User Guide

PB Texture Baker

Texture (and normal map) baking plug-in for NewTek LIGHTWAVE 3D Modeler.

http://www.blytools.com

Rev. 22 (2018-01-10)

1.0 Overview (aka Quick Start)

1.1 Installation

Copy the plug-in `PB_TextureBaker.p` (win) / `PB_TextureBaker.plugin` (mac) to the LW plug-ins directory, suggested location is `Plugins\model\`. The plug-in should now be found as `PB Texture Baker` under the command category `Additional` in the menu and shortcuts editors.

1.2 Low-Poly / Target Geometry

- Must consist of planar, convex polygons (triangles if you're unsure).
- Must be UV mapped/unwrapped and UV coordinates should be within 0 to 1 (100% in LW), anything outside will be ignored.
- Smoothing is used from LW surface settings, and should be used for smooth surfaces in order to produce good baking results, since the interpolated normals are used for tracing into the high-poly geometry.
- The baked textures will be named after the UV map(s).
- Can use several UV maps.
- Can be spread across several layers.

1.3 High-Poly / Source Geometry

- Must consist of planar, convex polygons (for objects with SubPatches just make a temporary copy to a free layer and freeze it).
- Can be surfaced with (almost) anything the surface editor has to offer, that makes sense in the context of unlit baking. Procedurals, texture maps, UV projections/sources, bump* maps for bumpy detail to enhance the baked normal maps. Surface shaders are currently not supported. LW nodes however are supported. For more information about nodes, see section “Using LW[9]+ Nodes” below.
- Can be spread across several layers.

[*] There unfortunately is a limitation with bump maps using images, they don't work with the baker when they use `UV` as `Projection` source. If you need a `UV` mapped bump map, you have to use the node editor and the `PB Image` node instead (see section “Using LW[9]+ Nodes” below).

1.4 Baking

- Select one or more FG layers containing the low-poly / target geometry.
- Select one or more BG layers containing the corresponding high-poly / source geometry. **OR** Do *not* select any BG layer if the target geometry should be baked onto itself (ie. target and source are the same).
- Invoke the texture baker.
- Select the desired output maps (see section “Maps” below).
- Adjust front and back trace distances to suit the geometry (see section “Trace Distances” below), if any BG layer is selected. These are not used when baking geometry onto itself.
- Chose image file type and output directory where files should be saved.
- Configure each available UV map in “Show settings for UV Map”, disable maps that should not be baked.
- Click OK to bake.
- *For first time use it might be necessary to adjust normal map coordinate axis mappings etc. to get correct results (see section “General Options” below).*

It's possible to select layers from multiple LWOs at once using the Layers Panel in LW. Generally this should work with the texture baker, as long as surface names are unique for each LWO. However if there are surfaces with the same name in the different LWOs, results can be wrong. The solution in such a case would be to either rename the surface(s) or work with a single LWO.

Complex objects, or high settings for occlusion and accessibility maps, may take a while to bake, the progress bar in modeler will display progress. If baking takes more than just a few seconds, the plug-in will display a message box with the total processing time after it's done, this could be useful when leaving the computer but still wanting to know how much time it needed. On the other hand for simpler setups the whole process might be faster than you

can release the mouse button, the plug-in dialog will just seem to close right away, but all baked images should be available (unless an error message was displayed).

1.5 Using LW[9]+ Nodes

Nodes (in LW[9] and later) are supported to more or less the same degree as the parameters in the classic surface editor. That means shaded inputs (including the Material input in LW9.5+) aren't supported, as lit baking isn't supported and would make no sense in Modeler. The Displacement input isn't supported, but may be in future versions. Nodes that do ray tracing should be avoided as they likely won't produce the expected results.

IMPORTANT: Using built-in (or third party) nodes which use UV or Vertex maps do unfortunately *not* work with the texture baker. This mainly affects the nodes `Image` and `Normal Map` when using `UV Map` as mapping method, or any of the `Vertex Map` nodes. As a workaround the baker plug-in provides a set of replacement nodes, which can be found under the `PB_Nodes` category in the node editor. The replacement nodes for `Image` and `Normal Map` only support `UV Map`, for any other mapping method the built-in (or third party) nodes should be used. When trying to use built-in nodes with UV/Vertex maps under LW[2018] it will probably even result in LW crashing when trying to bake.

LW node support may be a bit unpredictable when using more “exotic” or third party nodes, so make sure to save the object before baking, and if it crashes or any baked results look wrong please leave some feedback about it. The replacement texture nodes also work for rendering in Layout like regular nodes, it's however likely that MipMapping calculations aren't identical to built-in nodes as there's no public information available on how LW calculates it.

NOTE: If you have v1.23 or an earlier version installed of the plug-in, you may have to manually add it again through `Add Plugins` in order for the replacement nodes to show up. If you can see the `PB_Nodes` category in the node editor then this isn't necessary.

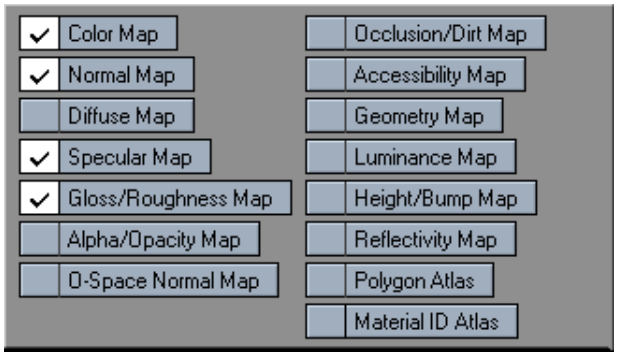
1.6 Notes On LW[2018]

As noted above, using built-in nodes to access UV or Vertex maps in LW[2018] will likely crash. *It's best to save before baking if there are any changes you don't want to lose.*

The baker currently only supports baking of surfaces that use the “Standard” or “Principled BSDF” materials. Complex surface setups with material mixers/switches or other material types are *not* supported.

The “Roughness” value in “Principled BSDF” materials will by default be converted to gloss/smoothness during baking. The conversion simply consists of inverting the values. This is done for consistency, in case a mesh contains both Standard and PBSDF materials. However as of version 1.41, the baker has a general option (see section 3.7 “General Options”) to output gloss as roughness instead. If the option is enabled then all values are baked as roughness, meaning that Standard materials will bake an inverted gloss value.

2.0 Maps

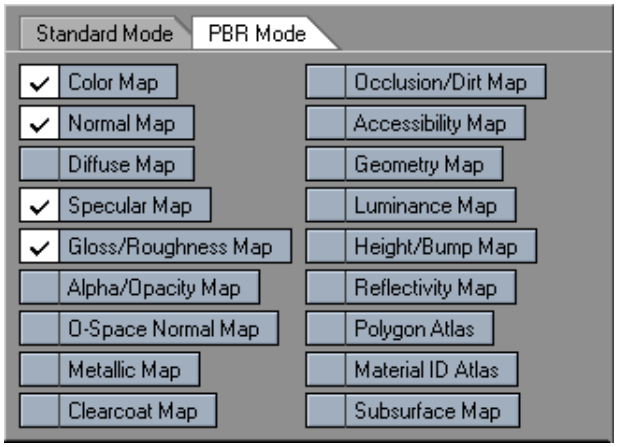


Below is a table of the available output maps, with a brief description of each and the identifier appended to the file name. The default base file name used, is the UV map name. For example if a map called `MyObject` is used to UV map the low-poly geometry, the resulting normal map file would be named `MyObject_n`. Optionally a custom name may be specified per UV map.

Map	Filename suffix	Image type	Description
Color		RGBA	Regular color/texture map (as defined per “Color” in the surface editor). When saved to an image format that supports alpha, the alpha channel contains a mask of the UV mapped areas, not an opacity map (use the separate Alpha/Opacity option for that). The alpha mask is mainly intended as a helper for post processing in an image editor.
Normal	`_n`	RGBA	Tangent space normal map. When saved to an image format that supports alpha, the alpha channel contains a mask of the baked high-poly geometry. Can be useful for post processing, for example as alpha mask for sprites.
Diffuse	`_d`	GREY	Diffuse contribution map (as defined per ”Diffuse” in the surface editor).
Specular	`_s`	RGBA	Specular contribution map (as defined per ”Specularity” in the surface editor). If “Color Highlights” in the surface editor is non-zero, the RGB portion will contain a colored specular map. The alpha component always contains an uncolored specular map, which is identical to RGB if color highlights is zero. In LW[2018]+ the counterpart for “Color Highlights” in “Principled BSDF” materials is “Specular Tint”.
Gloss/Roughness	`_g`	GREY	Gloss map (as defined per ”Glossiness” in the surface editor). Alternatively it can output Roughness instead, which can be enabled in “General Options”. Note that the “Roughness” value in LW[2018]+ “Principled BSDF” materials will be converted to gloss/smoothness, unless the general option to output roughness is enabled.
O-Space Normal	`_on`	RGBA	Object space normal map. Alpha component same as for “Normal” map.
Alpha/Opacity	`_a`	GREY	Alpha map, the inverse of transparency (as defined per “Transparency” in the surface editor). See section “Misc Options” below, for an additional option to automatically mask the alpha with the high-poly geometry mask.
Occlusion/Dirt	`_o`	GREY	Occlusion map, performs additional traces to determine how occluded each point is (aka ambient occlusion). NOTE: Because this map requires quite a few additional traces per pixel, baking times can become significantly longer, however it depends on geometry and output size so don't be afraid to try.
Accessibility	`_acc`	GREY	Accessibility map is sort of the opposite of an occlusion map, it focuses on ”finding” exposed edges. NOTE: The quality of the map varies greatly and may find limited use, as the algorithm is experimental.

Geometry	_geom	GREY	Similar to the geometry layer/channel of the PSD exporter in Layout.
Luminance	_l	GREY *RGBA	Luminance map (as defined per ”Luminosity” in the surface editor). [*] When setting the baker to “PBR Mode” under LW[2018]+, the Luminance map will generate an RGBA map instead, where RGB contains the luminous color and alpha the luminous intensity.
Height/Bump	_h	GREY	Height/Bump/Displacement map, height is based on hit location's percentage of total trace distance (back + front).
Reflectivity	_r	GREY	Reflectivity map (as defined per “Reflection” in the surface editor).
Polygon Atlas	_atlas	GREY	Wireframe polygon map, useful when manually painting textures.
Material ID Atlas	_mtlid	RGB	Material ID map, where each LW surface receives a random ID in the form of a color which is baked out. NOTE: The ID is based on the LW surface name, so the color stays the same as long as the surface name is the same. It is in theory possible that two surfaces can accidentally end up with the same color, in such a case changing the surface name ever so slightly should fix it.

LW[2018]+ has an additional bake mode selector, to choose between “Standard Mode” and “PBR Mode”. In “PBR Mode” it makes some additional maps available, that are relevant to PBR (the “Principled BSDF” material). Aside from the additional maps, “PBR Mode” affects the output format of the “Luminance” map (see table above).



The additional “PBR Mode” maps are listed in the table below.

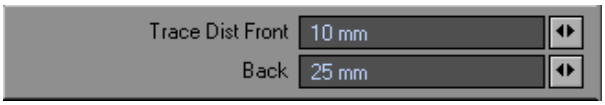
Metallic	_m	GREY	Metallic map (as defined per “Metallic” in the surface editor).
Clearcoat	_cc	RGBA	Clearcoat map. The RGB portion will contain the “Clearcoat” amount as a greyscale value and the alpha channel will contain “Clearcoat Gloss”.
Subsurface	_ss	RGBA	Subsurface map. The RGB portion will contain “Subsurface Color” and the alpha channel the “Subsurface” amount.

3.0 Basic Functionality

3.1 Misc Options

`Sprite Alpha Map`	Applies the high-poly geometry mask (which also can be found in the alpha channel of normal maps) to the “Alpha/Opacity” map. This can be convenient when working with for example sprites/billboards or decals, as it avoids the post processing step of manually applying the geometry mask to the baked alpha map in an image editor.
`No Material ID AA`	Disable anti-aliasing for “Material ID Atlas” maps. Has no effect when “Multi Sample” (see section 3.3) is disabled. Since anti-aliased pixels are blended colors, they no longer represent the original surface ID, which may confuse or cause problems when using the material ID maps in other applications. Enabling this option ensures that only the strict ID colors are used, which on the other hand can cause some jagged edges.
`Use FP Buffers`	Use floating-point image buffers during baking. Aside from the increased precision of FP, it also allows values outside the standard 0 – 1 range (HDR) when saving to an image format that supports it (like for example OpenEXR). *Use with caution as FP uses four times as much memory.*

3.2 Trace Distances



The Back and Front trace distances are crucial for baking one set of geometry onto another. *If there's no BG layer selected, meaning geometry is baked onto itself, the trace distances are disabled as they aren't used.* When baking a surface, the baker will do a ray trace from specified distance in front of the low-poly / target surface to the specified distance behind it. It expects to hit the high-poly / source surface somewhere between the start end end point, otherwise it will assume the pixel is empty. See figures below:

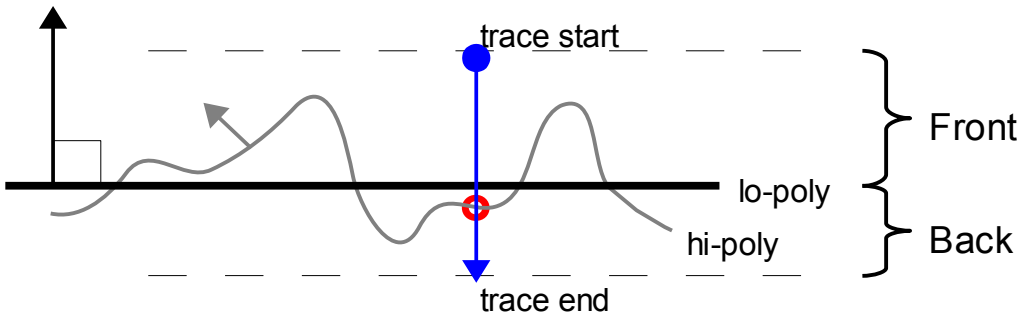


Figure 1: *Front and Back distances cover the entire range of the high-poly geometry with some extra margin.*

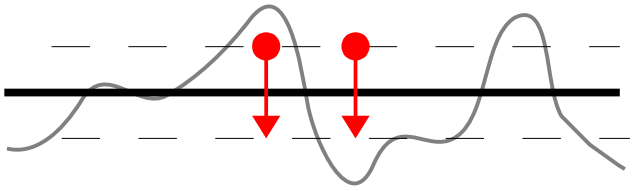
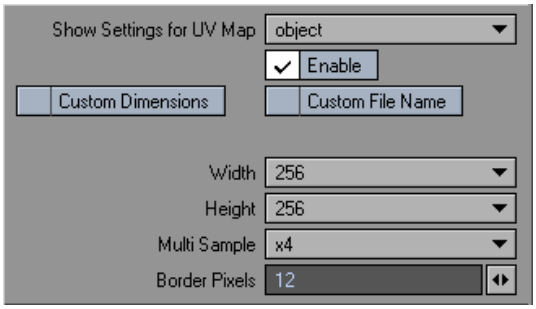


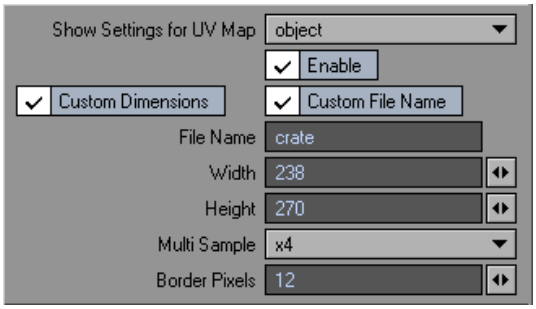
Figure 2: *Front and Back distances are both too narrow, the high-poly geometry gets "clipped" in several places, resulting in traces not hitting anything.*

Another way to describe it is that a thickness is applied to the low-poly geometry, and the high-poly geometry needs to be engulfed by the low-poly shell. The distances aren't meant to be exact, just a rough (over)estimate. Using unnecessary large values may affect performance and can in some cases cause artifacts, if "wrong" parts of the high-poly geometry are hit. Conversely if there are artifacts in "tight" spots, trimming the trace distances to be as narrow as possible around the high-poly geometry may help. If that doesn't help or isn't possible, grouping the problematic geometry or add blocker geometry may help (see chapter "Advanced Functionality" below).

When baking a Height/Bump Map, the total height/displacement range will be the Front + Back trace distance. That means when looking at Figure 1, "trace end" will have a height value of 0% and "trace start" will have 100%.

3.3 UV Map Settings





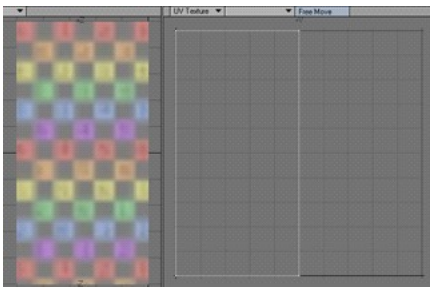
Each UV map in LW will have its own settings in the plug-in, where the map can be enabled/disabled for baking. It's good practice to go through the list, first time an object is processed, and disable all maps that aren't meant to be baked. Enabled maps have following options:

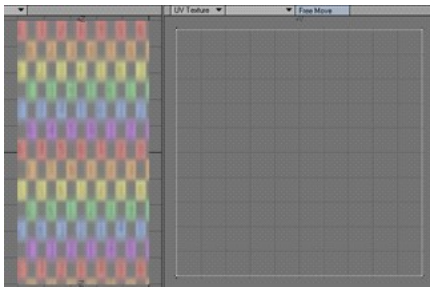
`Custom Dimensions`	Enable custom entry of exported image size, instead of selecting a predefined power of two value.
`Custom File Name`	Enable custom entry of file name for exported images, instead of using the UV map name. The file name may be up to 63 characters long and should not contain any of the following characters `:\/*?~<>{}#&%$!'"@+`|=`
`Width` / `Height`	Exported image size. A value between 16 and 8192.
`Multi Sample`	Over sampling amount (anti-aliasing). Disabled (lowest quality, fastest) up to x8 (best quality, slowest).
`Border Pixels`	Due to texture filtering (and MIP map use) on graphics hardware, it is usually desirable to expand the border pixels of non-contiguous areas of the map. The amount of pixels needed depends on number of MIP map levels that will be used. Generally it doesn't hurt to specify more pixels than needed.

NOTE: Using large image sizes with high Multi Sample settings (and/or many enabled output maps/channels) can use a *lot* of memory. For example a single 4096x4096 RGBA map with x8 multi sampling would require 4

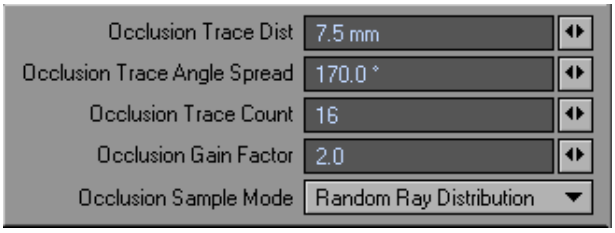
gigabytes of RAM during processing, which wouldn't even be possible in the 32-bit version. A 8192x8192 map with x8 multi sampling would require 16 gigabytes, that's not practical in most cases. If floating-point buffers are enabled then that would quadruple the memory use.

When the output dimensions of a map are non-square, the full UV range from 0 to 1 is still used. Which means if a checkered reference texture is used, it will look squeezed in one dimension in LW. Let's say, for example, we want to bake a 128 wide and 256 tall texture (ie. 1:2 image aspect ratio). We proceed to map it with even texture distribution (left image below), when we're done we can just scale the width by 200% to make use of the full UV range (right image below).





3.4 Occlusion/Dirt Map Settings



Selecting Occlusion/Dirt Map for baking will enable the occlusion map settings. These can be a bit tricky to get a feel for, and experimenting is the best way to get usable results. The occlusion value for each pixel is calculated by tracing a number of rays from pixel origin, on the high-poly geometry, spreading out in all directions. The occlusion value is based on how much of these traces hit something. Imagine being in a corner, a lot of rays will hit the walls, resulting in a more occluded result than something in the middle of a flat surface. Below is a figure and table describing the settings:

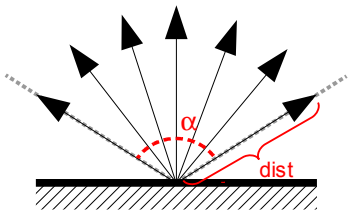
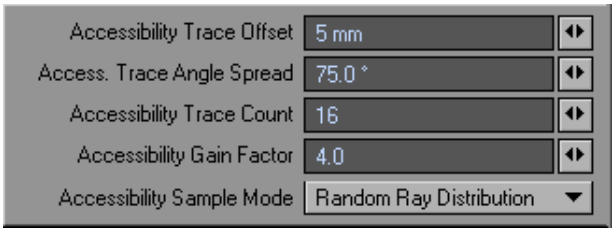


Figure 3: *Cone, with angle spread α, used to trace for occlusion.*

`Trace Dist`	Max length of rays when tracing for occlusion. Shorter distances (in relation to geometry size) generally results in “dirt” being drawn closer to corners/cavities, while longer distances will spread it outward more.
`Trace Angle Spread`	Cone angle that defines the interval in which ray directions are spread. 180° would be a hemisphere.
`Trace Count`	Number of rays to trace. More rays take longer time to compute, but

	gives more accurate results.
`Gain Factor`	Gain factor can be seen as an occlusion amplification or contrast factor.
`Sample Mode`	Determines the method used to gather samples. `Random Ray Distribution`: The ray directions, used for tracing, are randomized. Ensures that there are no pattern artifacts, similar to banding, but requires higher trace counts and/or multi sample levels to reduce noise. `Even Ray Distribution`: The ray directions are evenly distributed. Results in a smoother output map at lower trace counts, than random distribution, but will exhibit banding-like artifacts to some degree.

3.5 Accessibility Map Settings



Selecting Accessibility Map for baking will enable the accessibility map settings. It's an experimental algorithm that works in a similar fashion as occlusion maps, but instead of tracing for occlusion away from a point on the surface, it traces towards the surface from a point located in front of it. Due to the experimental nature not too much should be expected, also the algorithm is subject to change. Just like occlusion maps, this will require experimenting with values. Several parameters are also found in occlusion map settings. Below is a figure and table describing the parameters:

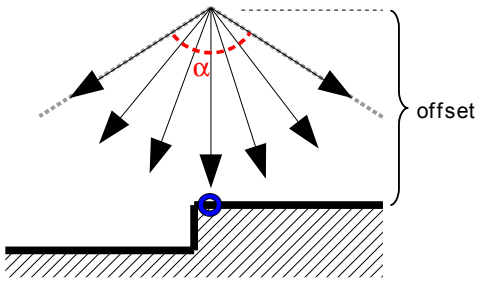


Figure 4: *Cone, with angle spread α, used to trace for accessibility.*

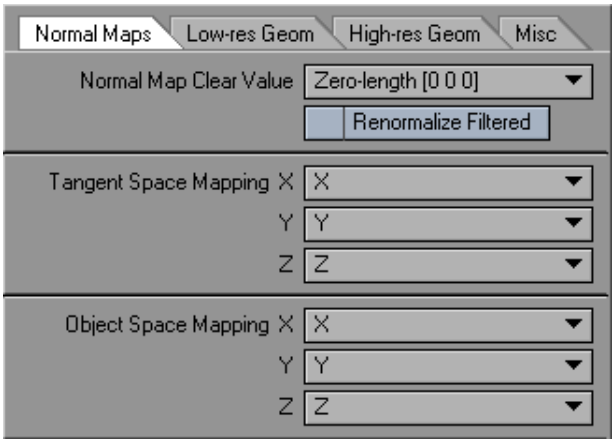
`Trace Offset`	Offset to the trace start point, in front of the surface.
`Trace Angle Spread`	Cone angle that defines the interval in which ray directions are spread.
`Trace Count`	Number of rays to trace. More rays take longer time to compute, but gives more accurate results.
`Gain Factor`	Gain factor can be seen as an amplification or contrast factor.
`Sample Mode`	Determines the method used to gather samples. For a description of

	each method, see “Sample Mode” parameter in section “Occlusion/Dirt Map Settings” above.

3.6 Baker Config

The baker maintains the config per lwo file. Meaning it will load/save config files in the same location and with the same file name as the lwo file. For example after baking something in `C:\LW\MyObject.lwo` it will save the config as `C:\LW\MyObject.bakercfg`. Any changes to the baker configuration will only be applied (and saved) after pressing OK. When moving an lwo file to another location, the bakercfg has to moved along with it, if the settings are to be preserved. *Note that if the object has never been saved (“Unnamed”)* ***no*** *bakercfg is saved.* The bakercfg contains the settings from the main baker window, but not the settings from `General Options`.

3.7 General Options



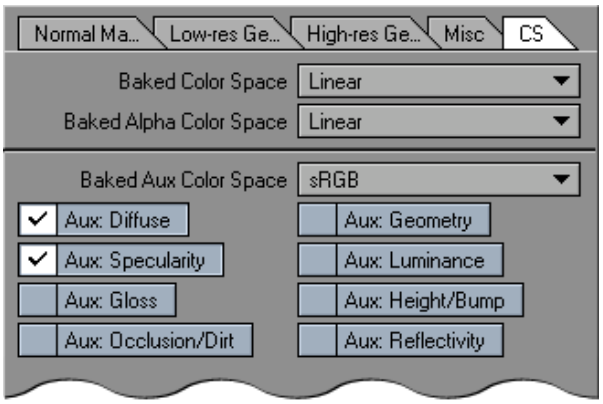
There are some configuration settings which apply to the plug-in in general, and are not saved per object in the bakercfg (see previous section). These affect such things as the way normal maps are calculated and geometry is processed. To make working on different projects, which may use different engines that require different settings, as smooth as possible, the general options are saved in the content directory currently set in LW. The last applied settings are also maintained and used as default when working with a fresh content directory. The following options are available:

`Normal Map Clear Value`	Specifies which value to use to initialize a normal map before baking, so that unused areas will have this value in the final map.
`Renormalize Filtered`	Renormalize multi sampled normal maps after downsize filtering.
`Tangent Space Mapping X, Y, Z`	Specifies how the coordinates axes are mapped in tangent space normal maps. The most common alternative to the default is inverted/flipped Y axis [`X , Inverted Y , Z`], but any weird mapping is possible. Keep in mind that mappings which don't include all three axes, will not work correctly with the renormalization option.
`Object Space Mapping X, Y, Z`	Same as Tangent Space Mapping above except it's used for object space normal maps.

`Smoothed Normal Weighting`	Specifies the weight scheme applied when calculating smoothed normals on low-res geometry. If the target environment also calculates smoothed normals by applying weighting, then enable the same option here for compatible normal map generation. `(none)`: Normals are averaged without weighting. This is the most common type, also used by LW. `Area Weighted`: Weighted by face area. This type is probably not used very often. For example nVidia's tool NVMeshMender has the option to use area weighting. `Angle Weighted`: Weighted by the angle between the two edges of each vertex in a face. This kind of weighting might be used by some 3DS tools for example.
`Weight Tangent Vectors`	Enables calculations, for smoothed tangent vectors, to apply the same weighting scheme as selected for normals.

`Smoothed Normal Weighting`	Specifies the weight scheme applied when calculating smoothed normals on high-res geometry. For more details see the same option for low-res geometry. Unlike low-res geometry this has no effect on the compatibility of generated maps, the option (in particular Angle Weighted) is available because it can help to get rid of smoothing artifacts or produce better smoothing.

`Use Single Layer Name`	When only a single FG layer is selected, and it has a name assigned, use the layer's name as a file name prefix. For example if the layer is named `Monkey`, the output color map would be named `Monkey_Texture.tga` instead of `Texture.tga`.
`Multithreading`	Number of threads to use by the baker. The `Automatic` option will use the same number of threads as the number of logical processors available (the number of CPUs reported by Windows, or CPU Usage in Activity Monitor on OS X). If the computer becomes too unresponsive with the `Automatic` option, you can manually set a thread count that is less than the number of available CPUs. WARNING: There may be problems with some nodes, or other plug-ins, that were not designed for multi-threading. If any problems occur, try using 1 thread.
`Bake Gloss as Roughness`	Output “Gloss/Roughness Map” as roughness values. The default is to output gloss/smoothness. Roughness is the inverted value of gloss/smoothness.



When running under LW[10] or later, there's an additional tab page named “CS” with color space settings. By default they're all set to `Linear` which means that the texture baker won't do any color space adjustements when saving the baked images.

`Baked Color Space`	Output color space for baked Color Map images and the RGB channel of Specular Map images. The color space for the monochrome specularity component, stored in the alpha channel of Specular Map images, can be set through Baked Aux Color Space.
`Baked Alpha Color Space`	Output color space for baked Alpha/Opacity Map images.
`Baked Aux Color Space`	Output color space for baked auxiliary map images. You can

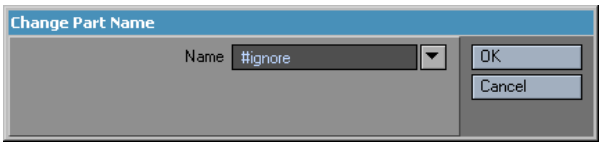
	configure which additional (monochrome) maps you want to be affected by the Aux color space with the checkboxes. Maps that aren't included will be saved unmodified, i.e. `Linear`.

Some maps are not subject to color space conversion as they contain data which isn't suitable for it.

4.0 Advanced Functionality

4.1 Excluding Geometry

Both low- and high-poly faces can be excluded from the baking process, by assigning them to a part named ”`#ignore`” (without quotes).



4.2 Mirroring Low-Poly Geometry

It's possible to have (partially) mirrored low-poly geometry, in order to save texture space, without messing up the baking process. This can be done without having to keep a separate copy of the geometry, for baking, where the mirrored faces are removed. This is achieved by telling the baker to discard the mirrored faces, using the ignore feature mentioned in “Excluding Geometry” above.

4.3 Grouping Geometry

It's sometimes desirable to be able to split up geometry in several groups during the baking process, so that the traces don't hit the “wrong” object, or if you want several objects to share one texture map. For this purpose chunks of geometry can be “grouped” by assigning them to parts. This needs to be done in both the low- and high-poly geometry, so that the corresponding geometry chunks have identical part names. Note that the part names `#ignore` and `#blocker` are reserved (see section “Excluding Geometry” above and “Blocker Geometry” below).

For example let's say you have bolt heads sticking out from a surface, which are modeled in both the low- and high-poly geometry. By assigning all high- and low-poly bolt polygons to the same part name, say “bolts”, the baker can process the base surface without any traces hitting a bolt, and it can bake the bolts separately without traces hitting any undesired obstacles.

4.4 Blocker Geometry

There are situations where different portions of an object are tightly together, for example fingers on a hand. Traces can hit another finger, causing artifacts in the baked image. You've reduced the Front trace distance (see section “Trace Distances” above) as much as possible, but there are still artifacts. Grouping geometry isn't an option because these aren't separable parts. Now what?

That's where blocker geometry comes in handy. You can insert polygons between the fingers, like separating “walls” or partitions. Just assign these polygons to a part named ”`#blocker`” (without quotes). These so called blockers will ensure that the trace front distance doesn't extend beyond the blocker.

It specifically works such that the baker will trace outward from the low-poly surface, and if it hits a blocker, it will start the actual baking-trace from the intersection point. When no blocker is hit, the baking-trace start point is always at Front distance in front of the low-poly surface. This also means that the blocker must be facing the target surface, or be double sided, for it to have any effect.

Blocker geometry must be located in a BG layer. You can either put it together with the high-poly mesh or keep it in a separate layer, as long as it's selected as BG during baking. Blockers are not group-able, so they affect all geometry, whether it's grouped or not. They do however not affect Occlusion/Dirt or Accessibility Map traces.

Note that including blocker geometry comes with a performance hit, since it's an additional trace that has to be done. If something can be solved using grouping, then that's usually preferable, but depending on the complexity, resolution and quality settings, the performance hit may or may not be noticeable.

5.0 Release Notes

5.1 Entering License Keys

After you have received a license key, that will look something like this:

```
[LICENSE ID="1111-2222-3333-4444"]
LsH60dRioGUE33UtuwDbyCJ4X2UeUSLr+QrO3OGxbwvimT10WlM1iqXijhYGy2ScNmyAoSkwqnCy
qktG5w187g==
[/LICENSE]
```

You can enter it into the plug-in by opening the `Global Options` dialog and clicking the `Enter License Key...` button located at the bottom. You can now switch to the mail program and copy the key (into the Windows/OS X clipboard) by selecting the entire key in the mail reader and pressing `Ctrl+C` (win) / ⌘+C (mac). Then switch back to Modeler and click `OK` in the license dialog. (It is also possible to copy the key to the clipboard prior to starting Modeler or running the plug-in.)

The `Enter License Key` dialog presents you with following options of where to store the license file:

`Globally`	The license file is saved in the same location as the plug-in, it applies globally to that particular LW installation, independently of which user is currently logged on. This probably is the preferred behavior in the majority of cases.
`LW Config Dir`	The license file is saved in the same location as the LW configuration files. By default this is the same path as `User Profile`, but can differ if an alternative configuration path was specified on the LW command line.
`User Profile`	The license file is saved in the user profile, normally something like `C:\Documents and Settings\<user name>` in Win2000/XP, `C:\Users\<user name>` in Vista/Win7 or `/Users/<user name>/Library/Preferences/LightWave3D` in OS X. Useful in environments where other users, who should not be allowed access to the license key, can log on to the computer.

NOTE: In order to access the `Global Options` dialog, the plug-in main window has to be opened, which requires you to have geometry and layer selections that the baker accepts. This is a little inconvenient, but entering a license key is usually something only done once so hopefully it does not cause too much trouble.

Alternatively, advanced users can copy-paste the license key into a (text) file named `PB_TextureBaker.lic`, which has to be located in the same directory as the plug-in file `PB_TextureBaker.p` (see `Globally` above), in the same directory where LW stores its config files (see `LW Config Dir` above) or in the user profile path (see `User Profile` above).

5.2 Additional Features

The goal is to be able to generate normal maps for a wide variety of target environments. The way smoothing and tangent space vectors are calculated, should be the same in both the normal map generator and the target engine, for best results. Should the already present configuration options not be enough to get correct results, you are welcome to contact me and we'll see if it can be fixed.

If there's any feature you would like to see, feel free to contact me (see section “Feedback” below). I cannot promise that a feature request will be fulfilled, but if I don't even know about it chances are a lot worse.

5.3 Feedback

Feedback, bug reports and questions are welcome. Either by using the contact information at http://www.blytools.com or using the NewTek forums http://forums.newtek.com and searching* for relevant threads (user name Myagi).

** For best results do a "Tag Search" for "pb_texturebaker"*

5.4 Copyright Notice