

## Materials

---

### General

The size of texture simply means in relation to other textures of the material. 100% means that it is max size (which is normal whatever size diffuse) has, while 50% means half of the max, size and so on. This information just general outlines though. It is nothing that needs to be followed, but is good for optimization needs.

File format is also just a guideline, although some formats like 3DC (for luminance-alpha), are almost requirements.

The suffix means how the file name should be. The base for the name should always be the name of the mat file if possible (unless shared). Then prefix means: `basename_[prefix].ext`, for example: `"mytexture_nrm.dds"`.

---

### Mipmaps

All textures that can (all textures in the dds format basically) must have pregenerated mipmaps in them. In the exporter tool (like [Nvidia Texture Tools](#)) it should be possible to do this. Having mipmaps is crucial as it improves the graphical quality and makes loading faster. Also, for all the compressed formats (which means pretty much all that are used) it is not possible generate mipmaps in the application, making it even more important that the created dds textures have mipmaps in them!

The filter used to generate the mipmaps is also important. Normally "box" filter is good enough, but when there are many smaller details, these can easily become too blurred or strange aliasing patterns might arise. It is then better to use something like "bessel", "cubic", "catrom" or similar. In the texture export tools there should be a list of these to choose from under mipmap generation. Best is to zoom out so it looks OK, and only if there are any strangeness do you have to think closer about what filter is being used. As said normally it is not a problem.

A list of all filters along with images of resizing can be found [here](#):

### Optimizing

The engine works by having many different hard-coded material types, that are changed by setting different textures and changing variables. Depending on this, the engine will generate a fitting shader to do the job. Because of this, artists do not have to worry about optimizing their materials as much as if another system would be used. However, there are still optimizations needed. Here is a few basic guidelines:

- Never add a texture unless it adds to the final material. For example, do not use black specular, flat blue normal maps, and so on. It is almost always better to just not set the texture at all.
- For textures that only use alpha, it is a good idea to embed it in the alpha of a texture. That way a single texture can fill up two texture slots. However, take note that if a texture with alpha is

used in many materials and/or can have a lower resolution than the one it is part of, it might be good to keep it separate.

---

## Solid diffuse

### Overview

This is the most common material used and is basically an all-round solid (as in non-transparent) surface shader. It works nicely for metals, plastics, wood, organic (that do not require anything special), etc.

Important Notes:

- A material having alpha and parallax on a mesh that where IsOccluder is true, will NOT be displayed correctly.
- An mesh MUST be set with IsOccluder=true for the dissolve effect to work!

### Blend Modes

Not used.

### Variables

**HeightMapScale** | This controls the amount how much the max depth (pure white) will offset the pixel.

**HeightMapBias** | Currently not used.

**FresnelBias** | Bias for Fresnel term. values: 0-1. Higher means that more of reflection is seen when looking straight at object.

**FresnelPow** | The higher the 'sharper' the reflection is, meaning that it is only clearly seen at sharp angles.

**AlphaDissolveFilter** | Currently not used.

**DetailUvMul** | The amount the UV maps are multiplied with. The higher the more the detail maps are repeated

**DetailWeight\_Diffuse** | Weight for the diffuse detail. Valid values: 0 - 2.0

**DetailWeight\_Specular** | Weight for the specular detail. Valid values: 0 - 2.0

**DetailWeight\_Normal** | Weight for the normalmap detail. Valid values: 0 - 2.0 (inf really, but not recommended)

**DetailFadeStart** | When the details start fading out **DetailFadeEnd** | When details have completely faded out

**SwayActive** | If the model should sway according to wind properties. NOTE: Heightmap must be used to set the sway amount for each model part.

**SwayForceFieldAffected** | If the model is affected by force fields

**SwayFreq** | The frequency of the sway movement.

**SwayAmplitude** | How much the sway offsets the model at maximum (in meters).

**SwaySpeed** | The speed of the sway

**SwayOctaveMuls** | The sway consists of the three (high - low) different freq movements, this is the factor for each.

**SwayYFreqMul** | This can be used so that sway changes along the y-axis. The higher value the more change. 0=off

**SwayForceFieldMul** | Factor for the force field induced sway.

**SwayForceFieldMax** | The max offset (after adding factor) can that be added from a force field.

**IlluminationBrightness** | The brightness of the illumination texture. Multiplies with the illumination map, used to make the illumination brighter than 1.0

**LiquidTrickleColor** | The color and alpha of the liquid

**LiquidTrickleSpecular** | Specular color and power of the liquid (RGB, Power). If set to (0,0,0,0) the original specular map value will be used

**LiquidTrickleLoopFade** | If the liquid animation should loop and fade in again after it has faded out

**LiquidTrickleFadeSpeed** | How fast the liquid fades in and then stays before fading out again. Unit is in seconds (Fade In, Fade Out)

**LiquidTrickleEdgeSize** | The size of the normal map edge generated by the trickling liquid

**LiquidTrickleDryness** | How long the generated liquid normal map should stay before drying back to the original normal map

**LiquidTrickleBlendMode** | Which blend mode that should be used when combining the liquid with the color. Specular always uses Alpha blending. With Illumination blend mode the liquid is added to the illumination output

## Diffuse

**Size:** 100%

**Channels:** RGB

**Type:** 2D

**File format:** dds, dxt1 or dxt5. (use dxt1 if it looks okay!)

**Suffix:** None.

This is simply the color and is needed for as good as all materials.

Note that diffuse is a perfect texture for adding some other map (that only use alpha) in the Alpha channel. For example: Alpha, Height and Translucency map.

## Normal map

**Size:** 100% (50% possible for very large and blurry features)

**Channels:** G and Alpha (must use special exporter for this!)

**Type:** 2D

**File format:** dds, 3dc.

**Suffix:** "nrm".

The normal map for the material and uses the standard format  $rgb = xyz$  and is in tangent space.

Note that this needs to be converted to a format of XY only though!

## Specular map

**Size:** 100% - 50% (50% should work in most cases, but 100% might be needed for slimy objects and similar)

**Channels:** RGBA

**Type:** 2D

**File format:** dds, dxt5.

**Suffix:** "spec".

The specular map handles the specular intensity and power for each pixel. The intensity uses the R,G and B channel and the power in the Alpha channel.

Intensity sets the amount of specularity (for each color channel) applied to a pixel, the higher value of the channel, the higher intensity.

Power sets the sharpness of the specular highlight, a low value give a dull and diffuse look, while a high value give a wet/shiny look.

## Alpha

**Size:** 100% - 50% (the smaller version only if alpha features are very large)

**Channels:** Alpha

**Type:** 2D

**File format:** tga, single channel

**Suffix:** "alpha".

Sets the alpha for each texture and for any lit surface this is a simply 0 or 1 decision per pixel since a lit pixel can never be half transparent, only fully visible or invisible.

Note that because of this, it is okay to use the Alpha channel of a DXT1 texture for this!

## Height map

**Size:** 100% - 50% (depends on the amount of small details needed)

**Channels:** Alpha

**Type:** 2D

**File format:** tga, single channel (greyscale).

**Suffix:** "height". ("sway" when used for sway)

Height map is more like a depth map actually. White = largest depth, Black = no depth and this is important to remember when creating it.

This map is also used for sway, so when sway is active, normal parallax does NOT work. During sway it act has a factor for the sway. Black = no sway, White = full sway and all inbetween for greyscale of course. So if doing a swaying plant, make it black at the base of the plant and whiter as it comes nearer the top.

## Illumination map

**Size:** 100% - 50% (only use 100% if very sharp features are required. Most illumination then to be blurry)

**Channels:** RGB

**Type:** 2D

**File format:** dds, dxt1. (dxt5 if alpha is required for extra map)

**Suffix:** "illum".

This map will be added additively on top of the model when rendering and will not be affected by lighting. It is useful for creating glowing surfaces such as lightbulb, ember, etc.

Like diffuse, Illumination is a perfect texture for adding some other map (that only use alpha) in the Alpha channel. For example: Alpha, Height and Translucency map. Especially so if the size of the other map is 50% or so (as this is normally what the illumination map is).

## Dissolve Alpha Map

**Size:** 100% - 50%

**Channels:** Alpha

**Type:** 2D

**File format:** tga, single channel (greyscale).

**Suffix:** "diss\_alpha".

When dissolving a mesh, this map sets how fast each part of the mesh does this. White = fastest, Black = slowest

Because of the way rendering works, it is almost always best to have this as separate texture (and as alpha inside another)

## Environment Map (CubeMap)

**Size:** NA, cubemaps take a lot of memory, so try and keep them small. Usually they are blurry and can be kept very small.

**Channels:** RGB

**Type:** CubeMap

**File format:** dds, dxt1 or dxt5 cube (as the map is usually blurry and distorted, dx1 should work fine in most cases)

**Suffix:** "env".

This will apply a cube to the mesh as if it is was reflecting it (hence called "environment map").

## Environment Alpha Map (CubeMapAlpha)

**Size:** 100% - 50%

**Channels:** Alpha

**Type:** 2D

**File format:** tga, single channel (greyscale).

**Suffix:** "env\_alpha".

Determines how much of each pixel on mesh should reflect. White=max reflection, Black = no reflection.

## Detail Diffuse Map

**Size:** NA, this is entirely up to how the texture is repeated

**Channels:** Alpha

**Type:** 2D**File format:** tga, single channel (greyscale). (or dxt5 if mipmaps are needed!!)**Suffix:** "detail\_diff".

A detail map is special in that it uses different UV coordinates from the rest of the textures (see variables). This map blends with the diffuse and specular (each according to weights, see variables section). The way it mixes, Grey (127) means that nothing is changed, darker (126 or below) makes the result darker, and brighter (128 or above) makes the result brighter).

There is a variable for fading this out at a distance, but in some special cases this might not look good. In that case use dxt5 and have custom mipmaps that fade to gray in an appropriate way. Only use this if really needed though!

**Detail Normal Map****Size:** NA, this is entirely up to how the texture is repeated**Channels:** G and Alpha (must use special exporter for this!)**Type:** 2D**File format:** dds, 3dc.**Suffix:** "detail\_nrm".

Detail normal works basically like the diffuse version, but on the normal map.

**Translucency Map****Size:** 100% - 50%**Channels:** Alpha**Type:** 2D**File format:** tga, single channel (greyscale).**Suffix:** "trans".

This allows part of the texture to shine through to the backside when lit. This is good for thin organic shapes like nose-wings, leaves and fish-fins. Note that for this to work, the mesh must not cast a shadow or it must be thin enough (how thin depends on shadow settings). White means that all light from the front comes out back and black means that no light does.

**Liquid Map****Size:** 100% - 25%**Channels:** G and Alpha (must use special exporter for this!)**Type:** 2D**File format:** dds, 3dc, tga for high quality.**Suffix:** "liquid".

This texture shows how liquid should fade in, move and fade out over an object. The red channel contains a mask for how liquid should fade in on the image, the brighter a pixel is the earlier the liquid will cover it. The green channel contains a mask for how the liquid should fade out. If a 3dc compressed texture produces poor results then a RGBA tga should be used with the red channel stored in the alpha channel and the green channel stored in the blue channel. If the liquid is very thin then the mip-map filter should be set to "Point" when exporting the texture.

# Translucent

## Overview

This material is used for any transparent surface, such as glass, some forms of water, etc. It is generally not affected by lighting and when, only mildly so by setting the variable `AffectedByLightLevel` to true (more on that below).

## Blend Modes

All available blend modes work for Translucent.

## Variables

**Refraction** | If the material has refraction (distortion of bg). Uses `NMap` and/or normals of mesh.

**RefractionEdgeCheck** | If true, there is no bleeding with foreground objects, but takes some extra power. This usually makes object look a lot better so should generally be used, but have in mind that removing it when there are several object with refraction, it can speed things up quite a bit.

**RefractionNormals** | If normals should be used when refracting. If no `NMap` is set this is forced true! If this is false (and `NMap` is set), then the material will only read from the `NMap` and use that as distortion. This can be quite nice for certain effects such as animated water, magical portals and the like.

**RefractionScale** | The amount refraction offsets the background.

**FresnelBias** | Bias for Fresnel term. values: 0-1. Higher means that more of reflection is seen when looking straight at the surface.

**FresnelPow** | The higher the 'sharper' the reflection is, meaning that it is only clearly seen at sharp angles.

**RimLightMul** | The amount of rim light based on the reflection. This gives an edge to the object. Values: 0 - inf (although 1.0f should be used for max).

**RimLightPow** | The sharpness of the rim lighting.

**AffectedByLightLevel** | The the material alpha is affected by the light level. This is especially useful for objects that are in an environment where the lightlevel changes.

How this works that for each light that intersect the Bounding box of the mesh, a light level variable is increased (according to color and attenuation). The final light level variable is then used to control the alpha.

**SwayActive** | If the model should sway according to wind properties. NOTE: Heightmap must be used to set the sway amount for each model part.

**SwayForceFieldAffected** | If the model is affected by force fields

**SwayFreq** | The frequency of the sway movement.

**SwayAmplitude** | How much the sway offsets the model at maximum (in meters).

**SwaySpeed** | The speed of the sway

**SwayOctaveMuls** | The sway consists of the three (high - low) different freq movements, this is the

factor for each.

**SwayYFreqMul** | This can be used so that sway changes along the y-axis. The higher value the more change. 0=off

**SwayForceFieldMul** | Factor for the force field induced sway.

**SwayForceFieldMax** | The max offset (after adding factor) can that be added from a force field.

**ScrollingNoiseActive** | If the alpha of the model should be affected by noise moving in the air

**ScrollingNoiseSpeed** | The speed of the scrolling noise

**ScrollingNoiseScale** | The size of the scrolling noise particles

**ScrollingNoiseDetail** | How detailed the scrolling noise is. The higher the value the more the alpha gets affected

**AngleFadeActive** | If the triangles should fade out at steep viewing angles

**SoftParticleActive** | Removes hard edges where the particle intersects geometry

**SoftPartSphereBillboard** | If the billboard should be thicker in the middle and thinner near the edges (Only works for billboards)

**SoftPartThickness** | The perceived thickness of the particle, measured in meters

**SoftPartAlphaBasedThickness** | The perceived thickness of the particle based on the alpha channel of the particle, measured in meters

**SoftPartDepthBias** | Moves the soft particle closer to or further away from the camera (Negative value only work for point billboards)

**LiquidTrickleColor** | The color and alpha of the liquid

**LiquidTrickleLoopFade** | If the liquid animation should loop and fade in again after it has faded out

**LiquidTrickleFadeSpeed** | How fast the liquid fades in and then stays before fading out again. Unit is in seconds (Fade In, Fade Out)

**LiquidTrickleEdgeSize** | The size of the normal map edge generated by the trickling liquid (Only used with refraction or EnvMap)

**LiquidTrickleDryness** | How long the generated liquid normal map should stay before drying back to the original normal map (Only used with refraction or EnvMap)

## Diffuse Map

**Size:** 100%

**Channels:** RGB

**Type:** 2D

**File format:** dds, dxt1 or dxt5. (use dxt1 if it looks okay!)

**Suffix:** None.

This is simply the color and is needed for as good as all materials.

Note that diffuse is a perfect texture for adding some other map (that only use alpha) in the Alpha channel. For example: Cube Alpha map..

## Normal Map

**Size:** 100% (50% possible for very large and blurry features)

**Channels:** G and Alpha (must use special exporter for this!)

**Type:** 2D

**File format:** dds, 3dc.



**Suffix:** "nrm".

Instead of changing the way light is shaded (like with solid material), here the normal map is used to distort the background (if refraction is set to true) and/or to change the way reflection is calculated. The normal map for the material and uses the standard format  $rgb = xyz$  and is in tangent space. Note that this needs to be converted to a format of  $XY$  only though!

### Environment Map (CubeMap)

**Size:** NA, cubemaps take a lot of memory, so try and keep them small. Usually they are blurry and can be kept very small.

**Channels:** RGB

**Type:** CubeMap

**File format:** dds, dxt1 or dxt5 cube (as the map is usually blurry and distorted, dx1 should work fine in most cases)

**Suffix:** "env".

This will apply a cube to the mesh as if it is was reflecting it (hence called "environment map").

### Environment Alpha Map (CubeMapAlpha)

**Size:** 100% - 50%

**Channels:** Alpha

**Type:** 2D

**File format:** tga, single channel (greyscale).

**Suffix:** "env\_alpha".

Determines how much of each pixel on mesh should reflect. White=max reflection, Black = no reflection.

### Height map

**Size:** 100% - 50% (depends on the amount of small details needed)

**Channels:** Alpha

**Type:** 2D

**File format:** tga, single channel (greyscale).

**Suffix:** "sway" (do NOT use "height").

This map is used for sway, so when sway is active and NOT for any parallax stuff. It act as a factor for the sway. Black = no sway, White = full sway and all in between for greyscale of course. So if doing a swaying plant, make it black at the base of the plant and whiter as it comes nearer the top.

### Liquid Map

**Size:** 100% - 25%

**Channels:** G and Alpha (must use special exporter for this!)

**Type:** 2D

**File format:** dds, 3dc, tga for high quality.

**Suffix:** "liquid".

This texture shows how liquid should fade in, move and fade out over an object. The red channel contains a mask for how liquid should fade in on the image, the brighter a pixel is the earlier the liquid will cover it. The green channel contains a mask for how the liquid should fade out. If a 3dc compressed texture produces poor results then a RGBA tga should be used with the red channel stored in the alpha channel and the green channel stored in the blue channel. If the liquid is very thin then the mip-map filter should be set to "Point" when exporting the texture.

---

---

## Decal

### Overview

Decals work a bit different from other materials in that they never write to the Z-buffer. Instead they simply write to the renderers G-buffer directly and what this means is that they need have some solid geometry behind them to be visible. This make them useful as a sort clutter to add to add extra details. As the resulting image is affected by lighting it can be easily used to add dirt to floor, blood stains, etc.

Important notes:

- The material require Alpha blend mode if normalmaps shall work!

### Blend Modes

All available blend modes work for Decal. However, alpha blending mode is best (if not needed!) to use if anything else but diffuse map is in the material.

### Variables

#### HeightMapScale

The scale of the height map.

#### HeightMapBias

Not used.

#### FrenselBias

Not used.

#### FrenselPow

Not used.

### Diffuse Map

**Size:** 100%

**Channels:** RGB

**Type:** 2D

**File format:** dds, dxt1 or dxt5. (use dxt1 if it looks okay!)

**Suffix:** None.

This is simply the color and is needed for as good as all materials.

Note that diffuse is a perfect texture for adding some other map (that only use alpha) in the Alpha channel. For example: Cube Alpha map..

## Normal Map

**Size:** 100% (50% possible for very large and blurry features)

**Channels:** G and Alpha (must use special exporter for this!)

**Type:** 2D

**File format:** dds, 3dc.

**Suffix:** "nrm".

Instead of changing the way light is shaded (like with solid material), here the normal map is used to distort the background (if refraction is set to true) and/or to change the way reflection is calculated.

The normal map for the material and uses the standard format  $rgb = xyz$  and is in tangent space.

Note that this needs to be converted to a format of XY only though!

## Specular Map

**Size:** 100% - 50% (50% should almost always work)

**Channels:** RGB

**Type:** 2D

**File format:** dds, dxt1 or dxt5

**Suffix:** "spec".

This will only apply the specular color/intensity and NOT the power. Because of this only 3 channels are needed!

## Height Map

**Size:** 100% - 50% (depends on the amount of small details needed)

**Channels:** Alpha

**Type:** 2D

**File format:** tga, single channel (greyscale).

**Suffix:** "height".

Height map is more like a depth map actually. White = largest depth, Black = no depth and this is important to remember when creating it.

## Detail Diffuse

**Size:** 100% - 25% (depends on the amount of small details needed)

**Channels:** RGB

**Type:** 2D

**File format:** DXT1

**Suffix:** "detail\_diff".

A detail map is special in that it uses different UV coordinates from the rest of the textures (see variables). This map blends with the diffuse and specular (each according to weights, see variables section). The way it mixes, Grey (127) means that nothing is changed, darker (126 or below) makes the result darker, and brighter (128 or above) makes the result brighter).

There is a variable for fading this out at a distance, but in some special cases this might not look good. In that case use dxt5 and have custom mipmaps that fade to gray in an appropriate way. Only use this if really needed though!

---

## Water

### Overview

Water is a special shader that not only can render the world reflection, but also contain a special algorithm to simulate ripples.

Important notes:

- When using world reflection, it is extremely demanding, so make sure there is not more than one visible at any time.
- Must be applied to a flat (or at least almost completely flat) and one-sided mesh.

### Variables

#### RefractionScale

The amount reflection and refraction is offset by ripples in water.

#### FresnelBias

Bias for Fresnel term. values: 0-1. Higher means that more of reflection is seen when looking straight at object.

#### FresnelPow

The higher the 'sharper' the reflection is, meaning that it is only clearly seen at sharp angles.

#### WaveSpeed

The speed of the waves.

#### WaveAmplitude

The size of the waves.

#### WaveFreq

The frequency of the waves.

#### ReflectionFadeStart

Where the reflection starts fading.

#### ReflectionFadeEnd

Where the reflection stops fading. 0 or less means no fading.

### **HasReflection**

If a reflection should be shown or not.

### **OcclusionCullWorldReflection**

If occlusion culling should be used on reflection.

### **LargeSurface**

If the water will cover a large surface and will need special sorting when rendering other transparent objects

### **HasVertexWaves“**

If the vertices are animated.

### **VertexWaveSpeed**

The speed of the vertex waves.

### **VertexWaveAmplitude**

The size of the vertex waves.

### **VertexWaveFreq**

The frequency of the vertex waves.

### **FadeWhenShallowMul**

This determines when the water should start fading when in shallow water. In meters.

### **FadeWhenShallowPow**

How fast the shallow water drops off, higher means faster, lower means slower.

## **Blend Modes**

A modulative blend mode is forced.

## **Diffuse Map**

**Size:** 100%

**Channels:** RGB

**Type:** 2D

**File format:** dds, dxt1 or dxt5. (use dxt1 if it looks okay!)

**Suffix:** None.

This is the basic texture that act as a base for the water surface. It is always modulatively blended with the background.

## **Normal Map**

**Size:** 100% (50% possible for very large and blurry features)

**Channels:** G and Alpha (must use special exporter for this!)

**Type:** 2D

**File format:** dds, 3dc.

**Suffix:** "nrm".

Instead of changing the way light is shaded (like with solid material), here the normal map is used to distort the background (if refraction is set to true) and/or to change the way reflection is calculated. The normal map for the material and uses the standard format  $rgb = xyz$  and is in tangent space. Note that this needs to be converted to a format of XY only though!

### Environment Map (CubeMap)

**Size:** NA, cubemaps take a lot of memory, so try and keep them small. Usually they are blurry and can be kept very small.

**Channels:** RGB

**Type:** CubeMap

**File format:** dds, dxt1 or dxt5 cube (as the map is usually blurry and distorted, dx1 should work fine in most cases)

**Suffix:** "env".

This will apply a cube to the mesh as if it is was reflecting it (hence called "environment map"). It will be used instead of the world reflection, thus making the water rendering ALOT faster!

From:

<https://wiki.frictionalgames.com/> - Frictional Game Wiki

Permanent link:

<https://wiki.frictionalgames.com/hpl3/engine/materials?rev=1335270100>

Last update: **2012/04/24 13:21**

