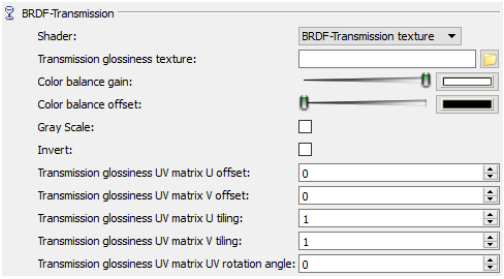
And indeed applying a texture does changen the shape of the effect. But I have no clue how to control orientation. Therefore, for me. Using a texture is unpredictable.

Fig 101d



BRDF Transmission - Controls the glossiness of the transmission (Transparency)

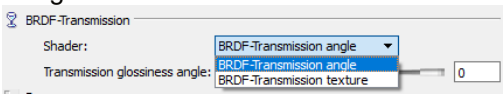
Fig 102a



As with most controls. There are two modes. Simple transmission glossiness angle. And transmission texture. Both will increase / decrease the glossiness (grain) of the transparency.

Prior to 2017 this setting was included as a single transparency option,
In 2017 it allows for a separate glossiness texture to be applied.
In 2018 the text was changed from glossiness to diffuse. But the function is the same.
In 2019 or 2020. The dialog was changed, to include colour balance etc.

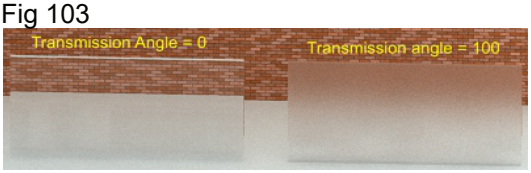
Fig 102b



The sub values, offset, tiling (scale) and rotation, operate the same as other sections in the material set-up for example transmission texture.

Fig 103,. Shows the affect of altering the transmission angle. Giving a frosted glass look.

Fig 103

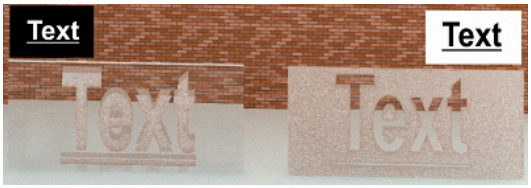


BRDF can be used to add a sort of frosted appearance using a texture. Where black or dark areas are transparent, white or light areas being grainy. This is similar to the etched shield we did in the normal transmission section.

To use, turn up the transparency colour to white.

Then apply the texture as a BDRF transparency texture. Apply the desired offset and scaling. Whilst very similar to adding just a transparency texture, using this, still allows the transparency colour to be altered. Which would not be possible if the image was applied as a transparency.

Fig 104

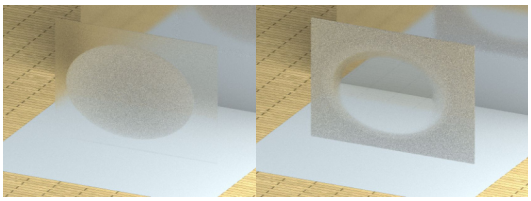


With 2019 or 2020. The extra controls for colour balance etc. allow fine tuning,

The invert function allows for the image to be inverted, when creating the glossiness settings.

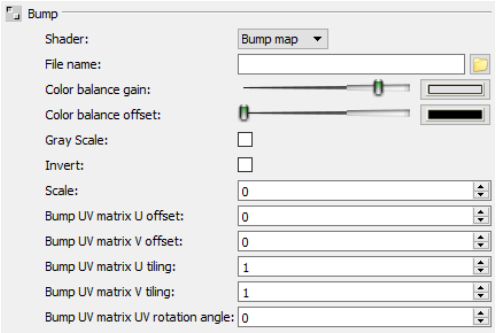
Fig 105a. Shows the original BDRF transmission. Fig 105b Shows the transmission after the invert box is ticked.

Fig 105a Fig 105b



Bump Map

Fig 106

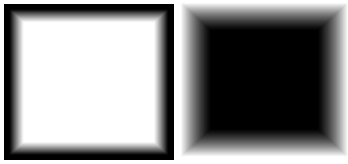


A bump map is a purely visual appearance, of a raised or lowered surface. Unlike displacement maps (discussed later) it does not alter the geometry. Instead it calculates shadows based on a bump map image.

When rendered, the program alters the areas, where it finds the difference between light and dark. Which is much quicker than actually altering the geometry (as displacement maps do).

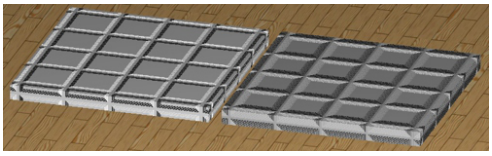
Bump maps are 2D pictures, which are calculated based on the RGB colour value. It is calculated to show white areas (RGD 255,255,255) as high areas and black (RGB 0,0,0) as low areas. With varying degrees in between these values.

Fig 107



Although any image can be used I prefer grey scale images (RGB) as its easier to judge which areas will be high and low. Users have a choice of using them on top of a plain colour, or over a texture map. You can use the same image for texture map and bump map. However it depends on the image. If the texture map has low contrast. It will likely make a poor bump map.

Fig 108



Why use a bump map. Two reasons. First. Many things are just not practical to draw. Take a piece of wood. If you wished to roughen up a piece of wood, then drawing all the wood fibres is not a job anyone would relish.

The second is speed. Taking the same example, applying a bump map takes minutes, drawing wood fibres possibly hours. Or in the case of human hair - probably days to draw every strand.

What images can be used. The best images for bump maps have good contract and shading. Whilst two colour black and white can be used, the results are often disappointing, rgb grey-scale or rgb colour give good results.

There is one drawback with bump maps, they can be sensitive to physical light and colour changes. Which could result in the map getting washed out.

Scale

The Bump Map input is very similar to other maps, with the same U and V scale and offset boxes. These should ideally be identical to the ones used in the pattern, if a pattern texture map is used. Indeed the only difference in the settings is the Bum map scale box.

The scale. tells the program how deep would would like it, to try and make the relief, I say try, because as the relief is purely visual, there is a limit as to what can be achieved. Although the scale will allow high numbers it can't work miracles.

Fig 109. Shows the image being used, Fig's 110 / 111 has two different scale factors.

Fig 109

Bump map image

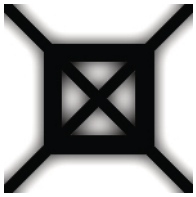


Fig 110 Scale = 0.5

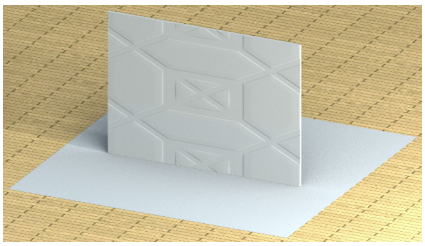
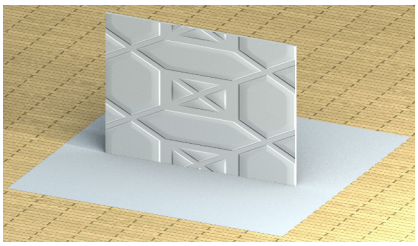


Fig 111 Scale = 2



he next example uses a coloured texture image.

Fig 112. Is the texture that will be wrapped, as a pattern image.

Fig 112

Fig 113. Is a plain render, without any bump map used.

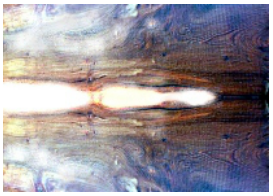
Fig 113

Fig 114. Is the same image used as the bump map, notice there is a little more definition. But because there is little difference between light and dark, the effects are minimal.

Fig 114

Fig 115. Is a new bump map. It Has been inverted and modified with some white paint added. Giving a much greater definition.

Fig 115



This results in the last Fig 116. Is at the same scale as previously used. As one can see, the bump map clearly enhances the grain pattern.

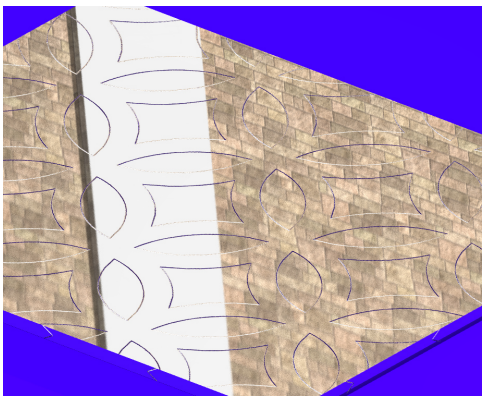
Fig 116

It produces a more realistic rough wood. than using the same image as the bump map. This is because there is a lot more high to low definition, than the original image.

Bump map and transparency.

Whilst bump map will actually work with transparency, the results are often disappointing, and best avoided.

Fig 116a



Bump map with reflectance.

This depends on the reflectance and what is being reflected.

As can be seen in Fig 116a, the bump map shows well on the light area, but not so good with the reflected brick

The difference in now a bump map texture appears can be seen in Fig's 116a and 116b.

Fig 116a has been blurred in a paint program, to give less sharp definition between the white and the black

Fig 116a

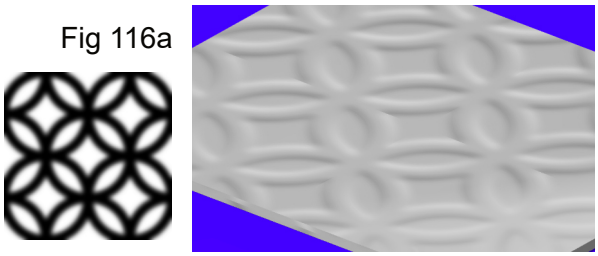
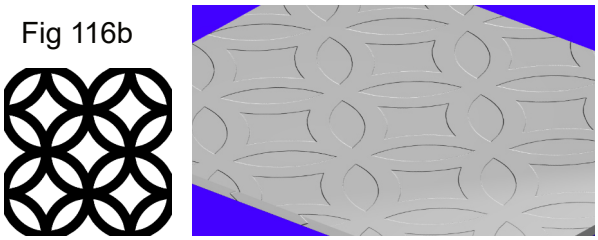


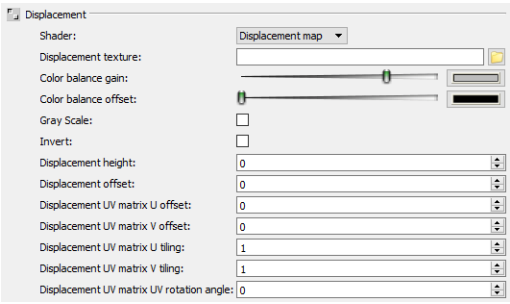
Fig 116b is the original black and white image, with a sharp definition between the two..

Fig 116b



Displacement map,

Fig 117



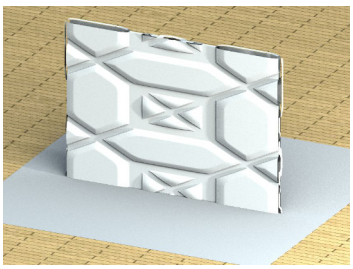
It should not be any surprise that the controls are like all the others. The only differences are that scale has changed to height, and a new box added called offset.

What's the difference from a bump map.

Displacement maps alter the actual geometry when rendering. (no worries though - this doesn't mean your actual part will suddenly acquire lumps). But it does mean the render will be less affected by light changes and allow for more definition. Sound good eh, but.

There is a downside. Because they are actually altering the geometry at rendering time, they can take significantly longer to render. This obviously depends on the complexity of the map and object.

Fig 118



Displacement maps use RGB values like bump maps from the low point black (RGB 0,0,0) to high point white (RGB 255,255,2550). It uses a special mapping system, however, fortunately we don't have to supply a special map. The program does it automatically.

Fig 118. Using the same image as for bump map, and a height (scale) of two. Comparing this with the bump map image of Fig.111. We can see the vast difference in the finished render.

Displacement works for both positive and negative effect, similar to bump maps. By changing the image the effect can be reversed.

As can be seen in Fig 119. and Fig 120.
The blue background in Fig 120 is just to make the white area stand out in this document, it is not part of the image.

Fig 119a

Fig 119b

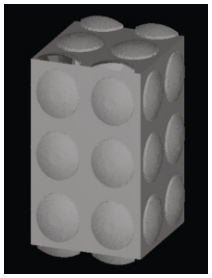
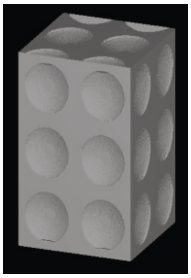


Fig 120a

Fig 120b



Colour balance, Gray scale and Invert. Work the same as for other material texture settings.

Height

Alters the actual depth between high and low points in world units. Which one needs to keep in mind when re-using materials, especially if the units are different from what you are used to. Also bear in mind to keep the scale boxes correct.

By this I mean, say you normally draw in mm, and draw a box 200mm x 100mm. Then at some point decide to change your space units to cm. You would need to go into the material and change the scale, so it will render correctly.

Fig 121 and Fig 122, Show the difference in height settings.

As can be seen, Displacement affects the face of a object. Fig 123. Shows a zoomed in view, and clearly shows gaps at the edges, where the displacement has lifted up the face.

The gaps are generally caused by the offset being left at zero. Thus the offset of zero means the low point is on the original face of the object, and height of 4 pushes the displacement above the face.

Fig 121

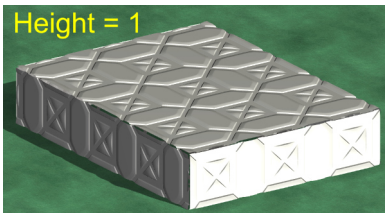


Fig 122

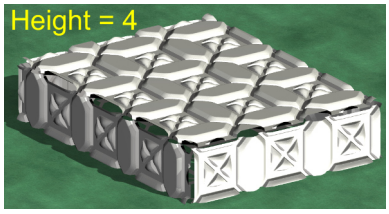
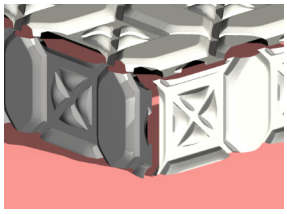


Fig 123



Offset.

This determines where the map will start to displace. Think of the original face being at offset ‘0’. Any height one uses pushes parts of the surface above this zero figure.

You can vary these figures for different effects. Indeed if the object is simply an extruded line (sheet object), it may not matter leaving it as 0, each scene will vary.

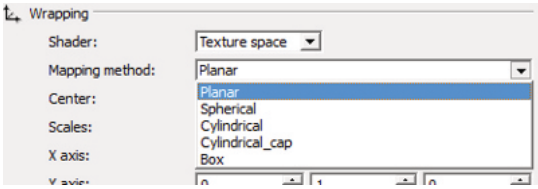
Fig 124. Was given a height of 2, but the offset was left at 0. To make the edges correct we need to use an offset of -2. A seen in Fig 125

Fig 124

Fig 125

Wrapping - Texture space

Fig 126

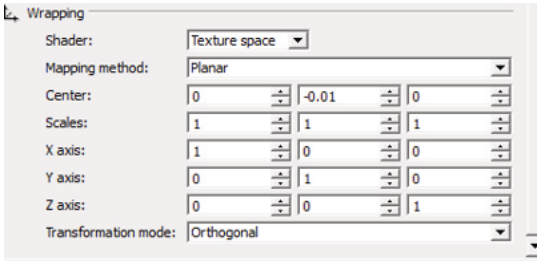


NOTE - At the time of writing, If both a pattern map scale u / scale v settings are use at the same time as wrapping, it may produce incorrect results. It is better to use scale u = 1, and scale v = 1, in the pattern section.

This section is completely different to the rest. Its function is to wrap the texture onto the object, based on the settings applied here, however everything you learned about scale u and scale v goes out the window.

With the scale U and scale V values in the pattern texture map, the smaller the number the larger the pattern. This works the opposite, and is actually more like normal scale. The bigger the scale, the larger the image. (More on that later).

Fig 127



Shader.

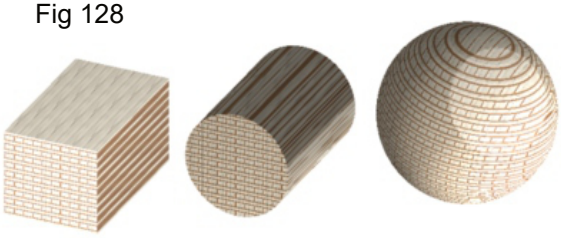
This is either None, or Texture space. Selecting texture space, reveals all the settings to which affect how a texture will look on the object.

The mapping method drop down box, allows one to choose the type of wrapping required. The default is Planar.

This mapping method affects how the image will be wrapped onto an object

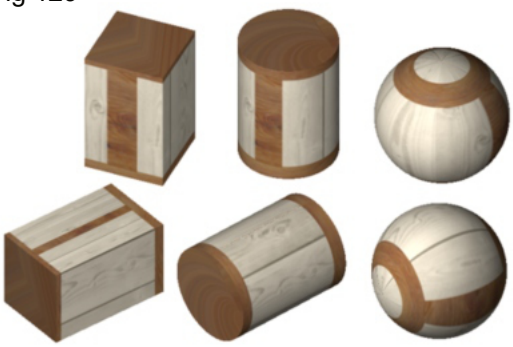
The mapping method has 5 options as follows

Fig 128



Planar. This lies the texture map onto the x / y plane of a single facet (object), by default. Note this is the entity coordinate system not the world.

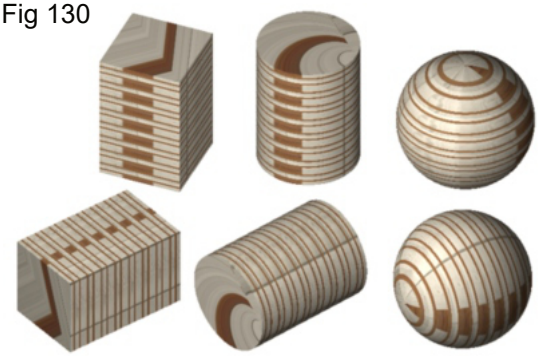
Fig 129



Spherical. This wraps the texture around a sphere it will also wrap around other objects, but its primary function is for a sphere. The image u axes is followed horizontally, whilst image v axes is wrapped vertically (object z axes).

Unlike planer, it is preset to wrap 1 instance of the texture around the object. Therefore under normal circumstances do not use a texture space scale u and scale v. i.e. set both to 1.

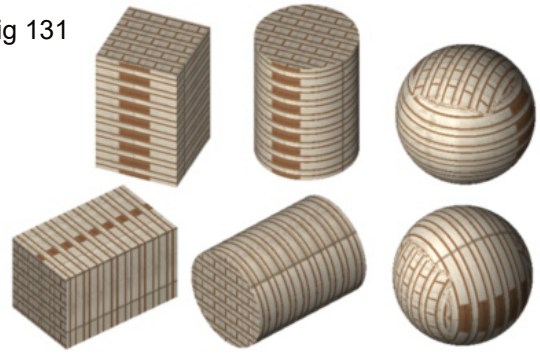
Fig 130



Cylindrical. This wraps an image around an object, based on stretching 1 instance of the texture, going round the object (x/y). I will use multiple copies vertically (Z) texture scale.

Cylinder Cap. Like cylinder, but it places / caps the end object with a planer image.

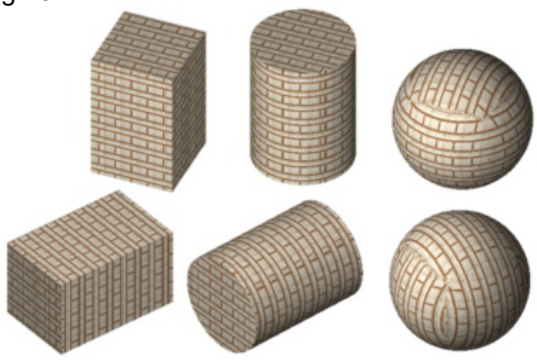
Fig 131



Box. The last option is box, which wraps a planer type image onto all surfaces of the object. The sides are wrapped, with the top and bottom being single planer surfaces.

As can be seen. It will not wrap correctly onto a sphere.

Fig 132

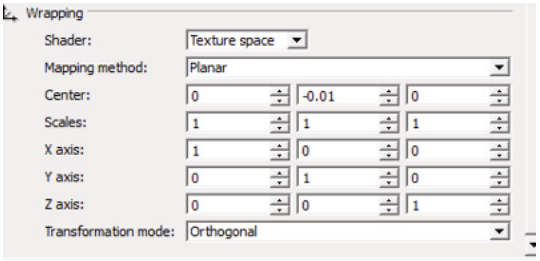


Back to the dialog, what are the other options.

One of the first thing you may have noticed is there are more boxes. And at first glance it may seem complicated. It is hoped that the following will give a little understanding

One point to note, the first part of this document will discuss using these boxes in conjunction with the other texture spaces as opposed to a standalone Setup area

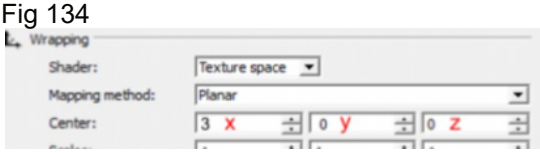
Fig 133



Center (Centre UK).

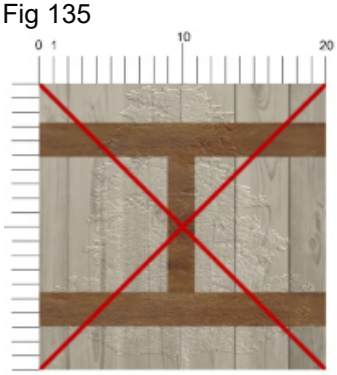
Centre, moves the texture x, y, z, in drawing units. This can be used to tweak your texture in any direction without having the alter the offset. And the beauty is that, if one has previously set up a texture space and bump map for example, the control will act on both, so no need to alter the offsets every time.

Fig 134



Center - planer mode uses x & y, as the image is not mapped around the z axes I planer mode, the z box is available but ignore it. the images on the right have been moved by typing 3 into the first box

Fig 135



Centre - Sphere mode, set the texture space scale u and v to 1 as this automatically scales the texture

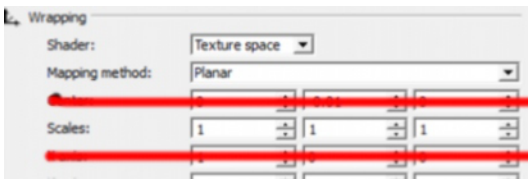
Centre - Cylinder mode, Centre - Cylinder Cap mode, Centre - Box mode, these can used in conjunction with texture scale u and v, and offset values,

Scale.

The scale boxes can be used in conjunction with texture map scaling or independently. It allows altering the scales in one place instead of each scale u and scale v boxes,

Scales works the opposite to the other scale u and v boxes, in that increasing the scale is the same as zooming in.

Fig 136



For example on a 20 unit by 20 unit square, a scale of one, will yield 20 textures wide and high as in Fig 137.

Fig 137

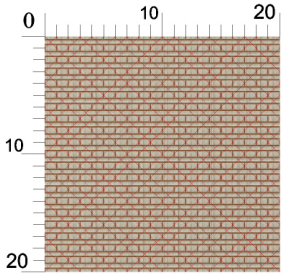
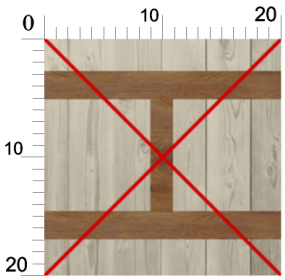


Fig 137a



To get one instance of the texture. One would use Wrapping scale x = 20, Y = 20, as in Fig 17a.. Rather than the - pattern texture map scale u = 0.05, texture map scale v = 0.05.

Scale - planer mode uses x & y, as the image is not mapped around the z axes, the z box is available but ignore it. the images on the right have been moved by typing 3 into the first box

Scale - Sphere mode, set the texture space scale u and v to 1 as this automatically scales the texture

Scale - Cylinder mode, Scale - Cylinder Cap mode, Scale - Box mode, these can used in conjunction with texture scale u and v, and offset values, can be used to sale an individual face though m

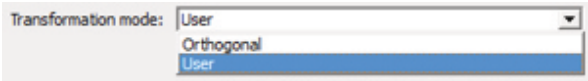
X Axes, Y Axes, Z Axes. hmm bit nondescript as a title,

IMPORTANT. At the time of writing there has been no definitive explanation from IMSI, nor anyone else as to the correct operation. Redway have a wrapping explanation, but not stated if Orthogonal or user., Therefore what is written below is a personal observation. ,And written with the best intentions. I Suggest doing your own experiments and decide yourself if the following is an accurate.

Fig 138

This appear to do two different functions, depending on whether the transformation mode is set to User, or Orthogonal. (Fig 139)

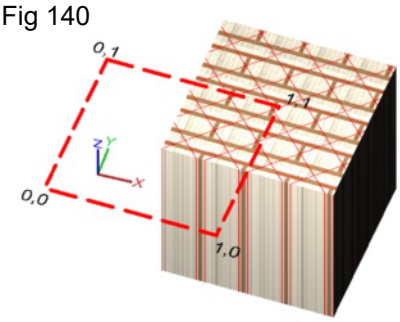
Fig 139



Orthogonal. - Planar.

When in orthogonal mode, it orientates the image, onto a single planer face. Although you can type in any figure into the boxes. I'd suggest using 0 & 1 at least until you are comfortable.

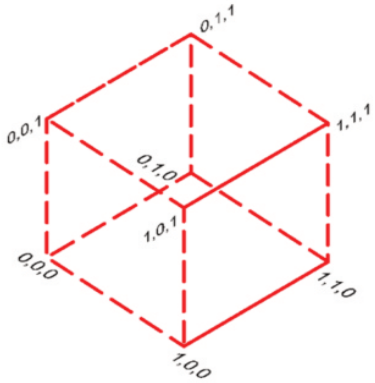
Fig 140



looking at the square which has only two numbers x,y. Planer mode places the texture using x,y. Hence the striping in the z axis. However, We can choose which planer face the image is wrapped onto.

Some examples. Remember as a planer mode, we are only using the three 'x axes' boxes, and the three 'y axes' boxes. And we are using the object coordinate system not world coordinates.

Fig 141



Planer mode, wraps onto opposite faces, i.e. 1). Top and bottom. 2). Left and right. 3). Front and back. To visualise the faces we will use the image on the right. Or more specifically the coordinates it contains.

All three wrappings can be accessed from a starting point of 0,0,0. What we are concerned about is the direction that x and y will be placed. From this origin, we have six orthogonal possible combinations. Three normal wraps, and three rotated 90 degrees.
(x and y separated with a plus sign for ease).

1,0,0 + 0,1,0.	1,0,0 + 0,0,1
0,1,0 + 1,0,0.	0,1,0 + 0,0,1
0,0,1 + 1,0,0.	0,0,1 + 0,1,0

Fig 142 is our image for wrapping. Note. In the next examples we are only interested, in placing the image orthogonal on a face of the object.,

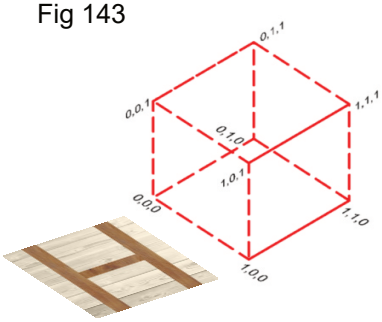
Fig 142

Y axis

X axis

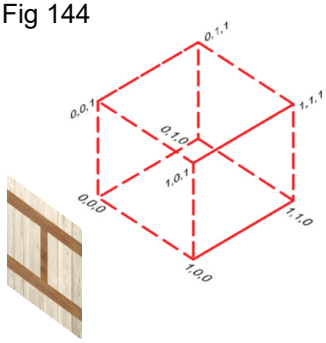
For the top / bottom.
The x axes goes to 1,0,0.
And the y axes to 0,1,0

Fig 143



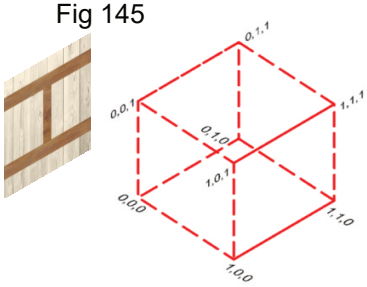
For the front / back.
The x axes goes to 1,0,0.
And the y axes to 0,0,1

Fig 144



For the left / right.
The x axes goes to 0,1,0.
And the y axes to 0,0,1

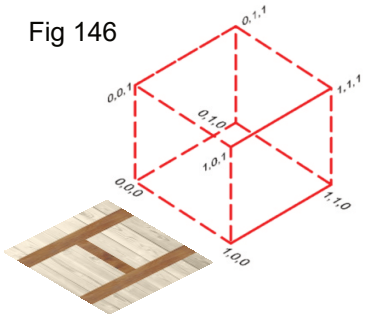
Fig 145



That leaves 3 possibilities left. From our original six. These will rotate the image 90 degrees before wrapping.

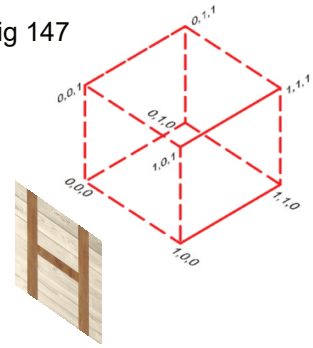
For the top / bottom rotated.
The x axes goes to 0,1,0.
And the y axes to 1,0,0

Fig 146



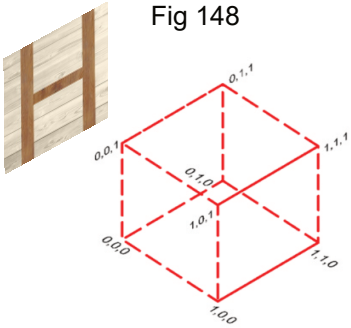
For the front / back rotated.
The x axes goes to 0,0,1.
And the y axes to 1,0,0

Fig 147



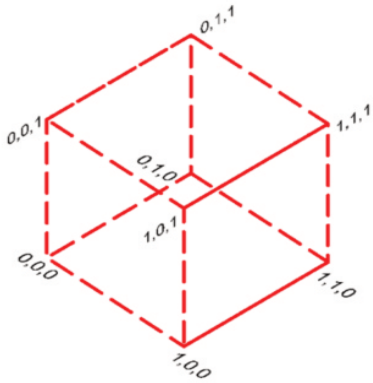
For the left / right.
The x axes goes to 0,0,1.
And the y axes to 0,1,0

Fig 148



What if we want the image to wrap diagonally.

Fig 149



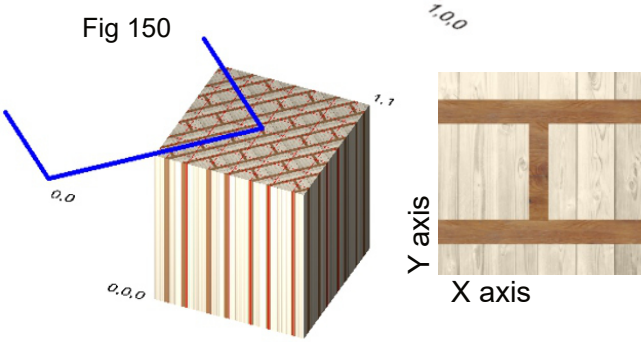
So far we have only put a single (1) into the X and Y axis boxes. Looking at the red diagram, we can see for a diagonal plane on the base. We would need to go from 0,0,0 To 1,1,0

This we can do in the X and Y axes boxes.

The image on the right, we have used
X axis = 1,1,0
Y axis = 0,1,0
It could also be written Y = -1,1,0. It seems the program will automatically align Y within certain parameters.

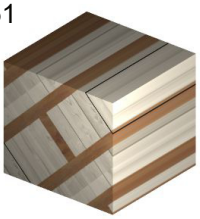
Therefore. I MUST stress again, that all the figures are provided through experimentation. I cannot say that this is exactly how TC is programmed.

Fig 150



We can do the same with other faces, For example. To wrap at a diagonal on the front face we would use. X axis = 1,0,1 and the Y axis = 0,0,1. Which gives us the figure on the right.

Fig 151

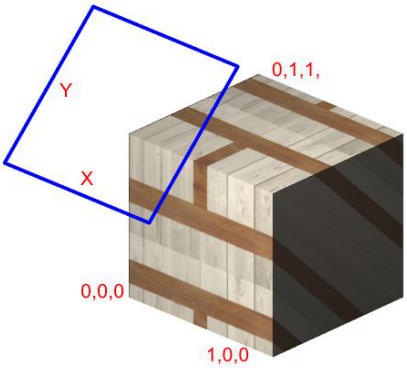


Suppose we want it from the opposite diagonals. Well. This is where we can enter a minus figure. If we change the X axis to 1,0,-1. We can change Y axis to 1,0,1. Thus giving up the opposite diagonal.

Fig 152

Last two planar examples.

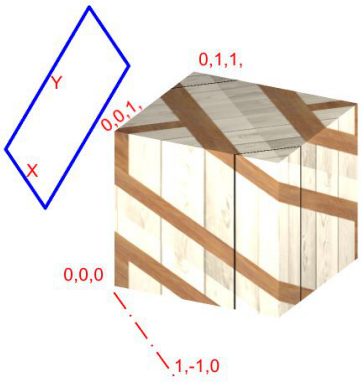
Fig 153



Up to now we have used flat planar faces with the side 'stripes' orthogonal. What if we want the stripe diagonal. We can do this, by giving X and Y, diagonal coordinates, However it can stretch the image as a result.

If we change the direction of the Y axis, to a diagonal. The program will wrap the image diagonally onto the faces, as can be seen from the stripes on the side. X axis would be 1,0,0 and Y axis 0,1,1

Fig 154

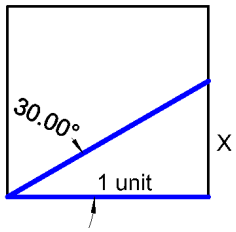


In this last example. We will attempt to wrap on a corner.

First we need to swing out the X axis, which is done with 1,-1,0
If we used a standard Y axis as 0,0,1 it would wrap the edge as the previous example. However, is we change it to 0,1,1. The program will rotate the Y axis. Giving is a corner to corner wrapping

Up to now. We have just used 1 and 0. You can use other figures. However I don't know all the calculations used. Therefore I may be trial and error, unless you are familiar with transformation.

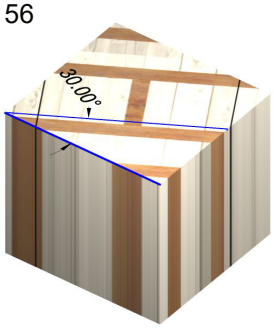
Fig 155



As an example, For the normal top / bottom planar wrap, we would use X = 1,0,0 and Y = 0,1,0. To rotate it we would need to swing the X axis away from 1,0,0. Towards 1,1,0. Looking at the diagram on the right, we need to find x. But we need Radian. So 30 degrees = 0.523599 Radian.

We can use X = 1 x (Tangent(0.523599)) = 0.57735.

Fig 156



Therefore we can use X axis = 1, 0.57735 ,0 and Y axes = 0,1,0. Giving us a wrapped image at 30 degree angle. As shown on the right. We could fully calculate the correct Y axis, But the program automatically assigns Y based on the X axis rotation. So in this case it is not necessary.

In practice. It is highly unlikely you would need (nor want), to calculate all the different scenario. All the previous examples are just done so one is aware of the possibilities. And hopefully give a slightly better understanding of what the X axis and Y axis boxes do.

Other Orthogonal modes.

As stated previously, some modes are pre-set in terms of wrapping, (sphere wraps a single instance of the texture). But the direction of the wrapping can be adjusted as per the planar adjustments discussed previously.

Texture space User Mode

Changing the drop down box from orthogonal to **User** mode, adds another variation.

Firstly it turns on the z axis settings, In orthogonal mode, we were mainly concerned with x and y axis.
In User mode we can also use the z axis box.
I
However one has to remember this is still a single wrapping tool, it is not a facet editing tool, nor a UV mapping tool, changing one face will affect the look of other faces.

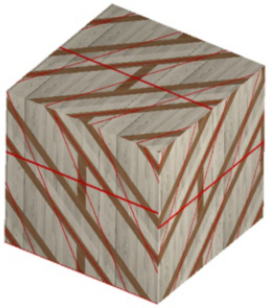
It is beyond the scope of this wiki to cover all the hundreds of different scenario's, its best just to have a play and see what you like, The modes planar, spherical etc. operate the same in user mode. Most of the time you will probably use box setting.

As a last example on the subject of scaling, the image in Fig 157, was wrapped using.

X axes = 3,3,0
Y axes = 0,3,3
Z axes = 0,0,3

As can be seen. As before tying two figures into one axes causes rotation and increasing the numbers causes scaling. unfortunately the 'magic' formulae for exactly predicting the wrapping is unknown.

Fig 157



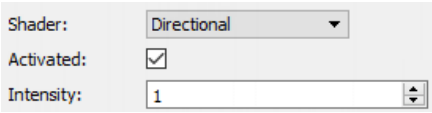
RedSDK Luminance

RedSDK Luminance has three categories, Non physical. Physical and Sky. Standard Luminance are Ambient, Area, Beam, Directional, Point, and Spot. Physical Luminance are Physical, Rectangular Physical, Spherical Physical. Lastly Sky Luminance.

Many of the setting are the same in each category. Therefore this first section is just a general introduction to luminance properties.

General Standard luminance Parameters .

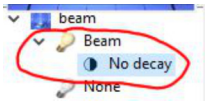
Fig 160



NONE (No intensity)
Intensity is 1 for the entire light source effect area. Applies to Ambient.

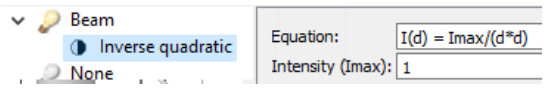
Intensity. Intensity may be found in two places, depending on the luminance used. For Directional. It on on the first screen.

Fig 161



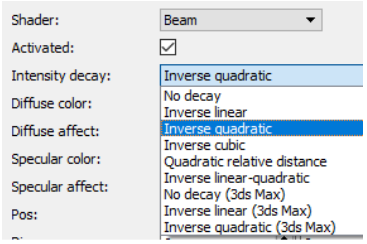
For most others, it is accessed by expanding the luminance in the left part of the dialog box. The decay name may changes to that chosen under the intensity decay option. Clicking on either No Decay, or the name, will open the right hand panel for altering the intensity value. Fig 161 and Fig 162

Fig 162



Intensity Decay.

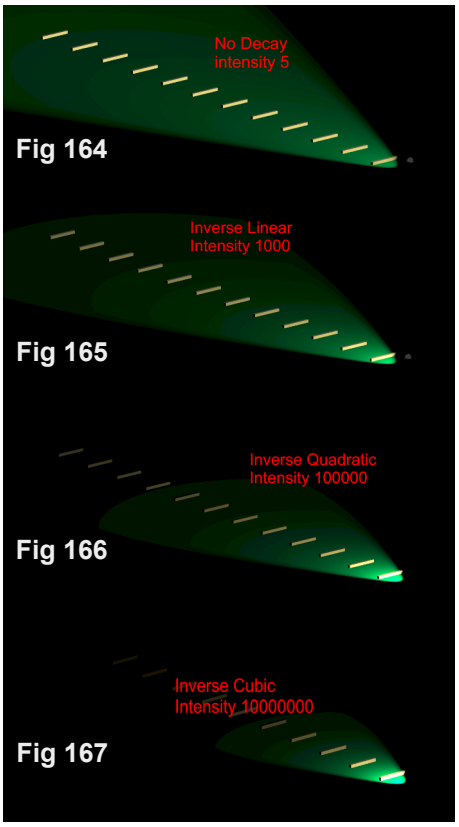
Fig 163



This Sets the rate at which the lights power diminishes with distance. For example No decay. Or Natural quadratic.

A light source in the Red engine may have a decay of its intensity, based on the distance between the lit point and the light source centre. The lit point is always the geometrical point at the pixel centre of the lit pixel. A light source may have a centre or not. If it has no centre, then it does not support the intensity decay calculation.

The actual options in the drop down box, changes depending on which luminance was chosen. Note. The images below are an example only. They may differ from your findings, due to other lighting factors.



No Decay. (Fig 164)
The luminance is spread evenly from the source. Without diminishing in intensity.

Inverse Linear. (Fig 165)
Luminance decreases as the lit point moves away from the light.
INTENSITY = IMAX / DISTANCE; where IMAX = ATT[0].

Good for decreasing luminance, whilst still keeping a good illuminated distance.

Inverse Quadratic. (Fig 166)
Intensity decreases using a quadratic equation.
INTENSITY = IMAX / POW(DISTANCE,2); where IMAX = ATT[0].

Requires a larger intensity. Allows good light control good for illuminating specific areas.

Inverse Cubic. (Fig 167)
Intensity decreases using a cubic equation
INTENSITY = IMAX / POW(DISTANCE,3); where IMAX = ATT[0].

Good for lighting specific objects, but required large intensity.

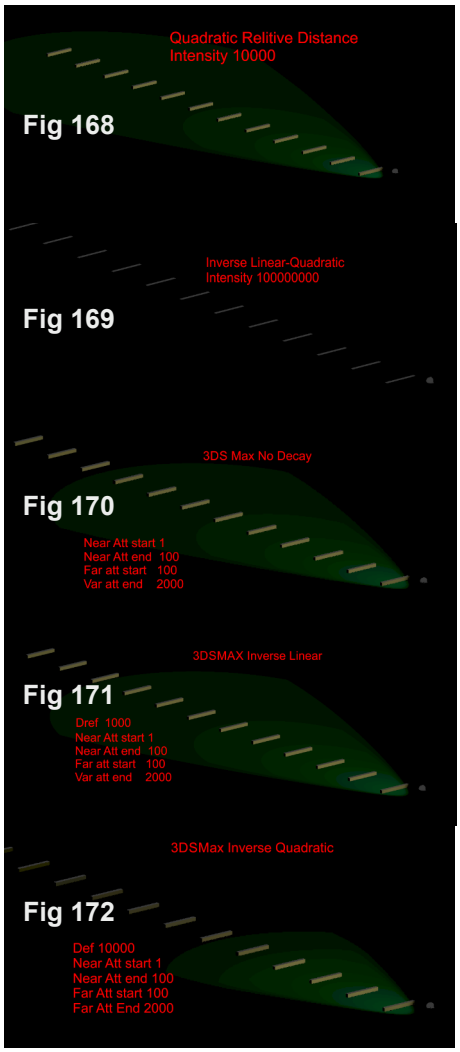
Quadratic Relative Distance. (Fig 168)
Intensity uses a 2nd order equation: INTENSITY = 1 / (1 + KLIN * DISTANCE + KQUA * DISTANCE); where ATT[0] = KLIN and ATT[1] = KQUA.

Inverse Linear Quadratic. (Fig 169)
Intensity decreases using a ratio between the current distance to the lit point and a reference distance. INTENSITY = 1 - POW(DISTANCE / DMAX,2); where DMAX = ATT[0].

No Decay 3DSMax. (Fig 170)
3DSMax decay. Near and far attenuation distances can be used, and the intensity does not vary with the distance besides that. INTENSITY = 1.

Inverse Linear 3DSMax. (Fig 171)
3DSMax decay. Near and far attenuation distances can be used, and the intensity decreases using the following equation: INTENSITY = min(1, DREF / D); where DREF = ATT[0].

Inverse Quadratic 3DSMax. (Fig 172)
3DSMax decay. Near and far attenuation distances can be used, and the intensity decreases using the following equation: INTENSITY = min(1, POW(DREF / D,2)); where DREF = ATT[0].



Diffuse color – Sets the diffuse colour of the light.
Diffuse affect – Sets the relative intensity (power) of the diffused light.

Diffuse colour and diffuse affect, do a similar job in black to white colour, the main difference comes when one uses a non grey colour. The diffuse colour midpoint of the slider is the selected colour, moving the slider toward 0% darkens the colour, moving towards 100% lightens the colour, eventually reaching white. The setting of these, is trial and error and personal preference. I look at it as. Diffuse color is the main setting, and Diffuse effect as a fine tuning.

Fig 173
Diffuse settings at normal

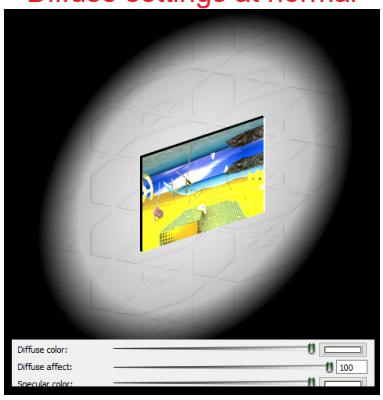


Fig 174
Diffuse color and affect reduced

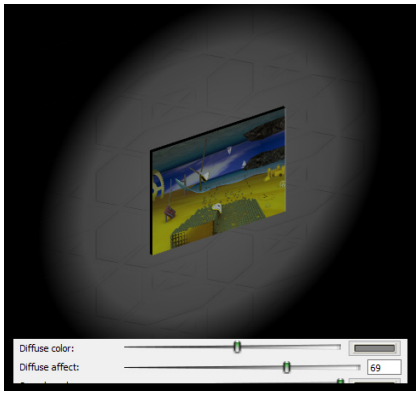
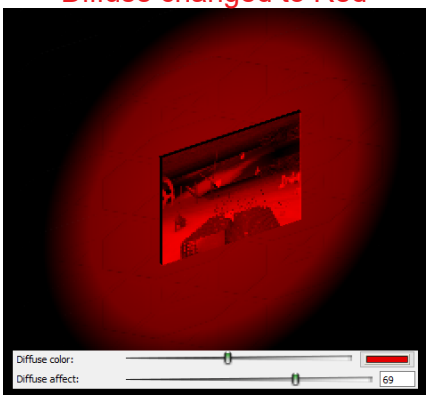


Fig 175
Diffuse changed to Red



Specular color – Sets the specular color of the light.
Specular affect – Sets the relative intensity (power) of the Specular light

Specular works on coloured objects, i.e. objects with no material attached, it is ignored when materials are applied, the effect of specular colour can sometimes be subtle or non existent.

Specular is effectively a 'brighter' spot on a non material object. Reducing the specular factor, gives a smoother finish.

Fig 176

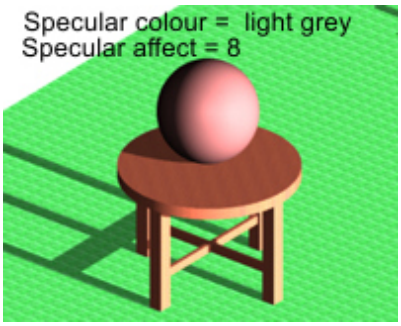
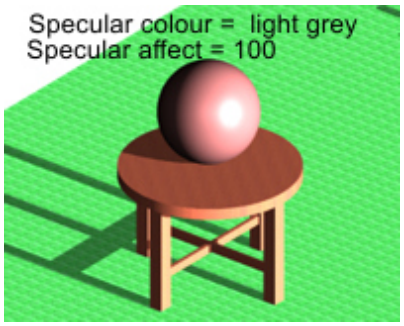


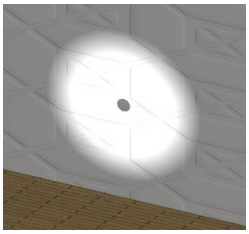
Fig 177



Pos – This Sets the X,Y,Z position of the light, relative to the object being used as the luminance.

There is an object luminance property setting, for **'Load for Luminance only'**. This Allows the luminance to shine, even if the position setting is at 0, 0, 0. With that setting turned off. The position should (almost) always be set to a value. For example. In Fig 178, the light emanates from the centre of an object. If the position was left at 0,0,0. Then when rendered, the shape of the source object may be seen.

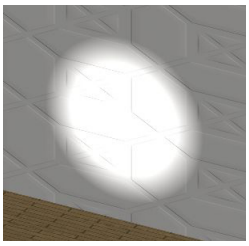
Fig 178



In the images on the right. The luminance was attached to a 10mm sphere. The Direction was set along the plus Z axes. Fig 178 clearly shows a dot in the centre, this is the source sphere, because the Position setting was left at 0,0,0.
In Fig 179, the Position was moved to 0,0,5. Therefore the luminance emanates from the edge of the sphere, and prevents the source object from showing up.

With some luminance types, a Pos setting of 0,0,0. Will actually prevent the light from showing up at all, (spot, being one such luminance). In these instances, the Pos setting must be moved to the edge of the source.

Fig 179



For all this document. The object property setting 'Load for Luminance only' is turned off.

Dir – Sets the X,Y,Z direction of the light relative to the position

Dir, determines where the light will be pointing. It uses the objects orientation, which may be different from world. I use just 1's and 0's. Though other figures can be used.

In general. One would just put a figure into the box representing the desired axis. For example. If the required direction was the objects Z direction, on would have 0,0,1. For a 45 degree axis one could use 1,0,1. Or -1,0,1.

In Fig 180. The Dir is set to positive Z axis (the blue arrow). Hence the setting of 0,0,1.

In Fig 181. The direction was change to positive Y axis (the green arrow). Hence 0,1,0.

Fig 180

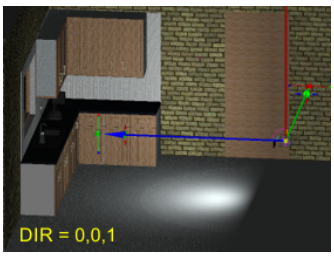
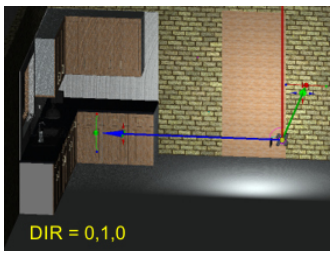


Fig 181



Top – Defines a second axis which is perpendicular to the direction, used for the rotation, if the lights display is not circular.

A beam light is circular. Therefore under normal circumstances it wont make any difference. Obviously do not use the same axis as DIR.

With a luminance that is rectangular, (or non circular). The Top setting can change the orientation.

Fig's 182 and 183, use a rectangular area light. The light direction is set to the plus Z axis. This allows for the X and Y directions to be used as the top.

Fig 182

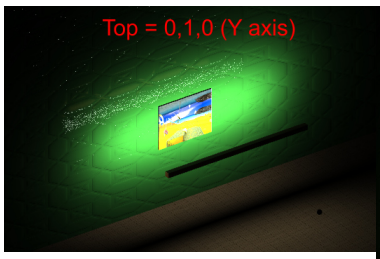
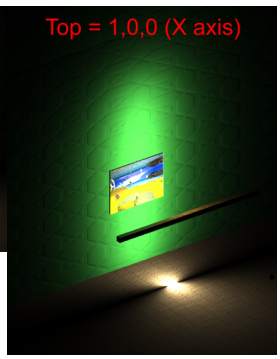


Fig 183



Inner Radius – Sets the inner radius for the light.
Outer Radius – Sets the outer radius for the light.
Width – Sets the width of the defining rectangle.
Height – Sets the height of the defining rectangle.
Spot Angle – Sets the angle of a spot type light.
Additional Spot Angle –Sets an additional angle for a spot type light.

All these settings control the width, height, radius of a luminance. Differing luminance uses two of these setting to determine the size of the light. Each is discussed on the appropriate page.

Fig 184

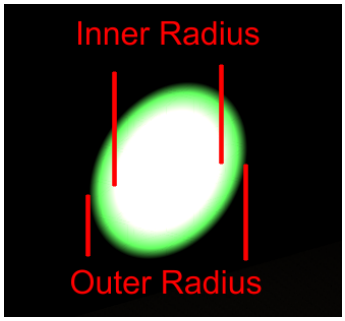


Fig 185

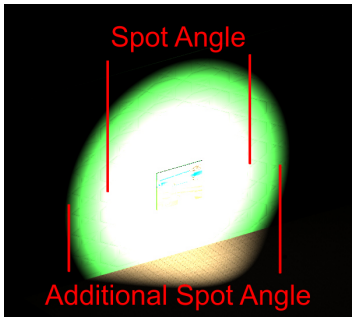
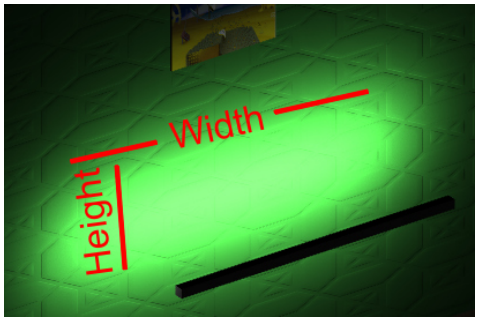


Fig 186



Shadows – Sets whether the light will generate shadows, or not.

The **Shadow** tick box, is used to turn on / off shadows. Shadows serve two purposes. The first is the obvious one. It allows shadows to show up in a render. The second is to arrest those luminance which would otherwise pass through objects.

Shadow Color – Specifies the color of shadows from the light.

The **Shadow color** would normally be left at grey scale. Though the program allows for the actual colour to be changed.

Shadow Map – Shadow maps are generated by testing whether or not each specific pixel is visible from the light source

Fig 187 - 189

Shadow Map Resolution – Sets the resolution of a shadow map. The Values must be multiples of 2.

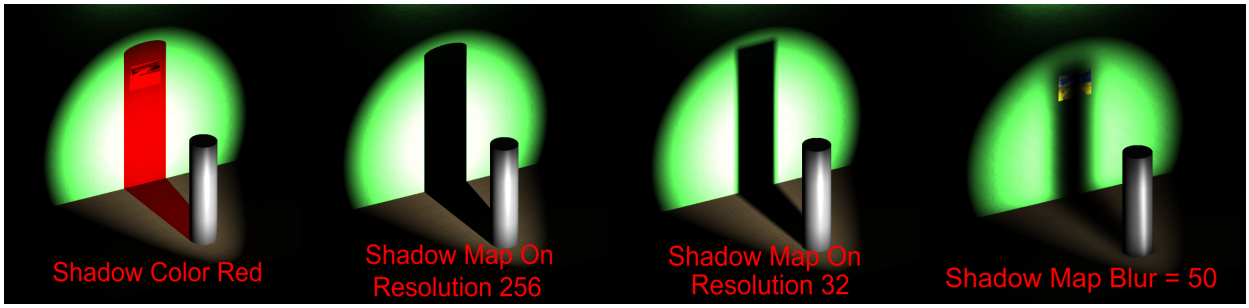
Shadow Map Blur – Sets a blur factor on the shadow map, reducing jagged edges + transitions

Shadow map pre-calculates the shadow before rendering the scene. This accomplished by comparing each pixel to a depth image (z-buffer) of the light source's view, stored in the form of a texture file. Shadow maps can accelerate the rendering of shadows, but usually as the cost of some quality.

Shadow images

A note regarding shadow colour grey. Normally this would result in a grey shadow. Except when the shadow colour slider is moved towards white. The object / texture colour will show in the colour of the shadow.

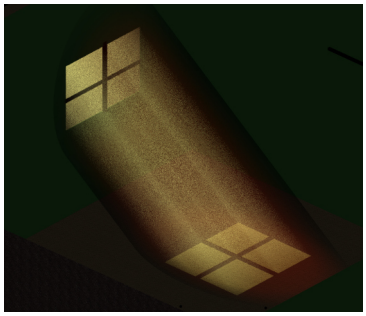
Fig 200 - 203



GI Baked.

Redway state. *"By baking, we mean pre-calculating a given information, storing it and restituting it, faster, as a later rendering phase".* It is used with global illumination rendering. And may not work correctly if global is set to use the GPU.

Fig 204

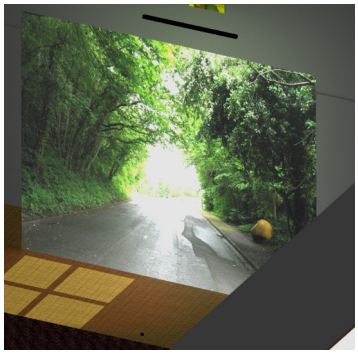


Enable Volume. Used with volumetric environmental effects (see RedSDK environment).

Enlighten volume. Used with volumetric environmental effects (see RedSDK environment).

Volume is a special effect (Fig 204). Which can show light rays passing through a space. Only certain lights can create a volume effect. This is covered in detail in the environment section.

Fig 205



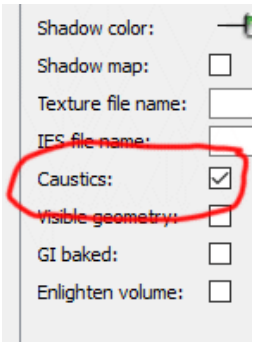
Texture File Name – Rather than using a single constant color for the whole light, you can setup a texture from which light color will be read.

Using a texture map, you can simulate projectors or complex light filters.

Fig 205 on the right, uses a projection, generated by a rectangular physical luminance.

IES File Name – Manufacturers measure the behaviour of their bulbs, and store their directional contribution into IES files. Those files are available for most of the bulb models directly from the manufacturers' web sites. Even if IES files store information about the shape of the emitter, they are not used here.

Fig 206



Caustics – Specifies whether the light will generate caustic effects.
See caustics under material settings. Which are the focused particles of light generated when Material caustics are turned on. It require a material with caustics options turned on. Plus a luminance capable of producing the effect.

The three physical luminance all have a caustic checkbox. Which needs to be turned on for the effect to appear.

Fig 207. Has caustics turned off. Fig 208. Has the caustic effect turned on. As can be seen, there is a big difference in the way the light passing through the glass hits the table. Note, this extreme effect was created simply to show the difference. Object material parameters can altered to change the effect. For example reducing the IOR values, Fresnel, etc.

Fig 207

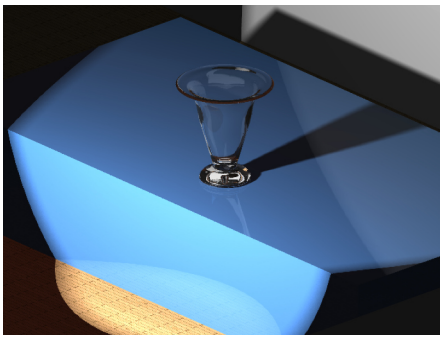
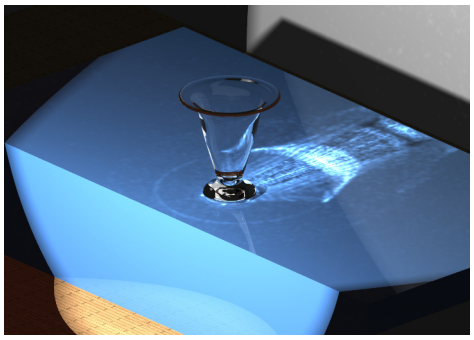


Fig 208



Visible Geometry - Specifies whether the physical geometry of the associate 3D objects will be seen. By this it means the light source visibility, rather than the actual physical source object.

For example, in real life. A turned on light bulb can glow white. This is simulated with luminance by turning on visibility (Fig 209). It creates a glow around the source object. Assuming the light diameter is larger than the object.

Fig 209

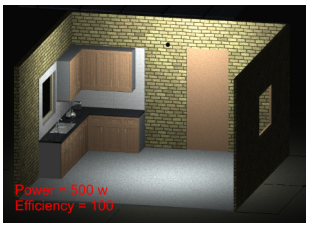
General Physical Luminance Parameters

PARAMETERS

Power – Sets the power of the light in Watts. Just like a normal house light. However some luminance have a fixed intensity decay. This can be used with Luminance efficiency, to vary the effect.

Fig 210

Fig 211



Luminous Efficiency – luminous efficiency of the light in percentage (%).

Intensity decay – Set the rate at which the lights power diminishes. For physical luminances, this is fixed at Natural Quadratic.

Fig 212

Color – Sets the color of the light.
Color Temperature – Alternate color value which supports definition of the color by temperature in Kelvin.

The colour selector and the slider allow for both grey scale and colour adjustment. When a colour Is selected, that slider will automatically be set appropriately. Moving the slider to the right (towards white) lightens the colour and scene. Moving to the left (towards black), darkens the light and scene.

The colour temperature is an alternative to using the slider. A setting of 0, defaults to the colour slider setting. Any number above 0, overrides the colour slider. i.e. A setting of 1 equals black.

Fig 213 White

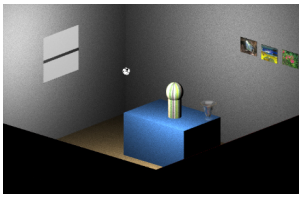


Fig 214 Red

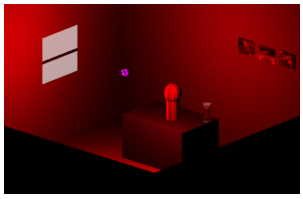


Fig 215 4000 k

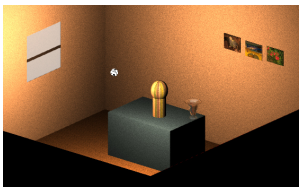
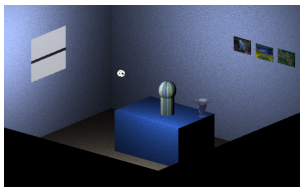


Fig 216 9000 k



However although its classed as an alternative. The normal colour temperature scale, does not include all colours. For example, it won't produce bright green. If that is required one would use the colour picker and slider.

The colour temperature is good for altering the 'mood' for example between 4000 warm and 9000 cool.

Color Affect – Sets the relative intensity (power) of the color light.

The Color Affect, is used with the colour slider (it is overridden by colour temperature). It is basically like a diffuse slider. Normally it is set at far right (100). Reducing the affect darkens the render colour.

Samples Count – Sets the maximum number of light samples.

Double sided - sets single or double sided surfaces.
When a physical luminance is applied to a surface/sheet. For example, and extruded line. I has a one two faces, front and back.

Double sided tells the render, whether the light emits from just one face of both faces. In Fig 217. Double sided is not ticked.
Fig 218. Has double sided ticked.

Fig 217

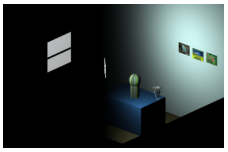
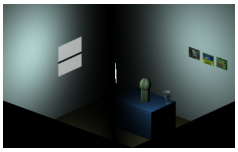


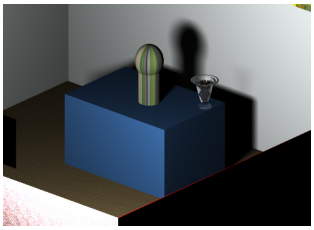
Fig 218



Pos – Sets the X,Y,Z position of the light relative to the object to which it is attached.
Dir – Sets the X,Y,Z direction of the light relative to the position.
Top – Defines a second axis which is perpendicular to the direction, used for the rotation, if the lights display is not circular.

Samples Count – Sets the number of samples used to populate the light surface. Must be greater than 0
Width – Sets the width of the defining rectangle.
Height – Sets the height of the defining rectangle.

Fig 219

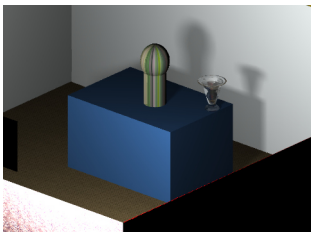


Shadows – Sets whether the light will generate shadows, or not.

A simple tick box to show or not show shadows in the render.

Shadow Color – Specifies the color of shadows created by the light.

Fig 220



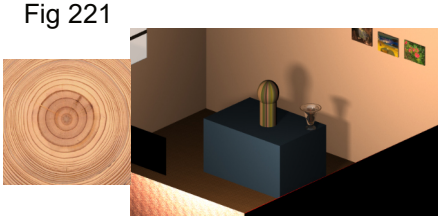
In grey scale, it is used to lighten or darken the shadows. Alternatively one can set a specific colour to use in the render.

Fig's 219. Shows almost black shadows. Fig 220 Has the shadows more gey

Shadow map – Shadow maps are generated by testing whether or not each specific pixel is visible from the light source

Using a shadow map instead of ticking 'Shadow' . In some cases, still allows for the shadow colour to be altered. Maps can produce a very different type of lighting and shadows.

Fig 221



Texture File Name – Rather than using a single constant color for the whole light, you can setup a texture from which light color will be read. Using a texture, you can simulate projectors or complex light filters.

Fig 221. Uses as a texture to completely alter the lighting colour. No other colour adjustment was made.

IES File Name – Manufacturers measure the behaviour of their bulbs, and store their directional contribution into IES files. Those files are available for most of the bulb models directly from the manufacturers' web sites. Even if IES files store information about the shape of the emitter, they are not used here.

IES files may require experimentation, as they can substantially alter the lighting,

Caustics – Specifies whether the light will generate caustic effects.

Visible Geometry - Specifies whether the physical geometry of the associate 3D objects will be seen

GI Baked.

Redway state. *"By baking, we mean pre-calculating a given information, storing it and restituting it, faster, as a later rendering phase".* It is used with global illumination rendering. And may not work correctly if global is set to use the GPU.,

Enlighten volume.

Used with volumetric environmental effects (see RedSDK environment).

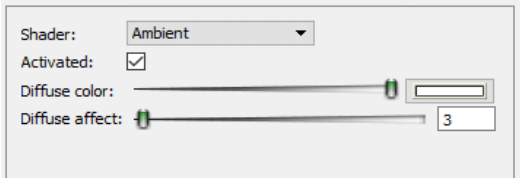
Volume is a special effect, which can show light rays passing through a space. Only certain lights can create a volume effect. This is covered in detail in th

Ambient Light

An ambient light adds a constant overall light source which affects the whole scene, the ambient light is generally used in conjunction with other lights, and can be used to eliminate black areas,

If overdone, Ambient will 'wash out' a render, it should (in my opinion) be used for subtle changes in lighting

Fig 222



PARAMETERS

- **Diffuse color** – Sets the diffuse colour of the light.
- **Diffuse affect** – Sets the relative intensity (power) of the diffused light.

The two settings are effectively interlinked and are used together, mostly the diffuse affect can be set to 1 or two, and the diffuse colour left at grey scale set towards the black (left) end of the scale, higher values can wash out the scene.

The image below, shows how easy it is to wash out a scene.

Fig 223

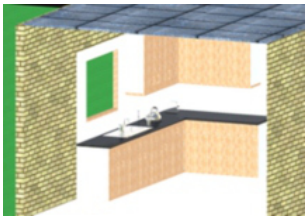
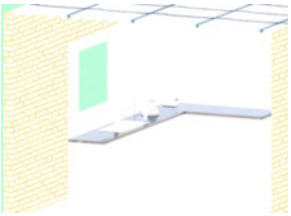


Fig 224



In the images below. Fig 225 has just a beam light, with no other illumination set.

Fig 226 has the bean light + added ambient set to Diffuse affect = 3. This allows one, to just make out the electric cord.

Fig 227 has the same beam + Ambient, but the diffuse colour was set to a brown. By adding a touch of ambient one can see detail not available with beam light alone.

Fig 225

Fig 226

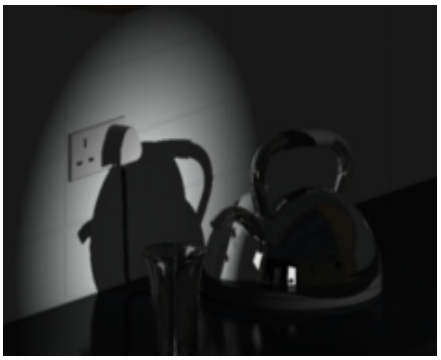
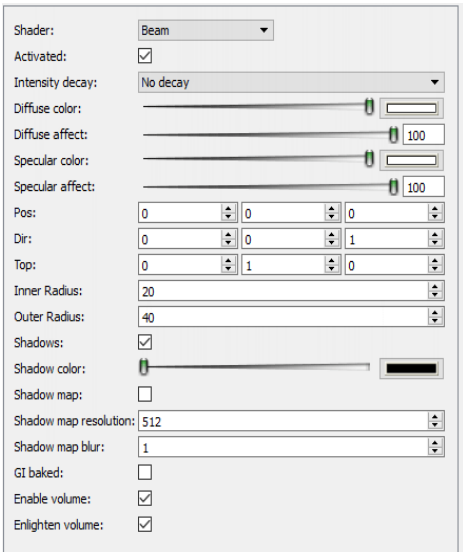


Fig 227

One important note. One would not normally use an ambient luminance in any scene where other luminance, like Physical or Sky are used. With this in mind. One would not normally use Ambient in an architectural scene.

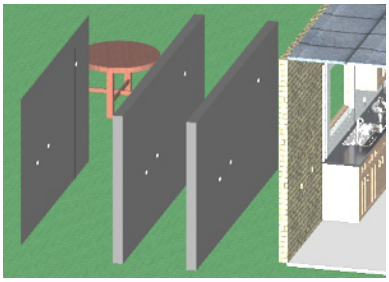
Beam Light.

Fig 228



A Beam light creates a single beam of light. Much like a laser. This light, in it's normal state, is of infinite length which can pass through objects, as can be seen from the white dots in Fig 229.

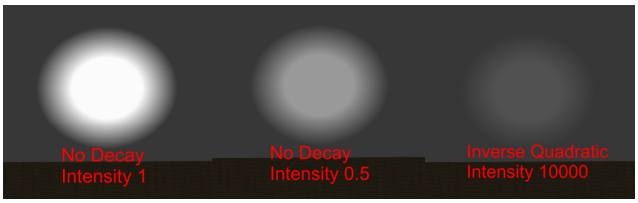
Fig 229



The light is held in check by use of the shadow tick box (with limitations). Or by using the Intensity decay. Intensity decay reduces the light power as distance increases.

PARAMETERS

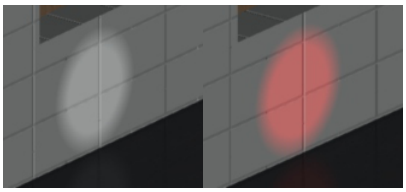
Fig 230 Fig 231 Fig 232



Intensity decay – This Set the rate at which the lights power diminishes with distance.
See page 30 for intensity decay types.

Diffuse color – Sets the diffuse colour of the light.
Diffuse affect – Sets the relative intensity (power) of the diffused light.

Fig 233 Fig 234

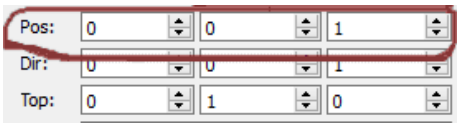


The diffuse colour midpoint of the slider is the selected colour, moving the slider toward 0% darkens the colour, moving towards 100% lightens the colour, eventually reaching white. The setting of these is trial and error and personal preference. Fig 234 changed to red

Specular color – Sets the specular color of the light.
Specular affect – Sets the relative intensity (power) of the Specular light.

Specular works on coloured objects, i.e. objects with no material attached, it is ignored when materials are applied.

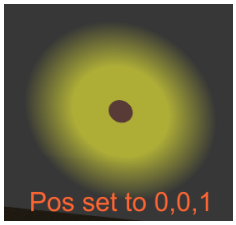
Fig 235

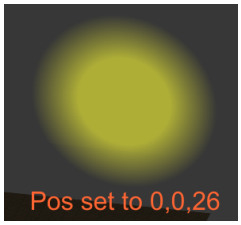


Pos – This Sets the X,Y,Z position of the light relative to the object being used as the luminance. **The following assumes 'Load for Luminance only' is not checked.**

This should (almost) always be set to a value. In Fig 236. The luminance object is a 50mm sphere, with the light direction set to positive Z. This shown an incorrect Pos setting (the dark dot in the centre).

Fig 236 Fig 237

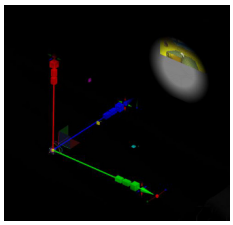


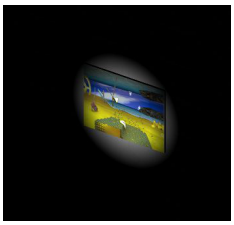


In Fig 237 the position was moved. Because the luminance is now coming from the edge of the sphere, no dot is present. In most situations the Position should be, at the edge of an object or greater. An exception would be fully transparent objects.

If one has the luminance on a fixed object. Which is not in line with the target. The light position can be altered to suit (In object/entity coordinate system).

Fig 238 Fig 239

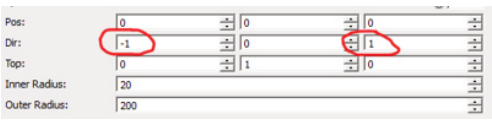




In Fig 238. The original object was out of position. Therefore In Fig 239. The light position was change to 62,0,5. 62 moves the light 62 units in x object direction (the red axis), the 5 moves light to object outer shell

Dir – Sets the X,Y,Z direction of the light relative to the position

Fig 240



Dir, determines where the light will be pointing. It uses the objects orientation, which may be different from world. Generally one would use 1's and 0's. Though other figures can be used.

In Fig's 241 - 243. The luminance is a fixed object. The main direction is set to positive Z (0, 0, 1). We could move the light Pos values, but we will alter Dir instead. A value of 0, 1, 1 would direct light 45 degrees, we want a lot less, so we use -0.18 and 0.19. It has the obvious drawback. In that angling the beam does elongate it slightly.

Fig 241

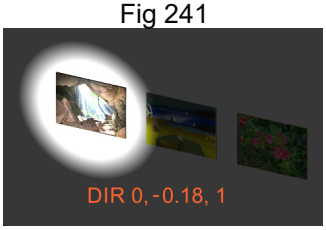
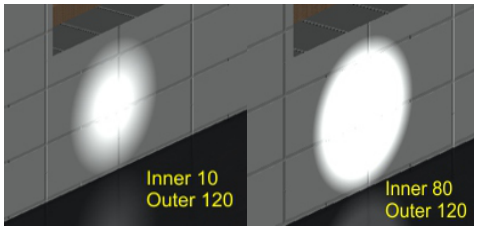


Fig 242

Fig 243

Top – Defines a second axis which is perpendicular to the direction, used for the rotation, if the lights display is not circular. A beam light is circular. Therefore under normal circumstances it wont make any difference. Obviously do not use the same axis as DIR.

Fig 244 Fig 245



Inner Radius – Sets the inner radius for the light.
Outer Radius – Sets the outer radius for the light.

The inner radius defines the bright spot, the light then decrease between inner and outer, with no light beyond outer radius. As shown in Fig's 244 & 245.

Shadows – Sets whether the light will generate shadows, or not.
As can be seen below, In Fig 246 we have no shadows the beam is constant as it passes through objects. In Fig 247, shadows is turned on with the shadow colour at medium, the beam / shadows are muted as they pass trough objects. In Fig 248 the shadow colour is set to black, the beam still passes through, but the shadow cast by objects / walls means the beam is no longer visible.

Fig 246

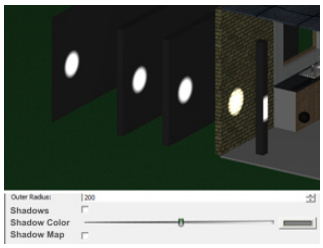


Fig 247

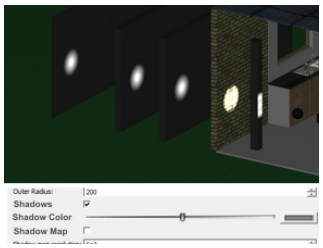


Fig 248

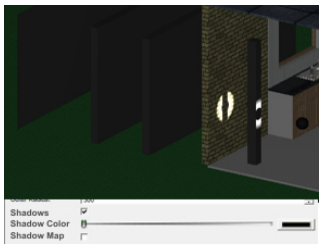


Fig 249

Shadow Color – Specifies the color of shadows from the light.
As can be seen above. If the colour is a grey colour it will act like diffusing the bean. However change the colour, and some unusual lighting can be achieved. In Fig 249 we change the colour to light yellow, which completely changes how the rendering will look. This setting is purely personal preference as to what colours one finds appealing. Turning on Shadow Map, will override this setting.

Shadow Map – Shadow maps can accelerate the rendering of shadows, but usually as the cost of some quality. Shadow maps can override any shadow slider setting.

Shadow Map Resolution –
Sets the resolution of a shadow map. Values must be multiples of 2. As shown in Fig's 250 and 251

Fig 250

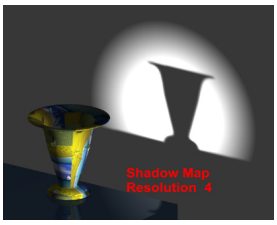
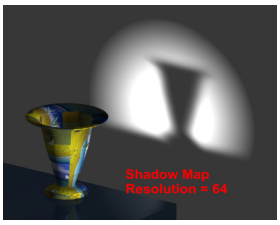


Fig 251



Shadow Map Blur – Sets a blur factor on the shadow map, reducing sharp and jagged edges + transitions. Fig 252 has the blur set to 1. Fig 253 has blur set to 80

Fig 252

Fig 253

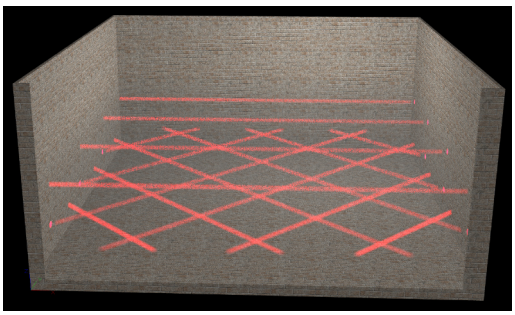
GI Baked.

Redway state. *"By baking, we mean pre-calculating a given information, storing it and restituting it, faster, as a later rendering phase".* It is used with global illumination rendering. And may not work correctly if global is set to use the GPU.,

Enable Volume. Used with volumetric environmental effects (see RedSDK environment).

Bean lights show volume effect really well. Although somewhat limited to a circular stream of light. The volumetric effect is discussed on page … . As an example of beam luminance volume. This simple laser alarm system uses 16 beam luminance, with a volume box within the building.

Fig 254



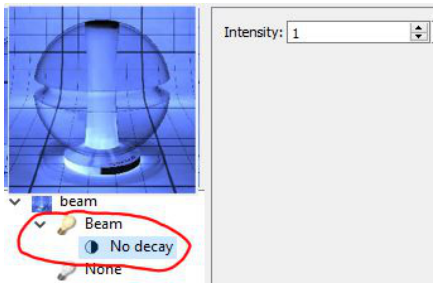
Enlighten volume. Used with volumetric environmental effects (see RedSDK environment).

In theory, this should only highlight areas within a volume made by other luminance, However in tests in TC 2020, it does the same job as enable volume. Unknown if this is machine specific or not. Also I have seen it where the volume didn't work if this was unchecked. Simply test and see what works best for you.

Intensity.

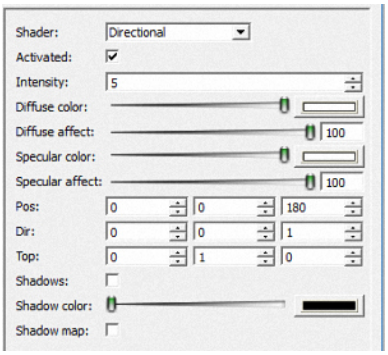
A last setting is Intensity**,** accessed by clicking the '+' sign against Beam in the dialog box, This reveals the 'decay' settings. The name will change to that entered as the Intensity Decay. This decay setting, increases or decreases the power of the light / luminance. It can be trial and error which value is best in any given situation. (See intensity decay page).

Fig 254b



Directional Light.

Fig 255



A Directional light shines along a path defined by Dir. And without shadows, continues infinitely in that direction. It is also the same intensity, with an infinitely wide light,

Directional luminance doesn't have any restrictions if shadows is not selected, its a bit like the light goes around the world an comes back to itself.

PARAMETERS

Fig 256

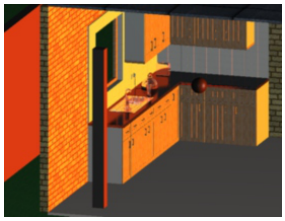
Fig 257

Intensity – Set the intensity of the light.

Unlike some other luminance. Intensity Is on the main dialog page. It alters how bright the light will shine. This should be adjusted with the diffuse affect. Fig 256 has intensity = 1. Fig 257 has intensity = 5. NOTE shadows is turned off.

As can be seen, the light can transmit though objects.

Fig 258



Diffuse color – Sets the diffuse color of the light.

The majority of the time one would leave this at a grey colour. Changing to another colour can produce unusual effects. In Fig 258, the colour was changed to red. (Diffuse affect set to 100).

Note - when selecting a colour the slider will be at the middle, moving the slider up or down makes the selected colour lighter or darker.

Fig 259

Diffuse affect – Sets the relative intensity (power) of the diffused light.

In Fig 259. The affect has been reduced towards Black.
Diffuse affect = 10.

Specular color – Sets the specular color of the light.

Fig 260

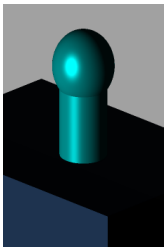
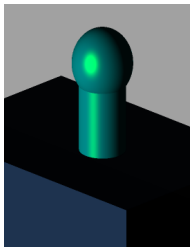


Fig 261



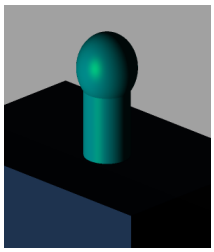
In Fig 260 the original specular colour. The object is set to cyan. With no material applied.

In Fig 261. The specular colour was set to yellow.. This changed the colour of the object, especially apparent in the specular bright spot.

Specular affect – Sets the relative intensity (power) of the Specular light.

Reducing the affect towards black diffuses the brightness of the specular affect. In Fig 262. It was reduced to 45 (default is 100).

Fig 262



The Specular affect is over-ridden by material settings, but shows up on non-material faces (pen / brush coloured objects),

Pos – Sets the X,Y,Z position of the light relative to the object to which it is attached.

Position is an odd one with Direction Light. The light starts at the extents, with the light being of infinite length and width. Therefore moving the position wouldn't make any difference. Indeed in tests, no difference could be found. It is unknown, if this is an fault on my computer or not.

Dir – Sets the X,Y,Z direction of the light relative to the source objects position.

In Fig 263, the direction was set at 0, 0, 1, the z axis being horizontal.

In Fig 264. The direction was changed to -1, 0, 1. Meaning the light is directed at 45 degrees downwards. Hence the light makes it over the wall and shines in the room. The light source object, is actually at the bottom of one of the cupboards. And as can be seen, the source position makes no difference to the actual lighting.

Fig 263

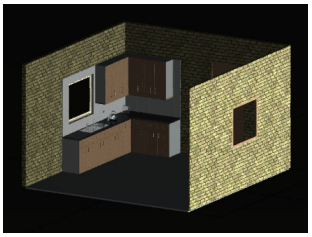
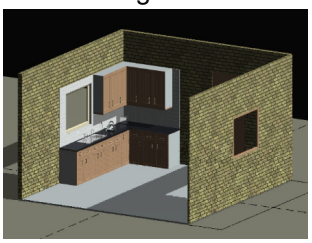


Fig 264



Top – Defines a second axis which is perpendicular to the direction, used for the rotation, if the lights display is not circular.

Again. As the light emits from outside the drawing, in infinite range. changing the top setting, may not make any difference.

Shadows – Sets whether the light will generate shadows, or not.

Shadow Color – Specifies the color of shadows created by the light. Shadows are relative to the drawing extents, rather than a specific luminance position.

As can be see from the two drawing on the right. The luminance is attached to a cupboard door. Fig 265 has Shadows not ticked. In Fig 266, shadows was ticked and colour changed to red. As shown, the shadows are created, irrespective of where the source object luminance is attached.

Fig 265

Fig 266

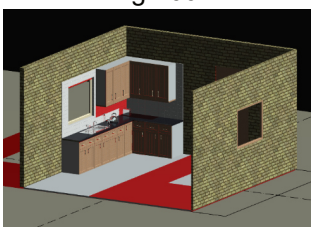
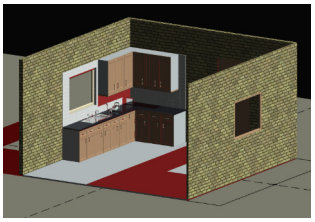


Fig 267



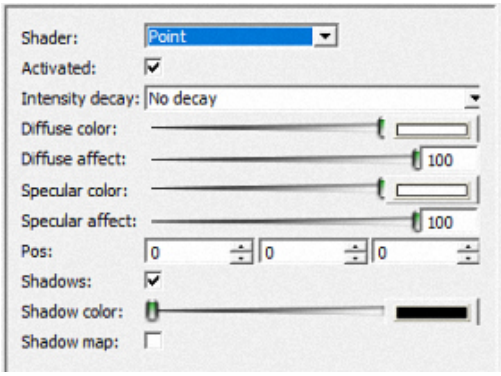
Shadow Map – Shadow maps can accelerate the rendering of shadows, but usually as the cost of some quality. Shadow maps can override a shadow slider setting. Shadow map uses the shadow colour in the calculations.
As in Fig 267.

GI Baked.
Redway state. *"By baking, we mean pre-calculating a given information, storing it and restituting it, faster, as a later rendering phase".* It is used with global illumination rendering. And may not work correctly if global is set to use the GPU.

Enlighten volume. Used with volumetric environmental effects (see RedSDK environment). The tick box for enable volume is missing. Therefore a directional light will only enlighten a volume created by another light.

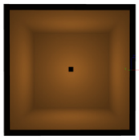
Point Light

Fig 268



The point light is a centric light. It produces lighting in all directions around the point light.

Fig 269

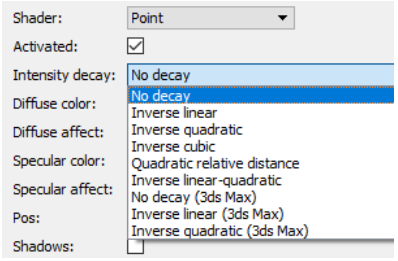


However the light is brightest on each plus or minus orthogonal axis. Even if No decay is set. As can be seen in Fig 269. The point luminance is attached to the black dot in the centre.

The light Is bright orthogonal, and diminishes towards the corners. Whilst understandable with quadratic decay, (the corners are further away than orthogonal areas). With No decay one may expect equal lighting.

PARAMETERS

Fig 270

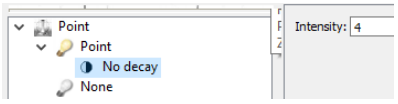


Activated. - Turns the light on or off

Intensity decay – Sets the type of intensity used.

Select the desired intensity decay from the drop down box. The actual intensity is set, by expanding the point selection, and clicking on No decay.. Fig 271

Fig 271



See separate Intensity page 30. For examples.

Diffuse color – Sets the diffuse color of the light.
Diffuse affect – Sets the relative intensity (power) of the diffused light.

Diffuse colour, can be used to change the colour of a light. But mostly it would be left as grey scale. In which case diffuse colour would be used to reduce the lighting effect.

Fig 272

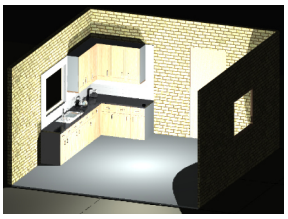
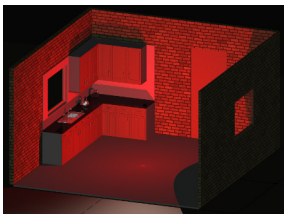


Fig 273



This combined with Diffuse affect act a bit like a dimmer switch. Both are used to fine tune the lighting. Fig 272 has the default setting. Colour - white, affect = 100. In Fig 273 . The colour, was changed to red, affect = 50.

Specular color – Sets the specular color of the light.
Specular affect – Sets the relative intensity (power) of the specular light.

Specular affects objects of pen colour, it I ignored if an object contains materials. In Fig 275, the affect was turned down.

Fig 274

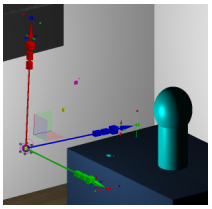
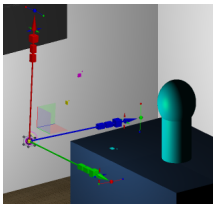


Fig 275



Pos – Sets the X,Y,Z position of the light relative to the object to which it is attached.

With the position left at 0, 0, 0. (Centre of the source object). And shadows not ticked. The light will will shine without restriction. Because the line shines greater on the orthogonal axes, moving the position will affect the illumination.

Fig 276 has a default 0, 0, 0. Fig 277 has the position changes to 0, 400, 100. This moves the light horizontally (sideways) by 400 units, and forward by 100 units.

Fig 276

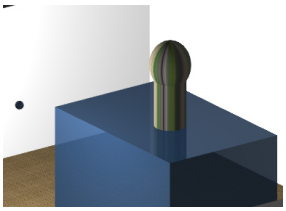
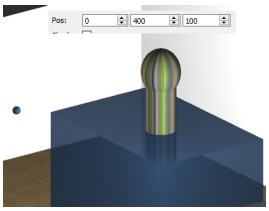


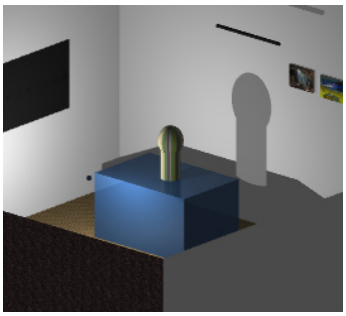
Fig 277



When shadows is turned on. The position needs moving to the shell of the source object. Assuming 'Load for luminance only' is not checked. To ensure the light will shine outside the source object.

Shadows – Sets whether the light will generate shadows, or not.

Fig 278



Shadow Color – Specifies the color of shadows created by the light.

Fig 278. Has shadows turned on. With shadow colour set to mid grey.

Shadow Map. Point luminance has a shadow map tick box, but no blur setting. The shadow colour can be used tone down the shadow map. Shadow maps can accelerate the rendering of shadows, but usually as the cost of some quality. Shadow maps can override a shadow slider setting.

GI Baked.

Redway state. *"By baking, we mean pre-calculating a given information, storing it and restituting it, faster, as a later rendering phase".* It is used with global illumination rendering. And may not work correctly if global is set to use the GPU.,

Enable Volume. Used with volumetric environmental effects (see RedSDK environment).

Rectangular Area Light

Fig 279

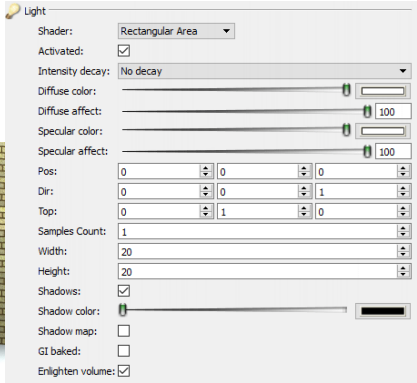


Fig 280

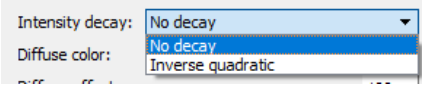
Area light sources are surface lights that emit light from a rectangle.

The settings are mostly the same as other lights. Except for the width and height of the rectangle used to calculate the light.

A rectangular area light provides a quick way to illuminate a scene. Including shadows.

PARAMETERS

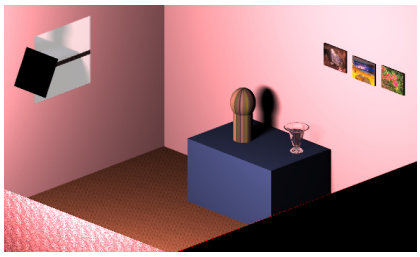
Fig 281



Intensity decay – Set the rate at which the lights power diminishes.

For rectangular area, the decay is limited to No decay, or Inverse Quadratic. No decay can, in some cases, produce unwanted bright spots.

Fig 282



Diffuse color – Sets the diffuse color of the light.
Diffuse affect – Sets the relative intensity (power) of the diffused light.

Diffuse colour, can be used to change the colour of a light. But mostly it would be left as grey scale. In which case diffuse colour would be used to reduce the lighting effect. Moving the slide towards the left will diffuse, or darken the tone or colour.

Fig 282, has the colour changed to pink, and the diffuse affect reduced to 75

Specular color – Sets the specular color of the light.
Specular affect – Sets the relative intensity (power) of the Specular light.

Specular affects objects of pen colour, it I ignored if an objects contain materials. In practice, on many occasions. One would apply a material to objects, rather than rely on a specular effect.

Fig 283

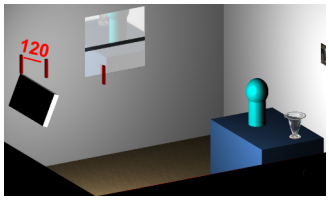
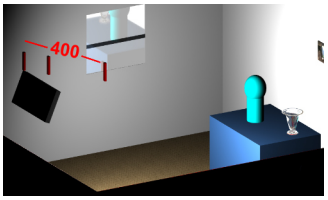


Fig 284



Pos – Sets the X,Y,Z position of the light relative to the object to which it is attached.

Pos uses the entity coordinates, Fig 283 is set to Pos z axis or 0, 0, 120. Fig 284 has the Pos increased to 400 units.

Fig 285

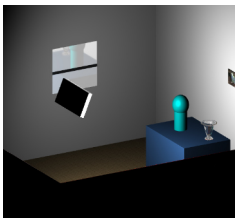
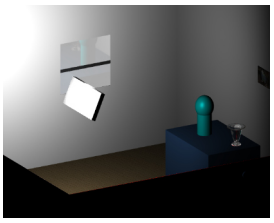


Fig 286



Dir – Sets the X,Y,Z direction of the light relative to the object.

In most cases one would put a '1' in the relevant x, y, or z boxes. Fig 285 is set to dir 0,0,1 (+ Z axis). In Fig 286, the Dir was moved to 1, 0, 0 (-X axis).

Top – Defines a second axis which is perpendicular to the direction, used for the rotation, if the lights display is not circular. This is one luminance where top can make a difference, in some circumstances.

Samples Count – Sets the number of samples used to populate the light surface. Must be greater than 0. In some luminance, increasing count improves render, at the expense of time. However in at est no obvious difference was found, between 1 and 32

Width – Sets the width of the defining rectangle.
Height – Sets the height of the defining rectangle.

Width and height affect the amount of light emitting. A larger rectangular area, means light is coming from a larger source. See Fig's 287 / 288.

Fig 287

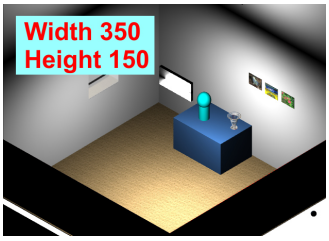
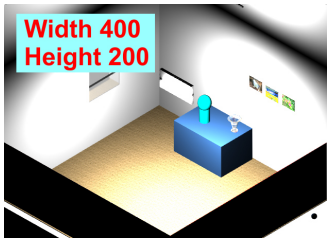


Fig 288



Shadows – Sets whether the light will generate shadows, or not.
Shadow Color – Specifies the color of shadows created by the light.

The images on the right show variations in the shadow effect.

Fig 289. Shadow ticked. Colour dark grey. This shows the lighter shadow from the glass.

Fig 290. Shadow ticked. Colour light grey. This simply lightens all the shadows.

Fig 291. Shadow map ticked. Colour dark grey. Note shadow map darkens the shadow of the glass.

Fig 292. Shadow map ticked. Colour set to light green. Which changes the whole shadow.

Fig 289

Fig 290

Fig 291

Fig 292

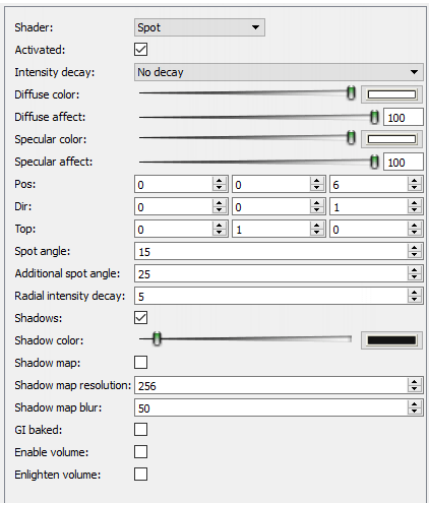
GI Baked.

Enlighten volume. Used with volumetric environmental effects (see RedSDK environment).

This determines whether the light will affect volumetric effects, produced by other luminance. (Beam, spot, point).

Spot light

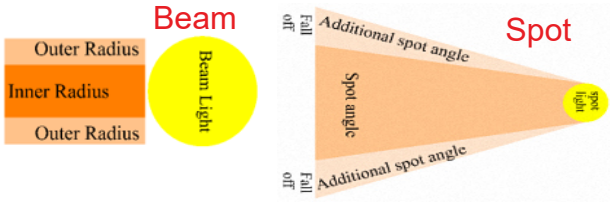
Fig 293



The spot light source is a centric light. It produces an isotropic lighting in all directions issued from the spot cone apex. The lighting emitted by the spot light decreases at the spot cone borders. A spot light is defined in a two steps code:

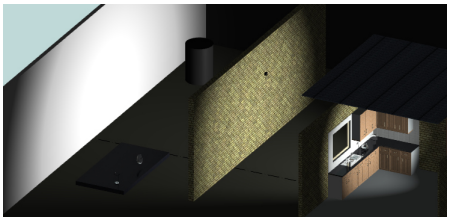
The main difference between spot and bean, is the angle. Fig 294

Fig 294



A spot light, without shadows, Is an infinate light. In that it shines through objects. As can be seen in Fig 295
The shadow setting can reduce the effect.

Fig 295



Spot light PARAMETERS

Intensity decay – This Set the rate at which the lights power diminishes with distance.

The required decay is selected from the drop down box.

The Intensity itself, can be set by expanding the spot luminance. As in Fig 297.

The option will change name depending on which decay has been chosen. See page 30 for decay types.

Fig 296

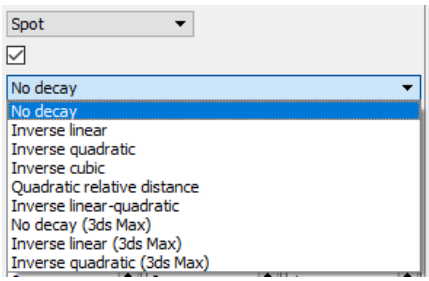
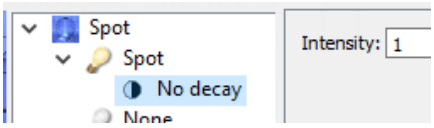


Fig 297



Diffuse color – Sets the diffuse colour of the light.
Diffuse affect – Sets the relative intensity (power) of the diffused light.

In grey scale. The diffuse colour and diffuse affect, is used to darken or lighten the luminance.

Using the colour picker (at the right of the slider), and the colour slider, to set the desired colour. And use the diffuse affect to lighten or darken the render.

Fig 298. Was given an orange colour, with the affect set to 75.

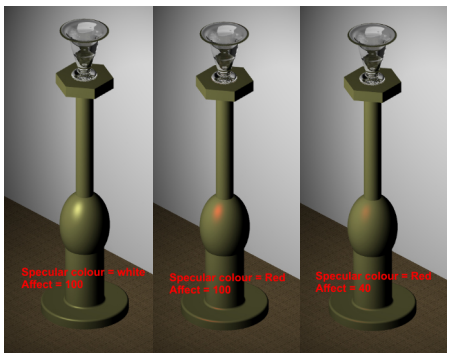
Fig 298

Specular color – Sets the specular color of the light.
Specular affect – Sets the relative intensity (power) of the Specular light.

Specular works on coloured objects, i.e. objects with no material attached, it is ignored when materials are applied.

Fig 299 is default.
Fig 300 is set to Red.
Fig 301 is set t Red with a reduced affect.

Fig 299 Fig 300 Fig 301



Pos – This Sets the X,Y,Z position of the light relative to the object being used as the luminance. **The following assumes 'Load for Luminance only' is not checked.**

Depending on the source object. If shadows or shadow map is ticked. Sometimes it may be necessary to move the position, to the edge or outside the source object. Most likely the position will need to be moved at least half depth of the source object.

An exception would be, if the source object is a single sheet (for example, an extruded line).

Fig 302. Shows a sheet and a Sphere (called a cylinder by mistake in the images). with luminance attached. The Pos is set at default 0,0,0. As can be seen, because the pos is inside the sphere the luminance is not showing.

Fig 303. Shows the result after the Pos is moved to the edge of the sphere. In this case it moved 20 units in the plus z axes.

Fig 302

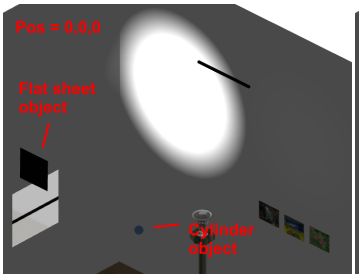
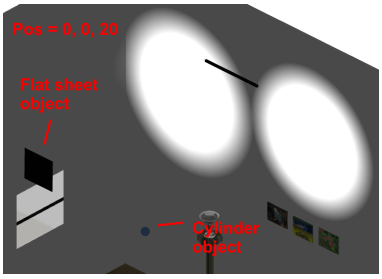


Fig 303



Dir – Sets the X,Y,Z direction of the light relative to the object.

Dir, determines the direction the light will shine from the source. Using the objects coordinate system.

In most cases one would put a '1' in the relevant x, y, or z boxes (0, 0, 1). To determine the direction the light will shine from the source. Other figures can be used, Including putting 1, 0, 1 for a 45 degree angle.
Fig's 304 and 305, show the difference in light position, when differing values are used.

Fig 304

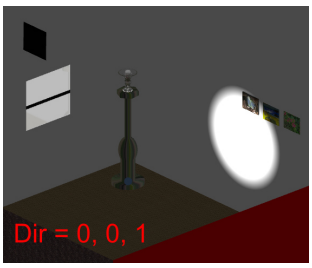
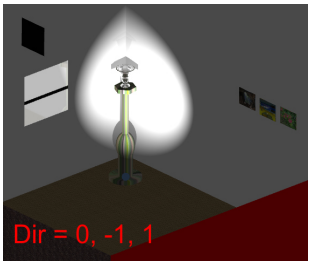


Fig 305



Top – Defines a second axis which is perpendicular to the direction, used for the rotation, if the lights display is not circular.

Spot lights are circular, therefore Top is unlikely to make any difference.

Spot Angle - Sets the inner brighter part of the light cone.

Allows for the bright area to be adjusted, Therefore allowing it to light a small object or a whole room.

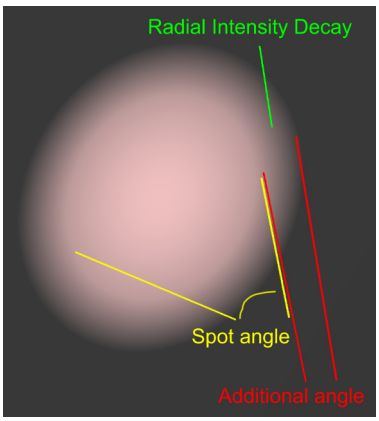
Additional Spot Angle - Sets an additional outer light cone.

Adds a second outer cone, which is used to fade the light before diminishing.

Radial Intensity decay - Sets the decay, between the inner and outer conses.

Defines the density of light in the outer cone, as it travels from the inner cone to it being extinguished. Essentially it fades the light rather than having a sharp cut off point.

Fig 306

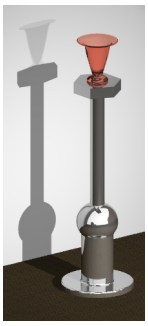


Shadows – Sets whether the light will generate shadows, or not.
Shadow Color – Specifies the color of shadows created by the light.

Allows for shadows to be created, and the colour adjusted. When left at a grey scale colour, the slider is used to lighten or darken the shadows.

The shadows generally have a 'hard' edge, rather than being blended into the background. But this depends on other settings, and any other lighting being used.

Fig 307

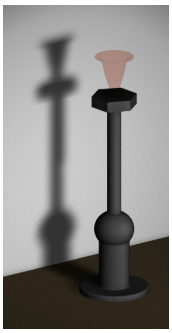


Shadow Map - Turns on the use of a shadow map.

Shadow Map Resolution – Sets the resolution of a shadow map. Values must be multiples of 2.

Shadow Map Blur – Sets a blur factor on the shadow map, reducing jagged edges + transitions.

Fig 308



Shadow maps can accelerate the rendering of shadows, but usually as the cost of some quality. The higher the map resolution, the closer it follows the edges of the object, Blur is just that. It blurs the shadows to blend in more with its surrounding. Fig 308

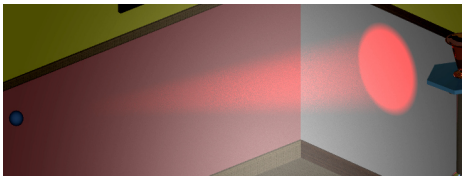
GI Baked.

Redway state. *"By baking, we mean pre-calculating a given information, storing it and restituting it, faster, as a later rendering phase".* It is used with global illumination rendering. And may not work correctly if global is set to use the GPU.,

Enable Volume. - Turns on the use of volumetric effect. Used with volumetric environmental effects (see RedSDK environment).

Enlighten volume. Used with volumetric environmental effects (see RedSDK environment).

Fig 309

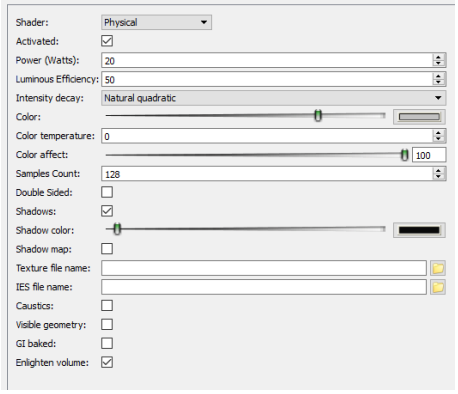


Volumetric effect is light bouncing off particles in the air. And work well with Beam and spot luminance.

Volumes also require an environment set to accept volume.

See RedSDK volume environment.

Physical Light



A physical light uses the definition of light sources from physical parameters issued from lamp manufacturers. A physical light can be associated to a mesh that defines the light surface that will emit light in the scene. Therefore a physical light can be of any shape.

Physical lights are rendered on the CPU using sampling and are approximated on the GPU. There are two possible approximations for a physical light on the GPU:

If the physical light is linked to a rectangular mesh, then the physical light is automatically turned into an Area light.

If the physical light is linked to a mesh with any other geometric shape, then the physical light is automatically turned into a Point light.

Physical lights only support ray-traced shadows on the CPU. Physical lights converted to point lights are set to use ray-traced shadows. (shadow maps for point lights can be expensive and are disabled in this case). Physical lights converted to area lights area set to use ray-traced shadows or maps.

PARAMETERS

Power – Sets the power of the light in Watts.

This simply increases or decreases the amount of light emitted from the source object. Fig 310 = 12 Watt.
Fig 311 = 25 Watt

Fig 310

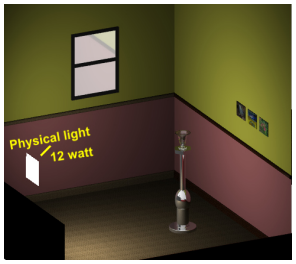
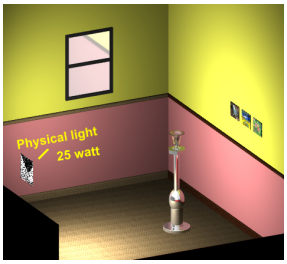


Fig 311



Luminous Efficiency – luminous efficiency of the light in percentage (%).

Like a diffuse setting. I look on it as fine tuning the luminance power, without having to keep altering the power setting.

Intensity decay – Set the rate at which the lights power diminishes. Intensity decay is preset at Natural Quadratic.

Note. When looking at the room images, it appears as though the corners are not lit correctly, This is assumed to be due to the quadratic equation. The short distance between the light and a wall, produces the brighter light. The corners are further away from the light source, and therefore are calculated to receive less light. In a circular room, all walls were lit with an equal amount of light.

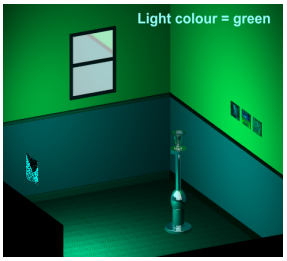
Color – Sets the color of the light.

The alteration can either be subtle or dramatic, depending on the chosen light colour.

Fig 311 = Normal grey colour. Fig 312 = Magenta
Fig 313 = Green. As can be seen, varying the colour allows one to create any lighting condition

Fig 312

Fig 313

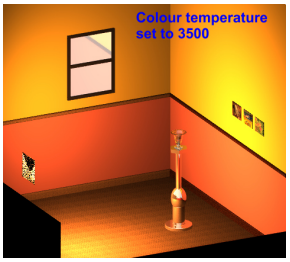


Color Temperature – An alternate color value, which supports definition of the color by temperature, in Kelvin.

Colour temperature can be used instead of Color. It can change the colour from warm to cool. Note. Some colours, like bright green, are not available as a colour temperature.

Setting the temperature to 0, switches to using the colour setting.

Fig 314



Color Affect – Sets the relative intensity (power) of the color light.

Colour affect can be used to diffuse the chosen colour, it does not affect colour temperature. Moving the slider to the left. Darkens the colour used.
Samples Count – Sets the maximum number of light samples.

Double sided - sets single or double sided surfaces.
When a physical luminance is applied to a surface/sheet. For example, and extruded line. It has two faces, front and back.

Double sided tells the render, whether the light emits from just one face of both faces. In Fig 315. Double sided is not ticked. With Fig 316, double sided is ticked.

Double sided can reduce the light power. Therefore it is often necessary to increase the power, when double sided is used.

Fig 315

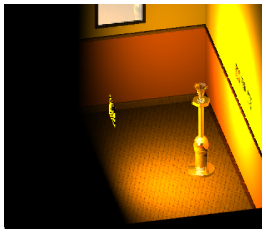
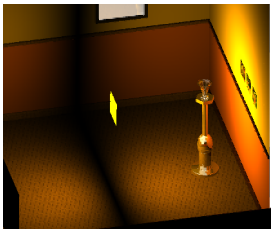


Fig 316



Shadows – Sets whether the light will generate shadows, or not.

Shadow Color – Specifies the color of shadows created by the light. In grey scale, it is used to lighten or darken the shadows. It can be used to change the colour of just the shadows.

The actual colour may differ depending on whether the light has been given a colour.

Fig 317

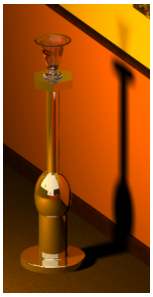


Fig 318

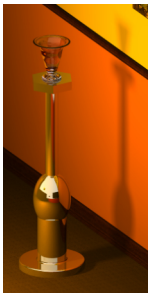
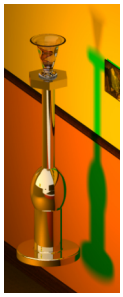


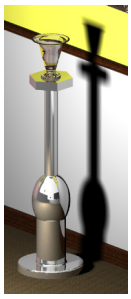
Fig 319



Shadow map – Shadow maps are generated by testing whether or not each specific pixel is visible from the light source

One difference from standard shadows. Is when an object has transparency. The shadow map may ignore the transparency and treats the object as a solid colour. Fig 320.

Fig 320



Texture File Name – Rather than using a single constant color for the whole light.

You can setup a texture from which light color will be read. Using a texture, you can simulate projectors or complex light filters.
Fig 321, is the texture applied.
Fig 322, is before the texture is used.
Fig 323, is after the texture is applied.

Fig 321

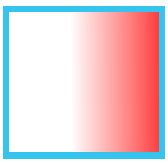


Fig 322

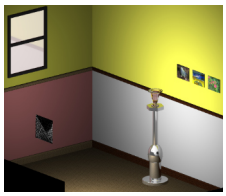


Fig 323

IES File Name – Manufacturers measure the behaviour of their bulbs and store their directional contribution into IES files. Those files are available for most of the bulb models directly from the manufacturers' web sites. Even if IES files store information about the shape of the emitter, they are not used here.

IES files can bypass some of the other settings. For example turning on double sided. The image in Fig 324 on the right uses an IES file from Phillips lighting.

Fig 324

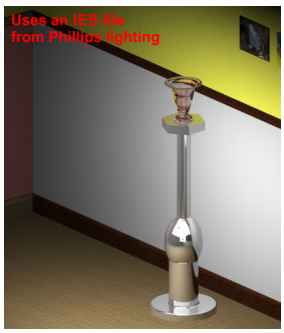
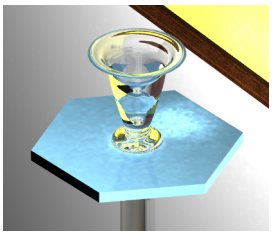


Fig 325



Caustics – Specifies whether the light will generate caustic effects.

Caustics are available with a physical luminance. However, as the standard physical luminance is generally used as an overall light source, the caustic direction is not controllable. As it is with spherical physical.

The caustic effect is shown on the blue table in Fig 325. As scattered particles of light.

Visible Geometry - Specifies whether the physical geometry of the associate 3D objects will be seen.

The effect may differ depending on what type of object the luminance is attached to. Experiment to see the effect on ones chosen source object.

GI Baked.

Redway state. *"By baking, we mean pre-calculating a given information, storing it and restituting it, faster, as a later rendering phase".* It is used with global illumination rendering. And may not work correctly if global is set to use the GPU.,

Enlighten volume. Used with volumetric environmental effects (see RedSDK environment).

Volume is a special effect, which can show light rays passing through a space. Only certain lights can create a volume effect. This is covered in detail in the environment section.

This setting determines whether the physical light can affect the volumetric effects produced by other luminance. E.g. Beam, Spot, Point.

The difference will vary depending on the intensity of the physical light.
Fig 326 has enlighten turned off.
Fig 327 has enlighten turned on. And as can be seen the volume effect is lighter, because it is also affected by the physical light.

Fig 326

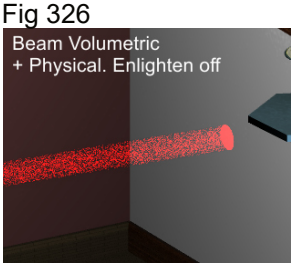
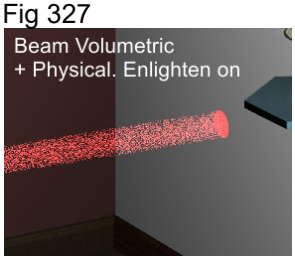


Fig 327



Rectangular Physical Light

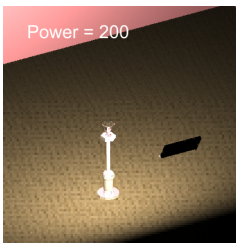
A Physical Rectangle light uses the definition of light sources from physical parameters issued from lamp manufacturers. A Physical Rectangle light uses a rectangle that defines the light surface that will emit light in the scene.

See Physical light for further notes on how physical lights are rendered.

PARAMETERS. Note. Many parameters are the same as for Physical, so only have a brief description. And a few images. See physical for image representations.

Fig 328

Fig 329



Power – Sets the power of the light in Watts.

Increases or decreases the power of the luminance.

Luminous Efficiency – Sets the luminous efficiency of the light in percentage (%).

Like a diffuse setting. I look on it as fine tuning the luminance power, without having to keep altering the power. Setting.

Intensity decay – Set the rate at which the lights power diminishes. Intensity decay is preset at Natural Quadratic.

Color – Sets the color of the light.

The alteration can either be subtle or dramatic, depending on the chosen light colour.

Color Temperature – Alternate color value which supports definition of the color by temperature in Kelvin.

Colour temperature can be used instead of Color. It can change the colour from warm to cool. Note. Some colours, like bright green, are not available as a colour temperature.

Setting the temperature to 0, switches to using the colour setting.

Color Affect – Sets the relative intensity (power) of the color light.

Colour affect can be used to diffuse the chosen colour, it does not affect colour temperature. Moving the slider to the left. Darkens the colour user.

Samples Count – Sets the maximum number of light samples.

Pos – Sets the X,Y,Z position of the light relative to the object to which it is attached.

This uses the objects coordinate system. Altering the position allows for the light to be altered, even if the source object cannot be moved. (Example. If attached to a fixed object like a cupboard).

Fig 330

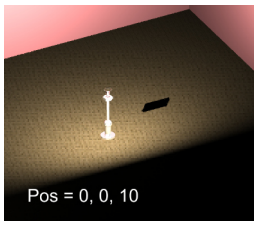


Fig 331

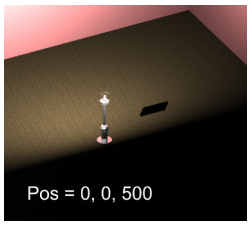


Fig 332

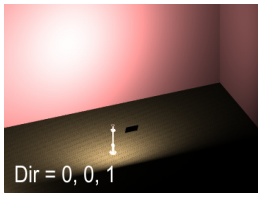
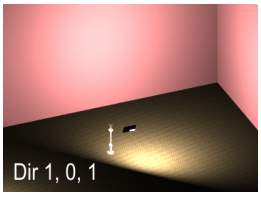


Fig 333



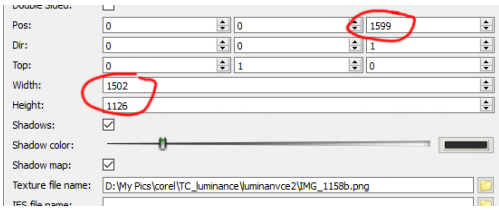
Dir – Sets the X,Y,Z direction of the light relative to the position.

Dir sets the axis of the source object. From which the luminance will emerge. Normally one would just put a 1 in the relevant axis,. For a 45 degree light, one could put a 1 in two axis boxes. Other values can also be used.

Top – Defines a second axis which is perpendicular to the direction, used for the rotation, if the lights display is not circular. Should not be the same axis as Dir.

Samples Count – Sets the number of samples used to populate the light surface. Must be greater than 0

Fig 334



Width – Sets the width of the defining rectangle.
Height – Sets the height of the defining rectangle.

These set the width and height of the light not the object. Increasing the width or height, can reduce the brightness. Therefore it may also require increasing the power, or colour slider to compensate.

Fig 335 and Fig 336. Shows different width settings, and as can be seen it can create a totally different lighting effect.

Fig 335

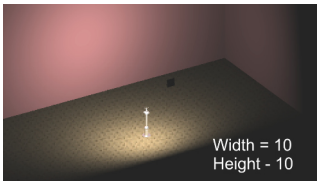
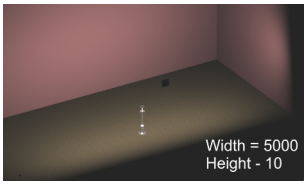


Fig 336



Shadows – Sets whether the light will generate shadows, or not. Same as other luminance

Shadow Color – Specifies the color of shadows created by the light. Same as other luminance

Texture File Name – Rather than using a single constant colour for the whole light, you can setup a texture from which light colour will be read. Using a texture, you can simulate projectors or complex light filters.

For a projection one needs the rectangle. The projection distance is on, or just short of the object the light will shine on. The rectangle on the luminance is situated at the source. Therefore it is necessary to alter the position setting to the correct distance.
In the Fig 337. the distance is 1600. Therefore we will set the Pos in the rectangular physical light to 1599
The result is shown in Fig 338.

Fig 337

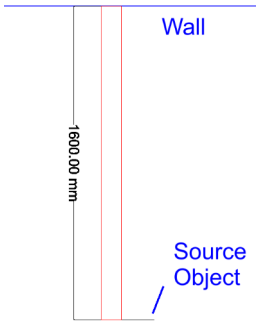
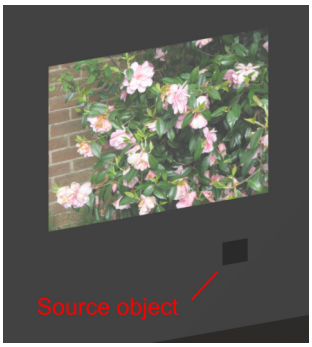


Fig 338



The width and height were set to the size of the projected image, Though this is something that can be altered to suit.

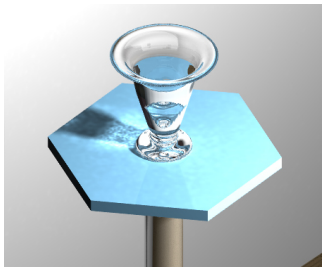
One also needs to balance the power and luminance efficiency, to prevent the light being too bright. And obscuring the image.

If only a white area is seen, the power may be too high to show the texture image.

Note. It may require visible geometry turned on. In tests, this seemed to depend on the type of render mode used. Advanced render, required visible geometry.

IES File Name – Manufacturers measure the behavior of their bulbs and store their directional contribution into IES files. Those files are available for most of the bulb models directly from the manufacturers' web sites. Even if IES files store information about the shape of the emitter, they are not used here.

Fig 339



Caustics – Specifies whether the light will generate caustic effects.

Caustics are particles of light scattered by a transparent object. Fig 338. Physical luminance can generate the caustic effect. The effect is controlled by the material settings, like IOR, anisotrophy, etc.

Visible Geometry - Specifies whether the physical geometry of the light itself will be seen. This is the actual light, not any object from which the light emanates. It can be used for turned on light bulbs,

As seen in fig 341, the luminance light itself shows as white,

To obscure the object itself, one would use object properties, 'load for luminance only'.

Fig 340

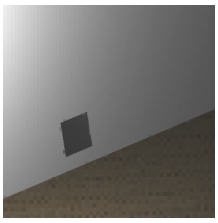
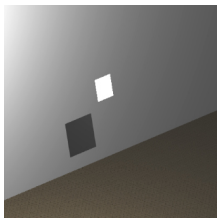


Fig 341



GI Baked.

Redway state. *"By baking, we mean pre-calculating a given information, storing it and restituting it, faster, as a later rendering phase".* It is used with global illumination rendering. And may not work correctly if global is set to use the GPU.,

Enlighten volume. Used with volumetric environmental effects (see RedSDK environment).

Volume is a special effect, which can show light rays passing through a space. Only certain lights can create a volume effect. This is covered in detail in the environment section.

This setting determines whether the physical light can affect the volumetric effects produced by other luminance. E.g. Beam, Spot, Point.

Spherical Physical Light

A Physical Spherical light uses the definition of light sources from physical parameters issued from lamp manufacturers. A Physical Spherical light uses, as the name suggests, a circle or sphere that defines the light surface that will emit light in the scene.

See Physical light for further notes on how physical lights are rendered.

PARAMETERS. Note. Many parameters are the same as for Physical, so only have a brief description. And a few images. See physical for image representations.

Power – Sets the power of the light in Watts.

Fig 342

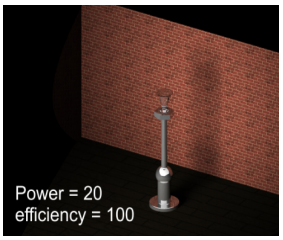
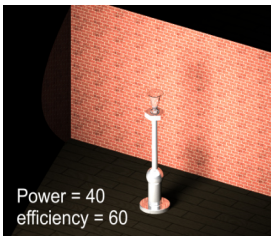


Fig 343



Increases or decreases the power of the luminance. Although the light is circular cone light. Unlike spot. It is a single light, not inner and outer cones. Also a small increase in power can easily produce too much light.

Luminous Efficiency – Sets the luminous efficiency of the light in percentage (%).

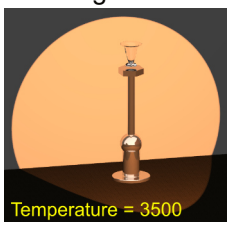
Like a diffuse setting. Used to tone down the light. Reducing the efficiency makes the light darker.

Intensity decay – Set the rate at which the lights power diminishes. Intensity decay is preset at Natural Quadratic.

Color – Sets the colour of the light.

The alteration can either be subtle or dramatic, depending on the chosen light colour.

Fig 344



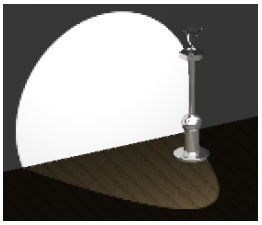
Color Temperature – Alternate color value which supports definition of the color by temperature in Kelvin.

Colour temperature can be used instead of Color. It can change the colour from warm to cool. Setting the temperature to 0, switches to using the colour setting.

Color Affect – Sets the relative intensity (power) of the color light.

Colour affect can be used to diffuse the chosen colour, it does not affect colour temperature. Moving the slider to the left. Darkens the colour used.

Fig 345

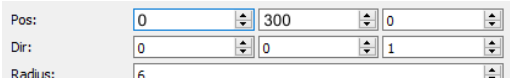


Samples Count – Sets the maximum number of light samples.

Pos – Sets the X,Y,Z position of the light, relative to the object to which it is attached.

This uses the objects coordinate system. Altering the position, allows for the light to be altered, even if the source object cannot be moved. (Example. If attached to a fixed object like a cupboard).

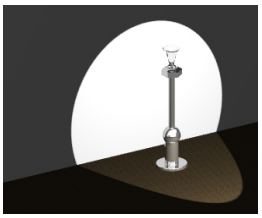
Fig 346



In the example right, the source objects y axis is horizontal. So to move it right, we need to add the distance to the y axis box.

Dir – Sets the X,Y,Z direction of the light relative to the position.

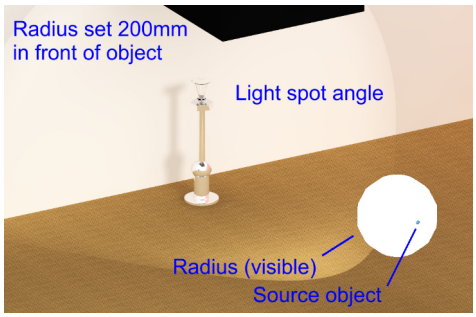
Fig 347



Dir sets the axis of the source object. From which the luminance will emerge. Normally one would just put a 1 in the relevant axis,. For a 45 degree light, one could put a 1 in two axis boxes. Other values can also be used.

Radius – Defines the radius of the light at the source object.

Fig 348

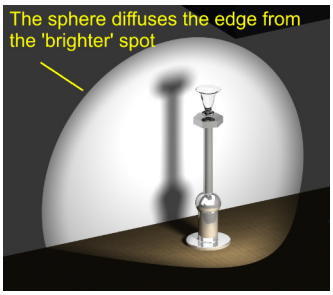


The radius is the geometry of the visible light sphere. It is not used for calculation of the light spot angle. It can obscure or blend the 'bright' spot in the light with a large radius. It can be used, for example. For displaying a turned on light bulb.

It requires visible geometry turned on, to display the sphere. Note. This light geometry radius, has nothing to do with the size of the source object, (for example a small sphere), as shown in the image right.
It is also used when a texture file is used, see below.

Spot Angle - Sets the angle of the cone used for the spot light.

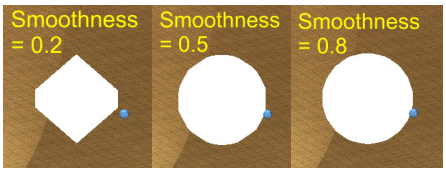
Fig 349



This is the angle of the spot light as it emits from the Pos setting. This may or may not be the source object. Depending on whether one has chosen to use the default 0, 0, 0 in Pos. Or if one has moved the position away from the source, example. Pos = 0, 0, 200.

The Spherical light only has one spot setting, unlike a spot luminance which has inner and outer. Because the light is spherical it will naturally 'diffuse' the edge, and thus does not need a second angle.

Fig 350



Smoothness - Sets the visible geometry

Smoothness affects the shape of the lights visible geometry. It does not affect the shape of the luminance which is always spherical.

Samples Count – Sets the number of samples used to populate the light surface. Must be greater than 0

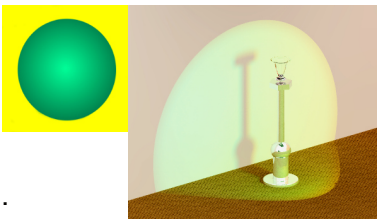
Shadows – Sets whether the light will generate shadows, or not.

Shadow Color – Specifies the color of shadows created by the light.

Texture File Name – Rather than using a single constant color for the whole light, you can setup a texture from which light color will be read. Using a texture, you can simulate projectors or complex light filters.

For a light texture, simply supply the relevant image path and name.

Fig 349 Fig 350



The example in Fig 349. Is the green circle inside yellow square.
Fig 350. shows that a texture can change how the lighting is displayed.

For a texture map (wrapping), it requires visible geometry. With the image wrapped onto the visible sphere. The example right shows the texture wrapped.

NOTE. The image is wrapped onto the light geometry. Not onto any object used, hence why the radius is set higher than one would normally use.

Fig 351

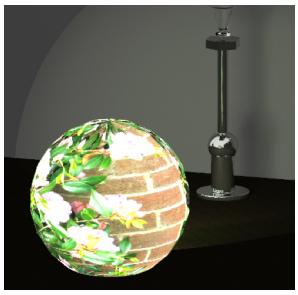


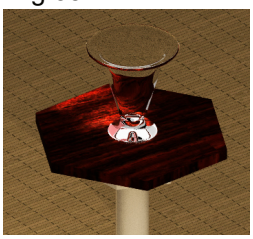
Fig 351 simply shows that is can work. But in honesty you will probably never use projection with a spherical luminance.

One will need to adjust power and efficiency, or colour, to compensate for any bright geometry.

IES File Name – Manufacturers measure the behavior of their bulbs and store their directional contribution into IES files. Those files are available for most of the bulb models directly from the manufacturers' web sites. Even if IES files store information about the shape of the emitter, they are not used here.

Requires experimentation, as IES data can override some of the settings, including direction.

Fig 352



Caustics – Specifies whether the light will generate caustic effects.

Caustics are particles of light scattered by a transparent object. Physical luminance can generate the caustic effect. The effect is controlled by the material settings, like IOR, anisotrophy, etc. Fig 352

Visible Geometry - Specifies whether the physical geometry of the associate 3D objects will be seen.

This is the light geometry, not the object from which the light emanates. To obscure the object one would use object properties, 'load for luminance only'. See Radius and Smoothness above.

GI Baked.

Redway state. *"By baking, we mean pre-calculating a given information, storing it and restituting it, faster, as a later rendering phase".* It is used with global illumination rendering. And may not work correctly if global is set to use the GPU.,

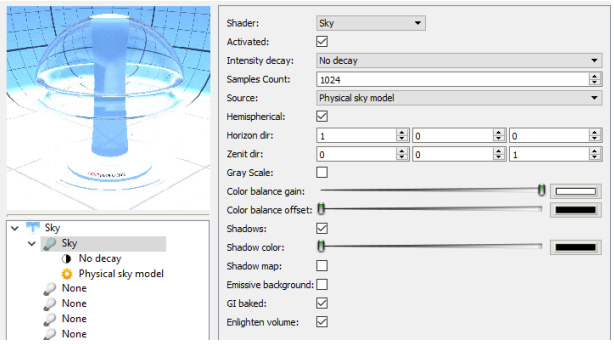
Enlighten volume. Used with volumetric environmental effects (see RedSDK environment).

Volume is a special effect, which can show light rays passing through a space. Only certain lights can create a volume effect. This is covered in detail in the environment section.

This setting determines whether the physical light can affect the volumetric effects produced by other luminance. E.g. Beam, Spot, Point.

Sky Light

Fig 352



NOTE - If you have TC Geo location - Include Sun, set up. And try to set up a manual sun direction in physical sky. This can add extra unwanted shadows, at a different angle than the desired shadows. Basically, this is not a bug, Just that you have set up two sun's.

This physical skylight shape generates a light source that casts light from a dome surrounding the scene. In addition a, skylight shape can create a physical sky texture that can be used in the rendering for direct visibility.

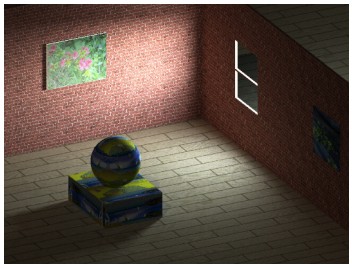
A physical skylight shape is rendered on the CPU and is rendered by a sampled skydome on the GPU. The GPU rendering of the skylight is a good approximation of the real physical skylight rendered on the CPU.

NOTE - Some settings only work properly when combined with other parameters. For example. For the image source in Fig's 354 and 355 below, it is combined with a render environment background, and Emissive background ticked.

PARAMETERS

Intensity decay – Set the rate at which the lights power diminishes. For Sky light. This is preset at No decay.

Fig 353



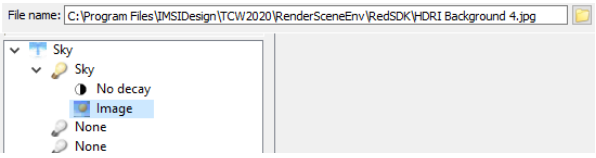
Samples Count – Sets the maximum number of light samples.

Increasign the count can improve the quality of the render. However this is at the cost of time. For example Fig 353 rendered at 16 samples, took 20 seconds. Rendered at 256 samples took 75 minutes.

One has to decide on a balance between quality and rendering time.

Source - Can be set to image, or physical sky model.

Fig 354



Image, NOTE. Sky with an image can be bland. So may need additional parameters or luminance.

When image is selected a new heading appears in the left side of the dialog Fig 354. Clicking this image heading opens up a place to browse for the desired image.

Different images can has a significant effect on the luminance, as can be seen in Fig 355 and 356, The insert in the bottom right shows the image used.

Selecting physical sky model, changes the heading in the left pane of the dialog. Clicking this new heading brings up many extra settings. These are covered later.

Fig 355

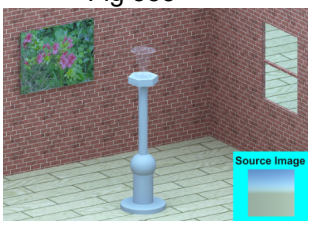


Fig 356

NOTE. The following settings were carried out, with Source set to physical sky model.

Hemispherical – When On the generated map will cover a hemisphere. When Off, the generated map will cover a whole sphere. This can make a difference in the light intensity on some renders.

Fig 357

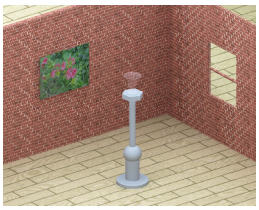


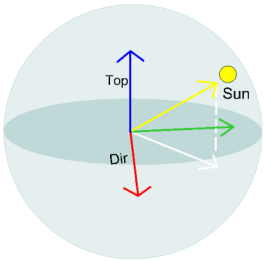
Fig 358

Fig 356 has Hemisphere turned off. Therefore it us a sphere. Fig 357 has Hemisphere turned on. Requiring an increase in intensity.

Horizontal dir - default is set to x axes

Zenit dir - default is set to z axes.

Fig 359



Gray scale - Over-rides the actual colour set in colour balance. And converts the colour to a gray scale equivalent. This still allows for the colour gain and effect, to alter the whole luminance and tone of the render.

Color balance gain -
Color balance offset -

Colour balance gain can be used to alter the overall lighting colour of the render. Colour offset can be used to change the tone of the colour. In most situation the colour would be left at grey scale, and used to alter the tone. However if one required a special lighting condition, altering the colour will do the job.

Fig 360

Fig 361

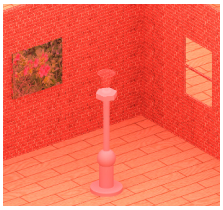
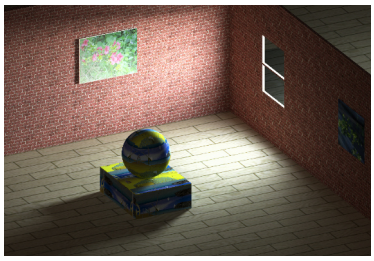


Fig 360 is the default grey scale adjustment. Fig 361 shows a dramatic change when the colour gain is set to Red.

Shadows – Sets whether the light will generate shadows. Fig 362
Shadow Color – Specifies the color of shadows created by the light.

Fig 362

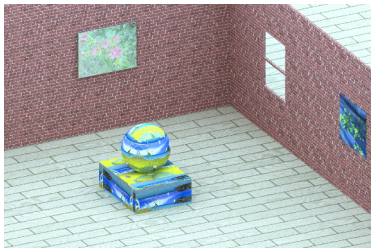


Shadow Map - Overrides shadows. Shadow maps are generated by testing whether or not each specific pixel is visible from the light source. This Can speed up render time, with slight loss in quality.

Shadows may not show correctly when Emissive background is turned on.

Emissive background - allows the background, whether simply a colour, or an image, to have an effect on the render. Fig 363.
Other luminance may need to be added to cast shadows. Or the render can look bland.

Fig 363

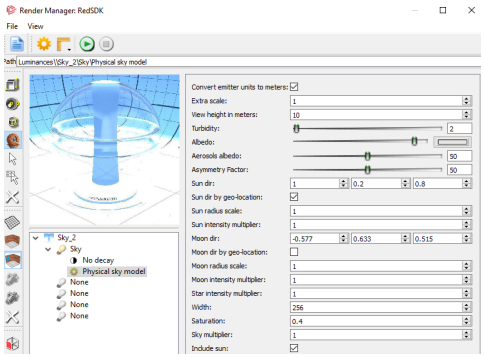


GI baked - Redway state. *"By baking, we mean pre-calculating a given information, storing it and restituting it, faster, as a later rendering phase".* It is used with global illumination rendering. And may not work correctly if global is set to use the GPU.,

Enlighten volume. Used with volumetric environmental effects (see RedSDK environment).

Volume is a special effect, which can show light rays passing through a space. Only certain lights can create a volume effect. This is covered in detail in the environment section.

Fig 364



Physical Sky Model

The physical sky model add settings to control sun, moon and stars. These are not an image, nor a physical object, but are used by RedSDK when calculating lighting.

Convert Emitter units to metres. -

Extra Scale -

View height in metres -

Turbidity – Sets the amount of particles present in the atmosphere. A turbidity of 1 means a perfect clear sky, while a turbidity of 5 means an average haziness sky. The valid range of turbidity values is [Integer: 0, 100].

Fig 365

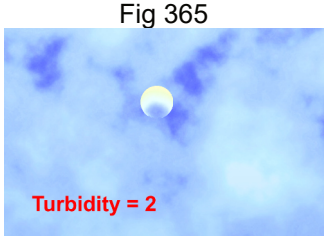
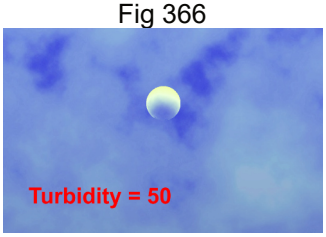


Fig 366

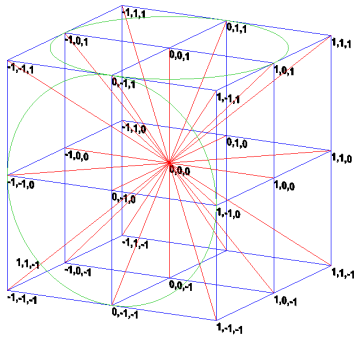


Turbidity effectively darkens the scene. In a similar way that a cloudy sky means it is darker than a clear sky.

Albedo – Sets the ground albedo defining how much light is reflected back to the sky by the ground. When grey scale is used (black to white), it gives an effective value of 1 to 100. One can also alter the

AlbedoColor. - Unknown.
Aerosols Albedo - Unknown.
Asymmetry Factor - Unknown.

Fig 367



Sun settings – sets direction to the sun relative to the scene or an associated 3D object. I suspect these are matrix transformation settings, bu the actual calculation is unknown to me. Below is surmised through experimentation.

If we looks at the standard matrix. Fig 367. We can visualise where we need to use either a plus figure or a negative figure.

But that in itself is not enough. We need to break down the matrix even further.

Fig 368

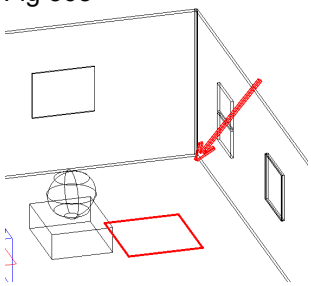
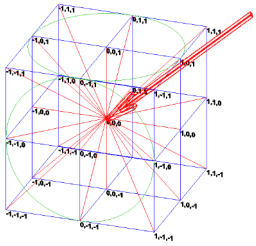


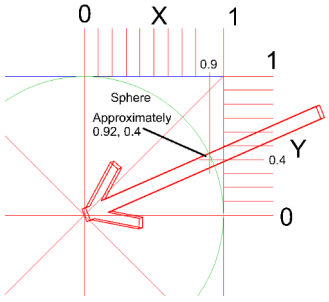
Fig 369



If we look at the image in Fig 368. We wish the light to go through the windows and land roughly in the red rectangle.

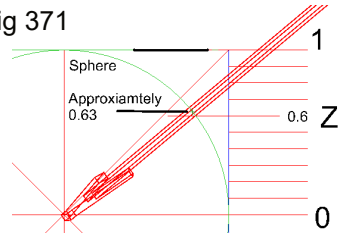
For this experiment we will use the red arrow as a guide. And move it to our matrix cube. Fig 369

Fig 370



First we will get the x and y values. See Fig 370. We want the pint where the red arrow crosses the green sphere. And as can be seen, this works out at about x = 0.92, Y = 0.4.

Fig 371



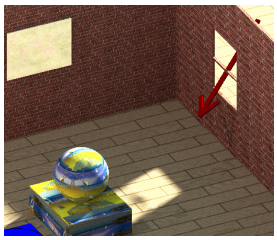
For the last figure, we look at Fig 371. (Front view, so we get the axes. Again where the arrow crosses the sphere. Which is about 0.63.

Using these figures we type into the sun direction boxes Fig 372.

Fig 372

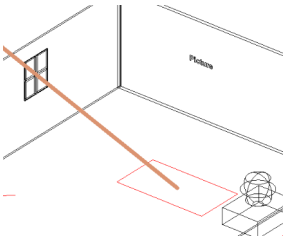
As can be seen in Fig 373. We were pretty close. Obviously because the light has to enter thought the window at an angle, to get to the previous red rectangle.
The light is skewed, which is to be expected

Fig 373



In this final example, we will use the opposite window. And use both minus and plus values. As can be seen in Fig 374 the light should come through the window and fall, roughly in the red rectangle, following the tan 3d line path.

Fig 374



Looking at fig's 375 and 376. The tan 3d line has been put into our matrix box. From this we can get the X, Y and Z values.

These equate to X = -0.98, Y = -0.06 and
Z = 0.39. Which we can then type into the relevant Sun direction boxes.

Fig 375

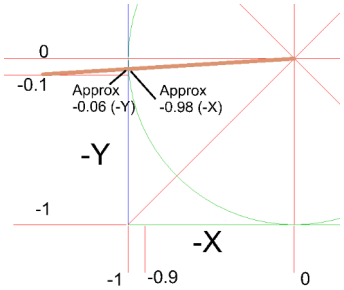


Fig 376

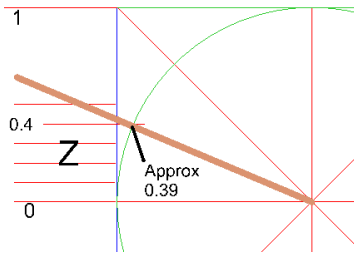
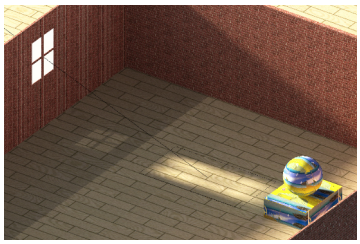


Fig 377

Looking at the image in Fig 378. We can see the light is hitting the floor, pretty much as required.

Fig 378



Final Note. As previously stated, this is an assumption. I am not a programmer. The sun direction settings are obtained through experimentation. I cannot guarantee that your settings will correspond then same.

If, for example. You chose to set the sky horizontal setting to correspond with the Y axes instead of the default X axes. Then the sun position settings will be incorrect. You will need to compensate for any other changes made within other Sky luminance settings.

Therefore the above sun direction settings are a guide only.

Sun by geo location - This can eliminate having to set the sun direction by ticking this box. This will use the settings one had previously set up in TC's geo location dialog. This overrides the figures in the sun direction boxes.

Sun radius scale- No idea really. From help. *Sets the scale factor to be applied to each pixel in the texture. Set it to 1.0 to create a physically accurate texture of the sky. Lower values can be used to simulate the effect of clouds or environment on the sky intensity.*

However it also appears to work the opposite way, like an adjustable torch head. Set to 1, the full sun brightness is directed at the drawing. Making the scale larger (moving the torch head adjustment to a larger bean), averages out the brightness over the larger scale. Thus the drawing gets darker.

Sun intensity multiplier - Increases the brightness (intensity) of the sun.

Moon settings - These mimic the sun settings, except the apply to a virtual moon.

Star Intensity multiplier -

Width – Sets the width of the defining rectangle. Requires experimenting.

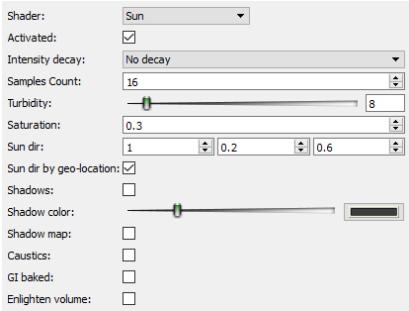
Saturation – From help file - *Sets the saturation factor applied to each pixel in the texture. Set it to 1.0 to create a texture with vivid colors. A lower value will make the sky look more gray and less colorful. The valid range of values is [0, 1]. This can be used to simulate the effect of clouds or environment on the skys intensity.*

Sky multiplier - increases the sky effect. But it can wash out the scene if a high value is used. For example, a value if just 8 can white-out the background. This obviously depends on all the other settings contained within the sky luminance dialog.

IncludeSun – When On, the direct sun contribution is added to the texture. Otherwise no sun is included (E.g in the case of a very cloudy sky).

Sun Light.

Fig 379



A sun light would normally be used alongside a skylight. It can also be used with TC's geo-location settings.

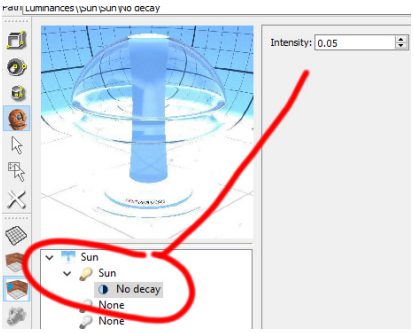
PARAMETERS

Activated - switches the light on or off.

Intensity decay – Set the rate at which the lights power diminishes. For Sun light. This is preset at No decay.

The intensity is set by clicking No decay in the left panel of the dialog box. This changes the right hand panel to accept the intensity.

Fig 380



Samples Count – Sets the maximum number of light samples.

Turbidity – Sets the amount of particles present in the atmosphere. A turbidity of 1 means a perfect clear sky, while a turbidity of 5 means an average haziness sky. A value of 50 could be very hazy. Fig's 381, 382

The valid range of turbidity values is [Integer: 0, 100]

Fig 381

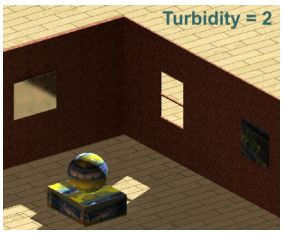
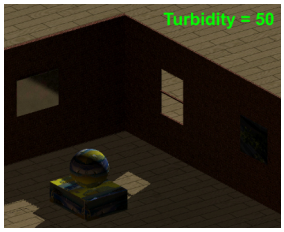
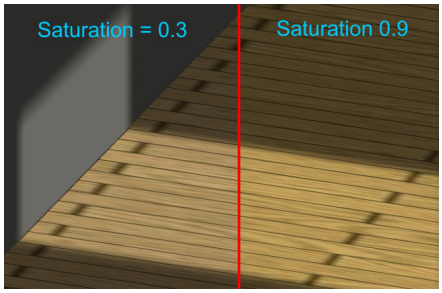


Fig 382



Saturation – Sets the saturation factor applied to each pixel in the texture. Set it to 1.0 to create a texture with vivid colors. A lower value will make the sky look more gray and less colorful. The valid range of values is [0, 1]. This can be used to simulate the effect of clouds or environment on the skys intensity.

Fig 383



SunDir – sets direction to the sun relative to the associated 3D object. See Physical sky model.

Sun Direction by Geo Location - allows the sun direction to access the geo-location, one it has been set up within TC.

Shadows – Sets whether the light will generate shadows, or not. Fig's 354 and 355.

Fig 384

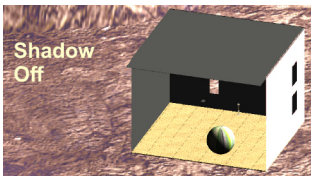
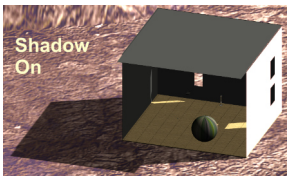


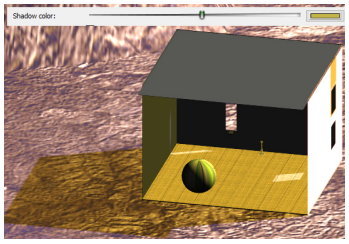
Fig 385



Shadow Color – Specifies the colour of shadows created by the light. The colour can either be left at black to white, which will act to difuse the shadow.

Or it can be set to a specific colour. For example, one could use yellow as in Fig 386.

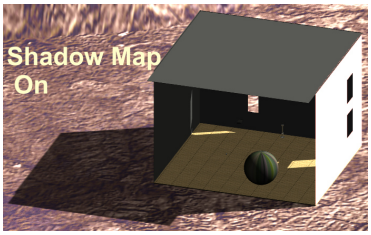
Fig 386



Shadow Map - Shadow maps are generated by testing whether or not each specific pixel is visible from the light source, Fig 387.

One difference from standard shadows. Is when an object has transparency. The shadow map may ignore the transparency and treats the object as a solid colour.

Fig 387

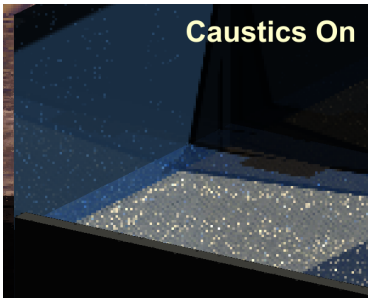


Caustics - Specifies whether the light will generate caustic effects.

Caustics are particles of light scattered by a transparent object. Fig 338. Physical luminance can generate the caustic effect. The effect is controlled by the material settings, like IOR, anisotrophy, etc.

With standard physical luminance, one can simply alter the luminance direction to change the effect, With sun luminance, altering for a specific effect will also alter the whole scene.

Fig 388



GI Baked.

Redway state. *"By baking, we mean pre-calculating a given information, storing it and restituting it, faster, as a later rendering phase".* It is used with global illumination rendering. And may not work correctly if global is set to use the GPU.,

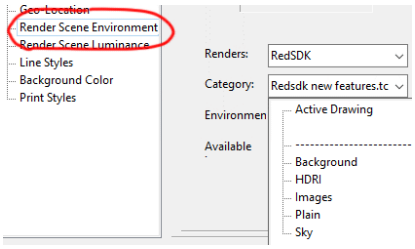
Enlighten volume. Used with volumetric environmental effects (see RedSDK environment).

Volume is a special effect, which can show light rays passing through a space. Only certain lights can create a volume effect. This is covered in detail in the environment section.

This setting determines whether the physical light can affect the volumetric effects produced by other luminance. E.g. Beam, Spot, Point.

Environments.

Fig 389



When selecting Render scene environment, and clicking the environment drop down box. One is greeted with Background, HDRI, Images, Plain and Sky. It may seem like plenty of choice. However apart from 'Plain', they are all just themes on the same environment map dialog box.

They simply differ on which fields are automatically filled in. The real difference comes when one changes the shader.

Shader

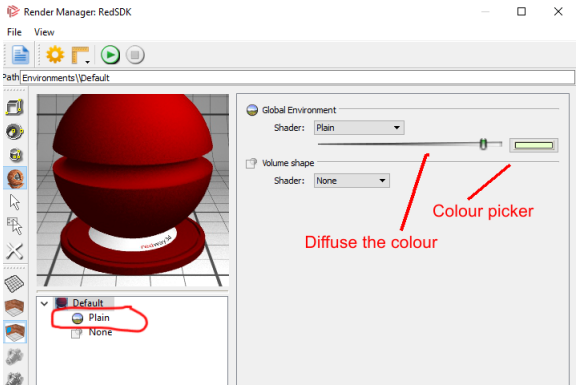
The number of options may differ between TC versions. But they all fall into 3 main categories. plus a sub category.

Plain. - Sets a plain colour as the background
Environment map. - Sets a background. Plus a cube map if required
Sky. - Sets sky and light source if required.

Volume shape. - A special volume shape to absorb light (light rays).
A volume shape is an effect that can show light rays, by reflecting the light off particles in the virtual air in a render. See seperate page for Volumetric effect.

The shader can be changed to whatever is required irrespective of which environment is originally chosen.

Fig 390



Plain.

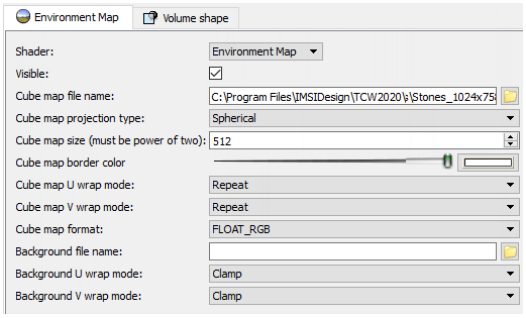
Selecting plain, allows for a single background colour to be assigned.

A colour picker can be used to select the desired colour.
And the slider used to lighten or darken the selected colour.

Volume shape is discussed separately.

Setting a plain colour, avoids having to alter the drawing background colour, each time a render is carried out.

Fig 391



Environment Map

The Environment map use two components, both optional.

Cube Map - This generates a 3 dimensional image projected surround the centre of the drawing at an infinite distance.
Background - This generates a virtual image which is used to generate background imagery in reflective materials.

PARAMETERS

Cube Map File Name – Name and Location of the Cube image.
If you choose one of TC built in environments like HDR. TC will automatically fill in the path and name.
If using ones own image. Enter the full path and name for the file.

Whilst almost any image can be used. The use of a proper environment map can give better results.
These are generally known as HDRI maps,

Below are two images wrapped onto a sphere. Fig 392 is a proper environmental map. Fig 393 is a simple image from TC environment images. As can be seen. The environment map is a rectangular image, designed to be wrapped onto a spherical environment dome. A normal TC environment image, whilst it will wrap. Is not a a true wrap, and some images will show a 'seam'.

Fig 392

Fig 393

Fig 394

Cube Map Projection Type – Type of the environment to project.
SPHERICAL – Environment mapping type. The source image must be a latitude / longitude spherical map. The (u,v) = (0,0.5) texture map point intersects the X axis of the cube map; the (u,v) = (0,1) texture map point intersects the Z axis of the cube map.

Fig 395

CYLINDRICAL – Environment mapping type. The source image is wrapped as a cylinder to define the cube map. The cylinder axis is aligned with the Z axis of the cube map. The (u,v) = (0,0.5) point is mapped along the X axis of the cube map.

Fig 396

HEMISPHERICAL – Environment mapping type. The source image must be a latitude / longitude hemispherical map. This is similar to the ENV_ SPHERICAL mapping considering only the part of the map above the line of horizon. The (u,v) = (0,0) texture map point intersects the X axis of the cube map; the (u,v) = (0,1) texture map point intersects the Z axis of the cube map.

Cube Map Size – Size in pixels of the cube map faces (must be power of two). Default is 512.

Note. Using a small image can have a detrimental affect on the how good the reflections look in a finished render.

Fig 397

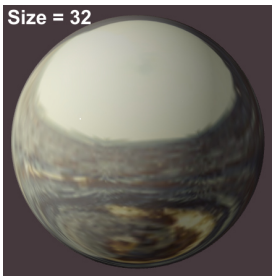
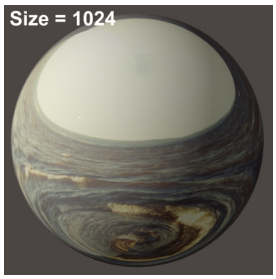


Fig 398



Cube Map Border Color – Border color of the source image. That was from the help files, To me, this sound different to what is actually achieved.

In Hemispherical and Clamp to border modes. It uses this colour to add a background colour outside the range of the wrapping. In Spherical and Cylinder modes, the background colour is taken from the actual image. (Assuming that one has not set a background map in this dialog box).

Note - because Cube map wrap modes do not show, except as reflective surfaces. See Background wrap mode, on next page for example images.

FROM THE HELP FILES.

Cube Map U Wrap Mode –U wrap mode of the source image.
REPEAT – Repeat the texture pattern out of the texture's limits.
MIRRORED REPEAT – Repeat the texture pattern with an inversion of the texture out of the texture's limits.
CLAMP – The texture sampling result is zero outside of the texture's limits.
CLAMP TO_EDGE – The texture sampling results remain slightly inside the texture when it's out of the texture's limits.
CLAMP TO_BORDER – The texture sampling results are set to the texture's border color when it's out of the texture's limits.

Cube Map V Wrap Mode –V wrap mode of the source image.
REPEAT – Repeat the texture pattern out of the texture's limits.
MIRRORED REPEAT – Repeat the texture pattern with an inversion of the texture out of the texture's limits.
CLAMP – The texture sampling result is zero outside of the texture's limits.
CLAMP TO EDGE – The texture sampling results remain slightly inside the texture when it's out of the texture's limits.
CLAMP TO BORDER – The texture sampling results are set to the texture's border color when it's out of the texture's limits.

Cube Map Format – Pixel format of to be used in calculating the Cube.

RGBA – 8 bits per component, 32 bits per pixel, RGBA in memory.
RGB – 8 bits per component, 24 bits per pixel, RGB in memory.
FLOAT – s1e8m23 IEEE floating point format. 32 bits per pixel.
FLOAT RGBA – 1 float per component, 128 bits per pixel, RGBA in memory.
FLOAT RGB – 1 float per component, 96 bits per pixel, RGB in memory.
L8 – 8 bits per component, 8 bits per pixel, L in memory. Loaded as a (L,L,L,1) texture on the GPU.
A8 – 8 bits per component, 8 bits per pixel, A in memory. Loaded as a (0,0,0,A) texture on the GPU.
RGBA DXT1 – Compressed image format (lossy). Uses one bit for alpha values. It is the largest compressed form along with
RGB DXT1
RGBA DXT3 – Compressed image format (lossy). Uses a better encoding for alpha values. Uses more memory than RGBA
DXT1
RGBA DXT5 – Compressed image format (lossy). Uses the best alpha encoding that is available for these compressed formats.
Uses the same memory amount as the RGBA DXT3
RGB DXT1 – Compressed image format (lossy). It is the largest compressed form along with RGBA DXT1.
HALF FLOAT – Half-float format. 16 bits per pixel.
HALF FLOAT RGBA – Half-float RGBA format. 16 bits per component, 48 bits per pixel.
HALF FLOAT RGB – Half-float RGB format. 16 bits per component, 64 bits per pixel.
FLOAT RGBA AA – Float RGBA format for anti-aliased images. 128 bits per pixel.

Background File Name – Name and Location of the Background image.

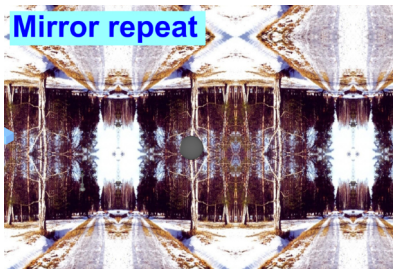
The path and name of an image file. bmp, tif, jpg, png, pcx, hdr tga. The size of the image makes a difference as it how it is displayed.

Background U Wrap Mode – Pixel U wrap mode of the source image.
Background V Wrap Mode – Pixel V wrap mode of the source image.

MIRRORED REPEAT – Repeat the texture pattern with an inversion of the texture out of the texture's limits.

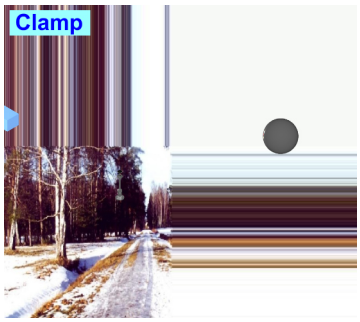
Note. This may throw up an error by TC, I have no idea why. But it still appears to work despite the error.

Fig 399



CLAMP – The texture sampling result is zero outside of the texture's limits.

Fig 400



CLAMP TO EDGE – The texture sampling results remain slightly inside the texture when it's out of the texture's limits.

Clamp and clamp to edge, produce very similar results.

CLAMP TO BORDER – The texture sampling results are set to the texture's border color when it's out of the texture's limits.

Fig 401

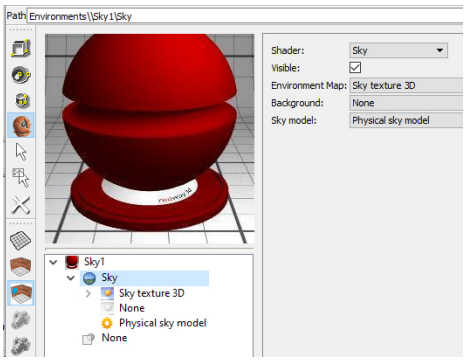
In the images shown, the TC window size was about 1600 pixels wide. The first three images are at a small size of 512 x 512 pixels. This means the texture cannot fit the screen.

The last one in Fig 402, had the image size increases to 1600 x 1400 in a paint program. This allows the background to correctly fill the screen.

Fig 402

Sky Environment.

Fig 403



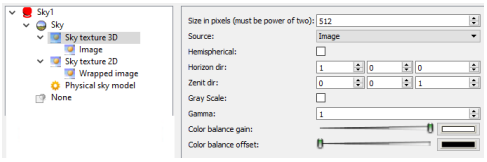
The last environment option is Sky environment. This uses much of the same settings as shown for Sky luminance.

The odd thing is that this is an environment, not a luminance. So these settings don't actually provide luminance over and above what a cube map would supply.

Visible - turns the environment on / off.

Environment map - Two options. None and Sky texture 3d.

Fig 404a

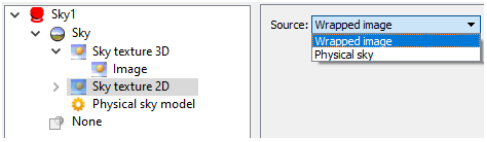


Sky texture 3d, opens up a new option in the left hand pane. Which when clicked reveals further settings in the right hand pane.

Source. Image. Allows on to select an image to use as a cube map.

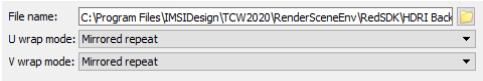
Source. physical sky. It appears (though I cannot be certain), that the 3D plus physical sky model, creates a cube map style environment. Based on the physical components one selects. And as such is probably best used in conjunction with sky and sun luminance.

Fig 404b



Background - Two options, None and Sky texture 2d.

Fig 404c



This creates the background to a scene. Similar to the standard background image environment. When the options is set to Sky texture 2D, it allows one to choose wrapped image or physical sky.

Wrapped image allows one to select any image one requires as a background.

Although the options allows one to select mirror repeat. On my computer it doesn't work. The background gets fixed at clamp. Therefore if one wants the full background, one may need to enlarge the image in a paint program before using as a background.

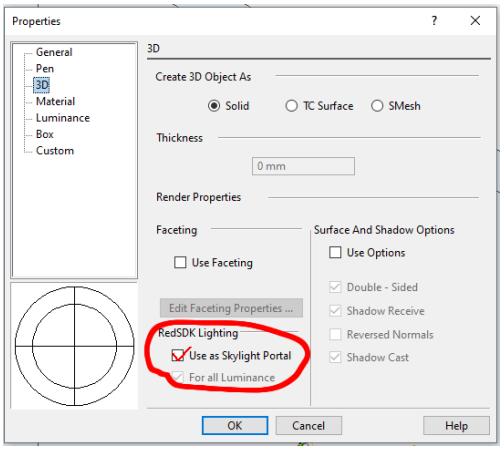
Physical sky model. For me, it is just a black background. I'm guessing it is more than that 'under the hood' . But I'm not going to just guess at what it could be.

Sky model - Preset at Physical sky model. (No option)

In tests I have not found what difference this makes to a render. Therefore all I can suggest is you do your own tests to determine if this will add any advantages to you particular render scene.

Portals (Skylight).

Fig 405



A portal is effectively like a window. Generally used for interior scenes, it tells RedSDK where you want light to enter the room.

Whilst just adding glass to a window frame would work, instead of a portal. In some instances the lighting may be incorrect

First. In order for portals to operate, a 3d object must be created to represent the portal. And in the objects properties, Use Skylight Portal, must be ticked

From the TC help files.

Portals are useful for interior scenes for example, where sky light is entering by small openings which are difficult to sample efficiently. By marking explicitely the openings using portals, the engine can render difficult situations at a much higher quality in less time.

For all Luminance: If On the portal will be used for all luminances with skylight in scene, no matter how this sky light was created (drawing luminance or graphic luminance). If Off the portal will be used for this objects luminance only. The option is enabled if option only if Use as Skylight Portal is On. The option is 'On' by default.

Using portals
Portals are objects (surfaces) that control the flow of photons from the light source. More precisely: the spatial configuration of the photon's flow.

The simplified algorithm is the following: if the photon has not passed through the portals, it is not considered in the calculation of illumination and the CPU time is not spent on it. Instead of it the next photon is launched (generally at random). This continues until the number of photons passing through the portals will not be equal to the target parameters of the light source. In our case it is a sky light (RedSDK supports only sky lights for the portals). In the case of absence of portals all launched photons are involved in the calculation of the illumination.

So, portals allow you to concentrate the quality and the intensity of the light in the right places of the scene. This facilitates the creation of a scene with proper lighting and improves the speed of calculation.

The main purpose of the portals is using them for the openings through which the outer sky light penetrates. Portals need not to be rectangular necessarily. The portal may consist of any set of surfaces. At the same time you should not make the portal very complicated (for example as an ADT window), because of the additional time is required to clarify whether the photon passes through a portal or not. Ideally it should be 1 rectangular facet.

In TurboCAD any 3D primitive can be a portal. This simplifies creation of portals. It should be noted that in this case the geometry of this primitive is not loaded into the scene. In other words an option 'Use as skylight portal' is similar to the option "Load to render luminance only".

Example.

Fig 406

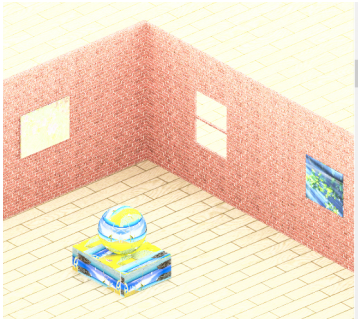


Fig 406 shows a ‘bright’ skyight luminance (without a portal).

Whilst portals would probably be used alongside other luminance, to enhance a scene, and give more realistic lighting coming in from outside.

In this instance, it is just an example of how different a render can look, when a portal is used.

Therefore rather than enhance a scene, we will use a single portal to completely change the scene.

Note, the only thing done was add a portal, and increase the skylight intensity a touch. No other luminance are used, and no other skylight settings were adjusted.

For the portal we need a an object. In Fig 407 we have a rectangular box (Red). This is covering the window to simulate light coming in from outside.

We then select the object, and in object properties tick ‘Use as Skylight Portal’. As shown on the previous page (Fig 405).

Sometimes it is also necessary to increase the skylight intensity. In this instance, in the skylight luminance intensity box, we increased it from 2, to 4.

The scene was then rendered with Global illumination render.

As can be seen from Fig 408. The lighting is nothing like the original render in Fig 406. The skyight is now coming directly from the portal, rather than from a sphere or hemisphere surrounding the scene, as is the case with normal skylight.

As stated earlier. This scene was just an example of the portal effect.

In a proper architectural scene, one would use portals to enhance light entering a room, and create a much brighter and realistic indoor scene.

There is a video on Youtube regarding the use of portals, at.

https://www.youtube.com/watch?v=I05LXRQZNrI

Fig 407

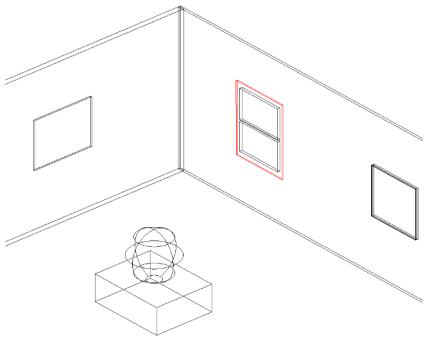
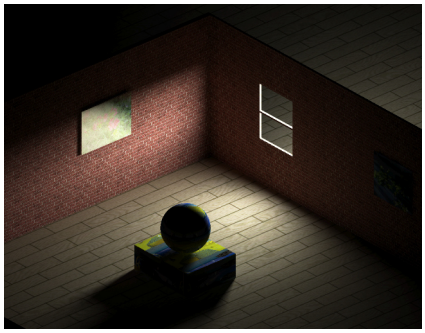
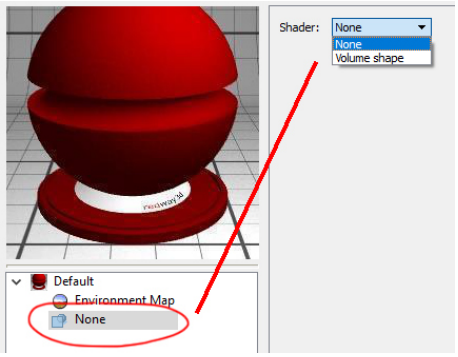


Fig 408



Volumetric effect.

Fig 409

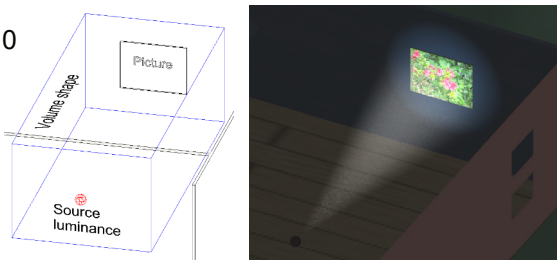


A first important point, Volumetric effect will in many cases severely slow down the rendering process depending on the settings. It is cpu intensive and does not render on graphics cards.

A volumetric effect is light passing through a medium, or particles in the air. Like shining a torch in a dusty cellar.

Volume effects require a luminance capable of producing it, (Beam, Spot, Point). It also requires an environment setting up, which can produce the final effect.

Fig 410



The image right. Simply simulates a torch illuminating a picture, using volumentric effects.

The first section of the volumetric effect.

Fig 411

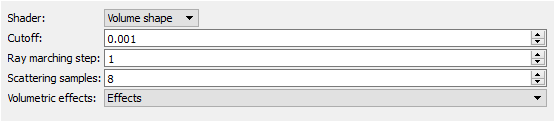


Fig 412

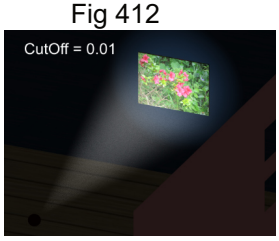


Fig 413

Ray cutoff - is an optional value to stop the ray propagation when the result value is below the threshold.

Ray marching step - is the size of a single volume step along the ray used when sampling the medium. Decreasing this value will increase the rendering quality and time, especially for heterogeneous. media.

This setting can have a significant affect on rendering times. Though the savings depend on the scene, and distance between the source object and the target.

As with difference in time. The difference in quality will vary depending on what is in scene between the source and target.

Fig 414

Fig 415

Scattering samples - defines the number of samples that will be used to sample the volume during its illumination process with surrounding lights. Increasing it increases the volume rendering quality and time. Opposite to Ray marching.

Decreasing the value results in less quality, but quicker rendering. With the two images on the right, the Ray Marching was set to 2.

Fig 416

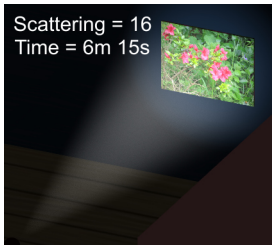
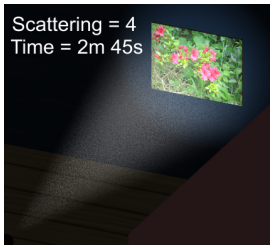
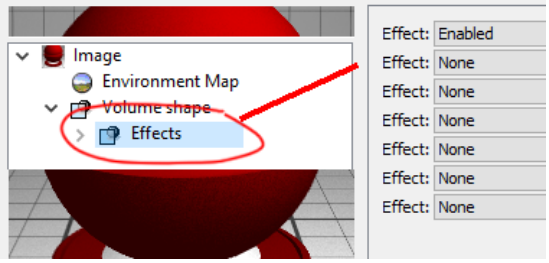


Fig 417



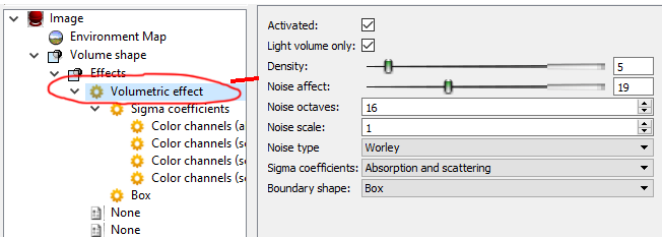
Effects - It is possible to set up multiple effects, however be careful with the settings. Setting multiple effects, using quality settings, can really slow down the rendering process.
Once an effect has been enabled, it allows for extra settings to be accessed.

Fig 418



Volumetric Effect.

Fig 419



Activated - switch on/off the volumetric effect. Turning off the effect, allows one to render the scene without delays caused by the effect. The effect can then be activated for final rendering.

Light volume only - render light volume only.

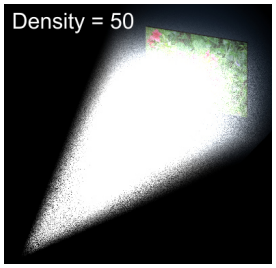
Density - density of the medium (0 – 100%).

The number of particles contained in medium, and thus the amount of light that gets reflected.

Increasing of the density causes more light to be reflected. This can require a reduction in luminance intensity, to prevent the scene from 'whitening out'.

Fig 420

Fig 421



Noise affect - noise factor (0 – 100%)
Noise octaves - Increasing the number of octaves, increases the level of noise details.

Noise scale - noise zoom

Noise type - noise algorithm (Perlin or Worley).

In tests. Perlin appears to be the quicker of the two options.

Fig 422

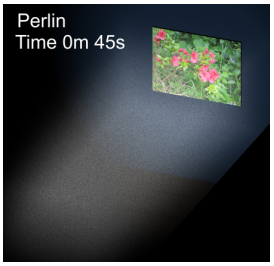
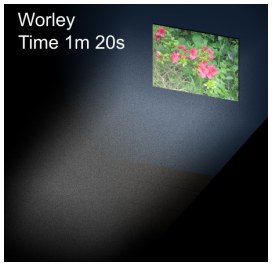


Fig 423



Sigma coefficients - 'Sigma coefficients' shader. The only option is Absorption and scattering.

Boundary shape - boundary shape (None, Box, Sphere).

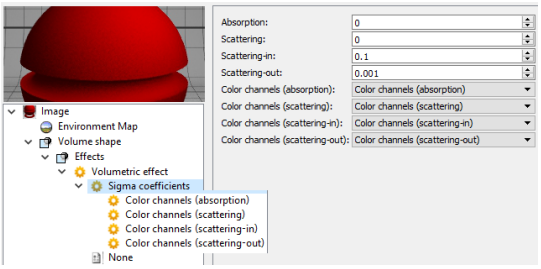
None. This renders the whole view as a volumetric effect. This can seriously slow down a render.

Due to the amount of calculations the cpu has to achieve, to make the effect. However for some scenes it is desirable rather than set up a specific container.

Box. Renders the volumetric effect contained with user defined a box shape.

Sphere. Renders the effect contained in a use defines spherical shape.

Fig 424



Sigma coefficients Shader.
The sigma coefficient tab. Releases further options for setting individual channels.

Absorption - absorption coefficient. How much it will absorb the light entering it. The higher the figure.equals the darker the effect. as more light is absorbed.

Scattering - scattering coefficient. how much it will in-scatter and out-scatter the light entering it.
Even small increments can have a large effect on the volumetric effect.
RedSDK docs state "sigma s allows to enlighten the volume where it receives a lot of light and darkened it elsewhere."

If Scattering coefficient is set to '0', The render takes into account the separate Scatter-in and Scatterout settings. In / out are ignored if scattering is greater than '0'.

Fig 425

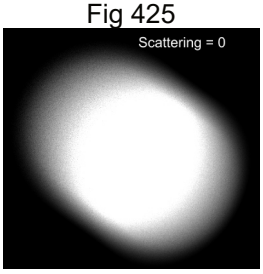


Fig 426

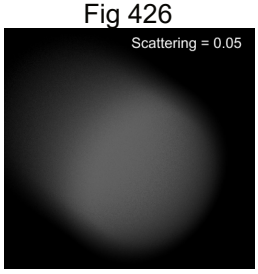
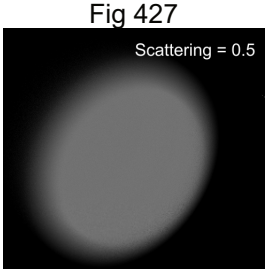


Fig 427



Scattering in and out, can be used independently or together.

Fig 428

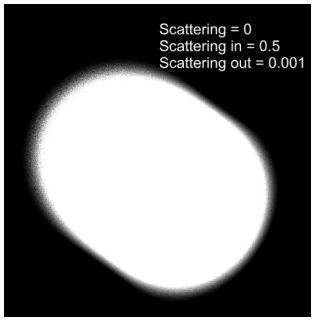
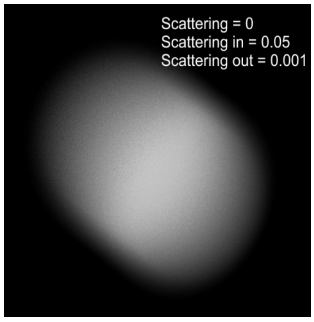


Fig 429



Scattering-in – scattering coefficient, (used only if Scattering = 0)

Increasing just the 'scattering in' setting, enlighten the volume. Use in small increments, or it can 'white-out; the volume effect..

Fig 430

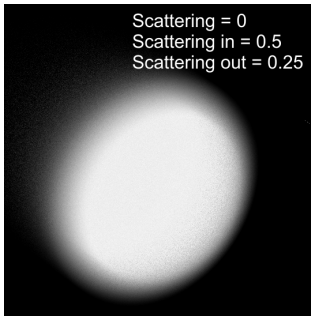
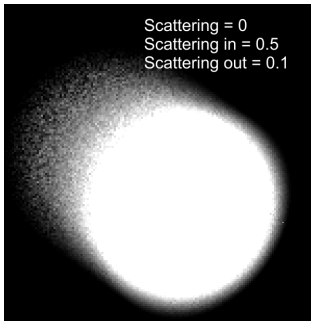


Fig 431



Scattering-out – scattering coefficient, (used only if Scattering = 0)

Increasing just the 'scattering out' setting, darkens the volume.

The four colour channels below the scattering settings, are preset in this section. But they are available under their respective headings, as sub menu's of the sigma coefficient.

Color channels (absorption) – 'color' shader for Absorption (used only if Absorption = 0)

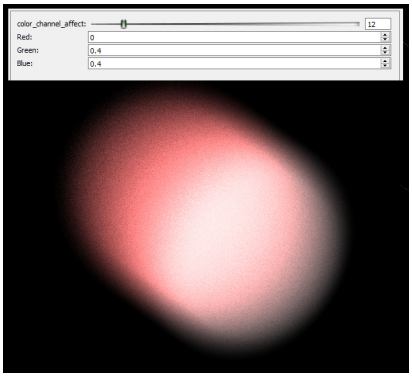
Color channels (scattering) – 'color' shader for Scattering (used only if Scattering = 0)

Color channels (scattering-in) – 'color' shader for Scattering-in (used only if Scattering-in = 0)

Color channels (scattering-out) – 'color' shader for Scattering-out (used only if Scattering-out = 0)

Each of the above headings contain the following settings.
Affect - color affect (0 – 100%)
Red – red channel
Green – green channel
Blue – blue channel

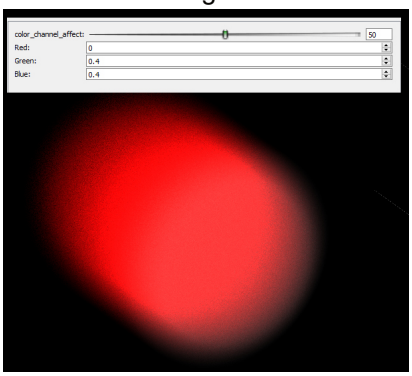
Fig 433



If Red, Blue and Green are all at 0, then the slider (affect) does not alter any settings. If all colours are equal above '0', The effect is grey-scale adjustment, i.e. from light (white) to dark (black), If the colours are set differently, it creates an inverse effect.

For example. In the image right, green and blue are set to 0.4. In a paint program setting both green and blue, but leaving red at zero, creates cyan, The inverse of cyan is Red.

Fig 434



When at least one of the RGB valuse are above zero. The slider alters how much the colour settings, affects the volumetric effect.

Note, Values are not standard RGB.

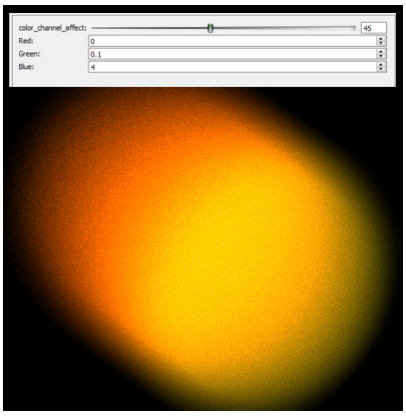
The affect slider determines how much the volume effect is affected by the colour which is set.

In the image right, the colour is the same as above, but affect slider is moved to a value of 50.

As can be seen this saturates the volume with the red colour.

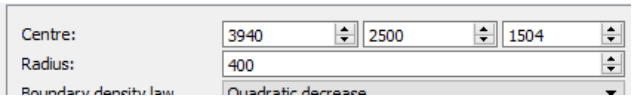
Various effects can be produced by changing the values

Fig 435



Boundary shape

Fig 436



Sphere specific parameters

Centre - This is the Centre point, x, y, z coordinates of the volume sphere, in world units.

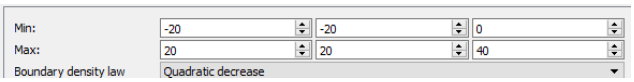
A simple method of finding the correct values. Is to, draw a 3d sphere of the required size. Adjust its position and read the world coordinates from the inspection bar.

A Sphere can be used as a simple container for the light beam to create a volumetric effect. Or it could be used to create a a glow effect around a 3d spherical object. For example the glow around a sun, or a planet.

Radius - The sphere radius in drawing units.

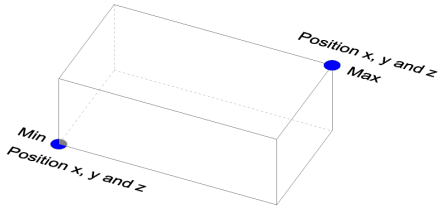
Box specific parameters.

Fig 437



Min: - This would normally be the front, bottom left hand corner of the box shape, in world units., One could draw a 3d box, position it where required. Move the ref point to the bottom corner and read the world values from the inspection bar.

Fig 484



The three entry boxes are X, Y and Z positions.

Max: - Normally the back, top right hand corner of the box shape.

General settings.

Boundary density law – the law of the density changing near the surface of the boundary shape.

Laws include. Linear decrease. Quadratic decrease. Cubic decrease. Linear increase. Quadratic increase. and Cubic increase.

Fig 439

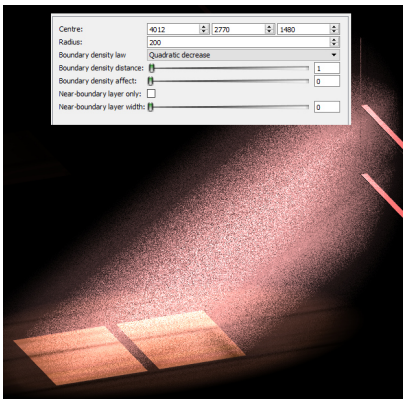


Fig 440

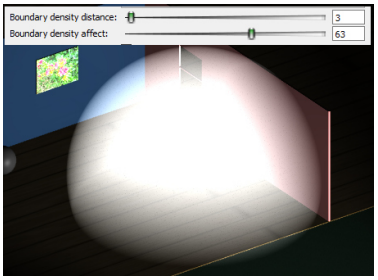
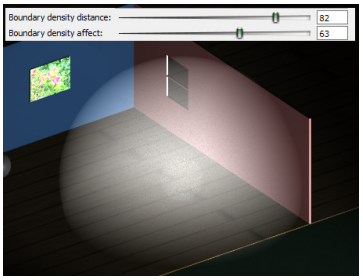


Fig 441



Boundary density distance – distance to the surface of the boundary shape (0 – 100%).

This requires Boundary density affect to be set above ‘0’

Fig 442

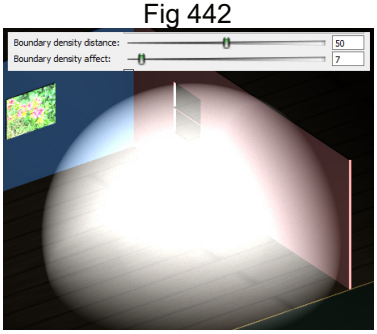
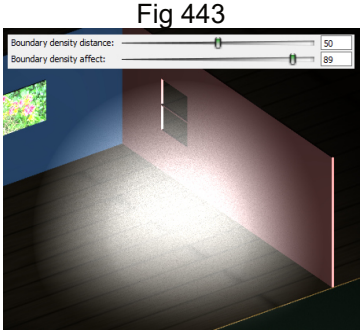


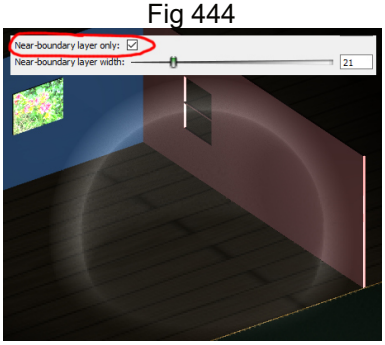
Fig 443



Boundary density affect - affect of density’s change (0 – 100%)

Used alongside density distance.

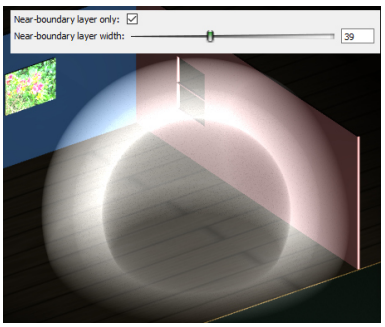
Fig 444



Near boundary layer only - switch on/off taking into account only the layer near the surface of the boundary shape. Used in conjunction with near boundary layer width.

This can be used to create a halo effect.

Fig 445



Near - boundary layer width – width of the layer near the surface of the boundary shape (0 – 100%)

Used with Near boundary layer only switch. This alters the width of the halo effect. Higher numbers increase the width.

Global Illumination.

Global illumination adds a lot of user control. And extra illumination algorithm to enhance photo realistic rendering.

The first image (right), is standard ray tracing. The second image is standard global ray trace illumination.

Fig 446

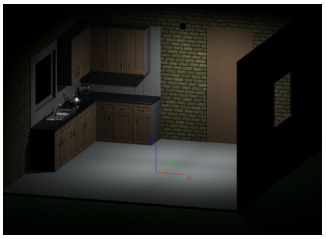
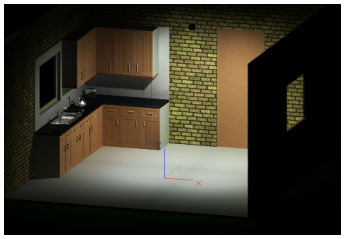


Fig 447

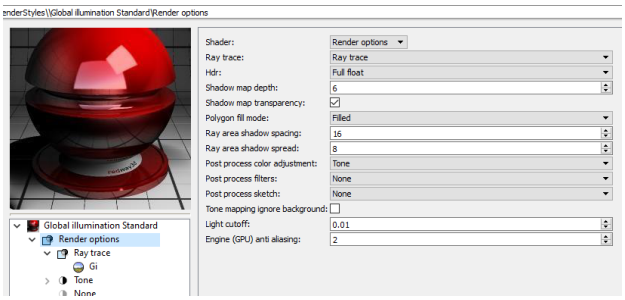


Global illumination, is one of the most complicated things of quality rendering. Unfortunately there is no 'one size fits all' setting. The Defaults mentioned below are taken from the Redway RedSDK help documents. As a good starting point, for probably most renders.

Some things will need manual alteration. For example. If one is using shadow maps, and wish to show behind glass, you may need to turn shadow map depth up to 2.

Also some settings can be turned off, Like Volume, Caustics, and Displacement, if not of these are used in the drawing.

Fig 448



Render Options

Shader. - Render Options is the only selection in this box.

Ray Trace. - Only two settings, None and RayTrace.

None. None, simply turns of the ray trace options in global illumination. A drawback is that in some cases turning off ray trace can actually increase rendering times.

It still allows tone changes in the rendered image.

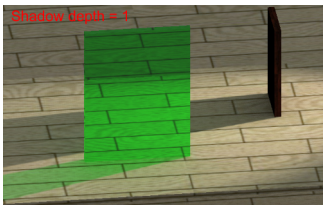
Ray Trace. Turns on a lot of additional settings for complete control of render.

Hdr.
OPTION: Disabled, Half Float, Full Float
Default: [0]
Enable or disable the HDR rendering pipeline for the specified window.
HDR is directly performed using the hardware capabilities. if the hardware has the appropriate blending functions available. Half float HDR consumes less memory than full float HDR (50% less), but has a lower accuracy resulting of 16 bits half floating point values being used to store color components.

Shadow map depth.

OPTION: [integer]: [0,inf[.
[default]: 1 (shadow map applied on first skin).
Defines the rendering depth for which shadow maps are applied on scene elements. A depth of 2 enables shadow maps to generate shadows in a reflection image for example.

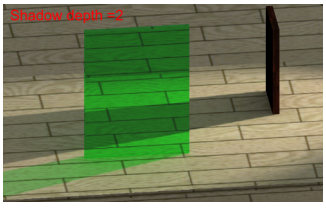
Fig 449



Enable or disable shadow map casting, Shadpow maps pre-test each pixel visibility from the light source, And can reduce render times. However there can also be some loss of shadow quality.

The shadow map depth, is the amount of rays used to calculate shadows. Under normal circumstances (solid objects) only one ray is needed. However looking through transparent objects may give incorrect results.

Fig 450



In Fig 449 we are looking through a glass object, and as can be clearly seen there is no shadow from the solid object showing behind the glass.

In Fig 450 We increase the depth to two, causing a second ray to be fired. And displays the correct image.

Shadow map transparency.

[boolean]: [false, true].
[default]: false (transparent objects cast shadows).

Defines the behavior of shadow mapping versus transparent objects. Because shadow mapping is a shadowing method based on the distance of an occluder from the light source, transparent objects cast black shadows and the amount of transparency can't be considered in the shadowing. This option is used to change this behavior. Whenever set to true, shadow maps will ignore all transparent objects that won't cast shadows anymore.

Fig 451

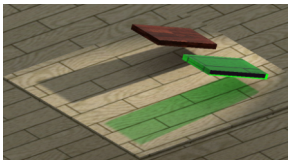


Fig 452

Shadow maps are not normally transparent. However RedSDK has been programmed to allow shadow map transparencies in global illumination.

This simple example, is just two boxes, one solid colour and one with (glass) transparency. Illuminated with spot and sky. Fig 451 shows the render with Map transparency not ticked. Fig 452 shows no shadow for the glass box, when map transparency is ticked.

The example on the Redway website, shows an object on a base, which when rendered normally, results in a black shadow and no reflectance off the base. Turning on transparency, allows a green reflection to show up.

Polygon filled mode.

The fill mode is an odd one. Normally when rendering one wants solid objects, These modes provide also for lines, and points.
Solid. The standard rendering mode for solid objects.
Lines. Unfortunately is not hidden line, but form building edges. These show triangular sections, which may or may not look good.
Points. Show the vertex points of an object, which may look fine on round objects, but only shows 8 points to depict a cube, and are generally not noticeable, on computer monitor screens with a high resolution.

The images below, are Solid (left), Lines (centre), points (right).

Fig 453

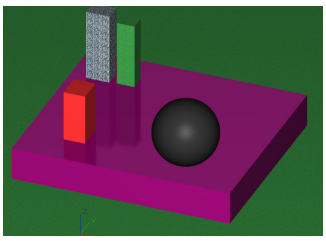


Fig 454

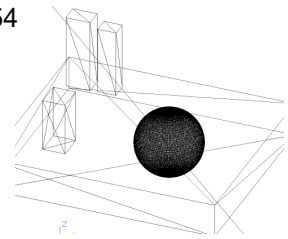
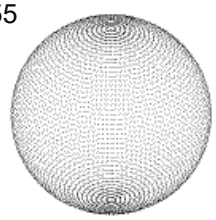


Fig 455



Ray area shadow spacing. (GPU)

OPTION: [integer: 1 to Infinite].
Default: [16]
This is the average screen size distance in pixels between two samples used to sample area shadows for the GPU ray-tracer.

Ray area shadow spread. (GPU)

OPTION: [integer: 1 to Infinite].
Default: [8]
This is the coverage factor of GPU area shadow samples on screen. A larger spread value generates more blurry shadows while a smaller value tends to generate sharper shadows.

Post process colour (Tone) adjustment.

OPTION: Tone, None
Specifies Whether tone mapping will be used post render. For example. Brightness, Contrast, Saturation etc.

Post process filter.

Opens the filter selection part of the dialog. For example. Sharpen, Blur, Glow etc.

Post process sketch.
Opens the sketch mode part of the dialog.

Tone mapping ignore background.
OPTION: [Boolean: false, true].
Default: [false (disabled)]
This option lets the user control whether the contribution of the background images or clear color is considered during the tone mapping calculations or not. If the option is enabled, all pixels of the image that are viewing the background are discarded during tone mapping calculations: they don't have any impact on the image luminance and their colors are not modified. If the option is disabled, the VRL's clear color or background images will participate to the tone mapping process, both in luminance calculations and tone mapping.

Note: *that this option is not supported if the tone mapping is performed by the GPU and if the engine anti-aliasing is activated at the same time. In this specific configuration, an aliasing border may be visible in the resulting image.*

Light cutoff.
[double 0.0 to Infinite].
Default: [0.003]
This value defines the minimal lighting intensity a light must emit to be taken into consideration. This parameter helps improve rendering performances in environments where many lights are set. In such environments, it's not rare to see that many lights have a negligible influence over the shading at a given point. This cutoff value helps the engine discard negligible lights, whose color contribution to the shaded point is below the cutoff value.

Please note that this value is empirical and arbitrary: it's an absolute intensity cutoff threshold applied to the contribution of a single light, regardless of its environment. In physical HDR environments, the real contribution of a physical light goes infinitely far away. Therefore, if the cutoff is set too high in dark scenes, the application of tone mapping can reveal the light cutoff as a smooth frontier between lit and non-lit areas.

The cutoff works for both physical and non-physical lights. Very small cutoff values can be ignored for non-physical lights rendered on the GPU to preserve some numerical accuracy in the decay calculations performed by rendering shaders.

The three images below show the effect of light cut off using Natural Quadratic light decay.

Fig 456 shows the cut off at 0.001. The ight is virtually undetectable about the red line mark.
Fig 457 shows the cut off at 0.1. As can be seen there is almost no difference, but the render does not have to calculate anything lower than 0.1. In certain situations with lots of lights, it could make a difference in rendering speed.

Fig 458 shows the cut off set to 1. As can be seen the lights are cut off closer to the source. Therefore setting the cut off too high can have a detrimental effect.

Fig 456

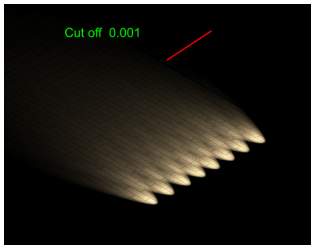


Fig 457

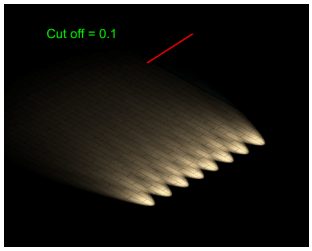
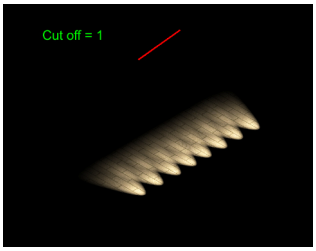


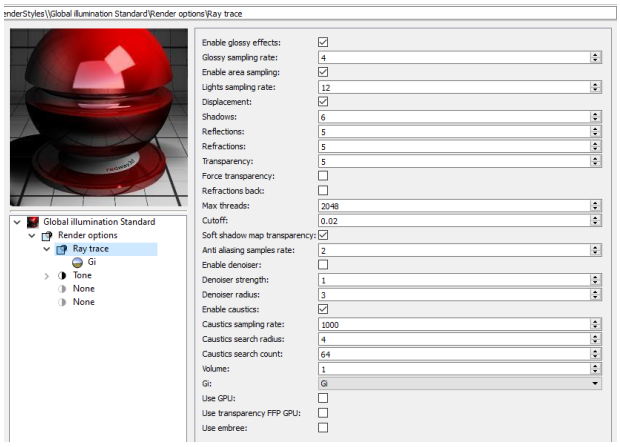
Fig 458



Engine (GPU) anti aliasing.
Range – [1-5]
By default – 1
Use if Ray trace\Use GPU is On.

Ray trace Options

Fig 459



Enable glossy effects.

OPTION: [Boolean: false, true].
Default: [True]
The value of this option lets the user to enable or to disable the calculation of glossy ray-traced effects.

When off, whatever may be specified by ray-tracer shaders, only one reflection ray and only one refraction ray is fired to render reflection and refraction effects.

Glossy sampling rate.

OPTION: [integer: 0 to Infinite].
Default: [8]
Set the number of samples to be taken while sampling glossiness (reflection or refraction). A value of 0 means that the number of samples set to each material is taken. Any other positive values overwrite the materials settings.
The real number of samples taken is the square of the option value. For example, if the option is set to 4, 16 samples will be taken.

Material glossiness, is set in the Anisotropy settings of BRDF reflection (in later RedSDK versions). Which creates noise. This noise shows in the glossy sample rate when glossy effects is ticked.

The examples below show three different glossy settings. Object 'A' has both U and V (in material properties) set to 0.001. Object 'B' has U and V set to 0.02. Object 'C' has U = 0.02 and V 0.05

Fig 460 is the render with Glossy effects turned off.
Fig 461 Glossy effects is on, and rate set to 2. This clearly shows the noise created by the effect.
Fig 462 is Glossy turned on, and rate set to 8. This extra rate (8 x 8 = 64 sample). Smooths out some of the noise.

Fig 460

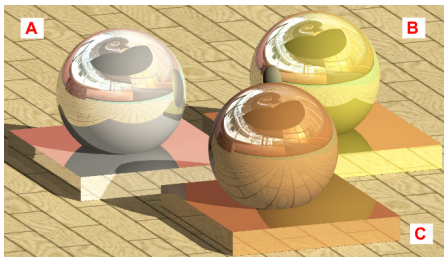


Fig 461

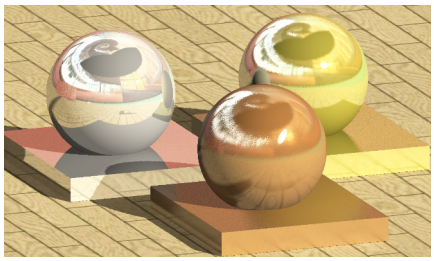


Fig 462

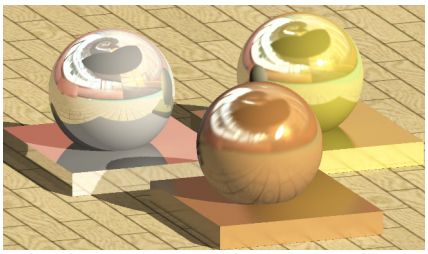
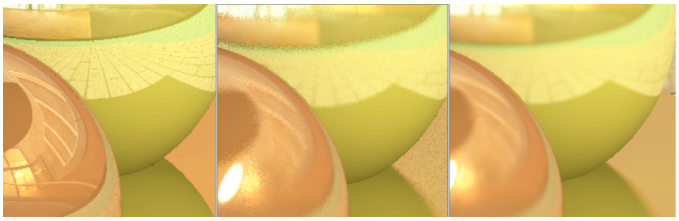


Fig 463 is a closeup showing the noise created, for each of the three scenes above.

Fig 463



One can go higher, for example a rate of 16 (16 x 16 = 256 samples). But you will likely impact on rendering times. in many cases 4 should give reasonable results. A Redway indoor scene example has the rate set at 4. For lower quality and 8 for medium quality. And 16 for high quality. Higher quality = slower render.

Enable area sampling.

OPTION: [Boolean: false, true].
Default: [true]
The value of this option lets the user enable or disable the calculation of area light sampling effects. If turned off, then all area lights are replaced by point lights using one single sample. This result in faster lighting and shadowing, for example suitable for all pre-visualization purposes.

Fig 464

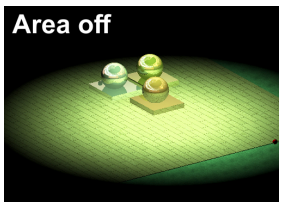
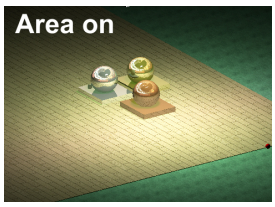


Fig 465



The next example uses a rectangular area light. The effect of turning on the area sampling in this particular case. Is to sample the larger area, rather than just the area lit by the light. Results may differ depending on the rectangular area light properties that had been set.

Light sampling rate.

OPTION: [integer: 1 to Infinite].
Default: [8]
Set the number of samples to be taken while sampling scene lights. A value of 0 means that the number of samples set to each light is taken. Any other positive values overwrite the lights settings. The real number of samples taken is the square of the option value. For example, if the option is set to 4, 16 samples will be taken.

The three images below show a difference, especially with the shadows. Too high a number will increase render times. Example, value = 32 squared = 1024 sample = very good quality, but significantly slower render time.

Fig 466 shows Light sample off. This then takes tha sample count from the light's themselves.
Fig 467 has a low sample of 2. And the shadows suffer from noise, and jaggies.
Fig 468 has sample at 8. This is a good improvement without excessive render time.

Fig 466

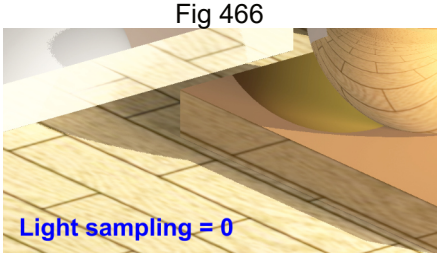


Fig 467

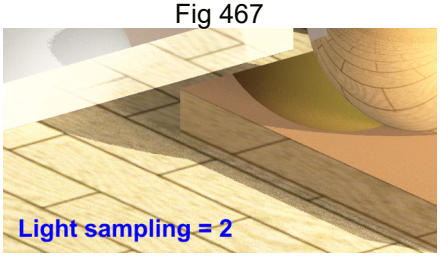
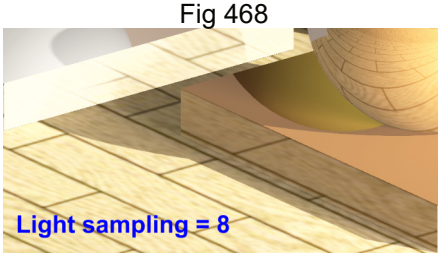


Fig 468



A Redway example has 8 for lower quality. 16 for medium quality, 32 for high quality.

Displacement.

OPTION: [Boolean: false, true].
Default: [true]
Activate software ray-tracer displacement mapping calculations when enabled. If the option is turned on (default) then displacement maps are used to modify the rendered geometries. If the option is turned off, then no displacement occurs.

Applicable if one has used a Displacement map in the material properties. See ....

Shadows.

OPTION: [integer: 1 to Infinite].
Default: [0]
Activates shadow casting for all lights that can cast ray-traced shadows when the specified value is greater than or equal to 1. Activation of shadows forces the activation of the primary ray option. The value of this option indicates the depth of the shadow casting process in the ray-tracing tree.

Operation can depend on the luminance used, and if shadow maps are used.

The two images in Fig 469 an Fig 470 use a spotlight. In Luminance properties, shadows was ticked, and shadow map not ticked. (Shadow maps may override this setting). When set to '0' no shadow is rendered. Setting to '1' allows shadows.

If one has shadows ticked in luminance properties, this should be set at one or above. If one has shadows unticked, in properties, but instead have just shadow map ticked, this can be left at 0.

Fig 469

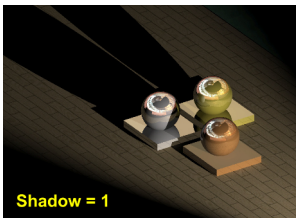
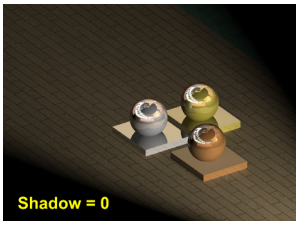


Fig 470



Reflections.

OPTION: [integer: 1 to Infinite].
Default: [0]
Activates the calculation of ray-traced reflections for objects that can reflect light when the specified value is greater than or equal to 1. Activation of reflection forces the activation of the primary ray option. The value of this option indicates the depth of the reflection calculation process in the ray-tracing tree.

Similar shadows above, Setting reflections to '0'. Prevents reflection from showing up. Fig 471 Setting it to '1' allows reflections to appear. Fig 472.

Fig 471

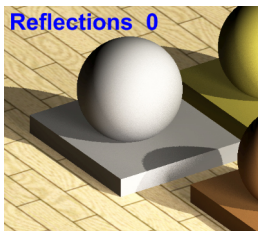
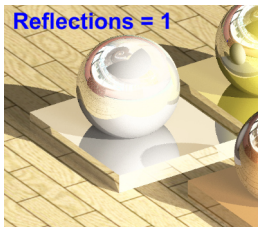


Fig 472



The floorboards are unaffected. As there is no material reflection added to it. Generally a material pattern will not be affected unless given a reflectance, like a varnished finish.

Refractions.

OPTION: [integer: 1 to Infinite].
Default: [0]
Activates the calculation of ray-traced refractions for objects that are refractive. Activation of the refractions forces the activation of the primary ray option. The value of this option indicates the depth of the refraction calculation process in the ray-tracing tree.

Switches on refraction rendering. A setting of '0' disables refractions, Fig 473. A setting of '1' enables refractions. Fig 474. As can be seen on the refractive lens the light bends withing the object, with no refractions it is flat.

Fig 473

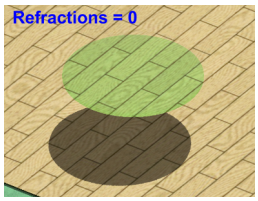
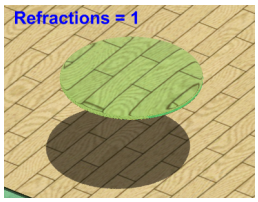


Fig 474



Transparency.

OPTION: [integer: 1 to Infinite].
Default: [1]
Activates the calculation of ray-traced transparencies for objects that are transparent. Activating this option does not force the activation of the primary ray option. The value of this option indicates the transparency depth that will be rendered for transparent objects.

This setting may be overridden if Refractions is set to 1 or above. With a plain transparency as in Fig 473 above, a setting of '1' shows the transparency.
A setting of '0' turns the transparent objects off. As in Fig 475

Fig 475

Force Transparency.

OPTION: [Boolean: false, true].
Default: [false (disabled)]
This option controls the behavior of the ray-tracer vs. the number of transparency layers that are rendered. By default, this option is disabled so that when a primary ray hits an opaque surface, its propagation is stopped. Turning this option on let the software ray-tracer pursue the propagation of a ray, even behind opaque surfaces. The need to see behind opaque surfaces arise in rendering configurations that are using a non-natural depth buffer setup. For example, rendering methods to see backward edges, section cutting, custom depth ordering, or material priorities are rendering setups that may use an exotic depth buffer setup.

Have no clue what the above means.

Reftractions back.

OPTION: [Boolean: false, true].
Default: [False]
When this option is off, the last refractive rays of the rendered ray- tree pass through all refractive objects to only hit opaque objects. For example, if we consider a refractive surface sphere over an opaque ground, and render that scene with a refraction depth set to one, then refractive rays will show the inner face of the sphere if the option is set to true, or they'll show the background if the option is set to false.

Disabling this option results in better looking images - however with fake refraction rays for the last depth - whenever there's not enough refraction depth to get through all refractive layers.

Max threads.

OPTION: [integer: 1 to Infinite].
Default: [2048]
The Red engine ray-tracer uses every processor available in the host computer for its rendering. This option can be used to clamp this number to a maximal number of threads running in parallel for the ray- tracer, so that some CPU bandwidth can be reserved for other tasks.

Cutoff.

double 0.0 to 1.0].
Default: [0.02]
The value of this option is used as input for the ray-tracer to set the minimal intensity of a ray that is considered as having a valid contribution to the final image. Note that a 0.0 cutoff value removes all optimization effects that can be related to using a cutoff. The smallest meaningful cutoff value is at 0.005.

Soft shadow map transparency.

[boolean]: [false, true].
Default: [true (enabled)]
This option let the user control the behavior of software shadow maps. A shadow map that is rendered by the graphic card is not able to take into consideration partial occlusion resulting of transparent objects. It can ignore transparent objects or consider them as shadow blockers for the shadow map

This is not the same for software shadow maps, that can cumulate the lighting attenuation caused by partially transparent occluders found on the path from the light to the surface. This option is then used to turn on or off the contribution of transparent surfaces.

Antialiasing sampling rate.

OPTION: [integer: 1 to Infinite].
Default [2]
Sets the anti-aliasing used in the render. Higher numbers can smooth out the edges, at the cost of increased render time.

A value of '4' gives a reasonable balance, though if overall rendering speed is a factor, leave at '2'.

Fig 475b

Fig 475c

Enable denoiser.

OPTION: [Boolean: false, true].
Default: [false]
This option turns on the denoiser that can modify software generated images during a post calculation pass. The denoiser will try to remove noise in the image while preserving all its details.

Denoiser strength.

OPTION: [integer: 0.01 to 10].
Default: [1.0]
Defines the strength of the denoising effect applied to software generated images if ENABLE_DENOISER is activated. The strength can be assimilated to a kind of blur effect applied to the image. If set to a stronger value, the denoiser will smooth more details in the image, resulting in a less noisy result but at the expense of losses in the details in the original image.

Denioser radius.

OPTION: [integer: 0 to 100].
Default: [10]
Defines the effect radius of the denoising effect applied to software generated images if the ENABLE_DENOISER is activated. A larger radius means that the denoiser will analyze more samples to remove the noise in an image, increasing the time it'll take to denoise the image and possibly increasing the quality of the result. Increasing the radius will strengthen the denoising effect too.

The three denoiser controls work together to try and remove or smooth out rendering noise, without having to increase sampling rates. The drawback is that there may be some loss of quality, depending on the items rendered. As with many settings, increasing the strength and radius, can impact on render times. The radius controls the amount of pixels used to sample an area.

Fig 476 shows noise on the wall with low sample rates and denoiser off.
Fig 477 shows the affect after turning on denoiser, and setting a low rate of radius.
Fig 478 shows the effect after increasing the radius, so more pixels are used to sample the area of noise. This results in a much cleaner wall.

Fig 476

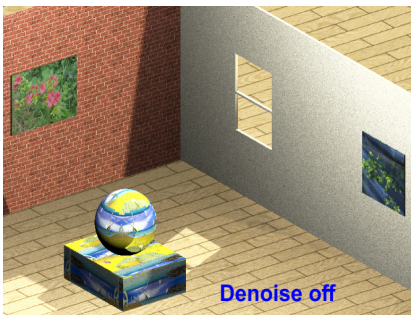


Fig 477

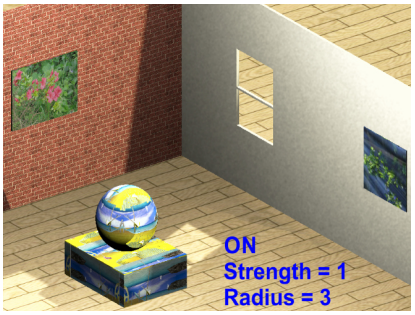
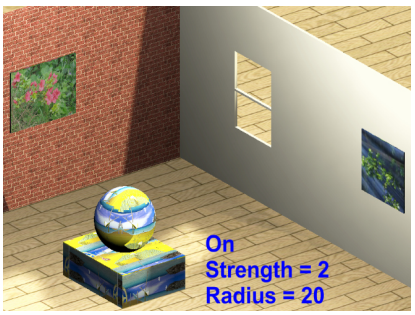


Fig 478



Enable Caustics.

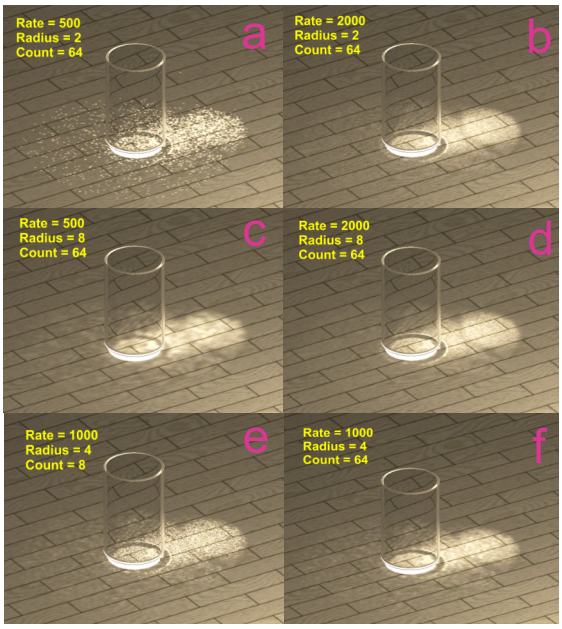
OPTION: [Boolean: false, true].
Default: [False]
Activate the calculation of caustics for transparent and reflective objects.
Note: that the generation of caustics depends also on the setup of the object materials and the lights.
Caustics are supported by physical lights only.

A switch to enable or disable caustics.

Caustic sampling rate.

OPTION: [integer: 0 to 100000].
Default: [1000]
Set the sampling rate of the caustics photons. The total number of caustics photons for the whole scene is the squared value of the sampling rate (default is 1000x1000 photons).

Fig 479



Caustic search radius.

OPTION: [double: 0.0 to Infinite]. = n
Default: [4.0]
When rendering caustics, the n nearest photons from the point being shaded are queried. This option defines the radius of the search area around that point in scene units. The bigger the radius, the smoother the caustics.

Caustic search count.

OPTION: [integer: 0 to 4096].
Default: [64]
Number of photons used for the caustics estimation. The photons search is limited by the value of CAUSTICS SEARCH RADIUS but if more than CAUSTICS SEARCH COUNT are found, only the CAUSTICS SEARCH COUNT nearest photons will be used to estimate caustics.
If the option is set to 0 then all the valid photons inside the search radius will be used to compute the caustics estimate.

Caustics require a material for the caustic effect.
(See .....). Plus a luminance capable of generating the effect. The three Physical luminance work fine, when using standard raytrace (non GI) render. But on has little control over the finished effect. Gi rendering allows more control over the caustic settings.

Especially allowing one to blur the effect, more in line what one would see from water.

Fig 479 'a' to 'f' shows some of the different settings all acting together to allw the desired effect

Volume.

OPTION: [integer]: [0,inf].
Default]: [0].
[scope]: All options instances.
Activate the calculation of ray-traced volume for volume shape containing volumetric effects when the specified value is greater than or equal to 1. The value of this option indicates the depth of the volume calculation to be processed in the ray-tracing tree.

A value of '0' disable the volume calculations.

Turns on the volume effect (a setting of '1' or above) If one has set up a volume in environments. And set the appropriate luminance to generate the effect.

Although the volume works similar in standard raytrace and GI raytrace. One advantage of GI, is that one does not have to set up an extra light, to get a subtle reflection one other walls / objects.
Fig 480 is standard raytrace, and nothing else can be seen.
Fig 481 Uses GI, and there is a small reflection of the light onto an adjacent wall. This is more realistic, and closes to real life.

Fig 480

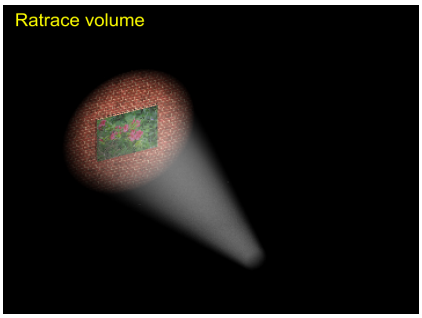
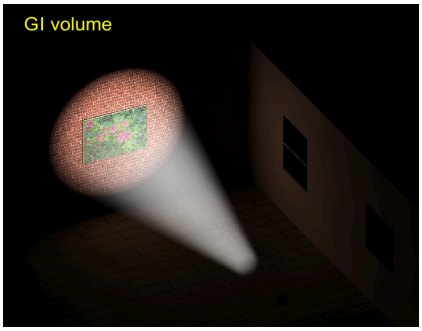


Fig 481



GI

OPTION: [Boolean: false, true].
Default: [false]
Enable or disable Global Illumination rendering.

A simple switch. Although one can use Gi rendering, just to alter the raytrace settings,(which cannot be done in standard raytrace quality rendering). Realistically. If one is choosing GI render mode, you will want to advantage of global illumination. And probably leave this set to GI.

Use GPU

OPTION: [Boolean: false, true].
If on the GPU will be used for the ray tracing in the render, if available.

This options tells the renderer to ‘attempt’ to use the Graphics processor for the render calculations. Because of the huge variation in graphics processors, not all the RedSDK functions may be availlable for a given Gpu.

Use transparency FFP GPU

[boolean]: [false,true].
This option rules the kind of GPU shaders used by the GPU hardware transparency. By default REDsdk's GPU ray-tracing shaders are using REDsdk's high definition viewing matrices that can provide an increased numerical accuracy during ray propagations. Turning on this option will force REDsdk's ray- traced GPU transparency to use legacy Fixed Function Pipeline matrices instead of high definition matrices. Turning on this option might be required in a mixed GPU rendering environment where different shaders have to co-exist for the rendering of a single frame, so that all shaders are based on the same numerical accuracy.

Use embree

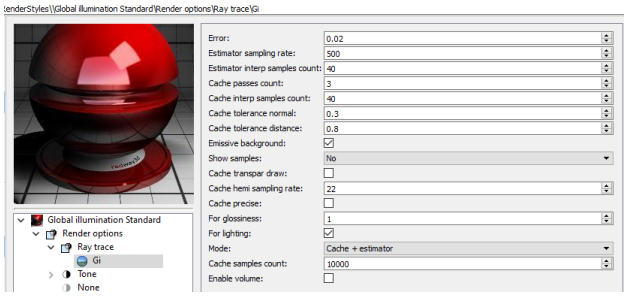
OPTION: [Boolean: false, true].
Default: [true]
Intel Embree is a collection of ray casting methods. Thanks to its careful handling of SSE instruction sets, it can provide an acceleration ratio between x1.2 and x2 to the rendering of images. However, it does not currently support lot of RED features: double precision matrices and geometries, lines, displacement...

When the option is turned on, the Embree ray-caster is used only for triangle-based geometries in 32bit float precision (vertex channel). Other types of geometries or geometries using higher precision vertex channels are still processed using the RED ray-casting mechanism. Please note that for geometries processed by Embree, the transformation matrices are casted to 32bit float (which may result in a loss of accuracy). If the option is off, the whole data is processed using the RED ray-casting algorithms. If GPU-based ray-tracing is used, the option value is automatically ignored and considered as false.

global illumination caches computed in world space and to shadow caches. Increasing the value will make finer caches that take a longer time to be processed.

GI

Fig 482



Error.

OPTION: [double 0.001 to 1000.0]
Default: [0.02]
Indicates the error tolerance of the global illumination solution. The error is inversely proportional to the global illumination precision. A good choice is too leave the error to its default value

Decreasing this error would enhance the final global illumination quality but at the price of a bigger memory consumption and a longer rendering time.

Estimator sampling rate.

OPTION: [integer 1 to 65535]
Default: [1000]
Indicates the squared root of the number of rays used to compute the estimator.
This is the sampling rate of the estimator, i.e the squared root of the total number of rays that will be traced to compute the indirect illumination estimator (first GI pass). You can increase this value for a better estimator convergence but the default value will be good most of the time.

Reducing the estimator figure, may reduce render time, but the quality will suffer.

Increasing the estimator, produces better quality, but as always increased quality results in increase render time (default is 1000) . As can be see in Fig 485. Increasing estimator, produces a clearer render in the shadow areas.

Estimator interp samples count.

Fig 483

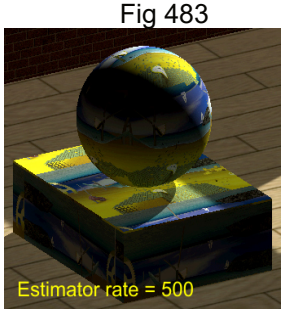


Fig 484

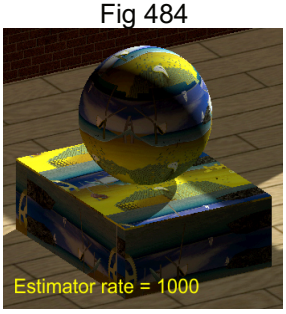
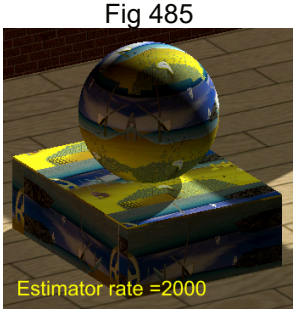


Fig 485



No information found in TC help, nor on RedSDK docs. A test increasing from the default of 10, up to 100, made no discernible difference. Except to increase rendering time.

Cache passes count.

OPTION: [integer -127 to 127]
Default: [3]
Indicates the number of indirect illumination cache refinement passes if positive; if negative, it is interpreted as a minimum convergence threshold.
The indirect illumination cache is built by iteratively refining an initial set of irradiance samples taken into the scene. The initial set a density proportional to the ERROR value. Then a fixed or variable number of refinement passes are processed to locally refine the cache following scene geometric and/or lighting variations. The fixed number of passes is given by the value of the CACHE PASSES COUNT option if positive. If negative, a convergence threshold is computed as: convergence threshold = 1.0 + 1.0 / CACHE PASSES COUNT. Then, the cache is refined until it converges to at least the convergence threshold value.

Cache interp samples count.

OPTION: [integer: 1 to Infinite].
Default: [20]
Indicates the number of samples taken while estimating the indirect illumination using.
This number is used after the global illumination cache computation, during the rendering phase. It influences the smoothness of the rendered indirect illumination. The bigger the interpolation samples count is, the smoother the result will be (meaning that small details will be lost). If a null or negative value is set, a value of 1 will be automatically assumed

Fig 486

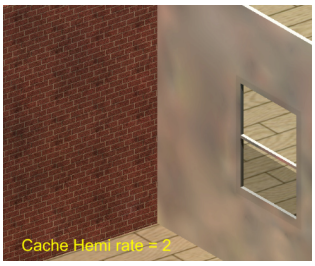


Fig 487

Cache tolerance normal. No known information

Cache tolerance distance. No known information

Emissive background. No known information

Show samples

OPTIONS: [integer: 0, 1, 2 or 4]
Default: [0 (none)]
This option is mainly for debugging purposes as it discards any shading operation to only render the position of the selected
GI samples. Its use should be reserved to expert users.
0 means show no GI samples. 1 means show GI estimator samples. 2 means show GI cache samples. 4 means show pixels
where GI cache samples were used to approximate glossiness (in magenta).

Cache transpar draw.

OPTION: [Boolean: false, true].
If this option's value is false, then an indirect irradiance cache set on a VRL won't be rendered for transparent objects for all cameras that have the option turned off. This saves render time in configurations where transparent objects are almost not diffuse (all common glasses) and for which the global illumination contribution can be neglected. By default, this option is disabled, so that we render faster ignoring transparency rendering passes.

Cache hemi sampleing count.

OPTIONS: [integer 1 to Infinite]
Default: [22 (22 x 22 = 484 samples)]
Indicates the number of samples to be taken while sampling the hemisphere. The real number of samples is the square of the option value.
This is the square root of the actual number of samples taken to estimate the global illumination at a given location during the global illumination cache computation. If a null or negative value is set, a value of 1 will be automatically assumed.

The default setting of 22 should be fine. Reducing the value can cause unusual GI effects.

Cache precise.

OPTION: [Boolean: false, true].
Default: [false]
Enable the high precision computation mode for the global illumination cache. If enabled, small geometric details are automatically detected by the engine and better GI estimation is performed. This is good to avoid interpolation artefacts over fine geometry variations like cabinet gaps, tiles edges, window frames...

For glossiness.

OPTION: [double 0 to 32].
Default: [1.0 (enabled)]
This option enables the use of the global illumination cache for the acceleration of glossiness computation. Because glossiness is a blurry effect, we can use computations of cheaper quality for rays being spread through a glossy reflection/refraction as they will finally be averaged together. Once turned on, the option lets the engine to reconstruct an approximation of the lighting reaching a point by requesting the global illumination cache instead of directly compute the illumination. This can dramatically reduce the time needed to render a scene with a lot of glossy materials.
The lesser the value, the better the glossiness quality (a value of 0 turns glossiness approximation off). If no GI cache has been computed, the option has no effect and is considered as disabled.

Increasing the effect, slightly reduces render time. With a slight loss in glossy quality. In some instances, this loss In quality will be barely noticeable, and totally acceptable.

Fig 488

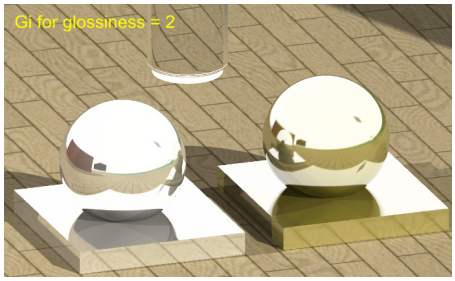
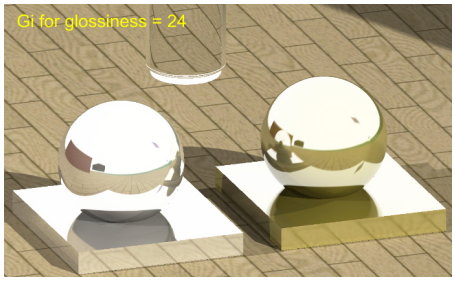


Fig 489



For lighting.
OPTION: [Boolean: false, true].
Default: [false]
This option enables the use of the global illumination cache for the enhancement of sampled lights rendering.
If turned on, the engine uses the global illumination cache to concentrate sampling on meaningful lights. This generally leads to an increase in lighting quality with no decrease in rendering time. However, the effect of the option is closely related to the quality of the GI cache. Hence, it performs best with default or higher GI cache settings.

If no GI cache has been computed, the option has no effect and is considered as disabled.
This option has no effect if path tracing is used to render the scene

Mode

Cache samples count. (World)
[integer]: [1,INT_MAX[.
[default]: 10000.
[scope]: All options instances.
This option rules the number of samples that'll be calculated for a given world space cache. This applies to global illumination caches computed in world space and to shadow caches. Increasing the value will make finer caches that take a longer time to be processed.

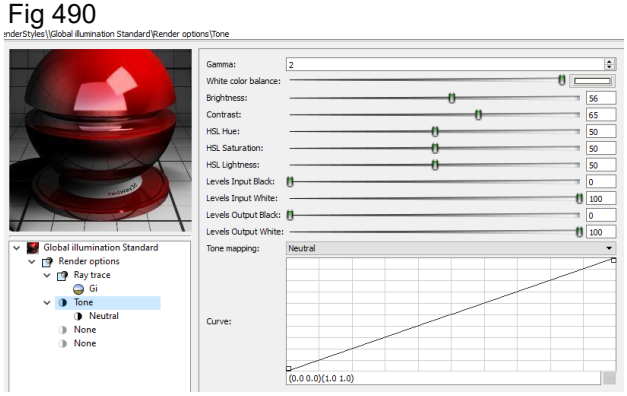
Enable volume.
[boolean]: [false,true] .
[default]: false .
[scope]: Alloptions instances.
Enable or disable the volume rendering during the global illumination process.

Volume effect is essentially light rays passing through a medium, (like dust in the air). Therefore it needs turning on in the ray trace section to make it work. However, also turning on under GI, may actually have a detrimental effect.

On my computer, turning Volume on, in the GI section. Appeared to cause GI to concentrate on the volume area, and ignore any reflections off the wall onto other surfaces. Feel free to experiment, with it turned on and off. As your finding may be different to mine.

Tone Mapping. Note. Also available in selection info palette for post processing.

Fig 490



The tone mapping section, is pretty much what one would see in a paint program for image adjustment.

See tone mapping on the Redway RedSDK website for a full explanation of tone mapping.

Gamma.

OPTIONS: [integer 1 to Infinite]
Sets the gamma value for image gamma correction. Due to the non-linear response of the monitors, gamma correction may need to be applied to image prior to displaying them onto the screen.
Gamma correction is applied as follows: output color = input color ^ (1.0 / gamma)

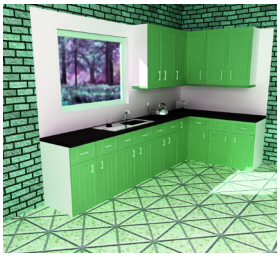
White colour balance.

OPTIONS: [color, as RGB/HSV]
Sets the white color to be used as reference when applying white balance.
Color balance is needed to compensate for un-natural looking of images rendered under some conditions. For example, an indoor scene by night lit only by artificial light sources may look too yellow. At the opposite, an outdoor scene may look too much blue.
A kind of auto white balance (as provided by consumer cameras) can be achieved by passing the average image color to SetWhiteReference. If not, the input color components are clamped to the [0,1] interval.

Brightness. Alters the overall brightness of the scene.
Contrast. Alters the overall contract

To set HSL back to default after alteration. The sliders need to be in the middle. An easy way is to use GI render style dialog, to alter the tone so the figures are 50 (this sets the slider mid way)

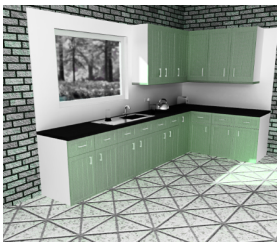
Fig 491



HSL hue. [Default 50 (midway]

Alters the main colour of the pixels in a scene. For example changing a scene from its normal hue to green.

Fig 492



HSL saturation. [Default 50 (midway]

Alters the saturation of a pixel, for example In using the colours within TC colours we have green_yellow - spring green - lime green

Fig 493

Hsl lightness. [Default 50 (midway]

Alters the lightness or darkness of a colour, The example right. We we can set the saturation to produce a black and white image, then use the lightness to adjust the greyscale

Level input black. [Default black]
Level input white. [Default white]

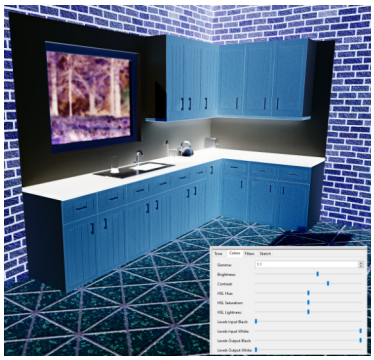
Fig 494

Similar to adjusting the start and end level of histogram tone adjustment. Adjusting the black level can make shadows appear darker. Adjusting the white level, will brighten the white areas

These two adjustments should be used with care, as it is very easy to take things too far and ruin an image, Personally I leave these at default.

Level output black. [Default black]
Level input white. [Defaul twhite]

Fig 495



As with the previous two levels, use with care. Small adjustments may help an image, large adjustments can completely change the image.

In the example, we set black over to the right (100), and set white to the left (0).

This will invert the image. In the same was as using the Invert command within a paint program. Why you would want to do this - No idea.

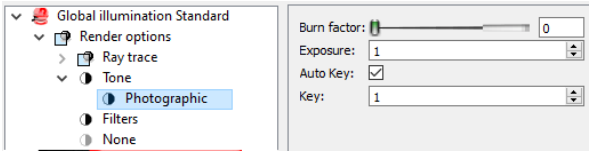
Tone mapping.

OPTIONS: [clamp, neutral, photographic]
CLAMP = RGBA color components are clamped to [0.0, 1.0].

NEUTRAL = This is the neutral 'no tone mapping' operator that does not change any input color.

PHOTOGRAPHIC = Tone mapping operator derived from photographs' best practices.

Fig 496



Photographic, releases further adjustment parameters for image adjustments.

Burn factor. One may think burn should make things darker. This works the opposite, it brightens the image.

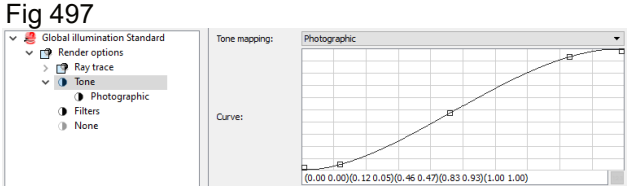
Exposure. The exposure starts at '1'. In photos one can use under exposure - normal - over exposure. To do that in RedSDK one would alter the exposure setting, up or down for example 0.8, 1, and 1.2. This can e done by typing in a value. Sometimes the up-down arrows only move it 1 whole unit.

Auto key. Greys out the key setting. In favour of auto mode.

Key. No idea of the actual definition of the key adjustment. It can be adjusted manually, in the same way as exposure. And seems to do a similar job.

Curve.
OPTIONS: [curve via points]
Sets the curve to be applied to image colors.
A curve can be defined which transforms input intensity. This is a common and easy way to apply intensity corrections to a whole image. For example, an S-curve is efficient to increase image contrast by lowering low intensity values and boosting high ones.

Fig 497



The curve is set by passing curve samples. Each sample is made of its position along the x-axis as well as the curve value at that position. No more than 65535 samples can be provided.

The curve is built back from the samples using Lagrange polynomials for interpolation.

The curve is altered by clicking the line, and moving the node that will appear, into the desired position. A variety of shapes can be used to alter the tone without having to alter the brightness or contrast.

To delete a node. Delete the relevant figures in brackets below the curve. The figures are x / y coordinates of a node, going from (0,0) to (1,1)

The images below, show different curve setups. Fig 498, is rendered with the the default straight line

Fig 498

Fig 499

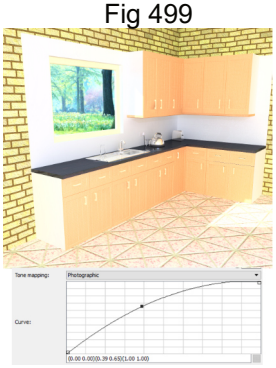


Fig 500

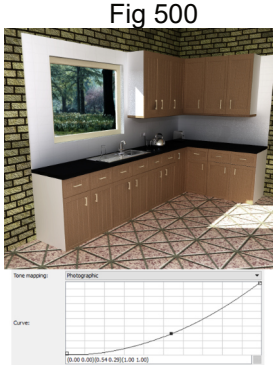
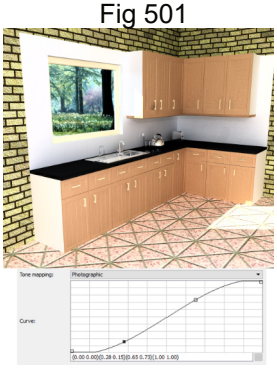
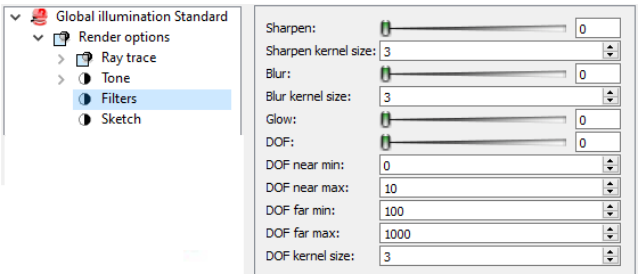


Fig 501



Filter.

Fig 502



Filter section, allows for post processing of the render. And includes alterations for sharpness, blur and depth of field.

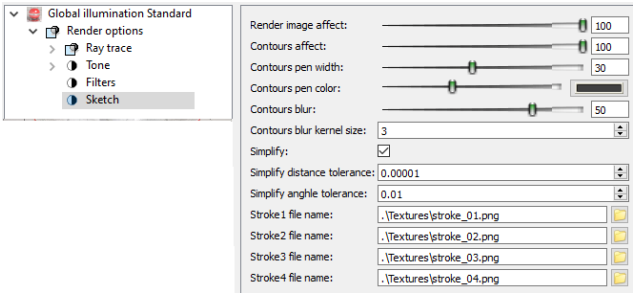
Post processing options are also available in the selection palette.

Sharpness. This adjustment is pretty subtle.

Blur. Blur's the image.

Depth of field. This blurs areas near the camera or far away from the camera (or both). Set the DOF slider, and the kernel size, to suit,

Fig 503



Sketch.

Sketch mode is probably more of a novelty than a serious CAD.

Some post processing features are available in the selection palette. At present the file names and colour are not available post process.

The images below are simple examples of different effects. Setting the image effect to maximum allows some of the rendering textures to show through

Fig 504

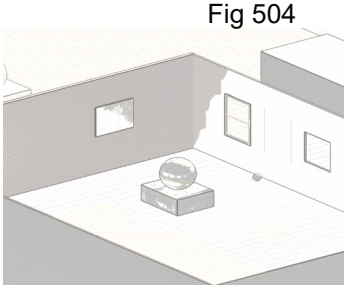


Fig 505

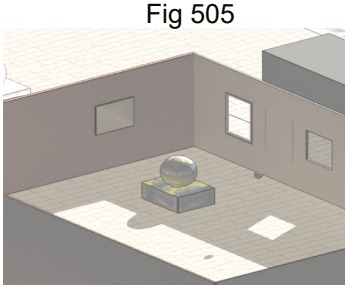
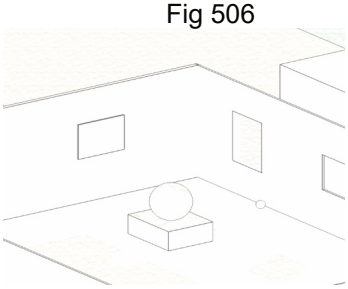


Fig 506



Some Redway suggestions for GI settings.

Low

RAY_GI_ESTIMATOR_SAMPLING_RATE	200
RAY_GI_CACHE_PASSES_COUNT	1
RAY_GI_CACHE_PRECISE	false
RAY_LIGHTS_SAMPLING_RATE	8 (default)
RAY_GLOSSY_SAMPLING_RATE	4
RAY_GI_FOR_GLOSSINESS	8.0

This 'low' quality setting is clearly focused on increasing the rendering speed. We reduce the amount of time spent in the global illumination in reducing the GI estimator quality and in reducing the number of cache passes. Clearly, this will reduce the quality. Then, we also discard all costly glossiness calculations by increasing a lot the glossiness interpolation. Sampling rates are low, so that there we may see noise in the resulting image.

Medium

RAY_GI_ESTIMATOR_SAMPLING_RATE	1000 (default)
RAY_GI_CACHE_PASSES_COUNT	3 (default)
RAY_GI_CACHE_PRECISE	false
RAY_LIGHTS_SAMPLING_RATE	16
RAY_GLOSSY_SAMPLING_RATE	8 (default)
RAY_GI_FOR_GLOSSINESS	2.0

This medium setting raises the GI quality to an acceptable level. Most images can be produced using these 'out of the box' settings, without having to raise them further. As a choice here, the light sampling rate has been increased. Depending on your own application scenery, if you don't have many lights, or only small lights, the light sampling rate value can be adjusted significantly. Glossiness are still interpolated, but less than before, therefore increasing their resulting quality.

High

RAY_GI_ESTIMATOR_SAMPLING_RATE	2000
RAY_GI_CACHE_PASSES_COUNT	5
RAY_GI_CACHE_PRECISE	true
RAY_LIGHTS_SAMPLING_RATE	32
RAY_GLOSSY_SAMPLING_RATE	16
RAY_GI_FOR_GLOSSINESS	1.0 (default)

Finally the high quality settings are all about generating a really high level image. The GI quality is strengthened a lot. If your models don't have many fine details to handle, you can consider keeping medium settings for the GI. These settings are meant to solve visual problems that could occur in very complex indirect lighting situations. So the cache precise option - which comes at a certain cost - can be ignored for many models. Similarly, we have increased a lot the light sampling rate, assuming that we may have many lights to manage. All these settings should be adjusted for each application. Some applications will not require such a high light sampling rate (for instance an automotive outdoor rendering application may require less samples due to the fact that the lighting is always sun / sky based); whereas some other applications (an object design application) may require a higher glossy sampling rate. So the key point here is that these presets should be just used as a starting basis for each application to tune.

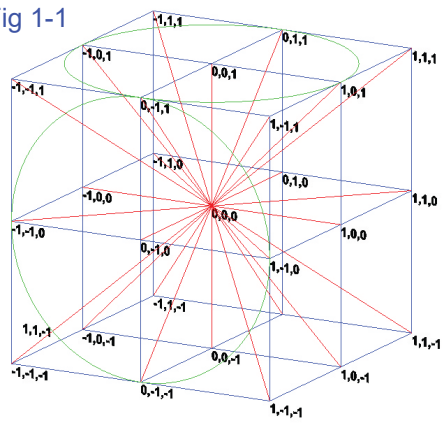
RAY_REFLECTIONS set to 3,
RAY_REFRACTIONS set to 3,
RAY_TRANSPARENCY set to 3 (mimics the refraction depth),
RAY_SHADOWS set to 4. Plus one here! The shadow ray is a ray cast at a greater depth than the ray it comes from!
OPTIONS_SHADOW_MAP_DEPTH set to 4 (mimics the shadow depth for lights casting shadow maps instead of ray-traced shadows).

Appendix 1 Vectors

Whilst the normal way of thinking of angles in TC is degrees or Radians, With material and light, including sun, there are also vectors. Although not always strictly necessary, often they need to be entered in the range of -1 to +1.

The origin, would be the centre of extents of the drawing (not an object). Classed as being at vector 0, 0, 0 (this is not world coordinates)

Fig 1-1

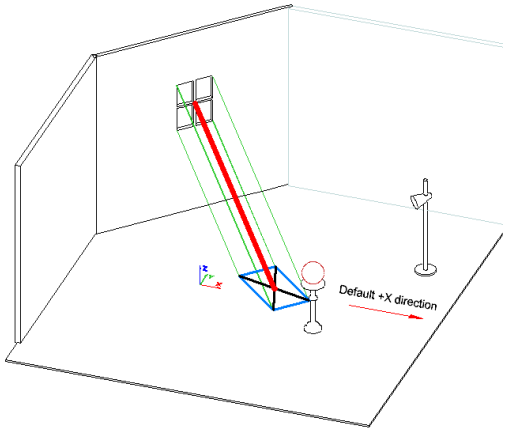


As can be seen in Fig 1-1, all directions can be contained within the -1, 0, +1 matrix. There are a few things in TC which can use this type of coordinates, Including RedSDK Sun direction. And for these examples, we will use sunlight in RedSDK.

Sun vectors are generally represented on a spherical surface, However they can be represented on two 2d planes, x,y. And x,z (or y,z).

There are a few ways we can get the correct vectors. One can use vector calculations (a later example).
Here are a few examples. They all start with the setup shown in Fig 1-2. In all cases we are trying to get the sun to shine at the blue rectangle drawn on the floor. We use the red 3d polyline as the direction needed.

Fig 1-2



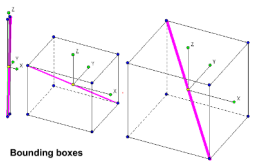
NOTE. Some images may be shown 'over complicated' . This is for illustration purposes only. And does not need to be included to obtain the correct vectors. But in all these examples,.
The red direction line is needed.
It is a 3D polyline drawn whilst the workplane is set to world.

If the sun is pointing in an orthogonal direction. i.e. pointing straight down, or in the direction of the orthogonal horizon. Then it is simply a matter of deciding whether the directions are + or minus. Other than that, we need some way of getting the x, y, and z vectors.

The quick method. (Selector Entity coordinate system to Extents)

First, set selector coordinate Entity system to Extents.

Fig 1-3

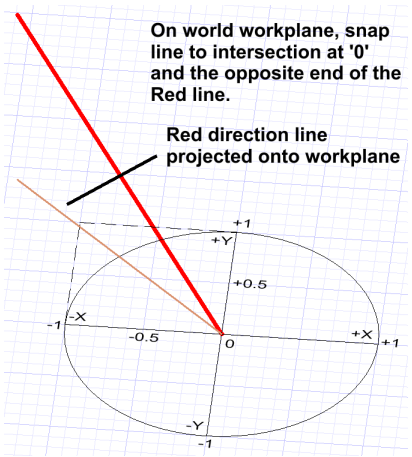


2D and 3D object have a bounding box. Irrespective or what direction is used. Fig 1-3.

What we need first is to decide on whether the sun x, y and z vectors are pointing at the origin point (0), from the positive of negative direction.

In Fig 1-4, looking at the red direction line, (I projected the tan coloured line onto the workplane for clarity). We can see the X direction is -X, Y direction is +Y, and Z is above the horizon, therefore +Z.

Fig 1-4

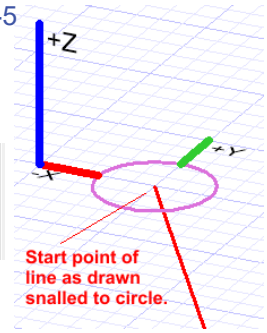


The middle image in Fig 3 above would be -X, -Y, +Z.
Draw a circle on the world workplane. This is strictly so that, if, when we re-size the red 3d polyline, it gets too small to see, or if we click away from the line. we can still easily find it again.

Make a copy of the red line, Relocate reference point onto the start vertex. If you cannot remember which way you drew it. The start vertex can be found by checking the start vertex coordinates, in the selection palette - metrics. If one uses the wrong end of the polyline, then when resized it could be a tiny line somewhere in space, difficult to find.

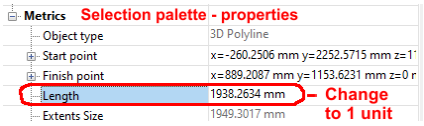
Snap the direction line to the centre of the circle. I drew the axis marker for visual reference only. Fig 1-5.

Fig 1-5



Access the selection palette - properties section, and expand metrics.

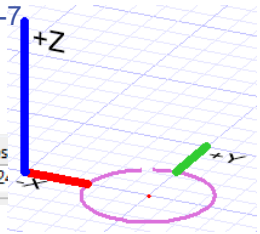
Fig 1-6



Change the length of the line to 1 unit. i.e. 1 mm, 1 inches etc. Fig 1-6. Do not use the inspection bar.

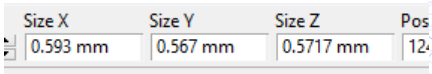
As can be see on the right. The line can become pretty small, hence why the circle makes it easy to find. Fig 1-7.

Fig 1-7



Select the line, and read the sizes in the inspection bar. Fig 1-8. Sizing the object in the selection palette down to 1 unit, means the sizes will fall into a 1 x 1 matrix.

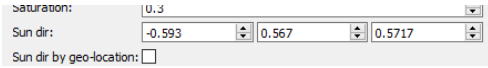
Fig 1-8



Next. Assign a positive or negative direction. As we found on the last page, X direction is -X, Y direction is +Y, and Z is +Z.

Therefore the three vectors we need are X = -0.593, Y = 0.567, and Z = 0.5717, Type these into the sun direction (Fig 8a). Rendering the scene, we can check if it is pointing as we would like. Fig 1-9.

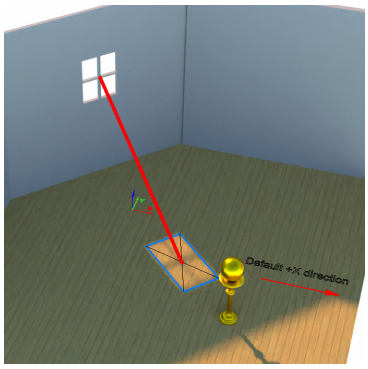
Fig 1-8a



IMPORTANT NOTE.

If one has applied a sun to an object rather than a scene luminance, then. The above may only works if the 3d line bounding box In selector extents = Entity, and it aligns with the x and y world directions.

Fig 1-9



For example. If the drawing (not the view) is rotated, the bounding box is also rotated Fig 1-10. This means the above procedure probably won't work. Therefore we need to get the objects bounding box, aligned. The simplest way is to select workplane by world, and snap a new 3d polyline into place.

Now we have the bounding box correct, Fig 1-11. We can look at the drawing and determine the new x,y,z directions.

From the strike point (the blue rectangle. We can determine the line (sun) is shining from -x direction, -y direction, and above the horizon so + z direction.

Comple the steps as previously stated.

Fig 1-10

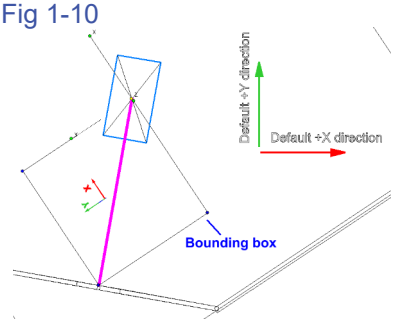
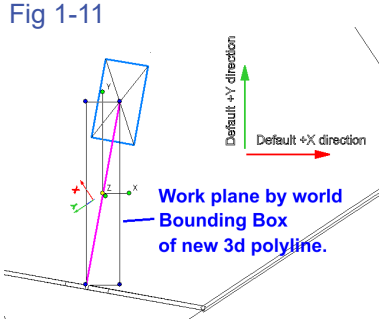
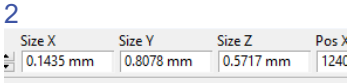


Fig 1-11



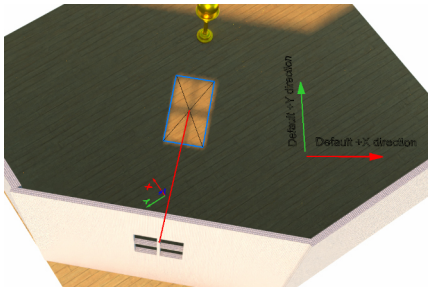
On the inspection bar we have X = 0.1435, Y = 0.8078, Z = 0.5717 Applying the + or - sign to the inspection bar size fields. In this case both x and y are in the minus directions.

Fig 1-12



We get vector sun direction of. X = -0.1435, Y = -0.8078, Z = 0.5717.

Fig 1-13



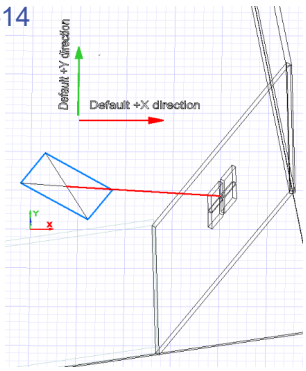
This first method of finding the vector is the simplest, providing on can get the workplane correct to start with.

However there may be times when one does not have a full bounding box. For example, if the red line is drawn on an orthogonal workplane, and then the drawing is rotated, It can show Z as zero.

One could then either re-draw the 3D polyline, or use a different method.

One could then use a fully manual approach. Where the coordinate system doesn't matter.

Fig 1-14



This is not totally accurate, but close enough, and quick onceits been done a couple of times. The same start drawing is used as previously, but rotated. Fig 1-14.

We will use the grid to get a 20 x 20 area on the world workplane.

Draw a circle to represent the spherical surface. Snapping the circle to a major grid line will make it easy to view as 20 x 20.
For reference, draw lines to represent +- X, and +- Y Fig 1-15.

In front view, we add a a line to represent + and - Z direction (shown blue). Fig 1-16.

Fig 1-15

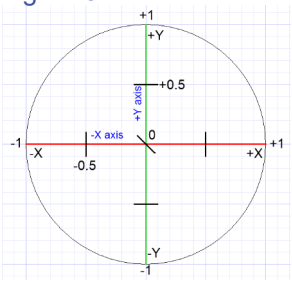
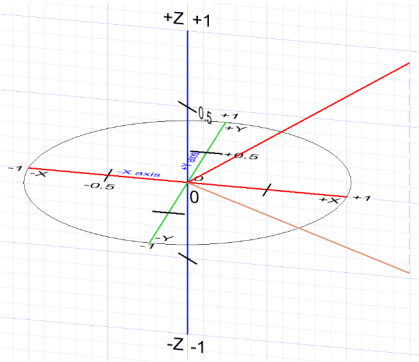


Fig 1-16



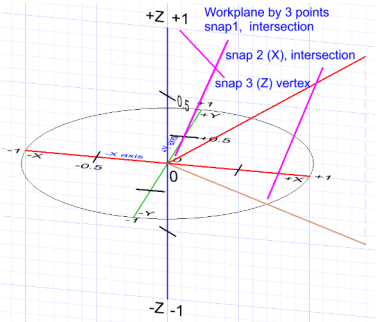
Make a copy of the red 3d direction line. The reference point is relocated to the target vertex. And snapped to the centre of the circle.

The diagonal tan coloured line in Fig 1-16. represents the red direction line as 2d on the world workplane. Simply switch to world workplane. 'C' snap a 2d line to centre of the circle, and V snap to the end of the red direction line. Because it is a 2d line, it will project the end point onto the workplane.

This is used for the z direction. To set the workplane.

I added the + and - markers for illustration purposes only.

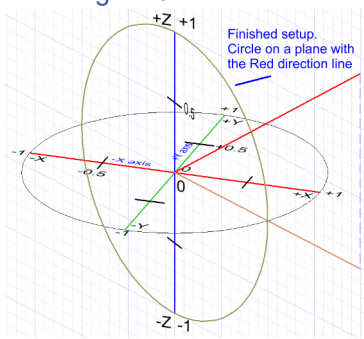
Fig 1-17



Set the workplane, to that of the red sun direction line. Fig 1-17.

Workplane by 3 points, snap to the centre of the circle (c). Intersect snap (i) where the 'Tan coloured' line crosses the circle. And finally vertex snap (v) to the top of the +Z line.

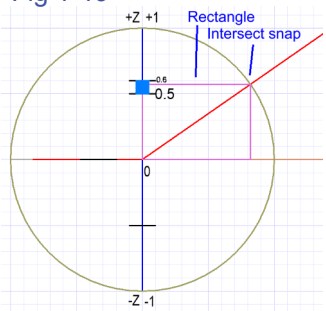
Fig 1-18



Snap a new circle on this workplane. Fig 18. The same size as the original circle, which can be typed in, or 'v' snap to Z, or 'I' snap to 'tan' line and circle, or snapped to grid (g).

From here there are multiple options. But as this is estimation, we will stick to 2d only.

Fig 1-19

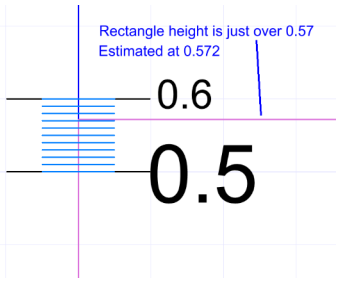


As we still have the workplane set to the direction line. Use View menu - camera - plan - by workplane. Or choose UCS plan (ctrl + shift + w) (Depending on ones TC interface)
.
Draw a rectangle, snapped to centre of the circle. And intersect (i) snapped to the junction of the red direction line and circle edge. (shown orchid). Fig 1-19.

Using the grid we can estimate the junction on the Z axis. (this is the z vector). Fig 1-20.

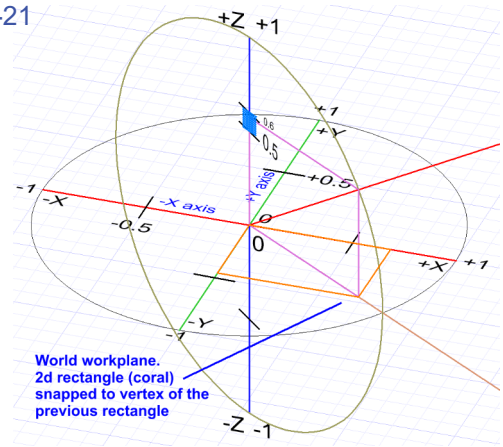
I drew some extra lines for clarity between the grid marks.

Fig 1-20



As can be see (Fig 20), this just makes estimating easier and more ac

Fig 1-21



Switch to ISO view and World workplane. draw a rectangle, snapped to centre of circle, and the vertex of the previously drawn orchid (z direction) rectangle. Shown coral in Fig 1-21.

Switch to world workplane.

Estimate the coral rectangle vectors, as shown below.
Fig's 1-22 to 1-24.

Fig 1-22

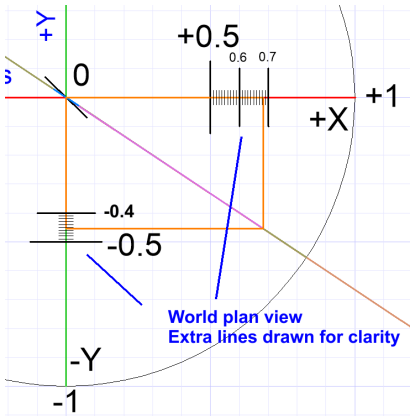


Fig 1-23

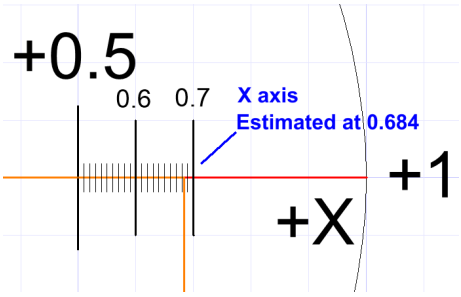
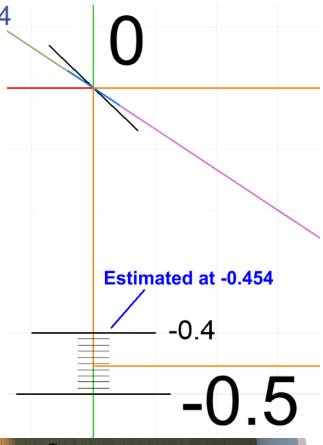


Fig 1-24



These give us our final vectors, X = 0.684, Y = -0.454, Z = 0.572.
Entering these into the sun direction settings, Fig 1-25.

Fig 1-26

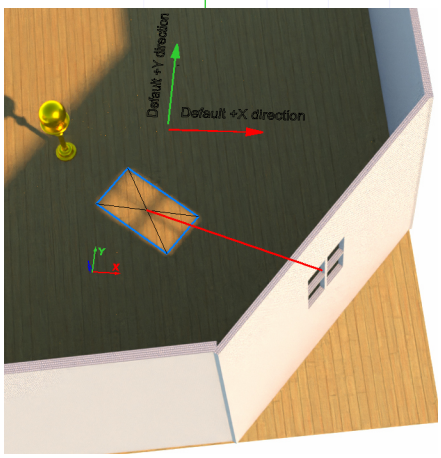
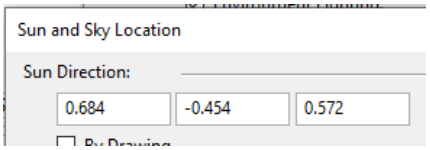


Fig 1-25

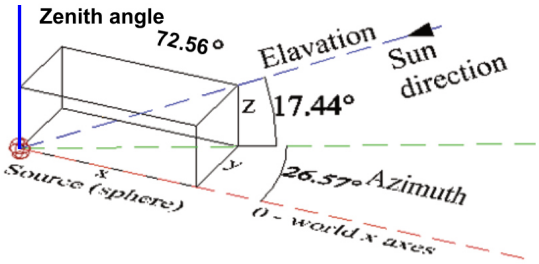


we can check our estimation.
Fig 1-26. Not an exact match to the blue rectangle, but pretty close, and suffice for most applications.

In case anyone doesn't wish to estimate. One can calculate using the grid and the rectangles. In our scenario, each grid space is 80mm, and each quadrant is 10 x 10 to grid squares.
Therefore = 10 squares x 80 mm = 800mm total.

X vector (coral rectangle length) is 546.6122 mm / 800 mm = 0.68327
Y vector (coral rectangle width) is 363.3633 mm / 800 mm = 0.4542 (change sign to minus)
Z vector (orchid rectangle height) is 457.3644 mm / 800 mm = 0.57171

Fig 1-27



The last example, is more mathematics.

And uses angles rather than drawing grid lines.

This relies on calculating the zenith angle, and the azimuth angle. Fig 1-27. And is the most accurate as it takes away much of the 'human error' factor, by reducing the amount of snapping required.

The angles required are from world +X in an anticlockwise direction to the sun direction line. 0 to 359.999 etc.

And the angle from +Z to the sun direction. 0 to 90. Assuming the sun is above the horizon.

We will use the original drawing for this example so we can compare figured to the 'quick method'. Fig 1-28.

Fig 1-28

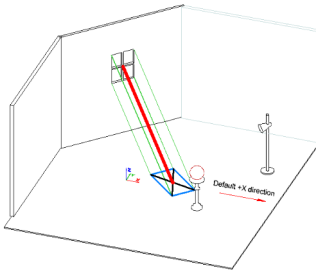
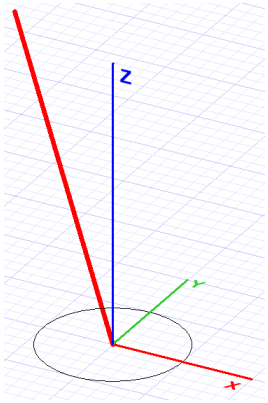


Fig 1-29



The setup is very simple. Fig 1-29.
We have our X, Y and Z indicators.
A circle, which is only for reference to the world workplane.
And our sun direction line, snapped to the XYZ intersection.

For the first angle, we need it from world +X to the direction line, anticlockwise. We can use the angle tool, with 'Angle node' selected (inspection bar, or right click menu) Fig 1-30.

Fig 1-30

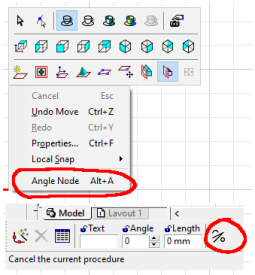
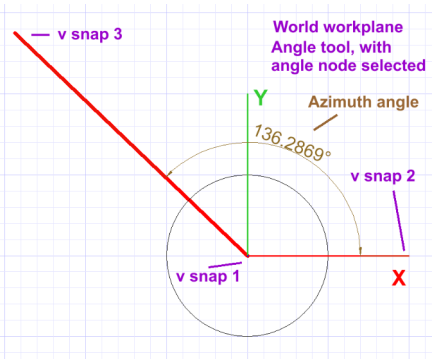


Fig 1-31



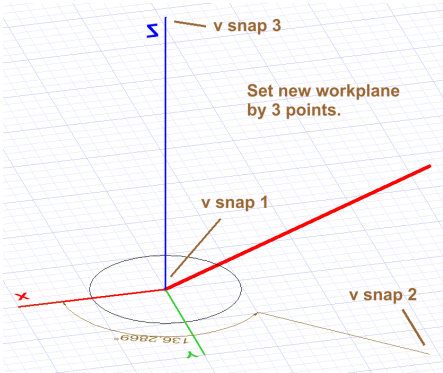
Or as previously, we can project a 2D line onto the world workplane, and measure as normal.

We can use three vertex snaps to obtain the angle. As shown in Fig 1-31.

This we will call angle A and is 136.2869 degrees

NOTE. There are multiple ways to obtain the Zenith angle. This is an example of one way. It may not be the simplest, nor the quickest way. But it works.

Fig 1-32

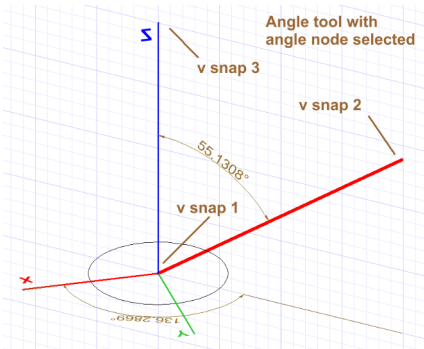


To get the correct zenith angle, we need a new workplane by 3 points, in line with the direction line.

For once, the angle extension lines come in handy as a snapping point. Snap the workplane as shown in Fig 1-32.

Last setup is to get the zenith angle. Again using the angle tool with angle node selected if necessary.

Fig 1-33



And that is all the setup we need. We now have the z angle for the vector calculation. 55.1308 degrees. Fig 1-33.
We will call this angle B

So we have our two angles. A = 136.2869. And B = 55.1308. (but we need them in Radians). One can convert first, or just add Radians into the calculation.

For the vectors, we will use three calculations. One each for X, Y and Z.

For 'c' in the following calculation we use 1 as the units size. Because we are only interested in direction, not magnitude. (I got the calculations via Internet).

The following calculations are for the x, y and z boxes in the sun direction entry dialog box.

X = c * sine (B) * cosine (A) - - - = 1 * Sin (Radians (55.1308)) * Cos (Radians (136.2869))

Y = c * sine (B) * sine (A) - - - = 1 * Sin (Radians (55.1308)) * Sin (Radians (136.2869))

Z = c * cosine (B) - - - = 1 * Cos (Radians (55.1308))

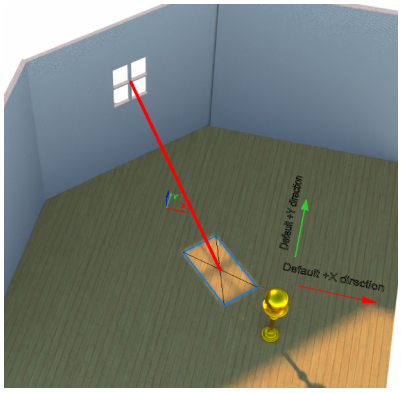
Therefore (figures rounded to 6 decimals)

X = Sin (0.96221398) * Cos (2.37865513) - - - = -0.593036

Y = Sin (0.96221398) * Sin (2.37865513) - - - = 0.566977

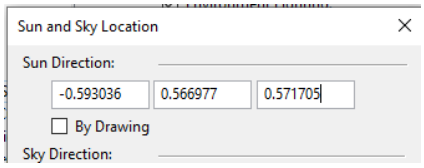
Z = Cos (0.96221398) - - - = 0.571705

Fig 1-35



Now its time to enter the coordinates in the Sun luminance direction boxes. Fig 1-34.

Fig 1-34



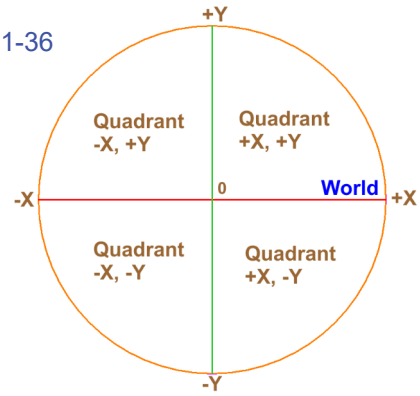
Cross ones fingers and hope our calculations were correct.

As can be seen from Fig 1-35. The sun shines down pretty good through our window, onto our floor blue rectangle.

As previously stated, there are other ways to get the vectors required. And probably much easier ways. This is document is just three ways to get one started.

Obviously if one is using Geo-Location, then none of this is necessary. This is for occasions where you need the sun to always point in a specific direction, or at a specific location.

Fig 1-36



One last thing about quadrants. When deciding the positive or negative x and y signs. Fig 1-36. Simply shows the sign needed in each quadrant .

Disclaimer. I've tested this many times, but I have not tested every combination of X, Y, Z coordinates. Therefore cannot guarantee the methods will work in every scenario.

Andy

Appendix 2 Texture scaling

Page	
About scaling	*A2-1*
Offset	*A2-2*
Size texture to fit object	*A2-3*
Size Object to fit image	*A2-3*
Close enough is OK	*A2-3/4*
Tile Image	*A2-4*
Tile to real world sizes	*A2-5*
End notes	*A2-6*

Ideally the texture should be the same aspect ration of the object it is being wrapped onto. This is to prevent undue scaling of the texture in one direction.

And there are some situations where this is important. A company logo, for example. If someone has spent ten's of thousands of pounds, getting their perfect company logo. You cannot just stretch it out of all proportion.

But many times some scaling does not matter, because a it will not be unnoticeable, concrete for example. On many occasions though, you simply need to choose the best method to ensure the texture wraps without too much distortion.

Fig 2-1

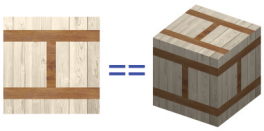


Image size in pixels. Unless one is designing an image to be a perfect size for an object, Then this only affects the quality of the wrapping. If the original image is 256 x 256. It will almost never look as good as 2048 x 2048. No matter how you try to apply it to the object.

In the past, textures were often 512 x 512 for square textures. Nowadays, one can have 1K (1024 x 1024), 2K (2048), 4k (4096) and even 8K (8192) textures for download,

If the scene is lots of small objects, you could get away with 512. For larger objects, and where an image is being tiled. 1K or 2K, may be needed. Wrapping very large objects with a single texture may require 4K. But I think it would be very rare that you would need an 8K texture in TC.

Fig 2-2

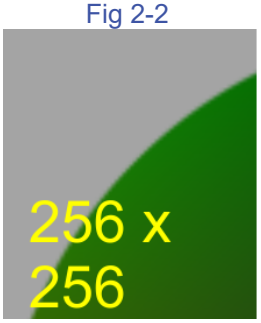
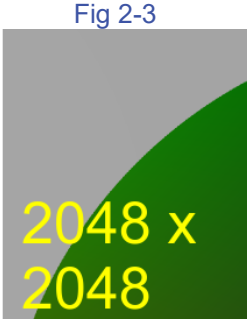


Fig 2-3



What happens during wrapping. In some rendering programs including RedSDK. One can think in terms of one wrap per unit. i.e. when using space units as mm, it is one texture wrap per mm. It doesn't matter what the units are, mm, inches, km, mile, or light years.

Pattern texture.

If we draw a 1mm square, with a wrapping scale of U and V = 1. It will wrap perfectly. If you draw a 10 mm x 10 mm square, at U, V scale of 1, it will place 10 images wide and 10 images high. (Fig 2-4) Because it is still wrapping 1 texture per drawing unit.

If we wanted the single texture to wrap once on the 10mm square instead of 10 times, we need to amend the wrapping scale U and V. Our single texture is one tenth of the object. So 1 / 10 = 0.1 Therefore entering U = 0.1, and V = 0.1, we will get a single texture covering our 10mm x 10mm square. Fig 2-5

Remember this is drawing units. So if units were feet, a single texture would be 1ft x 1ft, An the calculation is 1 / object wraps in feet.

To sum up. For any size of object, if we want a single texture to be applied across the whole object face. We simply divide 1 by the objects width for U, and 1 divided by height for V. (see tiling for a variation).

Fig 2-4

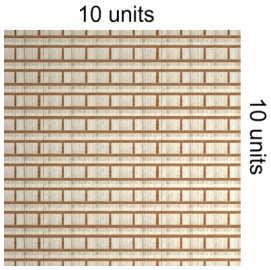


Fig 2-5

Wrapping - Texture space.

With texture space, the image is wrapped as a direct scale, no need to use the 1/object for the scale factor. See the pages on Texture space for more information.

NOTE, TC can wrap faces of an object differently. For example, on a cube, the top face may be wrapped differently than the side face. This is not a bug, just the ways the object mesh is unwrapped ready for the material to be applied. See last page for wrap on face.

What is the offset boxes for. **Pattern textures only**

Fig 2-5

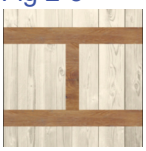
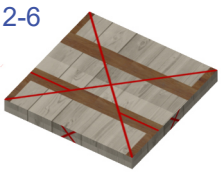


Fig 2-6



When a texture is wrapped, it isn't just dumped in the middle, and can be trial and error. For square / rectangular objects, it appears to wrap slightly different to non rectangular objects.

Looking at the Fig 2-6, we can see it has not wrapped in the centre of the square. This is often undesirable. And is where the offset comes in.

Fig 2-7

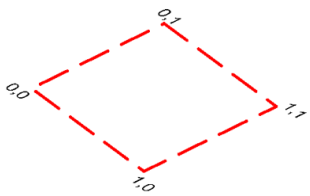
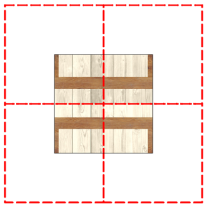


Fig 2-8



Offset can be thought of in terms of the image. '0, 0' being the bottom left, 1, 1 top right. In order to shift the image into position on the object, we need to move it over by ½ the image, in X and Y.

Fig 2-9

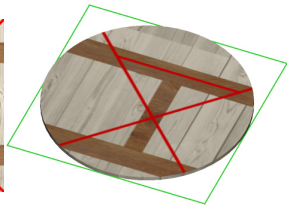
Therefore half way betweeo 0, 0 (x, y) and 1, 1 is 0.5, 0.5 . we can type 0.5 in both the U and V boxes. To push the texture into the desired place. Fig 2-9

However if using tiling, for example two or more copies of the image on the same face, this changes things slightly.

As a general rule, tiling an odd number 1, 3, 5 etc. one would offset by 0.5. Tiling even numbers 2, 4, 6 etc. offset = 0.

Fig 2-10

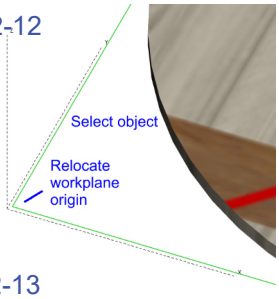
Fig 2-11



Now for a wrinkle. Non rectangular objects appear to be wrapped with reference to their own workplane not the object itself. To see this in action, the image right was wrapped with offset at '0'. As can be seen in Fig 2-11 the wrapping of not even centred in any way.

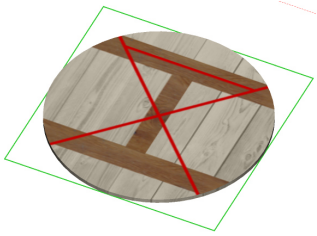
The green square is just a reference polyline.

Fig 2-12



To see the effects, we select the object and relocate its workplane origin. Snapping it to the bottom left corner of the green square.

Fig 2-13



Lastly, issue a 'place on workplane' command on the cylinder. As can be seen in Fig 2-13, the texture now wraps correctly. With the image being wrapped from the objects workplane.

Of course, instead of messing with the workplane origin. One would normally just adjust the offset, until the image wrapped correctly.

IMPORTANT. I cannot say this is how TC is programmed. Just what appears to happen on my computer. Your experiments may show a different situation. Therefore the above is just a guide only.

Texture scaling is discussed as 5 options.

1). Size the texture to fit the object. (aspect ratio).

Knowing the size of the object you have drawn, One can use a paint program to design or alter a texture, This is to ensure the same aspect ratio of the object. For example. You have drawn a box 500mm x 200mm. And using width divided by height, we get 500/200 = aspect ratio of 2.5 : 1.

Fig 2-14

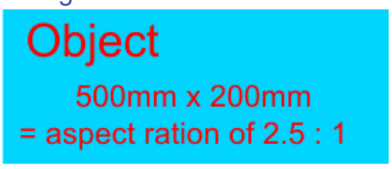
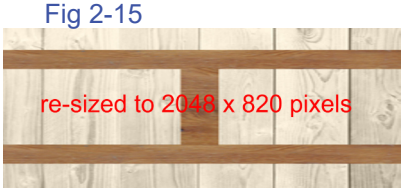


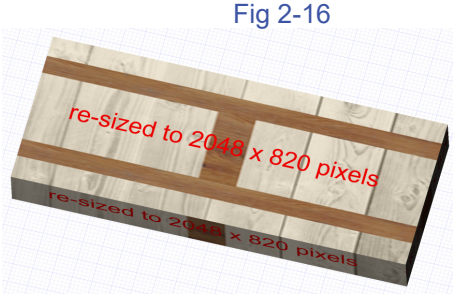
Fig 2-15



If we were using a texture of 2048 pixels wide, we could say 2048 / 2.5 = 820 (round up) pixels high. Designing the texture to 2048 x 820, should (no guarantees), look the same on the object as it does in the texture. In other words, no stretching or compressing of the image.

Pattern texture scale.

Fig 2-16



From there it is a simple matter to get Scale U and Scale V values. Here we just want a single texture to wrap once on the object. Therefore, for the width, divide 1 (U value) by the object width. And height, divide 1 (V value) by the object height.

For example. Our object was 500mm x 200mm, The texture size is immaterial now as we have got the correct aspect rations. So.
Scale U = 1 / 500 = 0.002
Scale V = 1 / 200 = 0.005.
As can be seen in Fig 2-16, The text looks good with no stretching.

Wrapping - Texture space.

Texture space - scale uses a different way of entering the figures. We don't use the 1 / object figure.

For a single wrap going off our 500mm x 200mm object size.

We Use 1 in both the scale U and scale V in the pattern texture scale boxes,

We then use 500, 200, 1 in the Texture space scale boxes

Fig 2-17

---

2). Size the object to fit the texture. (aspect ratio).

If one had a texture that cannot be resized in any way, without causing distortion. Then one may need to size the object so that the texture will fit, generally using the aspect ratio of the texture. A logo with text for example, may not be re-sizeable without distortion.

Fig 2-18

Example. We have a fancy logo that is 2500 pixels wide and 1200 pixels high. So, 2500 / 1200 = 2.0834 aspect (rounded up).

Our object is 800mm wide. This means we need our object height 800 / 2.0834 = 383.98 mm high (rounded down), to ensure correct aspect ration. Thus, we create a box 800 wide x 383 high. The object and texture should fit together nicely.

Pattern texture scale.

Fig 2-19

Therefore our object is 800mm, so scale U = 1 / 800 = 0.00125 and scale V = 1 / 383.98 = 0.0026.

Minor variations will often be caused because of rounding of the calculations, as a cautionary measure, make the object height on the lower side. Hence rounding the object size down. If it was a touch out, we could change the scale V to 0.00259 for example.

Wrapping - Texture space.

Fig 2-20

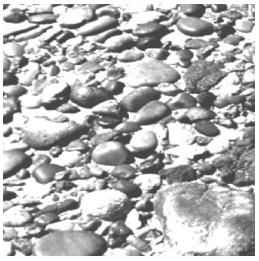
Texture space is the same as previously

We Use 1 in both the scale U and scale V in the pattern texture scale boxes,

Because we have resized the object. We use We then use 800, 383, 1 in the Texture space scale boxes

3). Close is good enough.

Fig 2-21

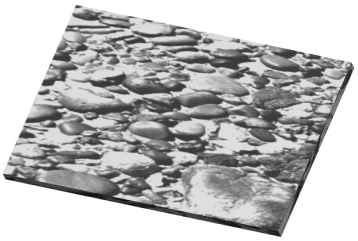


Many wrappings don't have to be exact, and whether the texture matches the object is of no concern. Marble for example. Has the impurities which give it the coloured effect. But assuming it is not man made, then every one is different. So it doesn't matter if the texture is stretched a bit.

Wood, stones, concrete, some leather, rust, clouds, etc. All can be scales a small amount in any direction without too much detrimental effect.
Obviously sometimes there are limits, like scaling wood too much across the grain can look poor. But many times it is acceptable.

Pattern texture scale.

Fig 2-22



And for this method, the wrapping, is very simple as before.
For the width, divide 1 (U value) by the object width.
And height, divide 1 (V value) by the object height.

For example. One has an object 400mm x 300mm x 20mm, The stones texture is 512 x 512. So obviously it won't fit. But, as we are not bothered about whether it stretches a bit. The texture size doesn't matter.

The calculation is simply U = 1 / 400 = 0.0025 and V = 1 / 300 = 0.0033 (rounded)

Wrapping - Texture space.

Fig 2-23

We simply enter the size of the object. For example. We have an object 420 mm x 380 mm.
Therefore we enter 1 in the pattern texture scale U and V.
We then enter 420, 380, 0 in the Wrapping - Texture scale boxes

Fig 2-18

4). Tiling (rough guess).

On many occasions image tiling is required. Many texture one creates, or downloads from the internet are 'tiled'. Which means the left side fits perfectly against the right , and same with top to bottom. Whilst this does mean repeating patterns, the same applies to a lot of things. Like wallpaper for example.

With many wrapped textures on large objects, it is not possible, or at least practical to avoid repeat patterns, There is little point in joining two images together in a paint program. Unless you specifically want to invert, or offset one of the images.

When deciding on tiling, there are two options. a). Decide on the amount of tiles you would like. Or. b). Use a fixed know element size within the texture. For 'b' see the next section - Scaling to a fixed element. For this section we will be scaling based on rough guess.

Fig 2-24

Tiling simply requires one to decide how many tiles are appropriate. For example, you have a 2000 x 1000 pixel image of wood grain (grain running the 2000px length). And wish to apply it to a 2500 mm round table top. Scaling the 2000 along the grain to 2500 is acceptable. But scaling the 1000mm up as far as 2500mm will unduly distort the image.

Luckily the top and bottom of the texture match, so it can be tiled without showing a join. But an actual answer as to how to proceed, is personal preference. And can sometimes be trial and error.

Here, as we are scaling the 2000 length, we can afford to scale the width a bit, so to avoid 3 joints, we will use two tiles. Effectively taking the 1000 to 1250.

Pattern texture scale.

Fig 2-25

For scaling the 2000 up to 2500. We simply use U = 1 / 2500 = 0.0004
For 1000 width. We use V = 2 tiles / 2500 = 0.0008

It is totally up the the user now many tiles looks fine. Sometimes it may be necessary to find a starting point, like in this case 0.008, and adjust it either way until one is happy.

Wrapping - Texture space.

Fig 2-26

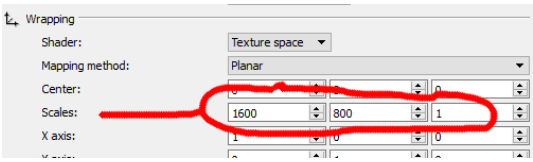
Here we will use the same texture on a 1600 mm square object.

Therefore we enter 1 in the pattern texture scale U and V.

As we want 1 texture wide and 2 texture high.

In the Texture scale boxes we enter
1600, (1600 / 2), 1 as 1600, 800, 1

Fig 2-26a

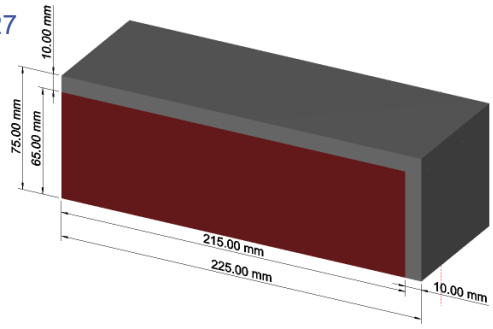


5). Tiling (fixed element - real world).

By fixed element it is meant a single known sized item, withing multiple items contained in a single image, For example bricks, tiles, chequer, grid etc. Taking bricks as an example, an single image may contain 7 bricks in width, and 20 bricks in the height, with mortar in between. Therefore a fixed element would be one brick wide + mortar. And 1 brick high + 1 mortar

This scenario doesn't rely on image size, nor object size. It relies on the real world dimensions of the element. **Note.** Using real world sizes, assumes the drawing is completed at a 1:1 scale in model space. If you are scaling model space, for example 1:10. Then the real world brick size, would also have to be scaled down by a factor of 10.

Fig 2-27

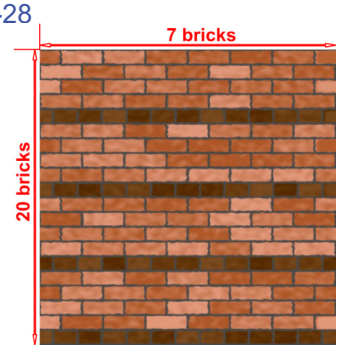


In this example, we will use a metric brick. Taking a rough dimensions of metric UK brick of 215mm x 65mm + 10mm mortar each, and having our drawing space units as mm.

We have 215 units wide, by 65 units tall.
Then allow for 10mm mortar, we get 225 x 75. Only include mortar once for top and one side.

Note. What is written below, may sound complicated at first. But it is very straightforward, and only takes a few seconds once you have done it a couple of times.

Fig 2-28



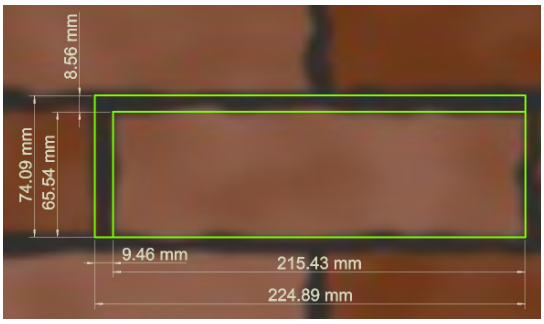
An important point. It will not be fully accurate, because we are guessing at the mortar size, and remember different textures may contain a different amount of brick wide and high.
For this example we will use the brick bond common texture. + don't forget to alter the bump map scaling as well as image map.

We know our texture has 7 bricks wide,
therefore 225 units x 7 brick = 1575.
And 75 units x 20 bricks high = 1500.

Pattern texture scale.

Using this information gained above, we can say
Scale U = 1 / 1575 = 0.0006349. And
Scale V = 1 / 1500 = 0.000666.

Fig 2-29



We can easily check how close we were by measuring a brick on the wrapped object. As can be see I didn't estimate the mortar correctly. But its reasonably close.

Using this method ensures the bricks will be roughly the real world size irrespective of what size the wall or texture is. (assuming your wall is big enough to accommodate them).

Using an template based on feet. and an American brick of 8 inch x 2 1/4 inch.
It would be 0.6666 ft and 0.1875 ft. + Mortar thickness of 0.03125 ft, gives 0.69785 x 0.21875.
Therefore 0.69785 x 8 = 5.5825, Scale U = 1 / 5.5825 = 0.17912 and
0.21875 x 14 = 3.0625. Scale V = 1 / 3.0625 = 0.32653
A second much simpler example. Would be to use a checker pattern for lino floor (so no need for mortar).

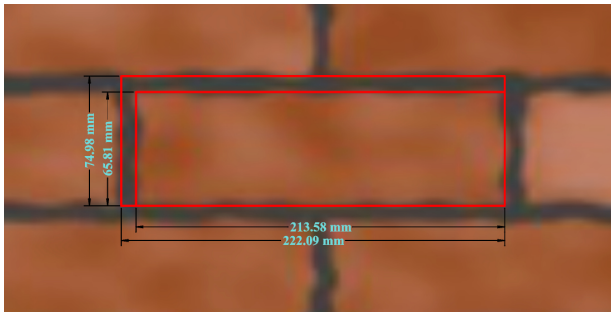
Wrapping - Texture space.

For using with Texture space scale units, Again we enter 1 in the pattern texture scale U and V.

For the Texture space we use the two figures calculated previously, i.e. We know our texture has 7 bricks wide, therefore 225 units x 7 brick = 1575. And 75 units x 20 bricks high = 1500.

We enter 1575, 1500, 1 in the Texture space scale boxes.

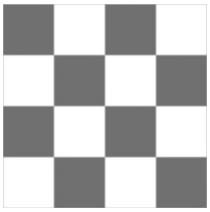
Fig 2-30



The result is close to being accurate.

This last example, uses a simple checker pattern.

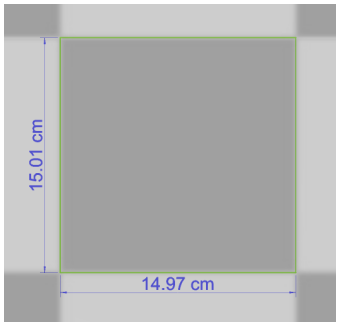
Fig 2-31



The checker pattern is 4 x 4, which makes it nice and simple. Our space units were set to Centimetres (cm). And we want each individual checker square to be 15cm x 15cm.

Pattern texture scale.

Fig 2-32



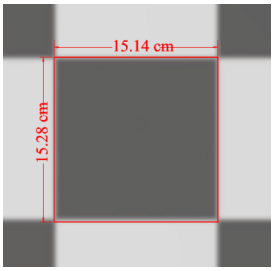
Because the chequer pattern consists of squares, the same value applies to both U and V scaling. Therefore it would be. 15 x 4 = 60. and 1 / 60 = 0.01666. For U and V.

Therefore we type in scale U and scale V boxes 0.01666, to get roughly the correct sized checker pattern. As can be seer right, Its pretty close when wrapped onto a box.

American users with a drawing in feet, would be.
6 inch squares = 0.5ft….. 0.5 x 4 = 2….. 1 / 2 = 0.5 in the scale U and scale V boxes.

Wrapping - Texture space.

Fig 2-33



We enter 1 in the pattern texture scale U and V.

With 15cm squares, and the 4 x 4 squares in the texture.
And scale units set to cm.
So 4 x 15cm = 60. We enter 60, 60, 1 in the texture space scale boxes.

With regards to images that have multiple element inside. For example, Brick, Roofing tiles, floorboards, Grid, Mesh, Tread (or chequer) plate etc. Providing one knows the 'real world' dimensions of a single element. etc. AND one is drawing 1:1 scale.

Remember, If using your own scale, for example a real world wall may get 8 metres, and you draw it 1200mm. You would have to take that into account when checking real world dimensions. And scale things to match, or the texture will not be correct.

Wrapping on a face.

All the above apply equally to wrapping on a single face of a 3d object. However if wrapping a material on say the side face of a cube, in a specific direction. It will generally be necessary to create a separate material, orientated to ones liking.

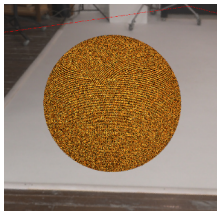
There will be times when normal wrapping won't work. On these occasions one will need to use the UV mapping tool, if it is available in your TC version.

UV wrapping.

TC's UV wrapping can be used with RedSDK. The setup is pretty much the same as previous UV wrapping.

Because the scaling of the map is handled by the UV mapping tools, this simplifies the image map properties. In that the u and v scaling can be set to 1 (or -1).

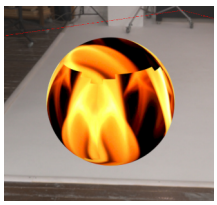
Fig 2-34



NOTE. This is not a UV mapping tutorial, see the help files, or internet for that.

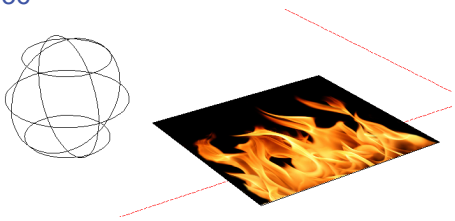
First a comparison image. This uses the traditional wrapped image (kd). And a u v scale of 1, in properties. As one can see at scale 1 there are lots of tiny wrapped images.

Fig 2-35



Altering the u v scale to make the wrap large (scale u & v = 0.003). Is trial and error. And doesn't really wrap good on a spherical object, This is a limitation of material wrapping.

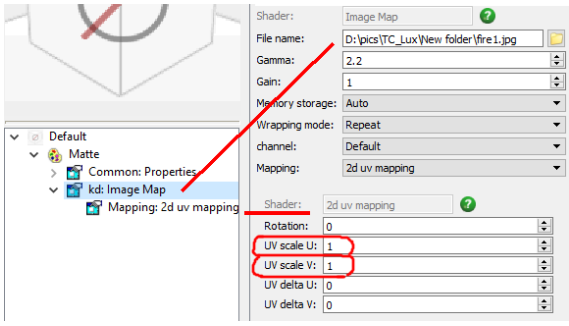
Fig 2-36



For the UV mapping tools.

First we will insert an image into the image list, and place in drawing.

Fig 2-37

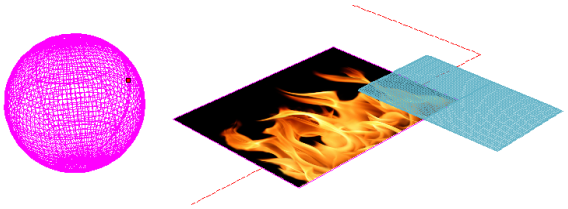


Whilst this can be done later, or using the UV wizard.

I created a default material and assigned the flame image. This was used in the first two images above. The U and V scales were set to 1.

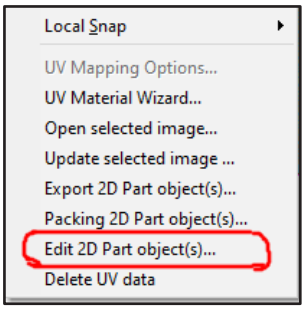
The UV mapping tool was selected.

Fig 2-38



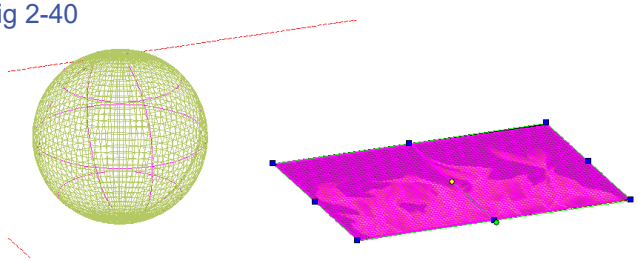
Selecting the image, then the object, will display the blue uv map.

Fig 2-39



Edit 2d part was selected so the map can be positioned.

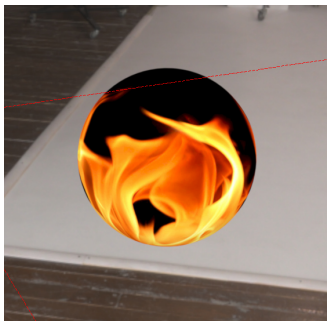
Fig 2-40



The map was edited to fit the size of the image.

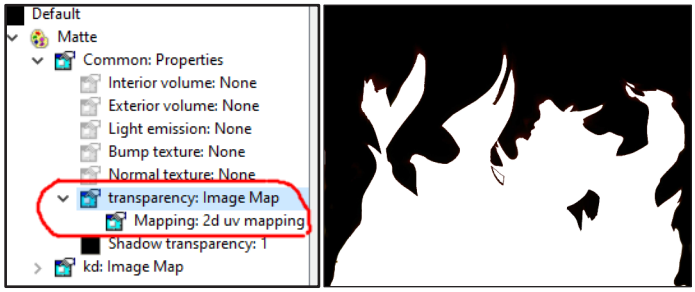
And Finish edit 2d part object selected

Fig 2-41



When rendered the flame wraps around the sphere. And will look a lot better than the standard imagemap wrapping.

Fig 2-42



It is also possible to add other effects, including adding a transparency map.

Transparency uses black as its transparent colour, therefore e create a black and white image, of what we wish the eliminate.

When rendered, the black parts of the transparency map are removed. Leaving just the flames in the finished render.

Fig 2-43

Appendix 3

Seeding 3d object.

Fig 3-1

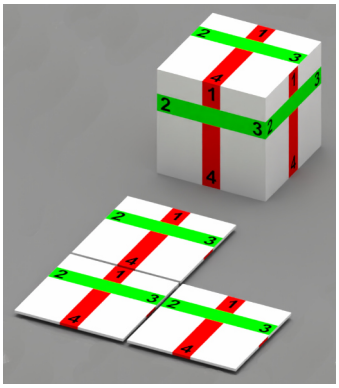
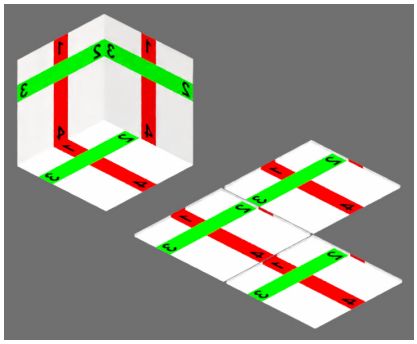


Fig 3-2



Material wrapping, depends on how the object was drawn, and on what work plane it had when it was drawn. This has nothing to do with the current work plane one may have set up since the object was drawn.

IMPORTANT…. I have no idea how TC was programmed.

There are a few ways software can wrap an image onto a 3d box, One way is to use a cross, just like wrapping up a gift. This is generally the way cube maps work.

In TurboLux it appears to wrap three faces of a box as positive images, Top, Front, Right (first image on right). And the other three faces are a negative image of the first three (Bottom, Back, Left). Far right image.

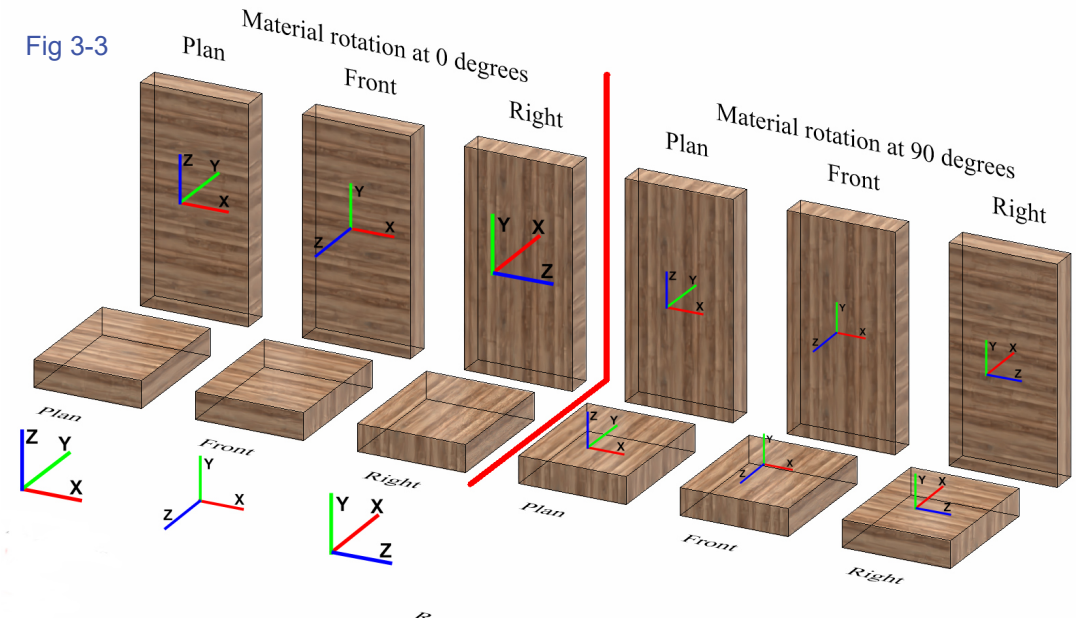
This is not new, and appears to be the default way TC wraps. This is possibly why, in RedSDK and Lightworks there were different wrapping modes. To counteract this effect.

At the time of writing, TurboLux only has the one default mode. What this means is, if one intends to add a texture which contains text, one needs to use TC's UV mapping tools. And not texture wrap.

As stated previously, wrapping depends on how one made the object. The image below shows the same texture applied to the same objects, which differ because of the workplane they were drawn on. As can be seen using the x, y, z, axes indicators. None have been rotated, they were simply drawn differently.

As can be seen, none of them have the correct grain orientation, in a situation where one is drawn flat on the workplane, and one is drawn vertically to the workplane. Example. The upright rotated plan object, has good front and side direction. But the flat rotated plan object, has an incorrect side grain direction.
An Ideal situation is to draw everything flat on the world work plane, and rotate the objects to suit.

Fig 3-3



Just like one would do in real life, when using plywood for example. This will ensure one only has to use one texture wrap to cope with all situations.

Life isn't always that simple, it is easy to forget and draw using the most convenient method. In these situations two texture s may be required.

Textures. My own personal preference, is to have the texture running top to bottom, as in the first image right.

For the next test we will assume the texture is running left to right (X direction). Image far right.

Fig 3-4

Scenario.

We have drawn lots of panels, and find just two of them incorrect. We know from a previous image, we can rotate the texture which will solve the vertical object, but we would have to create a third texture to also solve the flat object. Which we don't want to do.

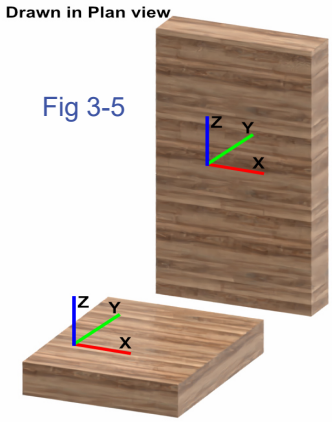


Fig 3-5

For this we can alleviate all problems by seeding the object to force the internal plane reference, to the direction we choose, thus no extra textures are required.

To get both direction correct we need two seeds orientated differently. One for the vertical object, and one for the flat object.

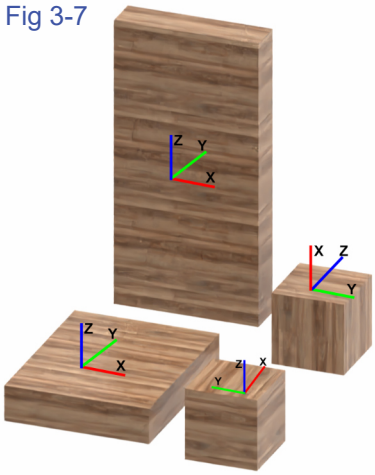
We could just look at the images on the previous page, and orientate the seeds to the same directions. But for this test we will use boxes.

If we look at the two boxes (right). We can simply re-orientate them to the directions we want the grain to run.

Fig 3-6

Plan view

Fig 3-7



Whilst in real life, a solid pieced of wood would have one of them end grain. When laminating (chipboard for example), we generally have long grain on all edges.

Far right, we have orientated the two boxes in the desired pattern. As can be seen the two boxes need to be orientated differently for each panel.

The boxes are the scaled down by 0.1 in all three scale boxes. And the boxes are them moved so that there completely inside the target object.

The last step is to add the box and target together. But first , ensure that options menu - advance preferences. 'Transfer material from operand on its facet' is NOT ticked.

Select the 3D add tool. Select the box first, then select the target object.

The material x, y, z axes of the target, is changed to that of the box.

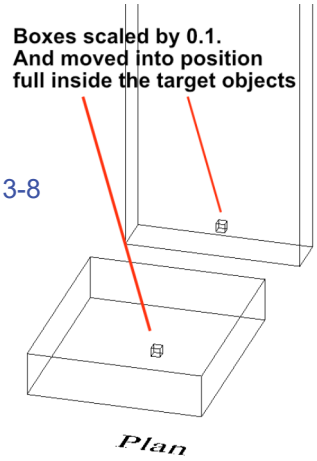


Fig 3-8

The final render is shown in the image right.

The same method can be used to orientate any object where the material direction is not what is required.

Fig 3-9

There may be occasions where many panels have incorrectly orientated materials. Providing two objects are NOT touching each other, one can boolean add the objects, apply the seed, then explode into separate parts.

Fig 3-10

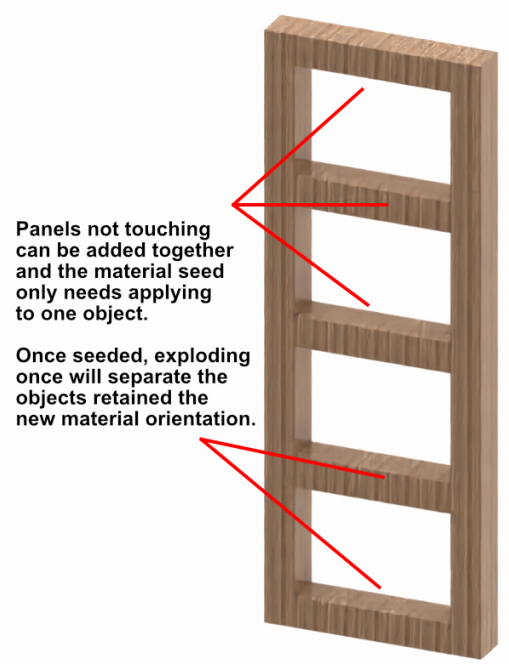


Fig 3-11

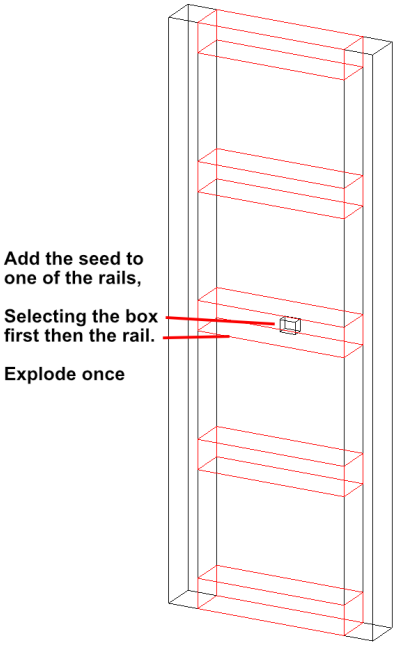
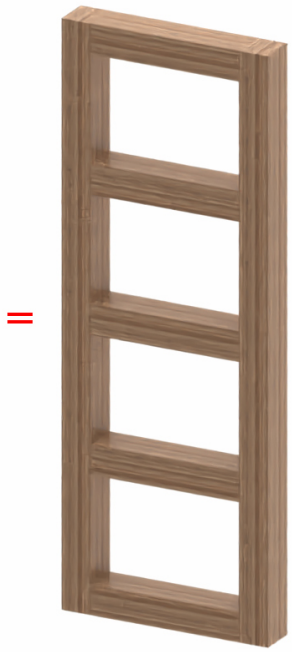


Fig 3-12



Render manager

(RedSDK functionality is available with Only with the RedSDK Plug-in)
Available in TurboCAD Pro, Platinum and Deluxe only

By applying materials to a 3D object, you can obtain a more realistic view of the model during a render. An object's materials can be seen in **Quality and Advanced** render mode, and in certain types of **Draft** rendering. Several materials and categories are provided by default. You can create a new material via the **Render Manager**

A quick note about the Differences between LightWorks and RedSDK materials

Whilst the representation, storage structures and all other elements are identical for LightWorks and RedSDK Luminance, Environments and Render Styles. There are a number of differences, which are described in more detail below: (taken from the TC help file)

- All information about LightWorks materials is stored in a special text file. Before the rendering process initiatea special code appends the LightWorks material definition into this text file. All information about the RedSDKmaterials is stored in *.RED files, and only a few special master properties are available for user to change.

  Therefore, RedSDK material stored in TurboCAD, are held as small text file, which contains the file path to the corresponding RED file. Without *.RED file materials will not work in RedSDK.
- In the case of LightWorks if a shader has a texture, the text representation of this shader contains only the file path to the image file. Only immediately before the rendering process is the image file loaded. In the case of RedSDK the RED file contains not only path to the texture, but the embedded image file itself. the path to the image file may not be valid, but the loaded texture inside RED file is always valid.
- RED files can be stored by the following paths:
  - Path to *propin.dat* (*Luminance.dat*, *sceneenv.dat* or *RenderStyles.dat*) / *RedSDK*
  - Path to *Default.dat / RedSDK (for materials, luminances, environments or render styles)*
  - Near active drawing file

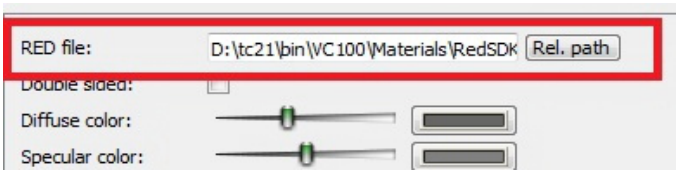
In all these cases a relative path to the RED file will be used. A user can use any other custom RED file, however, In this case, an absolute path will be used.

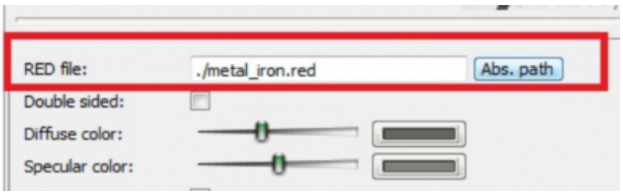
- When a user deletes a material, the text representation of the material also is deleted. In the case of the RedSDK material the *.RED file also is deleted except in cases when the RED file was custom or default (I.e. installed together with TurboCAD). In the case of a custom file you cannot delete the *.RED file because you did not create it.
- A user can create a new RedSDK material in two main ways.

  Create from a custom RED file.

  Create it from a special copy of the default RED file. Therefore, when the first new material is created in the Render Manager a new *.RED file is created as a copy of the default RED file.
- In the case of LightWorks, a text representation of the material is included in the *.tcw file. That is why it is possible to send just a *.tcw file to another user and that user will be able to reproduce a render.
- As mentioned, in the case of RedSDK for the *.RED file in needed for rendering. That is why it is necessary to use the e-pack functionality in the case of exchanging files that depend upon RedSDK.

There is a RedSDK material specific UI control which displays the path to the *.RED. file. Pressing the **Abs. path** button user can see the absolute path which is used for loading of the RED file. By pressing the **Rel. path** button the user can return to the state where the relative path is displayed.

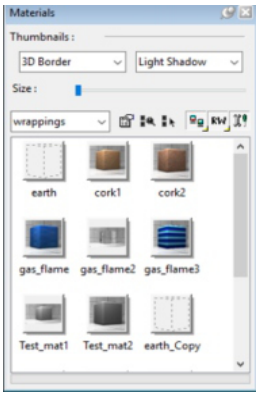


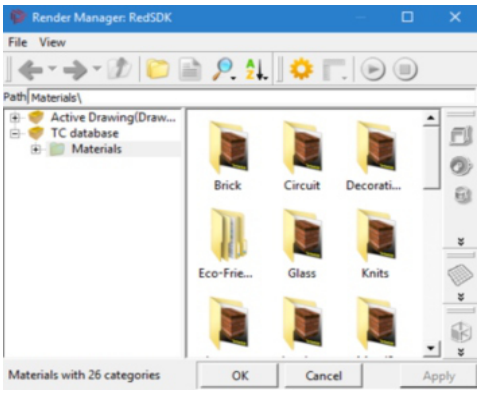


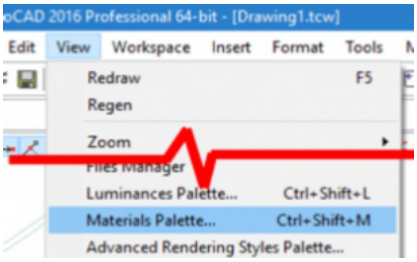
RedSdk Materials.

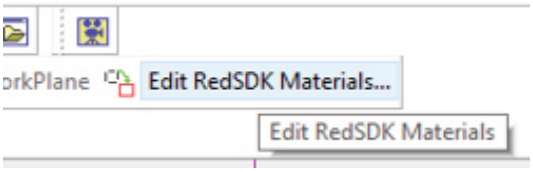
First an important difference between Lightworks and RedSdk, RedSdk is a lot less controllable as far as individual aspects of a material are concerned, using wood floor as an example, with Lightworks whilst it wasn't perfect, one could alter the board length, width, colour, gaps etc, with RedSdk its a fixed jpg image, no adjustment can be done in the program apart from scale, which when scaled quite large looks awful, therefore it is important to prepare the texture image in a paint program if non standard picture of size is required.

Materials can be accessed via a number of ways, the access path can change due to which workspace one has loaded, (in options - toolbars & Menus - options tab), and can also vary from TC versions,

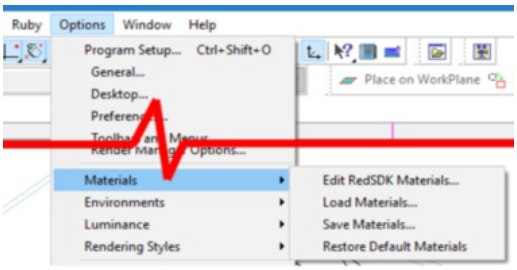




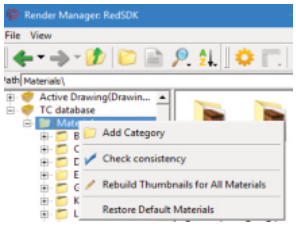




Materials palette



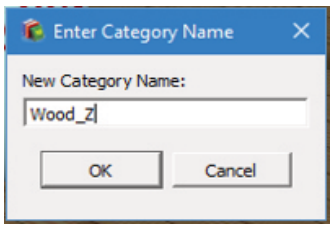
Using existing materials, this is the same as using Lightworks, and there are
a few ways to access them, just a few are listed here.
1). Open the material palette and select a material to use, drag it onto the object or onto a facet (with drag on facet ticked).
2). Right click the object and choose properties - Materials, and use the drop
down boxes to select the required material.
3). Select the object, Use the selection palette to expand 3D / materials select the required material.



Creating New RedSdk Materials,

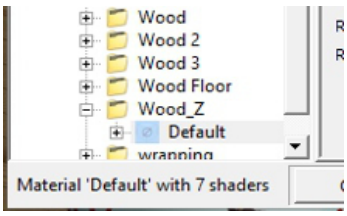
Unlike Lightworks from the past, RedSdk materials cannot be created from the materials property page, to create a new material one uses the Render Manager, accessed from the options menu or as I do from a custom toolbar

The first thing to decide is whether to use an existing category or create a new one, in this example we will create a new one as editing after that is the same as using existing,

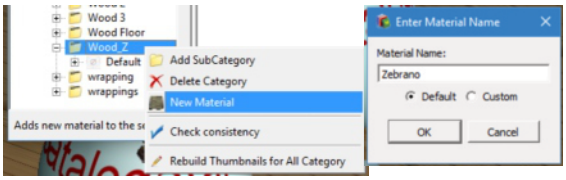


A Category is simply a folder which houses materials of a similar type for easier navigation, To create a category, expand the categories by clicking the '+' next to materials, Right click on the word Materials and in the dialog choose 'Add Category'.

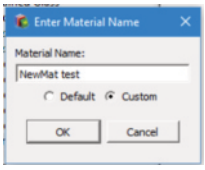
in the next box Type in your desired unique name and click OK, this will insert the category into the list and will normally open it at a default material. From here we can start to add our own material.



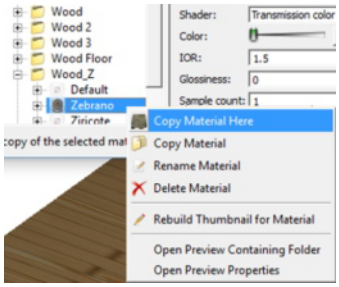
There are at least a couple of ways to add new materials.

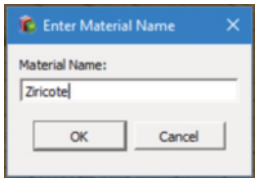


The first way to add a default new material to a category (the new category in this case), right click the category name and in the dialog select new material, a box for the name will appear, type in a name, leave the radio button set at default, clicking OK will add the new material.



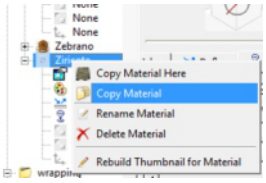
The other radio button in the New Material Name box is at present of little use to us TC users, selecting the custom radio button and clicking OK brings up the normal Windows find fil dialog, its purpose is to create a material from an existing .RED file, but at present few people will have access to new .red files.



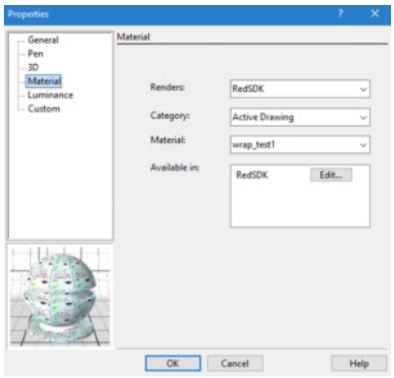


The next way is to copy an existing material which will pass all the settings from one material to a new one, in this instance we right click
on material name not the category, in the drop down box select Copy
Material Here.

In the dialog box, type in the desired name and click OK, this will add
the new material and pre-fill in the settings, however its highly unlike
it you would want exactly the same setting, so they can be customised as described later.



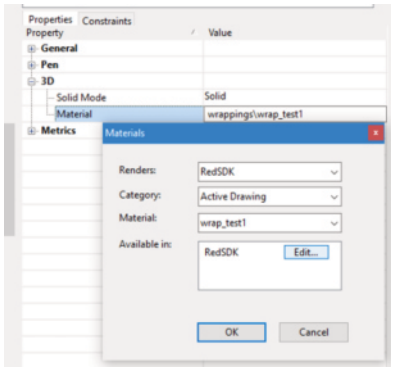
The other copy mode is simply copy material, this will copy a material to the computer windows clipboard, if one then right clicks another category, a paste option will be available, this is useful if a material is in the wrong category or simple cleaning up, but shouldn't be normally necessary as one can chose the right place from the start, one useful function of it is if one receives files from others and wish to organise the materials to your liking..



Editing materials

There are various ways to edit RedSdk materials which changed in 2016, global materials should be altered in the Render Manager accessed from the options menu (or custom toolbar), the reason for this is that when applying materials they are now copied into what is called an 'Active Drawing' category, these are edited independently of the main material and the settings saved in the drawing, there are options to get round this which will be touched on later.

One method of editing is the Render manager discussed in 'Creating a Material' above, or, via the objects property dialog, either right click the object select Materials in the dialog locate the desired material in the drop down boxes and click the 'Edit' button, or via the Selection Palette.

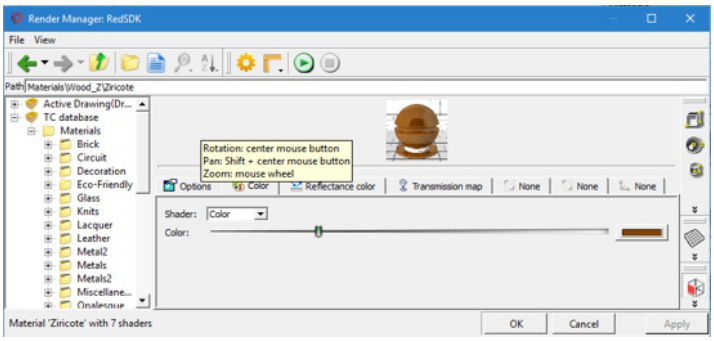


Whichever method is used one is presented with seven selections (as from 2015 only -
earlier versions had 5 options),

For this example we will enter using the Options - Materials - edit RedSdk Material

In the dialog the there are three main areas (not counting the toolbars), these are a preview image, a navigation area, and the Edit area, note - the actual layout will vary.

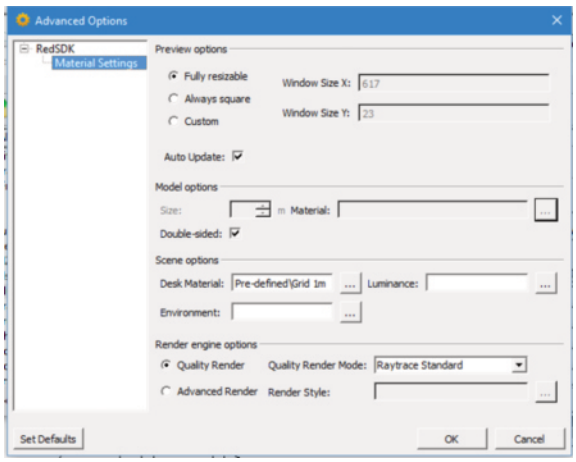
As can be see in the image below in the navigation area are the materials, and a part called 'Active Drawing', Previously it was mentioned about 'Copy Materials' which copies to clipboard, if one copies a material and pastes it into the Active Drawing, it will then be editable via the objects properties and be independent of the original material.



At the top are a number of buttons, Hovering over will reveal their use, the two most important ones are -

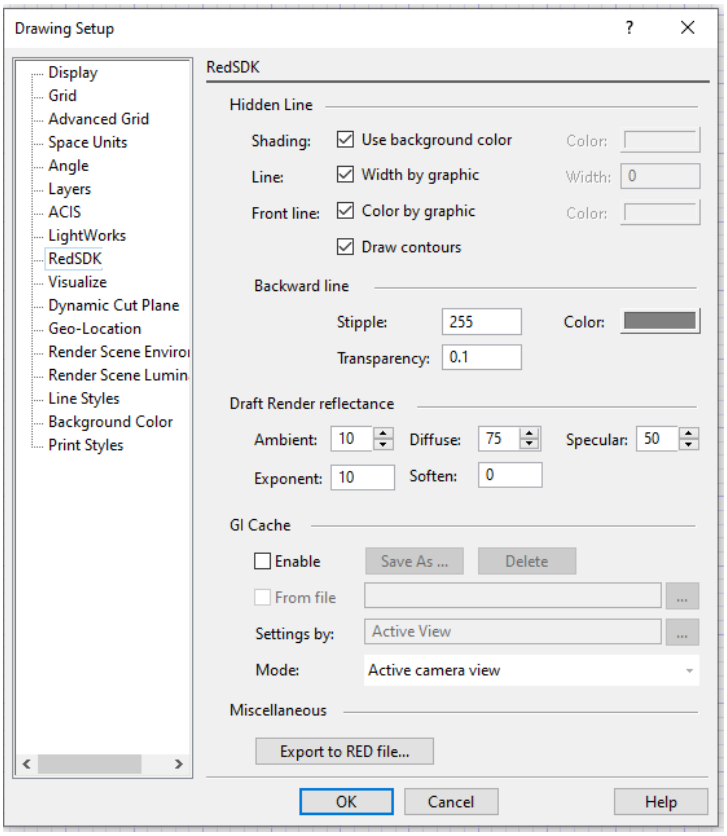
Tree View - if this is turned off the left navigation pane hides leaving either thumbnails or details, one can navigate using the toolbar arrows and by double clicking a category or material to get further options.

Options - This dialog lets one customise how the preview will appear, and I would suggest taking a look at it, as if one has no environment nor luminance's in the drawing it can give a false appearance of how the material will look, the settings I use are shown in the picture.



The environment and luminance drop down boxes have been changed to none, these settings are purely personal, every one will have their own favourite settings.

Redsdk drawing setup



The Redsdk settings allow you to control how object appear when using the Redsdk for draft rendering and hidden line rendering.

Shading

Use background color: When it is turned off (On by default), the option for editing internal color of objects will become available (by default the color is white).

Front line

Width by graphic: When this option is on the Width of the visible part of graphic's line is taken from the graphic itself. When it is turned off, the field for editing the width of the visible part of object lines will become available (by default the width is = 0).

Color by graphic:

When this option is on the color of the non-hidden part of an object's lines is taken from the graphic itself. When the option is
off the option for editing the color of the non-hidden part of an object's lines will become available (by default the color is white).

Backward line

Width by graphic: When on the Width of the hidden part of an object's line is taken from the graphic itself. When it is turned off, the field for editing the width of the hidden part of object lines will become available (by default the width is = 0).

Note: the width of hidden lines must be less than or equal to the width of visible lines.

Color: *This option defines the color of hidden lines (by default it is grey; R=G=B=128*

Transparency: *This option defines the transparency of hidden lines, with the range of 0 to 1 (by default it is 0.1)*

Stipple: *This option defines the type of dashed line used for hidden lines, with the range of 0 to 255 (by default it is 255)*

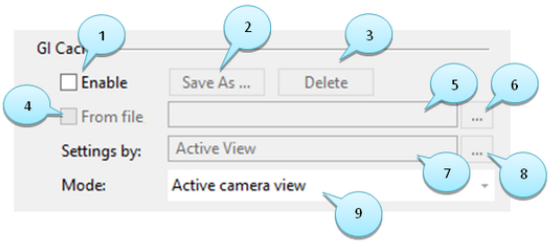
Draft Mode Settings

Ambient: *This field defines the light intensity factor for all light sources of Ambient type (by default it is 10).*
Diffuse: *This field defines the luminance factor (by default it is 75).*
Specular: *This field defines the glare factor (by default it is 50).*
Exponent: *This field defines the glare attenuation parameter (by default it is 10).*
Soften: *This field defines glare 'softness' (by default it is 0).*

GI Cache - Baking

By baking, we mean pre-calculating a given information, storing it and restituting it, faster, as a later rendering phase. Two types of data will be baked in our example:

The global illumination information of our scene.
The direct contribution of physical lights in the scene.
The baking process for this model takes between a few seconds up to a few minutes, depending on the quality of the baked signal. Then, the replay of the baked cache is either performed in real-time or in software.



*1. Enable - Turns on/off GI cache. By default – **Off**.*

2. Save As - Saves the GI cache to a RED file. Enabled – if GI cache was created or loaded.

3. Delete - Clears the GI cache. Enabled – if GI cache was created or loaded.

*4. From file - Turn on/off GI cache loading from file. By default – **Off** Enabled – if GI cache is **On***

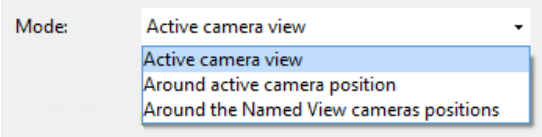
*5. File name - RED file name. By default – **Empty .** Enabled – if GI cache is **On** and From file is **On***

*6. … (Browsing RED file) - Browsing RED file. Enabled – if GI cache is **On** and From file is **On***

*7. Active View or Render Style name - Source of parameters for GI cache creating. By default – **Active View.** Enabled – if GI cache is **On** and From file is **Off***

*8. … (Browsing render style) Browsing render style. Enabled – if GI cache is **On** and From file is **Off***

*9. GI cache computation mode - Choosing GI cache computation mode. By default – **Active Camera View.** Enabled – if GI cache is **On** and From file is **Off***



Active Camera view The GI cache is computed for the given camera only.

Around active camera position The GI cache is computed for the whole set of directions around the camera position. The generated GI cache is then suitable to render panoramas.

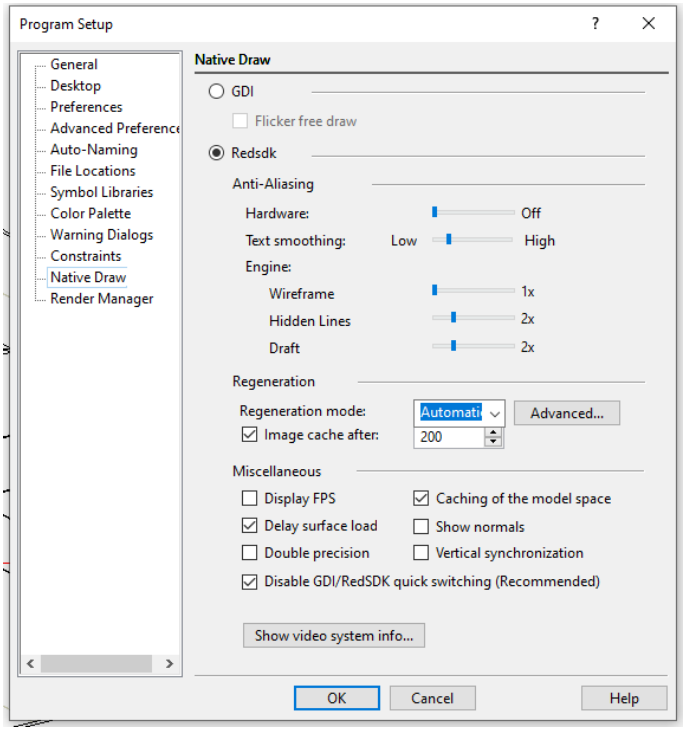
Around the Named View camera positions
The GI cache is computed for the whole scene, without taking the camera into account.
Miscellaneous

***Export to RED file:** This option allows you to save the current RedSDK settings.*
RedSDK Highlighting

Some 3D tools use RedWay Hilighting when working (e.g. Quick Pull, Thread3D, Copy Mirror with option By Face, and other tools which work with face selection)

RedWay Highlighting of PartTree nodes based on selection from Selection Info palette

Native Draw Redsdk Options



Hardware: Creates an anti-aliasing effect on 2D graphic elements. The values available are: Off, 1x, 2x, 4x, 8x, 16x. Turning this on will make the display smoother, but lowers the performance.

Vertical Synchronisation: Makes the display looking better, while moving, however it may higher memory usage.

Text Smoothing: Turning this on will make the display of text smoother, but lowers the performance.

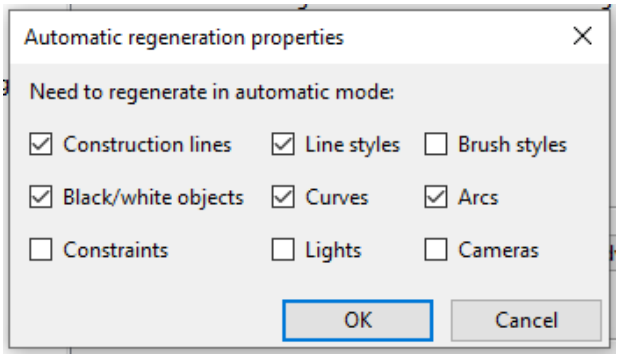
Engine:

Wireframe - Sets the engine anti-aliasing in Wireframe mode. Range – [1-5]. By default – 1. Enabled if Redsdk is On.

Hidden Lines - Sets the engine anti-aliasing in Hidden Line mode. Range – [1-5]. By default – 2. Enabled – if Redsdk is On.

Draft - Sets the engine anti-aliasing in Draft Render mode. Range – [1-5]. By default – 2.

Show Video System Info: Pressing this button will display the current video card setup on your computer. If your current driver for the video card is not the optimal driver for Redsdk you will be prompted to download the correct driver via a link within the window. If you do not download the optimal Redsdk will use a default driver.



Regeneration

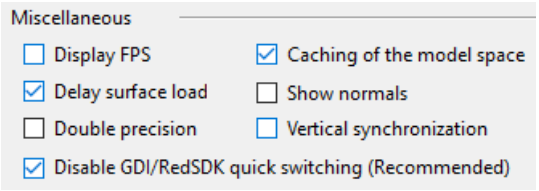
To optimise performance Redsdk will delay regeneration of some drawn elements during movement such as panning or zooming.

Regeneration Mode: You can specify that regeneration be Manual, or automatic. If set to manual all delayed elements will not be regeneration until you press F5 or select Regen or Redraw from the menu. The Advanced settings for Regeneration allow you to specify which items are optimizsed and which are not.

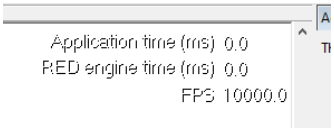
Image cache after: Specifies the size (in MB) after which rendered images will be cached.

Advanced: Settings for **Regeneration** allow you to specify which items are optimised and which are not.

Miscellaneous



Display FPS: If this option is on a display appears in the upper right corner of the drawing space. This display track the Frame Per Second to indicate the speed of the Redsdk drawing engine.

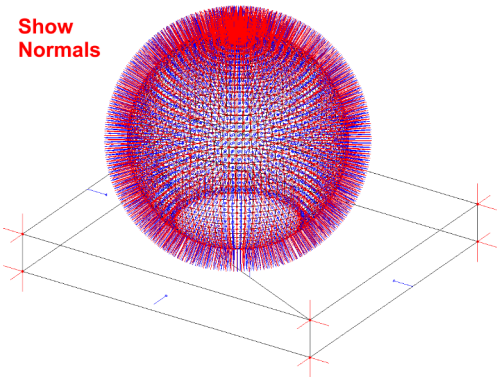


Delay Surface Load: Delays the loading of surfaces and objects until rendering to optimise memory usage.

Caching of Model Space: Store the contents of the model space in cache to optimise presentation.

Disable GDI/RedSDK quick switching (recommended): When checked, the program will be forced to restart when switching between **RedSDK** and **GDI** (either direction). This is the recommended setting. If unchecked, you can switch without restarting, but minor display problems, etc. have occurred when doing so.

Show Normals: When on this option causes Normals to be displayed on all 3D and Surface objects. Requires degenerative faceting turned off.

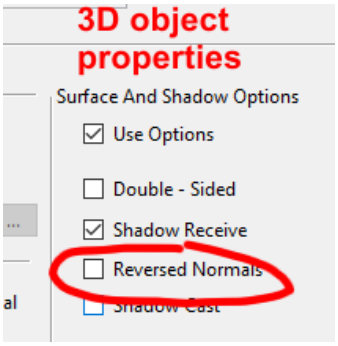


Note: Redsdk is dependant upon video cards which support OpenGL acceleration, and the features of that card. If you do not have an OpenGL video card you will not be able to use Redsdk. Switching between GDI mode and Redsdk will require a restart of TurboCAD.

Note: Redsdk conflicts with GDI's native support for OpenGL hardware acceleration If you switch to Redsdk you will be prompted to allow TurboCAD to turn native OpenGL hardware acceleration off. With Redsdk turned on the Camera Properties "Renderable objects" is always disabled, "Suppress Hidden Line" is always enabled for all pages except the Wireframe page in Camera Properties Dialog.

The Show Normals setting is found under Program Setup/Native Draw. It is Of' by default. The visualization of the normals make it easier to estimate the smoothness and orientation of the 3D objects, and to find artifacts/defects on those objects.

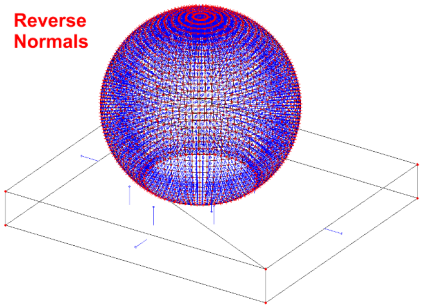
When this option is On the normals are drawn in red colur at the vertices of the 3D objects.



Normals are drawn in blue at the center of facets and show the direction each facet's is facing. The direction of each normal is determined by the path that the edges of each facet take as they wind around the face. The calculation uses the right-hand rule to determine the normal.

The size of the normals is approximately 1/10 the size of the surface bounding sphere's radius (the smallest sphere that will enclose the object). In other words the size of the normal is directly proportional to the size of the object.

The direction of the normals take into account the option **Reversed Normals** which is found in the 3D properties dialog of each object. The base polygon of the normals are display takes into account the option **Double Sided** which is found in the 3D properties dialog of each object. Double Sided objects show a double polygon.



Vertical syncronisation

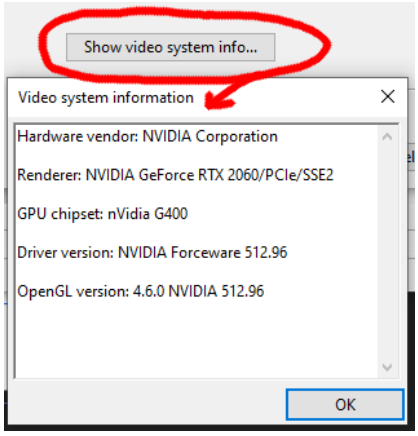
Turns on / off Vsync.

Makes smoother drawing motion, but requires more memory usage. May not work properly with some graphics cards.

See vertical synchronisation on the internet for further information.

Show video system information

Displays a small info box, with the current graphic card being used by RedSDK



The end.