shaderMeister

Software License and Limited Warranty Agreement

PLEASE READ CAREFULLY BEFORE DOWNLOADING, INSTALLING OR USING THIS SOFTWARE:

THIS SOFTWARE LICENSE AGREEMENT IS A LEGAL AGREEMENT BETWEEN YOU (EITHER AN INDIVIDUAL OR, IF PUR-CHASED OR OTHERWISE ACQUIRED BY OR FOR AN ENTITY, AN ENTITY) AND DB&W BORNEMANN UND WOLF GBR.

BY CLICKING THE "ACCEPT" BUTTON, OPENING THE PACKAGE, DOWLOADING, INSTALLING OR OTHERWISE USING THIS SOFTWARE YOU AGREE TO BECOME BOUND BY THE TERMS OF THIS LICENSE . IF YOU DO NOT AGREE TO THE TERMS OF THIS LICENSE, CLICK THE "DO NOT ACCEPT" BUTTON AND DO NOT DOWNLOAD, INSTALL OR USE THIS SOFTWARE OTHERWISE.

1.0 DEFINITIONS

"Licensee" means you, the User, who agrees to all of the terms set forth in this Software License and Warranty Agreement, by downloading, installing or using this software.

"Licensed Software" shall mean the shaderMeister plug in for LightWave 3D and documentation for shaderMeister.

"Software" shall mean the shaderMeister plug in for LightWave 3D and documentation for shaderMeister.

2.0 OWNERSHIP OF THE SOFTWARE

The Licensee acknowledges and agrees that all of the computer programs and associated documentation are owned exclusively by db&w Bornemann und Wolf GbR. You agree that the price paid by you for the Software is a license fee granting you only rights set forth in this License Agreement.

3.0 RESTRICTED LICENSE shaderMeister

db&w Bornemann und Wolf GbR grants Licensee a non-exclusive, non-transferable, non-assignable, non-sub licensable, revocable and restricted license to use Licensed Software, in machine-readable, object code form only. The licensee agrees to use the Software solely for the purposes as outlined in this Agreement. The License Agreement does not convey to the licensee any ownership rights or any other interest in the software.

4.0 SCOPE OF LICENSE

This Software is licensed to be installed and used only in connexion with one LightWave 3D license. A valid license must be purchased for each LightWave 3D license.

5.0 LICENSE RESTRICTIONS

No license is granted to Licensee for any other purpose and Licensee may not sell, rent, lent, lease, loan or sub-license or otherwise encumber or transfer the Licensed Software, in whole or in part, to any third party for any purpose.

5.1 Licensee may not duplicate or modify or translate or decompile or create derivative works of the Licensed Software, or disassemble or reverse-engineer script or protocols of the Licensed Software.

5.2 Licensee agrees to use all reasonable efforts to protect the Software from unauthorized use, modification, re-production, distribution or publication.

5.3 Licensee acknowledges that nothing in this Agreement gives Licensee the right to use any trademark, trade name, or service mark of db&w Bornemann und Wolf GbR or any third party from whom db&w Bornemann und Wolf GbR has acquired license rights.

5.4 The License and other terms and conditions set forth in this Agreement shall apply to any and all versions and components of the Licensed Software furnished by db&w Bornemann und Wolf GbR to Licensee during the term of this Agreement and all versions and components of the Licensed Software used by Licensee in accordance with this Agreement.

5.5 The Licensed Software shall be used only by Licensee and only subject to the terms and conditions in this Agreement.

6.0 TITLE, PROPRIETARY RIGHTS, AND DEVELOPMENT

6.1 The Licensee agrees that db&w Bornemann und Wolf GbR and/or its licensors own all right, title and interest in the Licensed Software and in all patents, trademarks, trade names, inventions, copyrights, know-how and trade secrets relating to the design, manufacture, operation or service of the Licensed Software. All rights not expressly granted are hereby reserved by db&w Bornemann und Wolf GbR. Unauthorized copying or use of the Licensed Software, or failure to comply with the restrictions provided in this Agreement, will result in automatic termination of this Agreement. Nothing in this Agreement shall change db&w Bornemann und Wolf GbR's or any of its licensors' ownership rights to their respective intellectual property, including but not limited to the Licensed Software.

6.2 db&w Bornemann und Wolf GbR is under no obligation to support the Licensed Software in any way, or to provide Licensee with updates, bug fixes, builds, or error corrections (collectively "Software Updates"). If db&w Bornemann und Wolf GbR at its sole option, supplies Software Updates to Licensee, the Updates will be considered part of Licensed Software, and subject to the terms and conditions of this Agreement.

6.3 The Licensee acknowledges that db&w Bornemann und Wolf GbR shall have the right, in its sole discretion and without incurring any liability, to modify the Licensed Software or discontinue its use or distribution at any time and for any reason.

7.0 TERM AND TERMINATION OF AGREEMENT

This Agreement will commence on the date on which Licensee downloads Licensed Software (the "Effective Date") and will remain in force until terminated.

7.1 Licensee might terminate the license at any time by removing the Software from the computer and destroying the original Software and all copies.

7.2 Upon termination or expiration of this Agreement, Licensee must cease use of and destroy the Licensed Software, any records or copies thereof and any related materials, and provide to db&w Bornemann und Wolf GbR a written statement certifying that Licensee has complied with the foregoing obligations within thirty (30) days.

7.3 This license will automatically terminate if the licensee breaches any of the terms or conditions set out in this license agreement.

7.4 Rights and obligations under this Agreement which by their nature should survive, shall remain in effect after termination or expiration hereof.

8.0 DISCLAIMER OF WARRANTY

8.1 Licensed Software IS PROVIDED "AS IS." ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS, AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

8.2 db&w Bornemann und Wolf GbR does not warrant that the operation of the Licensed Software will be uninterrupted or error-free.

9.0 LIMITATION OF LIABILITY

9.1 IN NO EVENT SHALL db&w Bornemann und Wolf GbR BE LIABLE FOR ANY DAMAGES WHATSOEVER, INCLUDING ANY OR ALL GENERAL, SPECIAL, CONSEQUENTIAL, INCIDENTAL, INDIRECT OR PUNITIVE DAMAGES ARISING OUT OF THIS AGREEMENT OR USE OF THE LICENSED Software OR CONFIDENTIAL INFORMATION, HOWEVER CAUSED, ON ANY THEORY OF LIABILITY, AND WHETHER OR NOT db&w Bornemann und Wolf GbR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. THESE LIMITATIONS SHALL APPLY NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY.

9.2 Notwithstanding the foregoing, any damages that you might incur for any reason whatsoever (including, without limitation, all damages referenced above and all direct or general damages), the entire liability of db&w Bornemann und Wolf GbR and any of its suppliers under any provision of this Agreement and your exclusive remedy for all of the foregoing shall be limited to the greater of the amount, if any, actually paid by you for the Licensed Software or U.S. $5.00. The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails its essential purpose.

9.3 Licensee will hold db&w Bornemann und Wolf GbR harmless from any claims based on Licensee's use of the Licensed Software for any purposes, and from any claims that later versions or releases of any Licensed Software furnished to Licensee are incompatible with the Licensed Software provided to Licensee under this Agreement.

9.4 Licensee shall have the sole responsibility to adequately protect and backup Licensee's data or equipment used in connexion with the Licensed Software. Licensee shall not claim against db&w Bornemann und Wolf GbR for lost data, inaccurate output, work delays or lost profits resulting from Licensee's use of the Licensed Software.

9.5 Neither party will be liable for any indirect, punitive, special, incidental or consequential damage in connection with or arising out of this Agreement, including loss of business, revenue, profits, use, data or other economic advantage, however it arises, whether for breach or in tort, even if that party has been previously advised of the possibility of such damage.

10.0 GENERAL TERMS AND INDEMNITY

10.1 Any action relating to or arising out of this Agreement shall be governed by German law.

10.2 Neither party may assign or otherwise transfer any of its rights or obligations under this Agreement, without the prior written consent of the other party, except that db&w Bornemann und Wolf GbR may assign this Agreement to any other company.

10.3 If any part of this Agreement is found void and unenforceable, it will not affect the validity of the balance of the Agreement, which shall remain valid and enforceable according to its terms. In the event that it is determined by a court of competent jurisdiction as a part of a final non-appeal able judgement that any provision of this Agreement (or part thereof) is invalid, illegal, or otherwise unenforceable, such provision will be enforced as nearly as possible in accordance with the stated intention of the parties, while the remainder of the Agreement will remain in full force and effect and the parties will be bound by obligations which approximate, as closely as possible, the effect of the provision found invalid or unenforceable, without being themselves invalid or unenforceable. The waiver of any breach or default shall not constitute a waiver of any other right in this Agreement or any subsequent breach or default. No waiver shall be effective unless in writing and signed by an authorized representative of the party to be bound. Failure to pursue, or delay in pursuing, any remedy for a breach shall not constitute a waiver of such breach.

10.4 Licensee agrees that the Licensee shall indemnify, defend, and hold harmless db&w Bornemann und Wolf GbR, and its respective officers, directors, employees, agents, successors, and assigns, from any damage, allegation, cost, loss, liability, or expense (including court costs and reasonable fees for attorneys or other professionals) arising out of or resulting from or in connexion with any breach or claimed breach of the terms of this Agreement or warranties provided herein.

Trademark and copyright information

TRADEMARKS

This paper may contain brands, company names and product names of third parties, and following are the requirements for identifying trademarks and registered trademarks of those third parties. This list is not exhaustive and the paper may contain brands, company names and product names of additional third parties. All brands, company names and product names are trademarks or registered trademarks of their respective holder.

LightWave® and LightWave 3D® are registered Trademarks of NewTek, Inc.

Acknowledgements

Software Design and Development

Michael Wolf

Management

Dagmar Bornemann

Documentation

Michael Wolf

Product Marketing

Dagmar Bornemann

Table of Contents

Introduction

Compatibility

shaderMeister is compatible with LightWave3D 9.3 up to LightWave 3D 9.6.1.

It currently runs with the Windows 32bit, Windows 64bit and Mac OSX versions of LightWave. It has been tested with Windows 2000, the 32bit and 64bit versions of Windows XP as well as Mac OSX. This is why you will see screen shots of both the OSX and the Windows port in this manual.

Please visit http://shadermeister.db-w.com for more up to date information.

Features

shaderMeister has been thoroughly tested in production by us and our trusty beta testers. The main features of shaderMeister are:

- Advanced Presets system
- Free point upgrades, free support
- Light Groups
- Item comment based variables exposed via nodes
- New selection types for the Item Filter
- Batteries not included![1]

New Feature
Version 1.1

---

1 And fortunately also not required.

Setting up shaderMeister

Installation

The download contains ZIP compressed files for all platforms supported by shaderMeister. Extract the appropriate ZIP file for your platform.

Windows 32-bit/64-bit

The *shaderMeister.p* file can be copied to any directory where you keep your plugins. We recommend using a manually created plugin directory to separate third party plugins from plugins shipped with LightWave 3D. This will simplify upgrades of LightWave 3D.

Now use the **Edit Plug-ins** panel (alt-F11) of the **Add Plugins** menu item to add the plugin file **shaderMeister.p** to LightWave 3D.

Mac OS X, UB

The OS X version of shaderMeister is included as a DMG (Disk Image). Double click on the icon to mount the disk image.

It also includes a folder of sample content as shown in the examples section of this manual.

Universal Binary

You can simply drop the *shaderMeister.plugin* icon onto the Install application to install it.

The plugin will either be copied to:

Current User installation option: *~/Library/Application Support/LightWave3D/Plugins*

or

All Users installation option:

/Library/Application Support/LightWave3D/Plugins

In both cases LightWave 3D 9.6 will automatically pick up the plugins once you launch it.

You are of course free to copy the plugin file to another location and add it manually.

The download contains ZIP compressed files for all platforms supported by shaderMeister. Extract the appropriate ZIP file for your platform.

The *shaderMeister.p* file can be copied to any directory where you keep your plugins. We recommend using a manually created plugin directory to separate third party plugins from plugins shipped with LightWave 3D. This will simplify upgrades of LightWave 3D.

Now use the **Edit Plug-ins** panel (alt-F11) of the **Add Plugins** menu item to add the plugin file **shaderMeister.p** to LightWave 3D.

To be able to use shaderMeister you will need to register and activate it with the activation code we will send you.

Registration
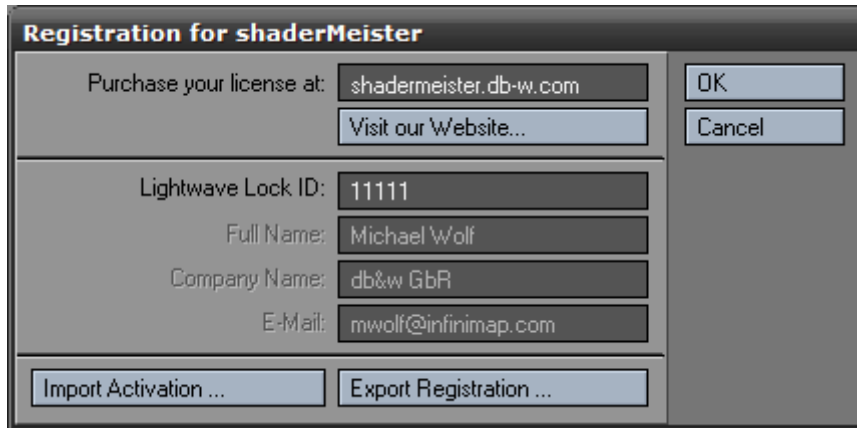
Once you've installed the plugin, apply the **shaderMeister** master handler plugin to your scene, and click on the "Register Plugin..." button.

Enter your full name, company name (if applicable) and e-mail address. Export the registration and attach it to an e-mail to **registration@db-w.com**. We will mail you an activation code within 24 hours after payment is received, depending on the time difference[2].

You can import the keyfile attached to the activation e-mail.

Your registration is now complete and you will see the interface of shaderMeister, allowing you to use it.

The registration file is stored in the same directory as the plugin, in the file shaderMeister.key. If you use multiple licenses of LightWave 3D from a single network share, the registration manager will only append/edit licenses locked to the dongle installed on the host machine. It will however manage multiple licenses allowing for a single registration file on your network share.

Included Plugins

shaderMeister is a collection of plugins for LightWave 3D. Currently it consists of the following plugins:

- shaderMeister (both a Surface Shader and a Master Handler)
- shaderMeister generic (can be assigned to a shortcut or added as a menu item)
- Filter Node
- Logic Node
- Surface Exposure Node

**2**   Our key generator needs some sleep every now and then ;)

What shaderMeister does ... and how it does it

shaderMeister allows you to override the shading of surface on a scene by scene basis. This is accomplished by a shader plugin that needs to be applied to all surfaces that should be overridden by shaderMeister.

> Originally Posted by **Exper**
> *"Parameter 1: use it to change the value of Parameter 1".*

When rendering the shader plugin will detect if a shaderMeister global shading graph is present and then evaluate it.
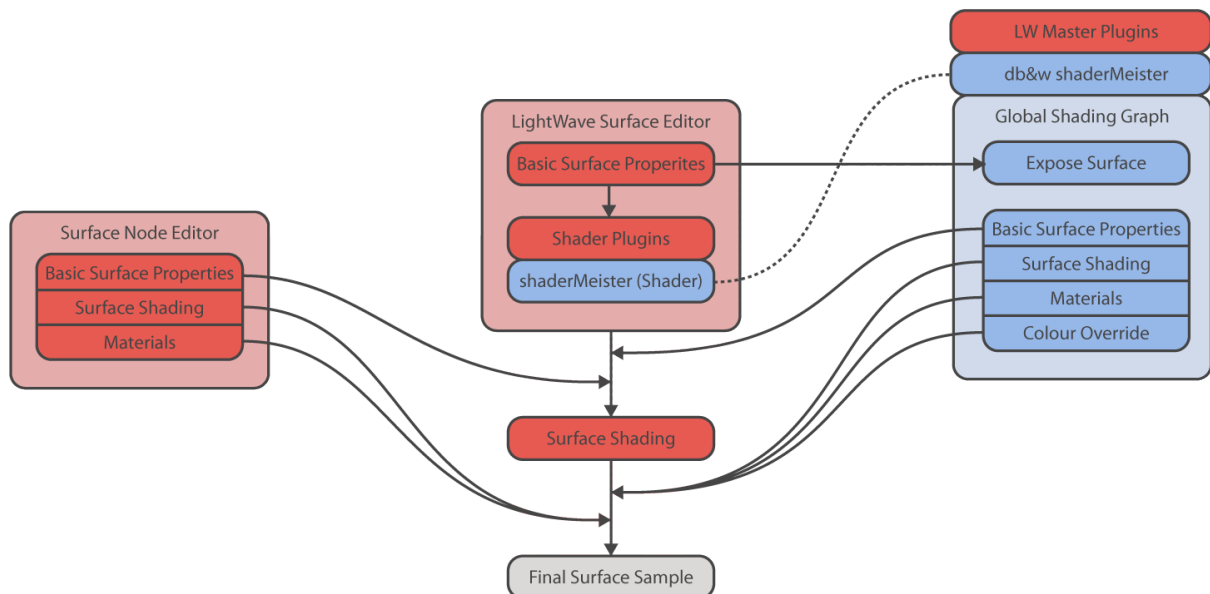
The shaderMeister global shading graph itself is saved with the scene. A master handler plugin manages it.

Once you have applied the shader to all surfaces in your current scene (which only needs to happen once and is automated by shaderMeister) you can thus override the surfacing on a scene by scene basis.

Unfortunately due to the nature of the LightWave 3D shading systems, things aren't as simple as they seem to be.

Since shaderMeister is applied as a shader, it has limited access to some of the nodal shading information and can also only override some of the nodal data.

The nodal shading inputs (diffuse shading, specular shading, reflection shading, refraction shading, material) are, unfortunately, applied after the shaderMeister shader plugin.

Pros and Cons of the method used by shaderMeister

Initially there were two ideas on how to design the functionality that shaderMeister should provide.

One was to implement it as a Pixel Filter (as some free plugins do), the other was to route the shading through a shader plugin (which is how it is implemented now).

There are a few reasons for that:

- If the Colour Override or the Material/Shading Inputs are used, then LightWave itself will not compute any shading at all – this can speed up rendering simple passes tremendously.
  A Pixel Filter will <u>always</u> compute the shading of the surfaces prior to applying additional effects.

- By using the a plugin shader to apply the global shading, shaderMeister will work in reflections and refractions as well. One can even limit shading using the current recursion level.

- The downside is that the shader needs to be applied[3].

---

3   Sigh, you win some and you lose some, but this at least happens semi-automatically by pressing a button.

shaderMeister – Master Handler Plugin

This plugin is the core of shaderMeister. It manages a global nodal surfacing graph that overrides LightWave's native surface via the Shader Plugin.

The global surfacing graph is loaded and saved via the master handler and thus **scene** specific. This allows you to easily create different scenes with different global shading overrides.

The Master Handler Plugin supports presets to facilitate this and allow you to create a library of overrides.

User Interface

Presets

Using the Presets pop-up you can apply any of your presets, save your current settings as a preset or delete existing presets.

shaderMeister has its own system to manage presets. The main reason for developing a custom system was to differentiate between user, global and project (content directory) specific settings.

user presets are stored in the same directory where your LightWave3D configuration files are stored, in a subdirectory called "SimplePresets".

global presets are stored in the installation directory of LightWave3D on Windows, on OSX UB they are stored in */Library/Preferences/LightWave3D/SimplePresets/.*

content presets are stored in the current content directory.

shaderMeister supports the concept of **default** presets. If you name a preset "**default**" then it will be loaded automatically if shaderMeister is added to a scene.

shaderMeister searches for default presets in the **user**, **global** and **content** presets in that order. This means that a default **content** preset overrides a **global** default which in turn has precedence over a **user** preset.

Save as Preset...

This option allows you to save the current shaderMeister settings to a preset. Choose any of the three types and enter a name for the preset. You can also use the small pop-up to select any of the currently available preset names.

Delete Preset...

This allows you to select and of the available presets and delete them from the hard drive.
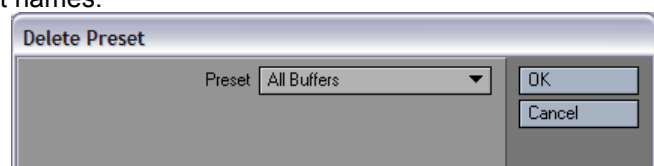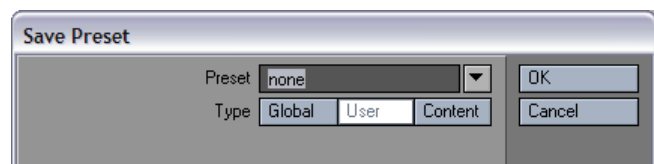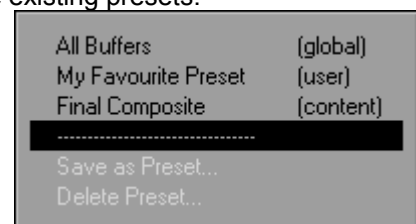
Global Shading Enabled

This is a switch to deactivate shaderMeister.

Nodal Graph...

This opens the nodal shading graph that serves as the override.

The inputs reflect the settings that are available via the normal surface editor. However, since shaderMeister is applied as a shader they may behave differently.

Disable Surface Nodes

This option disables all nodes that are applied to the original surface. There are two reasons to do so:

1) Speed up rendering

2) The shading and materials applied by surface nodes are applied after the shader has been evaluated. This overwrites the changes made by shaderMaster.
   Disabling Surface Nodes allows shaderMeister to overwrite nodal materials and shaders again.

Use Global Light Group

New Feature
Version 1.1
Enables the global light group for all inputs evaluated by the shaderMeister Surface node inputs.

## Edit Global Light Group

New Feature
Version 1.1
The global light group allows you to define a selection of lights that will be used when evaluating nodes.

The light group will be used for all nodes connected to the shaderMeister Surface node if **Use Global Light Group** is active.

There is also a **Global Light Group Node** which allows you to selectively restrict the lights visible to upstream nodes in any other nodal shading graph (such as those placed on a normal surface).

Clicking on the **Edit Global Light Group...** button will open a new panel.

Please have a look at the section covering the **Light Group Node** as well as the **Global Light Group Node** for a detailed explanation of the controls.
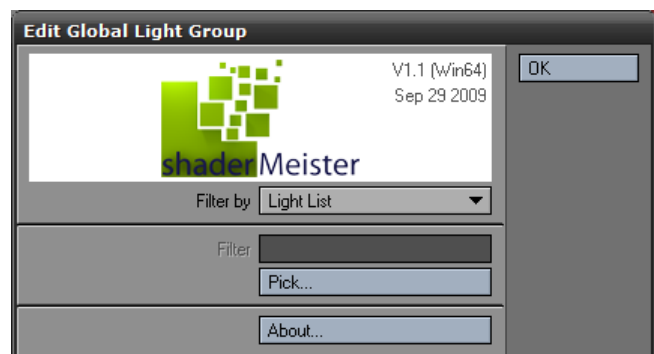
Scene Editing

This option allows for simple batch operations on the current objects.

Add All Shaders

This adds the shaderMeister shader plugin to all surfaces that don't have it applied yet.

Remove All Shaders

This removes the shaderMeister shader plugin from all surfaces.

Unfortunately this will crash if surfaces contain shaders in addition to the shaderMeister shader. This is a bug in LightWave 3D 9.3.1 to 9.6 and has been reported to NewTek and fixed for subsequent[4] versions of LightWave3D.

---
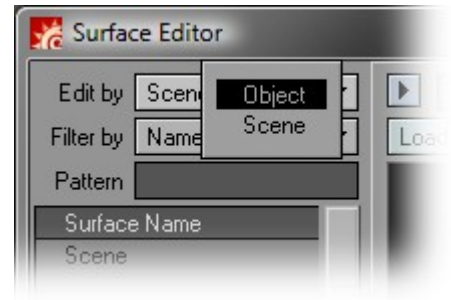
4   ...but currently unreleased...

Node Input Connections

Here is a description of the inputs available in the node graph. These inputs are evaluated from top to bottom. This implies that the lower inputs may override the ones further up in the node.

Colour, Luminosity, Diffuse, Specular, Glossiness, Reflection, Transparency, Refraction Index, Translucency

These inputs correspond to the shading properties as defined in the surface editor. They may be overridden by surface nodes.

Normal

This overrides the following bump input, allowing you to set the normal for the current surface directly as opposed to using a bump value.

Bump, Colour Highlights, Colour Filter, Diffuse Sharpness, Additive Transparency, Reflection Blurring, Refraction Blurring

These inputs correspond to the shading properties as defined in the surface editor. They may be overridden by surface nodes.

Override

This input is special and modifies how the following inputs behave.

It has been designed to work in conjunction with the **Logic** output of the **Filter Node** or the **Logic Node**, explained later in this manual.

If nothing is connected to **Override** then the following inputs will behave as expected.

If **Override** is connected and the value that is passed on to it is 0, the following inputs will be completely ignored (effectively displaying the original surface).

If **Override** is connected and a value other than 0 then the following inputs will be active.

Diffuse Shading, Specular Shading, Reflection Shading, Refraction Shading

These inputs correspond to the respective inputs in the nodal surface shading system.

They may be overridden by surface nodes connected to the respective shading inputs or the material input in a surface shading network.

Material

This input overrides the Diffuse/Specular/Reflection/Refraction Shading and Transparency. It may be overridden by surface nodes connected to the Shading inputs or the Material input.

Colour Override

This colour completely overrides the colour of the currently rendered pixel. This will be displayed as is in the final image.

If Colour Override is connected, LightWave 3D will not compute any shading on the surface by itself (unless it is a nodal surface). This can be a tremendous speed boost rendering out passes that do not rely on shading/lighting.

Unfortunately this may be overridden by shading and Material nodes when using nodal surfacing.

Node Input Connections and LightWave Buffers

Some of the shaderMeister inputs end up being visible in the buffers that LightWave 3D supports (unless they are further modified by surface nodes).

This also allows you to re-use inputs (and thus buffers) if a superior input is connected.

All inputs affect the Final Render buffer, which is the buffer that contains the "plain" rendered image.

Please note that some nodes (especially Material nodes) tend to also write into buffers that they're not directly connected to (this has been added by NT to make material based shading more compatible with the buffer system).

Here is a list of the inputs and the buffers affect by them:

Node Input	LightWave Buffer
Colour	Raw Colour
Luminosity	Luminosity
Diffuse	**Diffuse** (which contains the actual value) as well as **Diffuse Colour** which contains the shading due to the diffuse value.
Specular	**Specularity** (which contains the actual value) as well as **Specular Colour** which contains the shading due to the specular values (this includes Glossiness).
Glossiness	**Specular Colour** which contains the shading due to the specular values (this includes Specular).
Reflection	**Reflectivity** (which contains the actual value) as well as **Reflection Colour**, which contains the shading due to the reflection value.
Transparency	This affects the **Alpha** buffer as well as the **Refraction Alpha** buffer.
Refraction Index	This affects the shading stored in the **Refraction Colour** buffer.
Translucency	
Normal	Affects all shading buffers (since the normal changes how the surface is shaded)
Bump	Affects all shading buffers (since it changes how the shading is computed)
Colour Highlights	Affects all reflection and specular shading buffers.
Colour Filter	Affects the **Refraction Colour** buffer.
Diffuse Sharpness	Affects all buffers that depend on diffuse shading.
Additive Transparency	
Reflection Blurring	Affects the **Reflection Colour** buffer.
Refraction Blurring	Affects the **Refraction Colour** buffer.
Diffuse Shading	Directly writes into the **Diffuse Colour** buffer.
Specular Shading	Directly writes into the **Specular Colour** buffer.
Reflection Shading	Directly writes into the **Reflection Colour** buffer.
Refraction Shading	Directly writes into the **Refraction Colour** buffer.
Material	Directly writes into the **Diffuse Colour** , **Specular Colour**, **Reflection Colour** and **Refraction Colour** buffers as well as the **Alpha** buffer.
Colour Override	Directly overwrites the **Final Render** buffer (as well as **Pre-Effect Colour**)

shaderMeister – Shader Plugin

The shader plugin needs to be applied to all surfaces that you intend to override with shaderMeister.

If there is no shaderMeister global shading network in the scene the shader plugin will do nothing. In Modeler, it will do nothing as well.

The shaderMeister shader can be mass applied using the **Scene Editing** option in the master handler.
The objects will need to be saved after that to make the change permanent.

No other settings are saved with the Shader Plugin, it is just used if the shaderMeister Master Handler is present in the scene.

Nodal Graph...

This button opens the global shading graph from any instance of the shaderMeister Shader Plugin for your convenience. The global shading graph is still only saved with the Master Handler plugin.
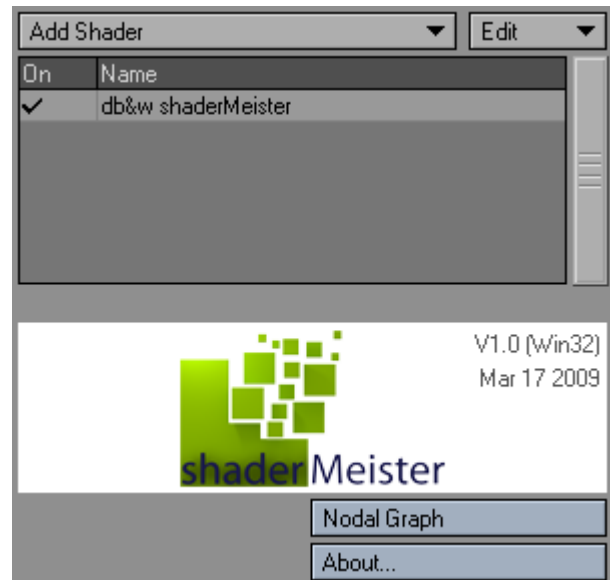
Filter Node

The Filter Node allows you to filter the currently shaded spot by:

- Surfaces
- Items
- Object name (as stored on disk as a .lwo)
- Weight Map Values
- Item Comments

New Feature
Version 1.1

These are filtered using their names or allow you to precisely select items/surfaces (with the exception of the Weight Map option). You can use wildcards to let the filter match more than one surface/item/object.

User Interface

Pass Through

This button completely disables the node.

The respective node inputs will be passed through to the outputs without change. If there is nothing connected to the inputs, the outputs will be either 0 or black, depending on the output type.

Logic Operation

This setting allows you to modify the output of the Filter Node with the output of another (Filter or Boolean) node.

The result of a boolean operation on the Logic input and the result of the filtering operation within the node will be used to determine the outputs of the Filter node.

This allows you to daisy chain and combine multiple filter nodes.

The boolean operations are:
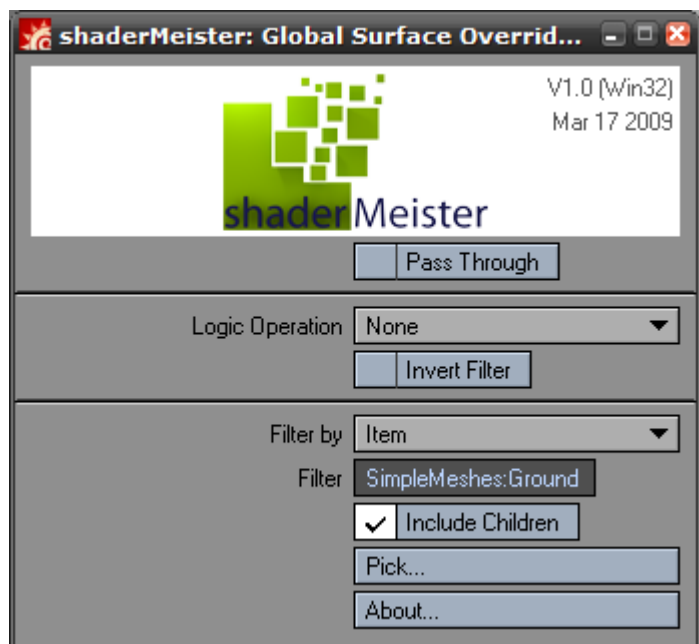
Logic AND Filter

If both the Logic input and the Filter result are the same, then the Logic output will be true and the new inputs will be passed through.

The **AND** Operation returns true if both inputs have the same value.

Logic Input	Filter	Result
True	True	True
True	False	False
False	True	False
False	False	True

Logic OR Filter

The **OR** Operation returns true if either of the inputs is true

Logic Input	Filter	Result
True	True	True
True	False	True
False	True	True
False	False	False

Logic XOR Filter

The **XOR** Operation returns true only if the inputs have different values.

Logic Input	Filter	Result
True	True	False
True	False	True
False	True	True
False	False	False

Invert Filter

Inverts the logic of the filter.

Filter by

New Feature
Version 1.1

You can filter the surface by a number of criteria. The options that are based on names are suited for presets if you adhere to a strict naming scheme within your scenes.

Surface List (by Object)

This allows you to open a panel using **Pick...** button that displays a list of all current objects and their respective surfaces.

The surfaces picked will only contribute to the filter for the objects they belong to. Surfaces with identical names but applied to different objects[5] will be treated as different surfaces.
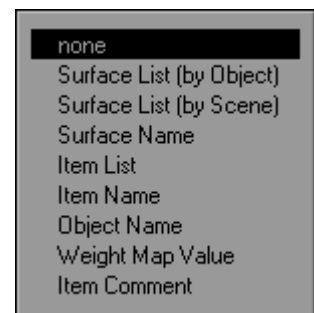
Surface List (by Scene)

This displays a list of all surfaces in a scene when you click on the **Pick...** button. Only surfaces are displayed, regardless of the objects they belong to.

Surface Name

This filters by the surface name.

Item List

This displays a list of all items in the current scene when you click on the **Pick...** button.

---

5 Object as in .lwo file that is.

Item Name

This filters by the item name as displayed in LightWave 3D. This also includes layer names.

Object Name

This filters by the object name as stored on disk (the .lwo file).

Weight Map Value

This filters by weight maps.

Item Comment

This option allows you to filter by item comments. This is extremely handy as item comments are saved within the scene file.

Include Children

(only if filter by **Item List** or by **Item Name** is active)

This will include the selected item as well as its children to the filter. This allows you to filter a group of items by only matching the parent item.

Weight Map

(only if filter by Weight Map)

This allows you to select the weight map used by the filter.

Weight Threshold

(only if filter by Weight Map)

Any weight map value above and including this threshold will be considered as being filtered, anything below will be considered as not being filtered.

If the weight map is not applied at all it will also be considered as not being filtered.

Filter

Supported wildcards

*

Matches any amount of characters (or none)

?

Matches any single character.

|

Allows you to concatenate multiple filters
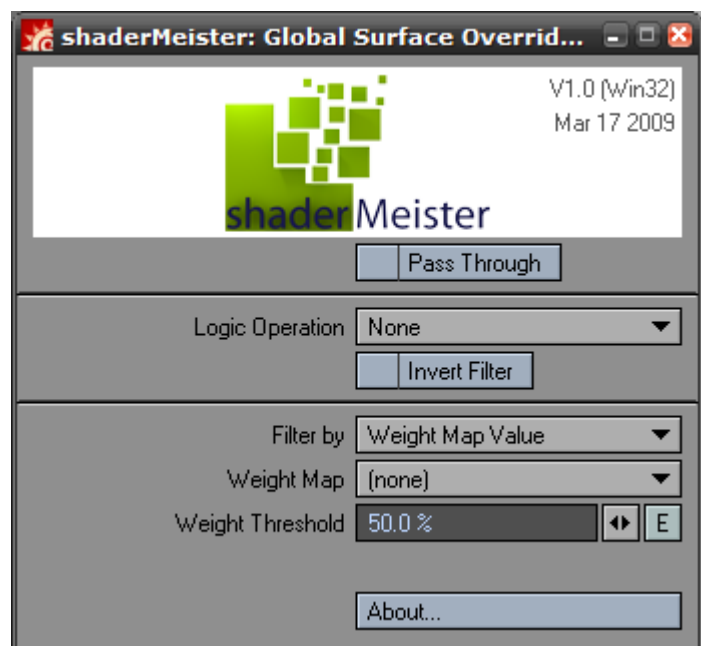
Examples

Let's assume we have the following surface names to filter:

Filter	green	greenish	almost_green	blue	blueish	almost_blue	blue01	blue02	blue3
green*	✓	✓	✓	✗	✗	✗	✗	✗	✗
*ish	✗	✓	✗	✗	✓	✗	✗	✗	✗
blue0?	✗	✗	✗	✗	✗	✗	✓	✓	✗
blue*	✗	✗	✗	✗	✓	✗	✓	✓	✓
blue*|green*	✓	✓	✗	✓	✓	✗	✓	✓	✓

Pick...

This button opens a new window that allows you to select a **Surface**, **Item** or **Object**.

The selection will either **Replace** the current **Filter** or **Append** to it (including the insertion of a "|" character to concatenate it to the existing **Filter**).

Node Input Connections

Logic

This integer input allows you to daisy chain multiple Filter or Boolean nodes.

It will be combined with the result of the filter using the **Logic Operation**.

baseVector, baseColour, baseScalar, base Material

These inputs will be passed through to the respective outputs if the filter is not **true**.

NewVector, newColour, newScalar, newMaterial

These inputs will be passed through to the respective outputs if the filter is **true**.

Invert

This integer input will invert the result of the filter if it is set to anything **but** 0.
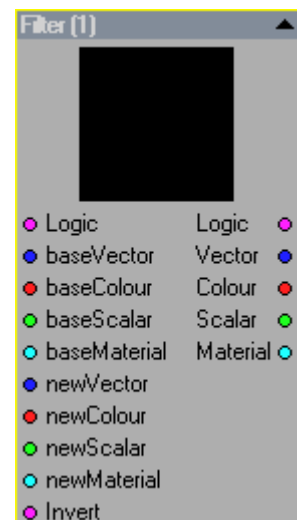
Node Output Connections

Logic

This output is 1 if the result of the filter and the Logic Operation on the Logic input result in a true.

Otherwise it is 0.

This allows you to daisy chain the Filter node with other filter nodes or Boolean Logic nodes to create complex ex-/inclusions.

Vector

Depending on the result of the filter this will either be the **baseVector** or the **newVector** input.

Colour

Depending on the result of the filter this will either be the **baseColour** or the **newColour** input.

Scalar

Depending on the result of the filter this will either be the **baseScalar** or the **newScalar** input.

Material

Depending on the results of the filter this will either be the **baseMaterial** of the **newMaterial** input.
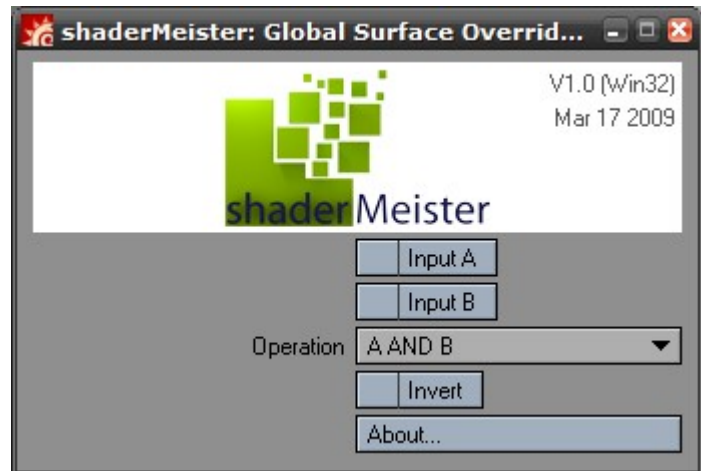
Boolean Node

This node allows you to combine two logical values using Boolean logic.

Using the Logic output of the Filter Node this allows you to design elaborate selections to narrow down the parts of a scene that shaderMeister affects.

Boolean logic deals with elements that can only have two values: true or false, 1 or 0 respectively.

User Interface

Input A

This button allows you to set **Input A** to either True (active) or False (not active) if nothing is connected to the respective input.

Input B

This button allows you to set **Input B** to either True (active) or False (not active) if nothing is connected to the respective input.

Operation

The Boolean operations are best explained with logic tables, as there is a difference to the usage of the words in English.

A and B

The **AND** Operation returns true if both inputs have the same value.

Input A	Input B	Result
True	True	True
True	False	False
False	True	False
False	False	True

A or B

The **OR** Operation returns true if either of the inputs is true

Input A	Input B	Result
True	True	True
True	False	True
False	True	True
False	False	False

A xor B

The **XOR** Operation returns true only if the inputs have different values.

Input A	Input B	Result
True	True	False
True	False	True
False	True	True
False	False	False

Invert

As expected this inverts the result of the Boolean operation.
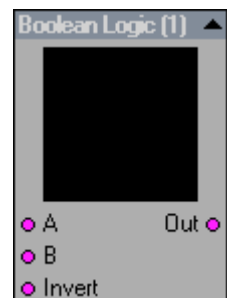
Node Input Connections

A

This is the first parameter of the Boolean operation. A value of 0 corresponds to **false,** any other value is **true**.

B

This is the second parameter of the Boolean operation. A value of 0 corresponds to **false,** any other value is **true**.

Invert

This inverts the result of the boolean operation. **False** turn to **true**, **true** to **false**.

Node Output Connections

Out

This is the result of the boolean operation. A **true** value will be exposed as 1, **false** as 0.

Surface Exposure Node

Please Note:

The Surface Exposure Node should only be added to shaderMeister nodal Surface Graphs applied using the Master Handler plugin.

It will fail to produce any meaningful output in any other Node Editor within LightWave 3D.

It is still very useful though ;)

The Surface Exposure Node makes the shading properties and channels of the currently shaded surface available in the shaderMeister Nodal Graph.

This allows you to modify existing surfaces as opposed to just replacing them.

The outputs correspond to the matching channels in the layered surface editor and include the effects of layers and envelopes on the surface.

These are basically the surface channel as seen by the shader plugin, the effects of nodal surfacing are ignored.

Node Output Connections

Colour, Diffuse, Luminosity, Specular, Reflection, Transparency, Refraction Index, Translucency, Colour Highlights, Colour Filter, Diffuse Sharpness, Additive Transparency, Reflection Blurring, Refraction Blurring

These outputs correspond to the surface properties in the surface editor. Changes due to nodal surfacing are not available due to SDK limitations.

Light Group Node

The Light Group Node can restrict the lights that are used by other nodes for their respective computations.

The basic idea is[6] quite simple. The node has one Colour and one Material input as well as one matching Colour and Material output.

Anything connected to either of these inputs will only be able to get lit by the lights defined in the Light Group Node, but only if you actually retrieve the Colour or Material from the node output.

In this sense it acts like a filter for the upstream nodes.

If you daisy chain Light Group Nodes, then the one further upstream will completely replace any existing light group (it is thus not restricted by the previous light group).

The node interface is quite similar to the Filter node.

The Light Group Node happily works in normal (per surface) shading graphs as well and thus a allows for per surface light exclusion.

Pass Through

If you enable this control then the light group will not be used upstream of this node.

Effectively the Colour input will go straight into the Colour output, the Material input will go straight into the Material output.

Filter By

There are three ways of selecting lights as part of the light group:

Light List

This allows you to use the **Pick...** button to open a new window with a list of included and excluded lights for the current group.
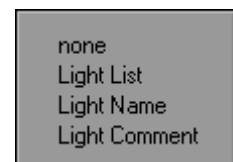
Light Name

This allows you to use names to include lights to the light groups. Just like using item/surface names in the filter node (see Filter) you can use wildcards.

The **Pick...** button will allow you to select light names that exist in the current scene to either replace or append to the current name.

Light Comment

This option allows you to filter the lights to be included in a light group by item comments associated to the light.

As with light names (and as explained in detail in the Filter Node section) wildcards are allowed.

**6**   Hopefully that is...

Filter

This control is enabled if the **Filter by** option is either set to **Light Name** or **Light Comment**.

This contains the string that is used to filter the lights. Wildcards are allowed as explained in the section about the Filter Node.

Pick...

This control is enabled if the **Filter by** option is set to either **Light List** or **Light Name**.

If the **Filter by** option is set to **Light List** the following panel will be displayed:
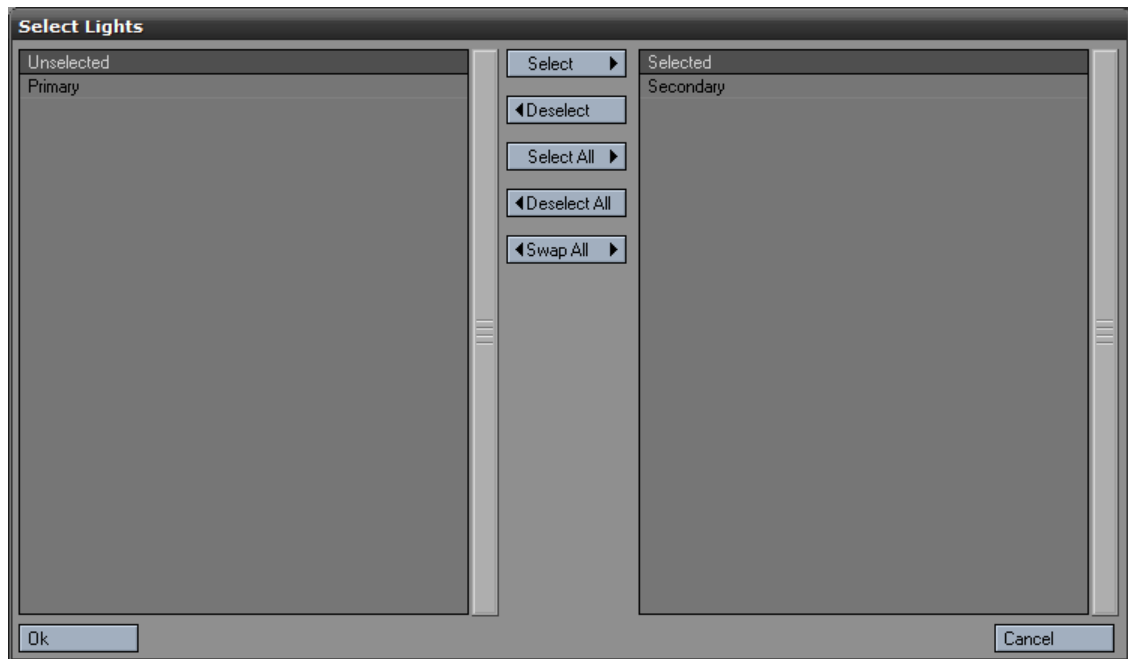
The lights in the left column show all the lights in the scene that are not a part of the light group, the lights on the right hand side are part of the light group.

The controls should be self-explanatory[7].

If the **Filter by** option is set to **Light Name** the Pick panel will be displayed as shown on the right.

The selection will either **Replace** the current **Filter** or **Append** to it (including the insertion of a "|" character to concatenate it to the existing **Filter**).

---
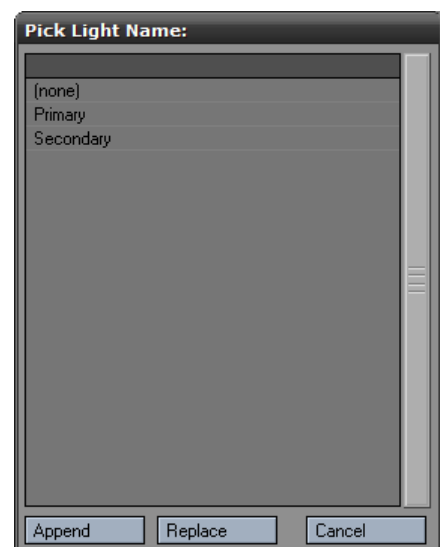
7 If they're not then we apparently screwed up the user interface design. In any case, if you have any problems feel free to contact us.

Global Light Group Node

New Feature
Version 1.1

Just like the Light Group Node, the Global Light Group Node will restrict the lights available to nodes that are upstream (connected to the inputs).

The main difference being that the Global Light Group Node shares the same light group with all other instances of the node as well as the main shaderMeister Surface node (if **Use Global Light Group** is enabled).

The global light group can be edited either from the Global Light Group Node itself or the main shaderMeister user interface via the **Edit Global Light Group...** button.

The global light group itself is only saved via the shaderMeister master plugin. This implies that the global light group is saved with the scene and not with any objects.

If no global light group exists then the Global Light Group node will automatically pass through.
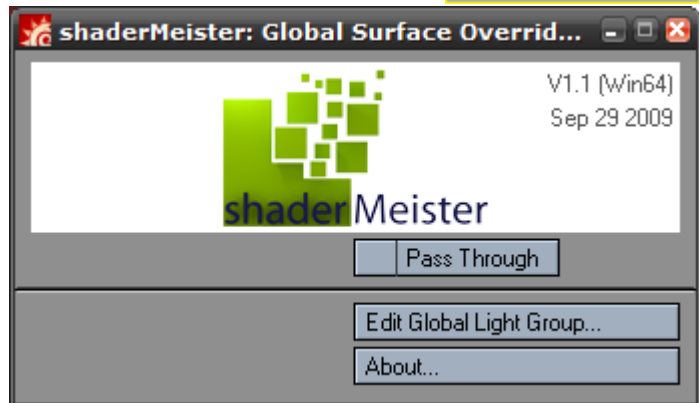
Pass Through

If you enable this control then the light group will not be used upstream of this node.

Effectively the Colour input will go straight into the Colour output, the Material input will go straight into the Material output.

Edit Global Light Group...

This opens a new panel that is virtually identical to the options present in the Light Group Node.
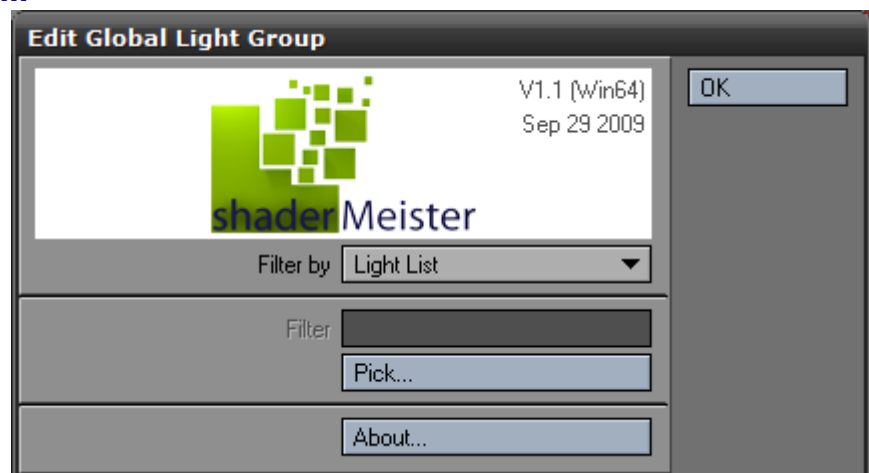
Get Integer / Scalar / Colour / Vector Comment Node

New Feature
Version 1.1

These nodes allow you to extract an Integer / Scalar / Colour of Vector value from the item currently being shaded.

One usage example would be tweaking the settings used by an ambient occlusion node on a per item basis for an ambient occlusion pass.

If the current item has no valid comment, then a default value will be used (or the value connected to the nodes input).

The values stored in the item comments may either be valid for the current item only, or for the item including all children.

Syntax

The syntax for item comment variables is as follows:

#type:name = value;

or

*type:name = value;

starts a variable definition that is valid only for the current item.

* starts a variable definition that is also applied to child items (unless they have their own variable of the same name, which overrides the previous one).

: separates the type from the name of the variable

= separates the name of the variable from the value

; denotes the end of the variable, anything following the semi-colon in the comment is ignored

Type can either be **i**, **s**, **c** or **v** for Integer, Scalar, Colour or Vector variables respectively.

The type is not case sensitive and only the first letter of the type is evaluated. This allows for **i**, **I**, **integer**, **Integer** or even **idunnowhattosay** as the type for an integer variable.

The **name** is not case sensitive either. It may also contain spaces, but (obviously) neither the **:** nor the **=** character.

Value is a single number in case of an Integer or Scalar variable, a comma separated group of three numbers in case of either a Colour or a Vector value.

Examples

#s:Luminosity=0.5;

This defines a scalar variable that is called Luminosity and has a value of 0.5 for this item only.

*Color : BaseCoat = 1.0, 0.0, 1.0;

This defines a colour variable called BaseCoat which is set to purple[8] for this item and all its children.

#c:basecoat = 0.0, 1.0, 0.0;

This defines a colour variable called basecoat which is set to green for the current item. Since variable names are **not** case sensitive, this would overwrite the previous BaseCoat variable for the current item. They thus have the same variable name as far as the Node is concerned.

*V:up_and_away = 0.0, 1.0, 0.0;

This defines a vector variable named up_and_away for the current item and all children, the value is 0.0, 1.0, 0.0 (X, Y, Z).

---
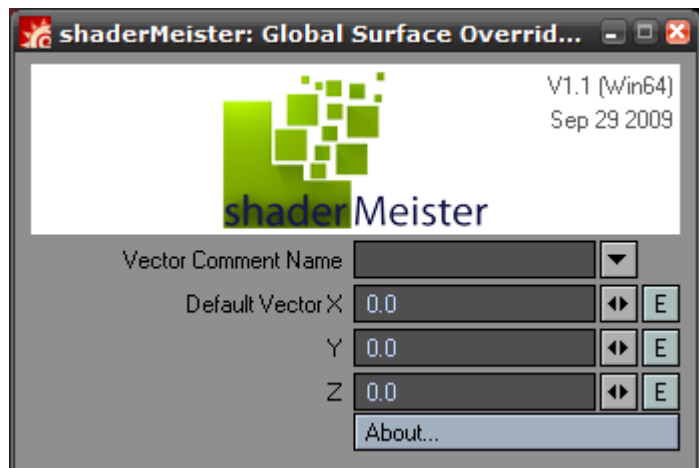
8   The three components are Red, Green, Blue in that order.

Examples

We've collected a few simple examples do give an idea of the possibilities that shaderMeister allows for.

These are available as Content Presets in the shaderMeister Sample scene which is included with the plugin.

Just load the sample scene, open shaderMeister, select any of the Content Presets and press F9 to render.

The image on the right is the sample scene without shaderMeister.
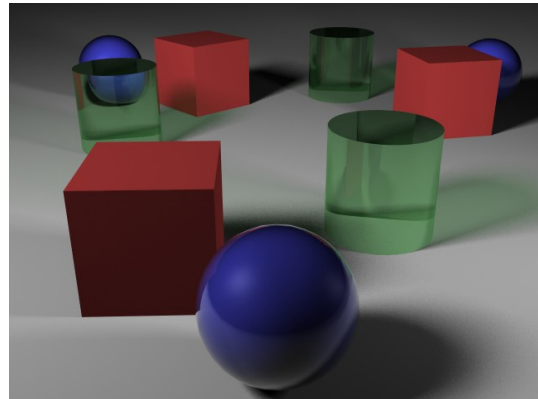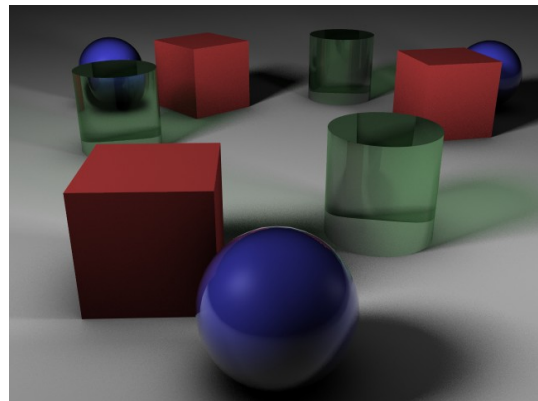
Ambient Occlusion Pass

This preset just renders a standard ambient occlusion pass to be used in compositing.
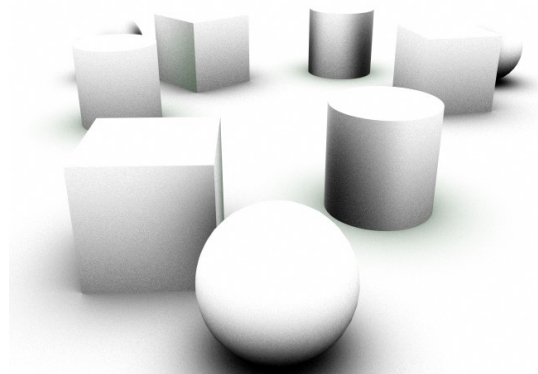
Global Ambient Occlusion

This preset mixes an ambient occlusion shader with all existing surfaces.

Please note that any nodal surfaces will override this preset.

Lit Ambient Occlusion Pass

This preset is a variation of standard Ambient Occlusion that takes normal lighting into account.

Lit areas are less affected by ambient occlusion.

This is designed to be used in compositing.
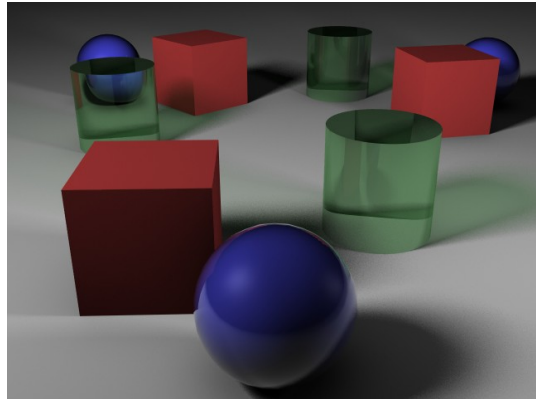
Courtesy of Guillaume Wyatt
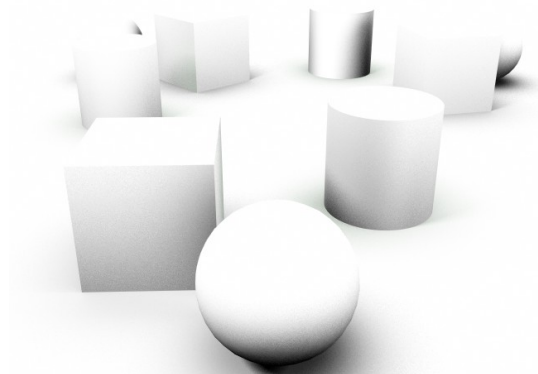
Global Lit Ambient Occlusion

This preset is a variation of standard Ambient Occlusion that takes normal lighting into account.

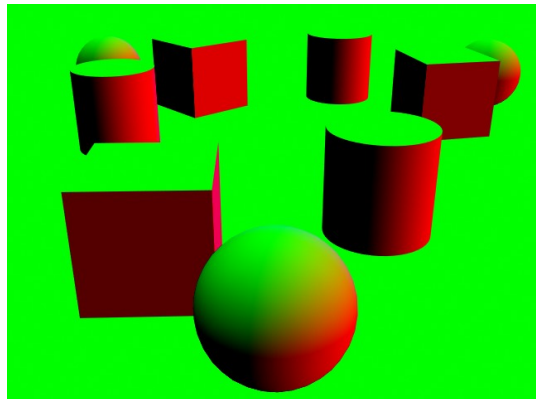This preset mixes the lit occlusion with the existing surfaces.

Courtesy of Guillaume Wyatt

Normal Pass

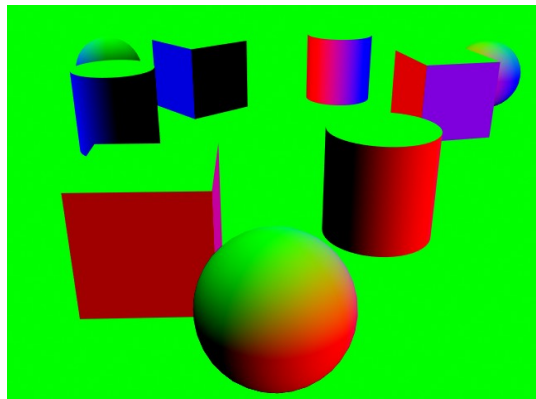This renders an image where the surface normals are encoded in the RGB channels.

Courtesy of Guillaume Wyatt

Object Space Normal Pass

This renders an image where the surface normals in object space are encoded in the RGB channels.
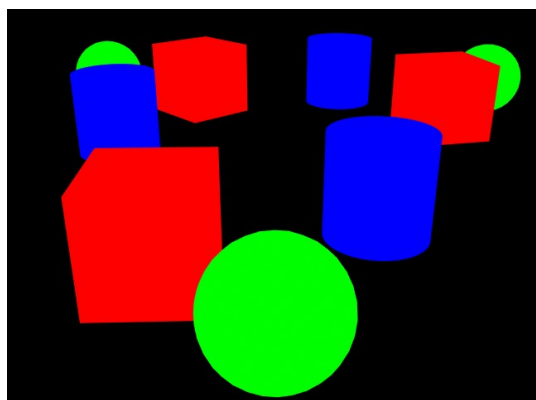
Courtesy of Guillaume Wyatt

RGBMatte

This preset assigns either red, green, or blue to the surfaces. This allows as compositing application to use a single RGB image as a matte for three different selections.
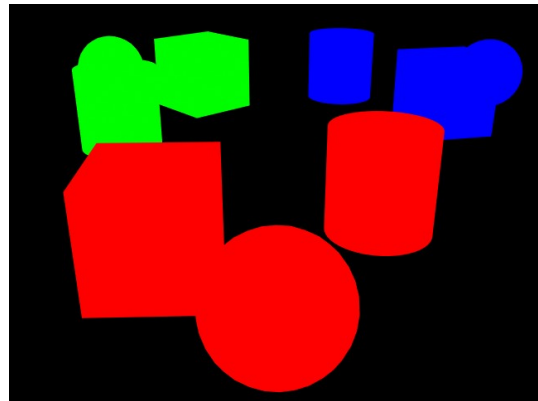
In a way this is like three alpha channels combined into a single image.
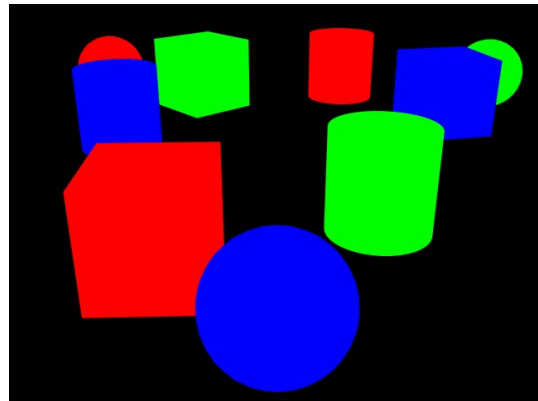
Courtesy of Guillaume Wyatt

RGBMatte using an Item Hierarchy

This is the same as the RGB Matte, in this case the parent Null objects are selected for the Matte in the Filter Nodes with the "Include Children" option selected.

RGBMatte using Comments

This is another variant of the RGB Matte, this time using Item Comments to filter the surfaces.
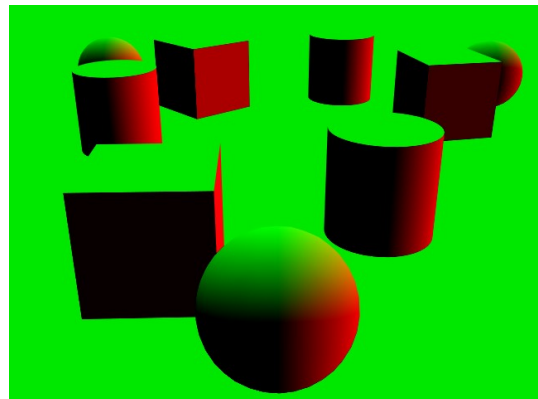

Camera Normals

This renders an image where the surface normals relative to the camera are encoded in the RGB channels.
"Not quite useful for relighting as the normals "stick" to the camera even when it orbits around the object, but has its uses for fake fresnel effects (blue channel) or as a displacement input (red and green channels) for a fake refraction effect."
Please note, if you apply this preset you are likely to change the current camera in the Item Node.

Courtesy of Guillaume Wyatt

Camera Normals, remapped

This renders an image where the surface normals relative to the camera are encoded in the RGB channels.
The result is remapped to the range from 0 to 1.0 – with 0.5 being the equivalent to 0. This is useful for compositing applications that can't handle HDR images.
Please note, if you apply this preset you are likely to change the current camera in the Item Node.

Courtesy of Guillaume Wyatt
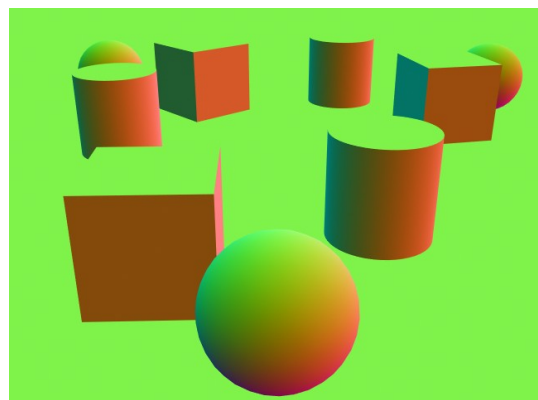
Appendices

Network Rendering

shaderMeister is ScreamerNet compatible and allows for network renders. You can install the plugin on any render node and it will render as expected. It has been tested with a variety of third party render controllers, including Butterfly NetRender and Spider.

Known Limitations

Incompatible with FPrime

Due to LightWave 3D SDK limitations FPrime can't render shaders. As these are crucial for shaderMeister, FPrime isn't able to render any modifications due to shaderMeister.

Incompatible with HDInstance

For the same reason as the FPrime incompatibility, shaderMeister won't be visible on items that are rendered by HDInstance.