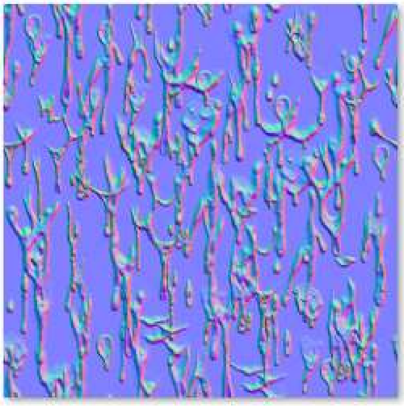
Normal Map Guide

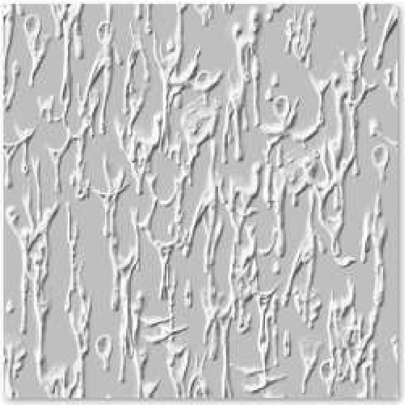
Wogrim's Brief Guide to Normal Maps

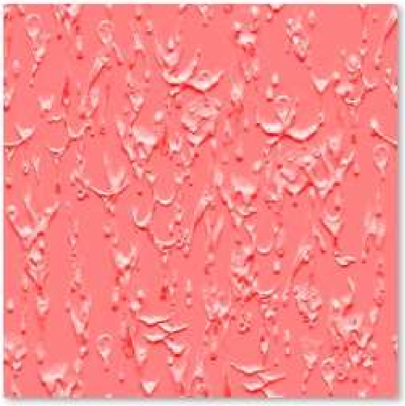
What is a Normal Map?

A Normal Map is a texture that a shader uses to modify a mesh's normals, for doing lighting calculations. This is used to make a surface look like it has more 3D detail than it actually has. This is most noticeable on shiny (specular) surfaces. The color values of a pixel in a Normal Map represent a vector that modifies the surface normal at that location on the mesh.

How Does it Work?







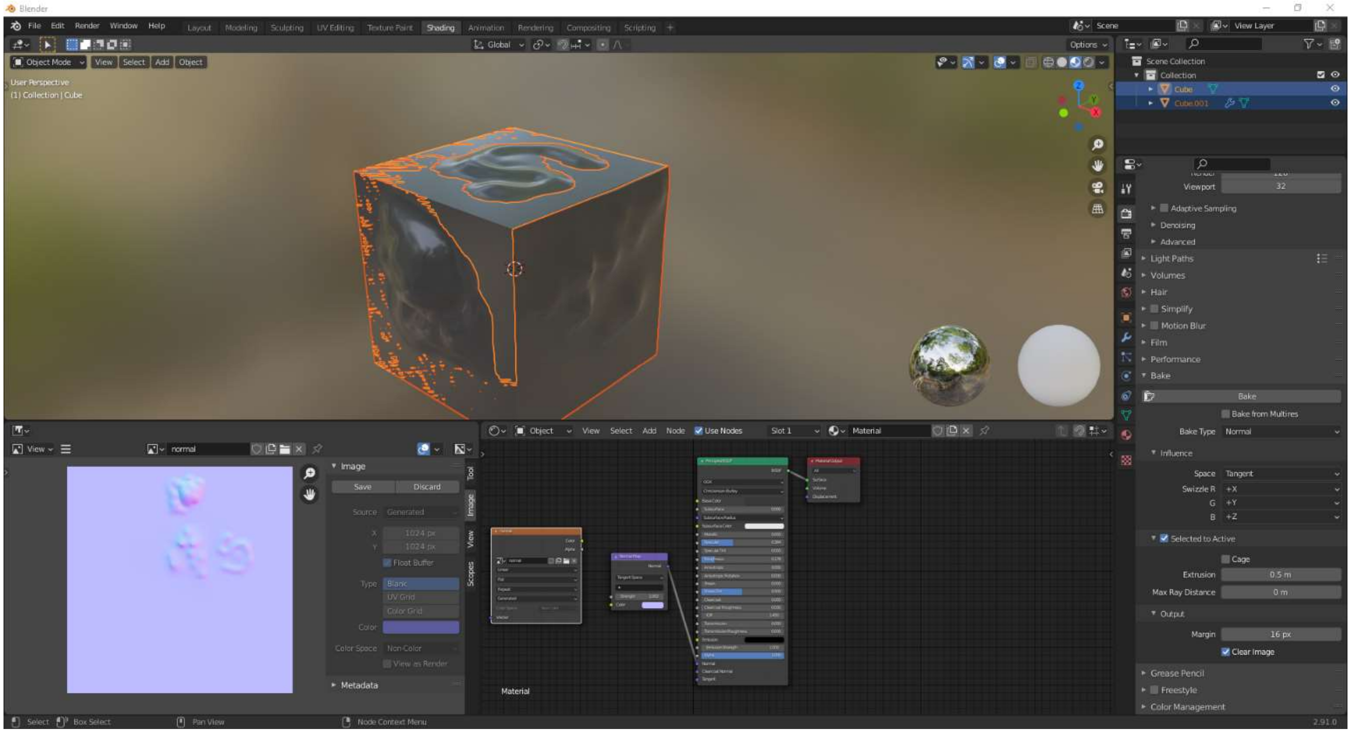
A "Regular" (blue-ish) Normal Map does this with 8-bit colors, with 0 to 255 being a -1 to 1 scale. Red is the X direction, Green is the Y direction, and Blue is the Z direction of what is supposed to be a normalized vector (normalized means the length is 1). Normal maps do not flip a mesh face, so Z should always be >= 0. This means Blue only goes from 127 to 255 so that it becomes 0 to 1, which is why Regular Normal Maps have a lot of blue in them. A "flat" normal map has X=0, Y=0, Z=1, so for a RNM Red = 127, Green = 127, Blue = 255. There is a variation of this format where the green channel is inverted, which in most software is wrong; you want Green = up (+Y).

But if you look at a "KK" (gray) Normal Map, it looks very different. They are all grayscale and use the Alpha channel. It turns out this is not a special format for KK, but rather a conversion that Unity does; Unity's algorithm for shaders to use a Normal Map requires it to be in this format, so when you import an image to Unity and specify it as a Normal Map, Unity secretly converts it. Here is what is different: the Y direction is Red, Green, and Blue. The X direction is in the Alpha channel. Z is not in the image, but because the vector is assumed to be normalized, Z can be calculated with $X^2 + Y^2 + Z^2 = 1$.

Later versions of Unity use a slightly different (red) version of this format, in which the red channel is full across the entire image; this is backwards compatible with the shaders for older Unity. So for forward compatibility, this format is recommended. KK Modding Tools has a script that automatically converts your Normal Maps from gray to red format.

How To Create A Regular Normal Map?

Technically you could hand draw a normal map, but this is not recommended. Normal maps are normally created by 3D modeling software. You have your model that you are using in-game, and a super-high-poly model that has all the gorgeous details. You "bake" the normal map texture which makes the software calculate the differences between the low and high-poly models, and store that as colors in the normal map. Exact steps for doing this depends on your 3D software, but it is quite easy to find YouTube videos for the popular ones. Blender is free and works fine. Here I've baked a normal map from a cube with some sculped details onto a flat cube.



Viewport 32

Image: normal

Source: Generated

X 1024 px

Y 1024 px

☑ Float Buffer

Type: Blank

Color Space: Non-Color

☐ Bake from Multires

Bake Type: Normal

Space: Tangent

Swizzle R: +X

G: +Y

B: +Z

☑ Selected to Active

☐ Cage

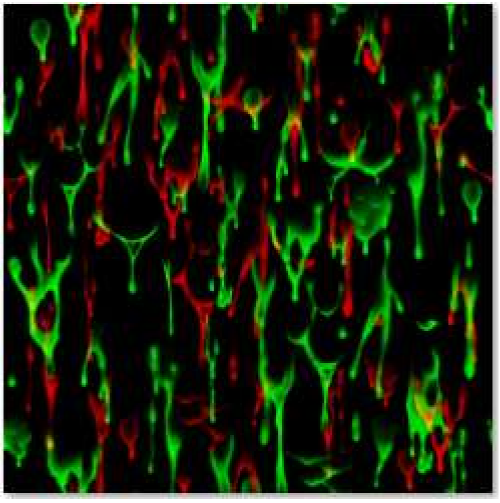
Extrusion: 0.5 m

Max Ray Distance: 0 m

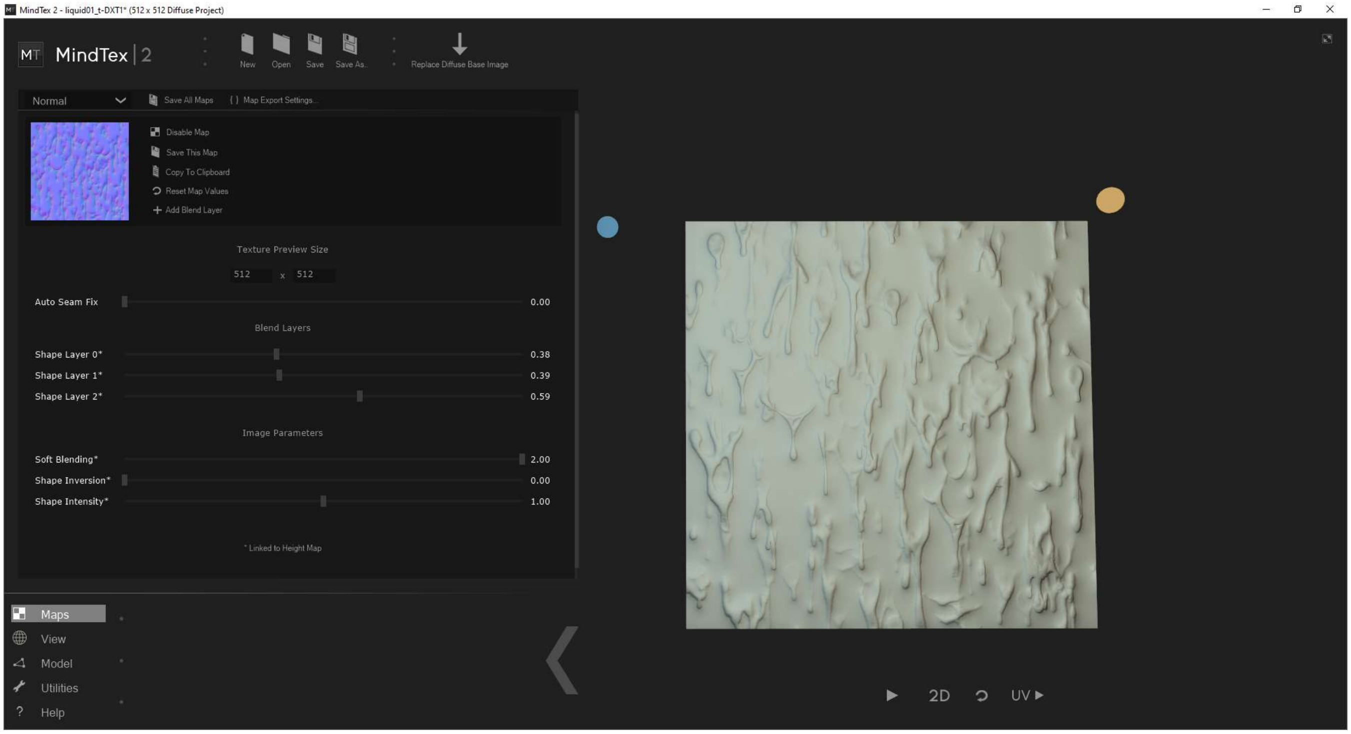
Margin: 16 px

☑ Clear Image

There is also software that generates a normal map from a 2D image (search "normal map generator"). It's mainly good for adding details to a flat surface, because it doesn't know what geometry you're putting it on. Give it an image:



And it'll generate a normal map and show how it will look on a surface. There are settings to tweak for making it look softer or sharper.


MindTex 2 - liquid01_t-DXT1* (512 x 512 Diffuse Project)

Texture Preview Size

512 x 512

Auto Seam Fix 0.00

Blend Layers

Shape Layer 0* 0.38

Shape Layer 1* 0.39

Shape Layer 2* 0.59

Image Parameters

Soft Blending* 2.00

Shape Inversion* 0.00

Shape Intensity* 1.00

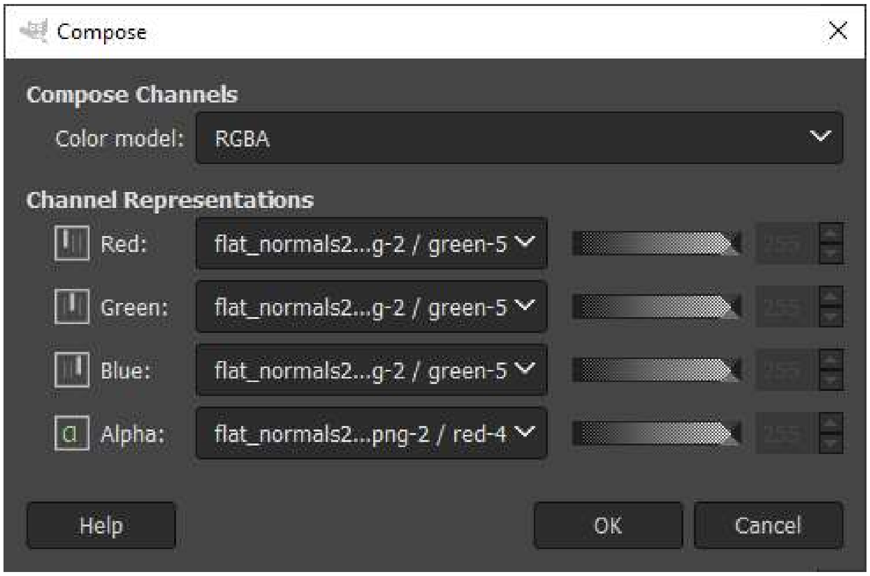
* Linked to Height Map

How To Convert A Regular Normal Map to KK?

Generally when making mods you just let Unity / KK Modding Tools do this for you, but if you're doing SB3U workflow or Material Editor, here is the process.

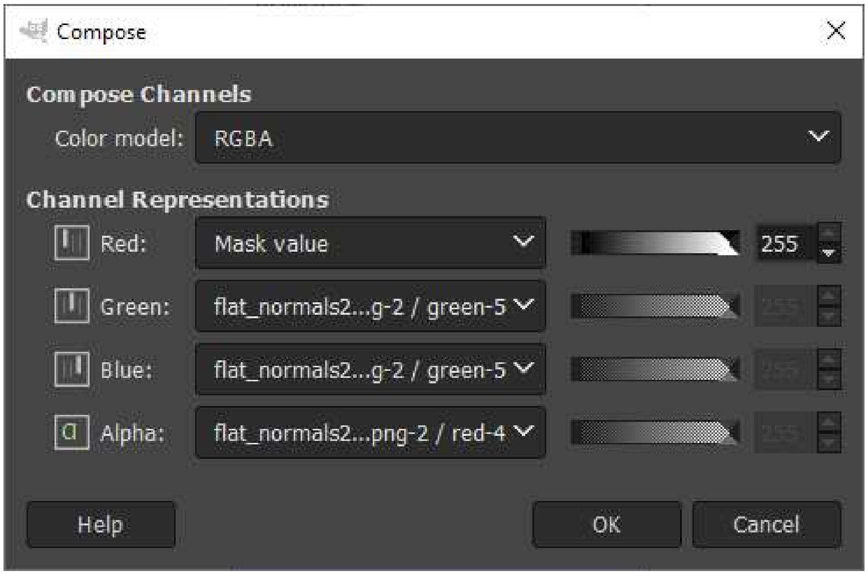
All you have to do is switch colors so that the X and Y are in their proper channels (and Z is not used). This can easily be done in GIMP, which is free. Steps for GIMP:

1. Open your normal map
2. On the menu, go to Colors -> Components -> Decompose
3. Decompose as RGB, to layers
4. On the menu, go to Colors -> Components -> Compose
5. Compose to RGBA, with Red layer in the Alpha slot and Green in the rest



Color model: RGBA
Red: flat_normals2...g-2 / green-5
Green: flat_normals2...g-2 / green-5
Blue: flat_normals2...g-2 / green-5
Alpha: flat_normals2...png-2 / red-4

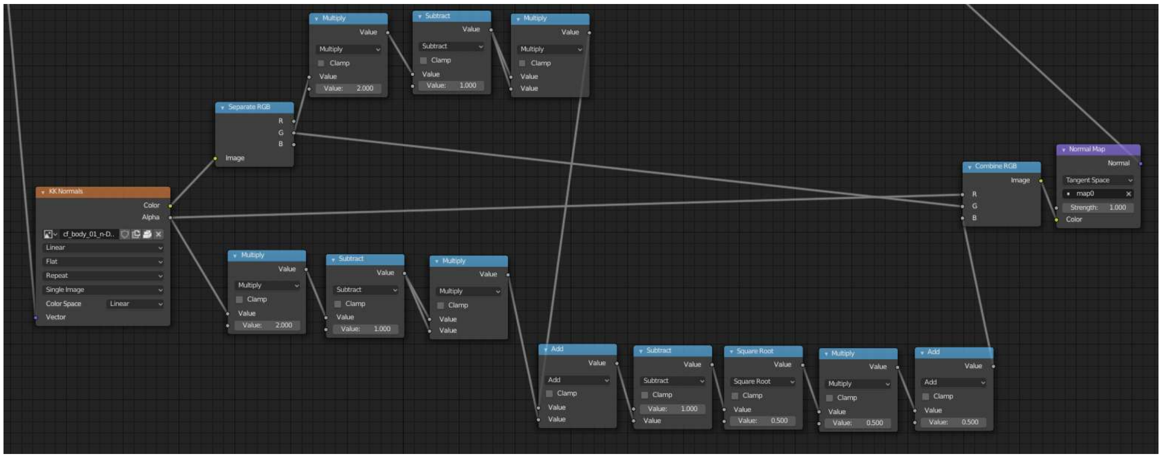
Or if you want the red version, change the Red slot to "Mask value" at 255.



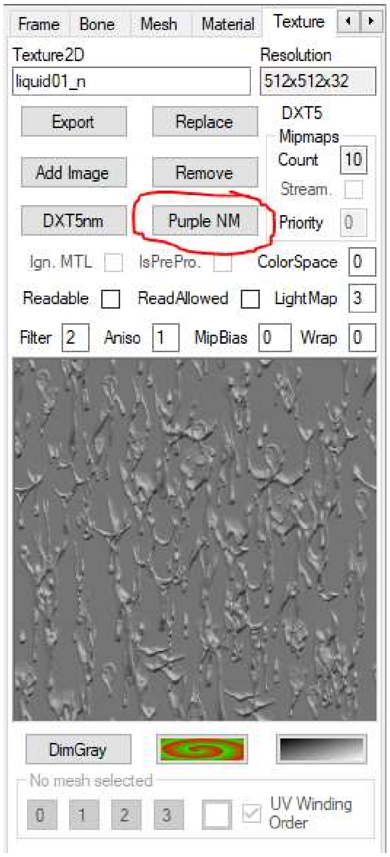
Color model: RGBA
Red: Mask value 255
Green: flat_normals2...g-2 / green-5
Blue: flat_normals2...g-2 / green-5
Alpha: flat_normals2...png-2 / red-4

How To Convert A KK Normal Map to Regular?

You need a program that recalculates the Z portion of the vector because it is missing from the KK Normal Maps. If you just want to see an item in Blender with the KK Normal Map, you can use Blender nodes to get the proper calculation, as seen in this picture (colors go from 0 to 1).



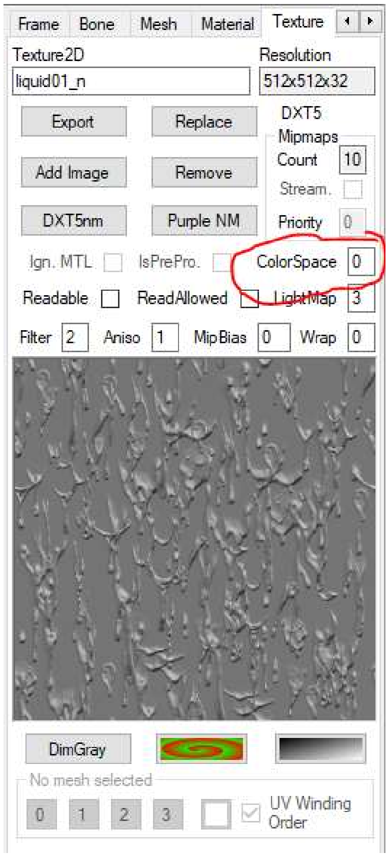
But a better way to do this would be a program that just converts the image. There is probably other software out there that does this (not KK-specific), but recent versions of SB3U come with a handy button ("Purple NM") to convert the format before you export. Only press it once, and make sure to close the AB without saving.



Texture2D liquid01_n
Resolution 512x512x32
DXT5
Mipmaps Count 10
Stream. ☐
Priority 0
Ign. MTL ☐
IsPrePro. ☐
ColorSpace 0
Readable ☐
ReadAllowed ☐
LightMap 3
Filter 2
Aniso 1
MipBias 0
Wrap 0

Other Technical Details?

Normal Maps are usually encoded in linear color space, whereas regular pictures are encoded in gamma space (sometimes called sRGB). See the Wikipedia page for "Gamma Correction" if you want more specifics. Many image editing/displaying software assume gamma space, so be careful. Material Editor lets you export textures, but they come out encoded in gamma space. Material Editor reads this properly if you give it back this way, but it does not read a linear space normal map correctly. Basically, don't use Material Editor for Normal Maps unless you really understand the color space thing and how to fix it. For SB3UGS, if you create a linear Normal Map and import it, the "ColorSpace" should be 0.



Texture2D: liquid01_n
Resolution: 512x512x32
DXT5
Mipmaps Count: 10
Stream. ☐
Priority: 0
Ign. MTL ☐
IsPrePro. ☐
ColorSpace: 0
Readable ☐
ReadAllowed ☐
LightMap: 3
Filter: 2
Aniso: 1
MipBias: 0
Wrap: 0

Troubleshooting

If there is a problem with your normal map, the light will look like it is coming from the wrong direction. This does not affect shadows, so you might not notice it at a glance. I recommend turning on specular to check. If you do find a problem, it might be one of the following:

- You used the wrong normal map format
- There is a color space problem
- There is a problem with the mesh, not your normal map (check without your normal map)