



User Manual

Version 1.6

©2005-2010 by db&w Bornemann und Wolf GbR

# 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 PURCHASED 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, DOWNLOADING, 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 infiniMap plug in for LightWave 3D and documentation for infiniMap.

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

## 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 infiniMap

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, reproduction, 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

© 2005-2010 db&w Bornemann und Wolf GbR. All rights reserved.

All brands, company names, and product names are trademarks or registered trademarks of their respective holder.

### 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.

### IP Acknowledgements

This plugin uses the ECW JPEG 2000 SDK by ER Mapper, some parts of this SDK are based on 3rd party open source libraries and projects:

#### 1) TinyXML - XML parsing for GML Geolocation Metadata

TinyXML is distributed under the zlib license. [www.sourceforge.net/projects/tinyxml](http://www.sourceforge.net/projects/tinyxml)

#### 2) LittleCMS - ICC Profile Management library - <http://www.littlecms.com>

LittleCMS is distributed under the MIT license:

##### The MIT License

Copyright (c) 1998-2003 Marti Maria Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

#### 3) J2000 - <http://www.j2000.org/>

The T1 and MQ code in the ECW JPEG 2000 SDK is based in part on highly modified versions of sections from the J2000 library From the J2000 website: "The J2000 codec was written in an effort to produce the cleanest and simplest implementation possible of the JPEG-2000 standard. We have put a particular emphasis on good architecture design and code simplicity, while at the same time providing an implementation as complete and efficient as possible. The source code for the codec is freely available for anyone to study or even for use in commercial programs. We hope that our open development process and our focus on clean, straightforward code will help make the J2000 codec become a reference implementation of the JPEG-2000 standard." J2000 Copyright (c) 2001-2002, David Janssens All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS 'AS IS' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 4) OpenEXR by ILM

OpenEXR: Copyright (c) 2002, Industrial Light & Magic, a division of Lucas Digital Ltd. LLC All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Industrial Light & Magic nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 5) zlib

zlib general purpose compression library

version 1.2.3, July 18th, 2005

Copyright (C) 1995-2005 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Jean-loup Gailly [jloup@gzip.org](mailto:jloup@gzip.org)

Mark Adler [madler@alumni.caltech.edu](mailto:madler@alumni.caltech.edu)

## 6) LightWrap++, a C++ wrapper for LightWave

Copyright (c) 2007-2010, db&w Bornemann und Wolf GbR All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of db&w GbR nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## **Acknowledgements**

### **Software Design and Development**

Michael Wolf

### **Management**

Dagmar Bornemann

### **Documentation and Tutorials**

Michael Wolf

### **Product Marketing**

Dagmar Bornemann

### **Original Logo Design**

Matt Gorner

Special thanks go to our team of dedicated beta testers. You've all done a great job and your feedback, time, criticism and bug hunting skills were invaluable.

Extra kudos to Phil for proof reading the manual!

We would also like to thank the teams at NewTek, NewTek-Europe, East View Cartographics and gadget for their kind support.

We also thank the LightWave community and the members of the LightWave Plugin Developers mailing list, especially Marvin Landis and Ernie Wright for their insight into the more obscure depths of the LightWave SDK.

---

db&w Bornemann und Wolf GbR  
Seyfferstr. 34  
70197 Stuttgart  
Germany

[www.db-w.com](http://www.db-w.com)  
[info@db-w.com](mailto:info@db-w.com)

# Table of Contents

---

<b>Introduction</b>	<b>1</b>
Compatibility.....	1
Features.....	2
Feature Comparison.....	3
Current Limitations.....	3
Important note for users of earlier versions of infinimap.....	3

---

<b>Setting up infinimap</b>	<b>4</b>
Installation.....	4
OSX UB only.....	4
Registration.....	5
Included Plugins.....	5

---

<b>Reference</b>	<b>6</b>
Common User Interface Elements.....	6
infinimap Procedural Texture.....	15
infinimap Shader.....	16
infinimap Node.....	17
infinimap Extended Node.....	18
infinimap Normal Mapper Node.....	19
infinimap Browser.....	21
infinimap Converter.....	22
infinimap Converter (standalone plugin, Batch).....	26

---

<b>Appendices</b>	<b>28</b>
JPEG 2000 and ECW conversion tools.....	28
OpenEXR related tools.....	28
Tips & Tricks.....	28
Network Rendering.....	29
Known Issues.....	29
LScript.....	29
Glossary of Used Terms.....	30
FAQ.....	30

---

<b>Alphabetical Index</b>	<b>31</b>
---------------------------	-----------

## Introduction

Even in the age of 64-bit computing RAM is a limited resource. While it is quite common to have workstations with gigabytes of RAM, images used to texture in 3D applications have grown in size as well. In certain situations image sizes of tens or hundreds of gigabytes are quite common. Higher render resolutions for HDTV, film or print also require highly detailed, high resolution image maps for texturing.

InfiniMap is a LightWave 3D plugin that allows the user to render virtually unlimited<sup>1</sup> size textures within a familiar interface.

This is accomplished by only loading the parts of an image into memory that are actually visible, in the resolution needed for the final image. To determine the optimum resolution of the image need for the render, infiniMap uses a number of variables, including for example the distance of the mesh to the camera.

We have rendered animations of the earth using more than 3GB of images in video resolution using only 300MB of memory, including the memory used by LightWave 3D, in the process. We have also rendered images of tens of gigabytes using infiniMap on computers with 500MB of RAM successfully.

A smart caching system makes sure that the performance hit on render times is minimal, especially when rendering out animations. The loading of scenes with infiniMap images is much quicker than using native images.

It has been our goal to integrate infiniMap as tightly into LightWave 3D as possible and to adhere to existing standards to make using it a smooth experience. If the user interface differs from the original LightWave 3D image layers, this is due to the SDK unfortunately not allowing for a 100% accurate replacement of the user interface.

The currently supported image file formats are ECW (from ER Mapper) as well as JPEG 2000 and tiled OpenEXR images.

You can use infiniMap to create ECW, JPEG 2000<sup>2</sup> or tiled OpenEXR images using our new batch converter or one of the tools listed in the appendix of this manual.

## Compatibility

infiniMap requires at least LightWave 3D 9.0 and has been tested with LightWave 3D up to version 9.6.1.

The Extended Node requires at least version 9.2 of LightWave 3D.

It currently runs with the Windows 32bit and Windows 64bit versions of LightWave. It has been tested with Windows 2000 as well as the 32bit and 64bit versions of Windows XP and Windows Vista.

### New Feature Version 1.6

The OSX Universal Binary has been tested with OSX 10.4 and 10.5 on both PPC and Intel processors. At least LightWave 9.6.1 is required to run it in 64-bit mode.

Upon demand we can also port a Linux render node version.

Please visit [www.infinimap.com](http://www.infinimap.com) for more up to date information.

<sup>1</sup> Well, not quite, the current limit is an image size of 4,294,967,296 x 4,294,967,296 pixels.

<sup>2</sup> Currently the OSX UB version does not support ECW or JPEG 2000 images.

## Features

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

- Full compatibility<sup>3</sup> to LightWave 3D's surfacing, allowing for an easy replacement of native image texture layers with infiniMap texture layers.
- Works as a shader, a procedural texture layer and a node
- Compatible with HDInstance
- A user interface that remains as close as possible to the native image texturing interface provided by LightWave 3D, resulting in a minimal learning curve.
- Complete support for multi-threading to make full use of multi-processor, multi-core and/or hyper-threading computers.
- Low memory footprint.
- Shared image caches: Multiple instances of the same image use a shared cache, reducing the amount of memory required during rendering for scenes that re-use identical images (or single channels thereof).
- FPrime support.
- Useful tools to make working with infiniMap images easier, such as an automatic OpenGL preview layer generation tool to make infiniMap images visible in Layout.
- Free network render nodes.
- Unlicensed plugins can be used to load, render and save infiniMap enabled scenes, there is no need to purchase a license for every copy of LightWave 3D in your facility<sup>4</sup>.
- ScreamerNet network rendering support, of course.
- Convert images to the ECW/JPEG 2000 and OpenEXR format from within LightWave 3D
- Replace images globally
- Simple Colour Correction for every plugin Instance
- Support for JPEG 2000 images with up to 28 bits per channel (a limitation of the JPEG 2000 standard).
- Free point upgrades, free support, free beer<sup>5</sup>
- 64-bit OSX version of infiniMap EXR
- A dedicated node for normal mapping
- Native support for PNG files to lower the memory requirements when converting
- Gamma support in the image converter

---

**3** As far as the SDK allows that is, unfortunately there are still some missing hooks. Limitations are detailed in the sections describing the actual plugins.

**4** Even though we won't complain if you do. We do offer volume pricing as well as site licenses.

**5** O.k., so we're joking about the beer...

## Feature Comparison

There are three versions of infinimap available.

infinimap Le is the free Version.

All versions support loading, saving and rendering scenes and objects created with any of the other versions<sup>6</sup>

Pro and Exr functionality is unlocked by the license file we send out after purchase.

	Le	Exr	Pro
Texturing plugins	Surface Shader	Surface Shader Texture Layer Texture Node Multi-Channel Node	Surface Shader Texture Layer Texture Node Multi-Channel Node
Projection types	Planar only	All image projections supported by LightWave 3D. The Texture Layer does not support UV mapping, the Shader and the Node do.	All image projections supported by LightWave 3D. The Texture Layer does not support UV mapping, the Shader and the Node do.
Native image format	Tiled OpenEXR JPEG 2000, ECW	Tiled OpenEXR	Tiled OpenEXR JPEG 2000, ECW
Supported platforms	Windows 32/64-bit, OSX UB 32/64-bit <sup>7</sup>	Windows 32/64-bit, OSX UB 32/64-bit	Windows 32/64-bit
Conversion options	Single Image converter. Convert on load if the source image is not native to infinimap.	Batch image converter. Convert on load if the source image is not native to infinimap.	Batch image converter. Convert on load if the source image is not native to infinimap.
Supported input image formats by the converter.	All image formats that LightWave 3D can read using plugins (not including: Targa, FLX and IFF), PNG OpenEXR ECW, JPEG 2000 (not available on OSX)		

## Current Limitations

- Only supports ECW, JPEG 2000 and tiled OpenEXR images natively. The OSX UB does not support ECW / JPEG 2000 images.

## Important note for users of earlier versions of infinimap

Please re-add the infinimap.p plugin file to LightWave using the Edit Plug-Ins panel. New plugins have been added which will only be visible if you add the plugin again.

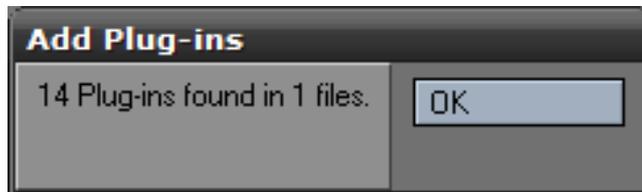
<sup>6</sup> With the exception of the OSX UB version which does not support ECW/JPEG 2000 images

<sup>7</sup> The OSX UB version does not support ECW / JPEG 2000 images

## Setting up infiniMap

### Installation

The *infiniMap.p* / *infinimap.plugin* 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 *infiniMap.p* to LightWave 3D (The OSX UB version is named *infiniMap.plugin*. LightWave will prompt you that it found 16<sup>8</sup> plugins in one file.

To be able to use all features of the plugin you will need to register it with us and then activate it with the activation code we send to you.

### OSX UB only

**New Feature**  
**Version 1.6**

The Converter relies on the image plugins that ship with LightWave3D. It tries to scan the plugin configuration file used by LightWave3D to find the actual plugins.

The mechanism for the plugin configuration file has changed with the UB version of Lightwave3D, plugins that are shipped with LightWave3D don't show up in the file any more. This means that the infiniMap converter may not find the image plugins any more.

To work around that, please add the image i/o plugins that ship with LightWave3D manually to your Lightwave3D installation once. They will then be listed in the appropriate configuration file and infiniMap will be able to find and use them for the conversion process.

<sup>8</sup> Yes, unlike the screenshot suggests it really is 16 plugins by now.

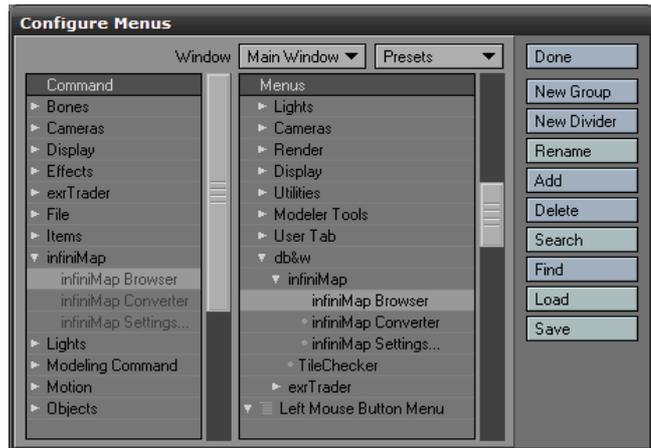
### Adding the helper plugins

Three of the included plugins can be added to the Layout menus.

Use Edit->Edit Menu Layout... (alt-F10) to open the Configure Menus panel in Layout.

On the Commands list on the left there is a group called infiniMap which contains three commands to launch the infiniMap Browser, the Converter or to apply infiniMap Settings<sup>9</sup> to the current Scene.

You can of course use Edit->Edit Keyboard Shortcuts... (alt-F9) to assign these commands to keyboard shortcuts.



### Registration

Once you've installed the plugin, open any object and either apply the infiniMap procedural texture, the shader, or either of the nodes to the surface. The interface controls will be disabled except for a button asking you to register.

You may also launch the Converter or the infiniMap Browser.



Clicking on the button will open the Registration panel as shown on the left. Enter your full name, company name (if applicable) and e-mail address. Export the registration and attach it to an e-mail to registration@infinimap.com . We will mail you an activation code within 24 hours after payment is received, depending on the time difference.

You can now import the keyfile and press OK to complete the licensing process.

The registration file is stored in the same directory where the plugin resides. 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 manage multiple licenses allowing for a single registration file on your network share.

### Included Plugins

infiniMap is actually a suite of plugins for LightWave 3D. Currently it consists of the following plugins:

- infiniMap Texture Layer
- infiniMap Shader
- infiniMap Node
- infiniMap Extended Node
- infiniMap Browser
- infiniMap Converter
- InfiniMap Settings

<sup>9</sup> Currently the infiniMap Settings only limit the maximum amount of memory used to cache ECW / JPEG 2000 images. It has no effect if you use OpenEXR images.

## Reference

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

### Common User Interface Elements

Both the procedural and the shader use many common elements in their user interface<sup>10</sup>.

The common user interface elements used by both plugins, from top to bottom, are:

- Projection
  - Planar, Cylindrical, Spherical, Cubic, Front, UV (not available in the texture layer).
- Depending on the type of projection, these additional controls may also be visible
  - Fixed, Time, Reference Camera (Front projection)
  - UV Map (UV projection, not available in the texture layer)
- Image
  - Image selection
  - Layer selection
  - Channel selection
  - Image Preview Area
- Options
- Texture Quality
- LoD Blending
- UV Tiles, UV Tiling and UV Offset to modify the projection.
- Editing (colour correction)

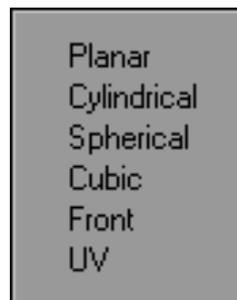
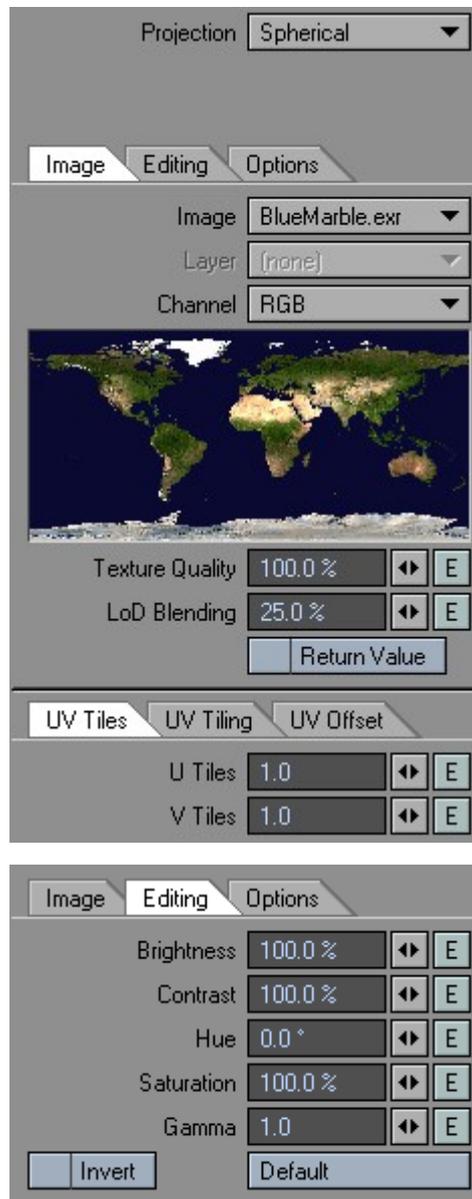
These are explained in more detail in the following sections.

### Projection Types

infiniMap supports all projection types that LightWave3D provides to normal image maps. Unfortunately, due to limitations within the SDK, UV mapping is only supported by the infiniMap shader and nodes.

The projection types are:

- Planar projection
- Cylindrical projection
- Spherical Projection
- Cubic Projection
- Front Projection (also known as Camera projection)
- UV Map (not available in the procedural texture)



<sup>10</sup> Actually, to a large extent they are based on the same source code as well.

Each one of these behaves exactly like the matching image projection type in LightWave 3D. infinimap also supports all the options, such as the UV tiling amounts and UV wrapping modes etc.

Please refer to your LightWave 3D documentation for more information on the projection types.

### Image Selection

This pop-up menu works just like the image selector in the image texture layer provided by LightWave.

You can use it to select any of the currently loaded infinimap images or load a new one.

If you attempt to load an image that can't be used natively by infinimap, but can be converted using the Converter, infinimap will open the Converter for you. You can then change the settings, convert the image and the result will be loaded into infinimap.

Loaded images actually do not use any memory until a render process is started, this can be a preview (including the preview sphere in the Surface Editor), a final render or a FPrime render.

infinimap keeps track of the usage of the images. If no plugin actually uses one of the images in this list, the image is automatically removed from the list.<sup>11</sup>

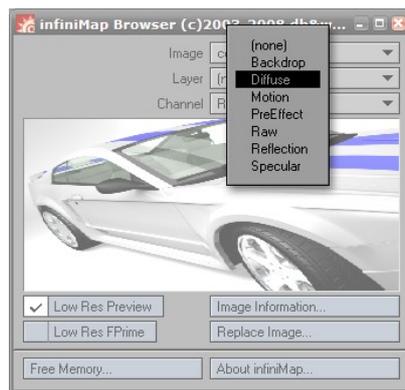
infinimap EXR will not allow you to pick a ECW/JPEG 2000 image, but you can still load, save and render objects that use ECW / JPEG 2000 images (except for the OSX version).

### Layer Selection

OpenEXR can store multiple layers in a single image file.

If the selected image contains more than one layer you may pick one of them using the Layer Selection.

The Extended Node even allows accessing multiple layers at once.



### Channel Selection

Unlike LightWave 3D, infinimap allows you to specify which channel of an image you wish to use for texturing. In some instances it may be useful to just apply the red channel as a bump map for example. This feature also allows you to just use the alpha channel of a RGBA image to mask out other textures for example.

The channel selector has to do a bit of guessing as to how to interpret the bands/channels stored in an image file. If there are 3 bands, it assumes them to be RGB in that order, if there are four it assumes that the fourth band is the alpha channel. Single channels are always available as luminance / grey scale values.

Images with three or more channels, such as RGB images, will also be able to output two additional "pseudo" channels that infinimap can generate on the fly. These are:

- **Average** – Just the average of the Red, Green and Blue Channel



<sup>11</sup> As always, there is an exception to the rule: Procedural texture layers, when copied using the 'Copy' button, still hold on to the infinimap image they use. So you may notice an image being in the list that you're sure isn't being used, because you copied an infinimap procedural texture layer that uses this image. No worries, next time you 'Copy' a texture layer, or restart LightWave 3D, it will be gone.

- **Luminance** – The Red, Green and Blue Channel balanced to produce the perceptual Luminance based on the SMPTE standard.

### Image Preview Area

If you have an image loaded, the image preview area will display a small thumbnail of the image.

If you have no image loaded, the image preview area will display the infiniMap logo as well as the version number of infiniMap.

Note: Due to a glitch in the LightWave 3D SDK, in Modeler the image preview area is not available for the procedural texture.

### Options

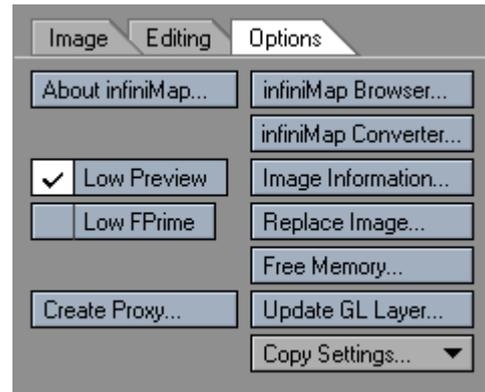
If you select the Options tab you will be able to access some bonus functionality of infiniMap.

The options visible depend on the plugin type, some options may not always be available.

Also, some options may only be visible if an image is actually selected in the Image Selector.

The Options are available in the procedural texture and the shader.

In the procedural interface some of the option names had to be shortened due to space constraints.



### About infiniMap... / About infiniMap...

This displays a panel with the credits for infiniMap, as well as the version number and the date the plugin was created.

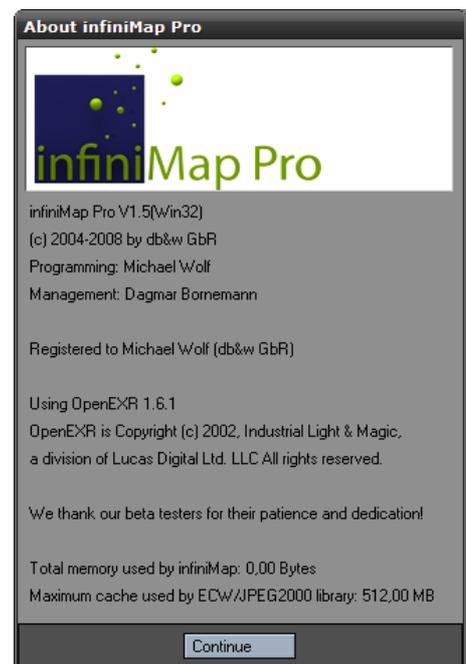
The version information is quite handy when you require support and check for updates.

As an additional note, we would once again like to thank our trusty beta testers for their patience and support.

The “About infiniMap” panel will also display the total amount of memory used by all images handled by infiniMap. This is especially handy if you’re working with FPrime.

It will also display the maximum amount of memory that the libraries used to handle JPEG 2000 and ECW images **may** use to speed up access to the image files on disk. This value is adapted to your actual RAM usage. The more available memory you have, the more memory **may** be used and vice versa.

Please note: this is a maximum value, and infiniMap is likely to actually use lot less than that.



### Open infiniMap Browser... / infiniMap Browser...

This is a quick way to access the infiniMap Browser from within the Shader or the Procedural Texture plugin.

### Open infiniMap Converter... / infiniMap Converter...

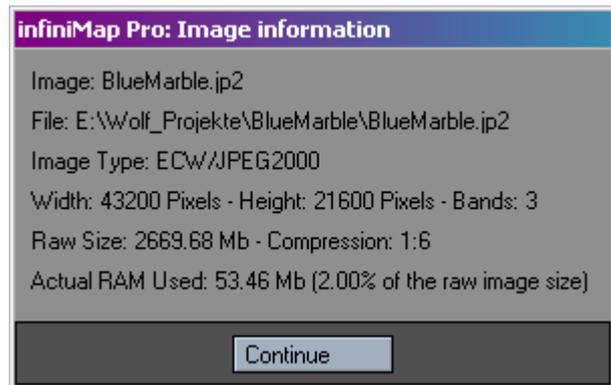
This is a quick way to access the infiniMap Converter from within the Shader or the Procedural Texture plugin.

## Show Image Information... / Image Information...

If the user interface displays a preview image, this option displays more verbose information about the loaded image.

This information includes:

- The file name and full path of the image
- The image format (currently only ECW, JPEG 2000 or OpenEXR)
- The width and height of the image in pixels, as well as the number of bands (or channels) included in the file
- The theoretical raw size of the image if it was uncompressed as well as the compression factor of the file.<sup>12</sup>
- The amount of RAM/memory currently used by the image. Please note, since images may be used multiple times, the memory is only used once per image.



## Replace Image...

This option allows you to globally replace the currently visible image. A file requester will open to allow you to select a new infinimap image.

## Update/Create GL Layer... / Update GL Layer...

Currently LightWave 3D cannot display infinimap applied images in the OpenGL preview. We have included an automated way to create what we call OpenGL Preview Layers, that reflect the state of infinimap using standard LightWave 3D texture functions. These preview layers are only visible in the OpenGL view ports and are hidden during renders.

OpenGL Preview layers are not visible if GLSL is turned on in the OpenGL options of LightWave 3D.

This menu item allows you to create a preview layer, or, if you have already done so, updates the parameters of the preview layer to those of the current instance of infinimap.

infinimap identifies OpenGL preview layers by the name of the image file used by the LightWave 3D image texture. If it contains the name of the base infinimap image (without the file name extension in both cases) it will assume that the layer is an OpenGL preview layer.

As an example:

Assuming your infinimap image is BlueMarble.jp2, infinimap will accept *Proxy\_BlueMarble.tga* or *BlueMarble\_myProxy.bmp* as a valid name for the proxy image, but not *My\_Blue\_Marble\_Proxy.tga*.

Please note: For the creation of an OpenGL proxy layer to work in the shader, you will have to manually activate the texture layers of the surface (just click once on the [T] button next to the base colour of the Surface). Unfortunately infinimap can not automatically do this yet due to limitations of the LightWave SDK.

### Creating an OpenGL Preview Layer

If no preview layer exists, infinimap will display the Create OpenGL Preview Layer panel allowing you to create a preview layer as well as a preview image.

<sup>12</sup> Please note, the raw image size does not signify how much memory infinimap will actually use during a render.

**Proxy Width  
Proxy Height**

Defines the size of the final image used by the OpenGL proxy layer. These sizes match the texture size options available in LightWave 3D.

**Size Presets**

These are some common presets for OpenGL preview textures.

**File Format**

This is the file format used to save the file to disk.

**Proxy File Name**

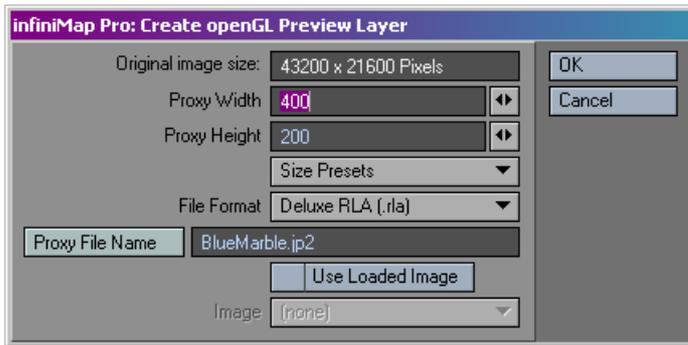
Allows you to change the name of the preview image file. Make sure that it contains the name of the infinimap image as stated further above. Currently you will also have to type in the correct file name extension manually (this makes no difference to LightWave 3D however).

**Use Loaded Image**

If you activate this button, instead of creating a new image file you can use an existing image file that has been loaded into LightWave 3D.

**Image**

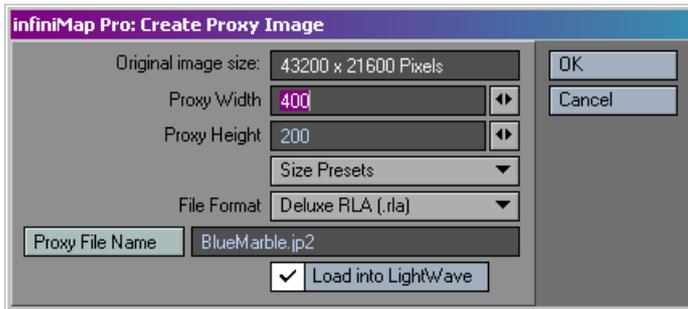
This item allows you to pick a loaded image to use as a preview image instead. Please note that the same naming rules apply for the image file for infinimap to recognize the OpenGL preview layer.



**Create Proxy Image... / Create Proxy...**

Sometimes you just wish to create a small version of the currently loaded infinimap image, or create a proxy layer manually.

Create OpenGL Proxy Image allows you to do that without actually creating an OpenGL proxy layer within LightWave 3D. The user interface controls have the same functionality as in the **Update/Create OpenGL Preview Layer** panel.



**Copy Settings from...**

This control is only available in the texture layer plugin. It allows you to copy the applicable settings of a LightWave 3D image layer to the current infinimap layer.

It will optionally also convert and load the image texture used by the original image layer.

This option should help to transition LightWave 3D texture layers quickly to infinimap layers.

**Low Res Preview / Low Preview**

Previews rendered with infinimap can take a while to start because infinimap needs to read the disk based image files for each rendering process.

This includes rendering the preview sphere in the Surface Editor and VIPER.

To speed up preview renders, you can turn on the Lores Preview option (which is active by default). It basically reduces the Texture Quality setting by a fixed amount, which in turn means infiniMap doesn't need to read as much data from disk for every preview render.

Final LightWave 3D renders are not affected by this setting at all. We recommend you leave Lores Preview on to maximize interactivity while surfacing. For fine details, or FPrime renders it should be turned off.

This menu items toggles low resolution previews for all instances of infiniMap in the current scene.

### **Low Res FPrime / Low FPrime**

This option is the same as the "Low Res Preview" option but for FPrime renders only. Please note that this settings affects FPrime final renders as well (using the "FPrime Render" plugin).

This setting is off by default, in case you want to quickly load a scene and render it with "FPrime Render".

### **Free Memory...**

infiniMap tries to manage memory as intelligently as possible, however there may be cases where this doesn't work as expected. If you work with FPrime extensively and preview on a single frame, but toggle the "Low Res FPrime" option a lot, or move the camera about, infiniMap may use more memory than it really needs to.

Normally it re-evaluates its memory usage when you change to a different frame but you can also use the "Free Memory..." option to tell infiniMap to release all memory.

### **Registration...**

This item opens up the license manager used during the installation. Once the plugin is activated you will not need to enter the license manager again, but we do provide this option to allow you to check the registration.

## **Texture Quality**

Texture Quality defines the resolution of the image texture when applied on the final render. While infiniMap tries to estimate the needed quality depending on a variety of factors, such as the size of the final render or the distance to the camera, this may produce inaccurate results in some cases. This settings allows you to tweak the resolution, and thus the sharpness of the image texture.

The higher the value, the more detailed the image texture will be (within the limits of the image resolution). infiniMap will also use more memory during rendering.

A lower value will blur the image texture more, but also lowers the memory usage.

In most cases the default value of 100% is appropriate, but you might want to tweak it after checking your final render or your memory usage.

## **LoD Blending**

When rendering, infiniMap creates a number of level of details (LoD), small parts of the image texture at different resolutions. These are applied to the surface depending on the Texture Quality setting and other factors such as the distance to the camera.

To smooth transitions from one level of detail (or image resolution) to another, infiniMap blends the level of details. LoD Blending specifies over what range this blending happens.

The higher the value, the more memory will be used, although the memory hit is not as significant as changing the Texture Quality.

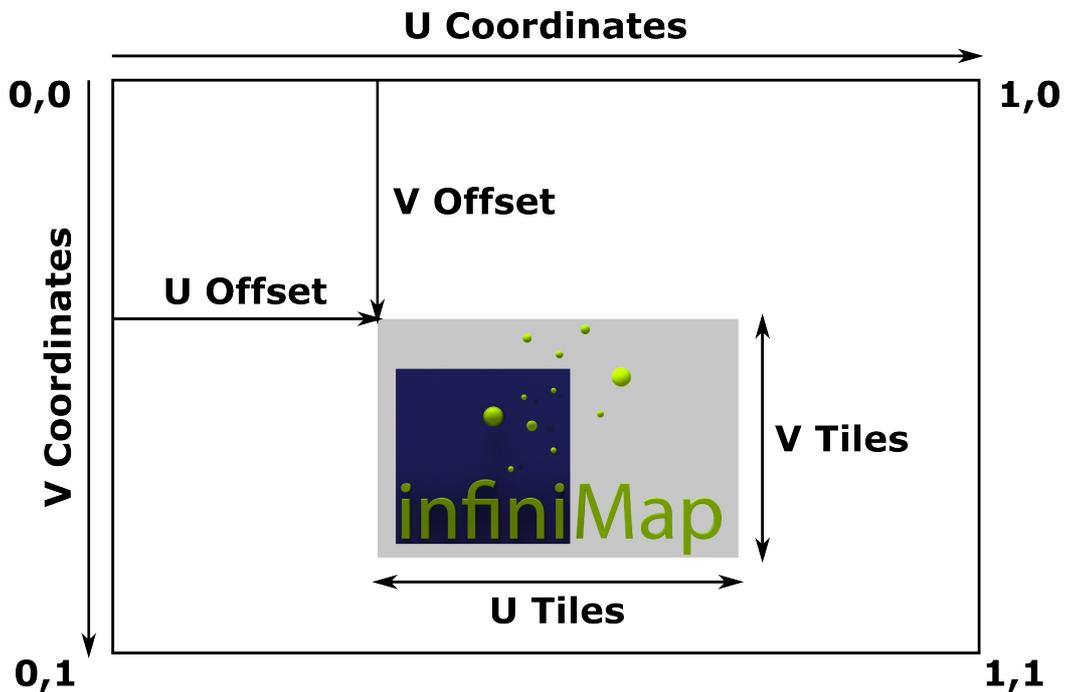
We recommend the default setting. If you can clearly see the different resolutions switch (this tends to be visible in animations, especially with a low LoD Blending setting) you should increase this value.

The useful range of LoD Blending values is between 0% and 100%.

### UV Modification

The UV modifiers allow you to change the UV coordinates before the image is applied to the surface.

These UV coordinates can either be created implicitly by the projection type (such as planar or spherical) or explicitly defined by a UV map if the UV Map projection type is used.



This diagram illustrates the relationship of the UV Offset and UV Tiles options in relation to the UV texture coordinate space.

#### UV Tiles

U and V tiles defines how many times the image is projected into the final UV space that ranges from 0 to 1.

A lower value means that the image will be smaller, a bigger value enlarges it. At a value of 1.0 the image will fit exactly into the UV space.



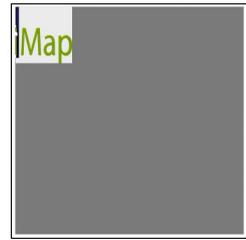
#### UV Tiling

The Tiling options define what happens with the image once it is repeated in UV space.

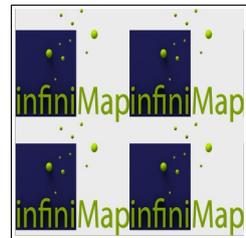
The sample images have been created with a planar map, using UV tiles of 2.0 respectively and an offset of -0.25 on U for the Reset and Wrap tiling option.



**Reset** The image is not repeated at all, outside of it's placement in UV space the image is not applied at all and the background colour will show through.



**Repeat** The image is repeated.



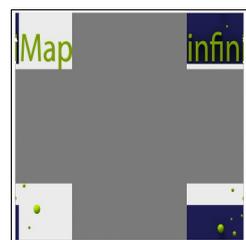
**Mirror** The image is mirrored and repeated



**Edge** The edge pixels of the image are repeated



**Wrap** The image is not repeated at all just like the Reset mode, but it will wrap around the UV coordinates.



**UV Offset**

The UV Offset allows you to position the image in UV space. The value is in relation to the original UV space and does not change if the UV Tiles changes.



## Editing (Colour Correction)

The Editing options allow for simple, on the fly colour correction that is applied to the infiniMap image while rendering.

The settings are similar to the ones supplied by LightWave 3D in the images panel.

### Brightness

Changes the brightness of the image. At 100% the image is not changed, values below that make it darker, values above 100% brighten the image.

### Contrast

This setting changes the contrast of the image. 100% is no change, anything below that lowers the contrast, anything higher (yep, you guessed it) raises the contrast.

### Hue

The Hue setting changes the Hue of the image, basically cycling the colours through the colour spectrum. It is measured in degrees. 0° is the default. A setting of 360° degree is a full circle and the same as 0°.

### Saturation

This changes the saturation of the image. 100% is the default value, at 0% the image will be greyscale only.

### Gamma

This allows you to add gamma to your image.

To “degamma” (i.e. remove the gamma of an image) you can type in  $1 / \text{gamma}$  – where gamma is the gamma of the image you'd like to neutralize. (i.e. If your image has a baked in gamma of 2.2, you can type in  $1 / 2.2$  which results in 0.4545 to remove the gamma and linearize the image).

### Invert

Invert inverts the colours of the image.

### Default

This resets the values to their default settings. Enveloped values can unfortunately not be reset due to SDK limitations.



## infiniMap Procedural Texture

The procedural component of infiniMap is probably the plugin you will spend the most time using.

Except for some special cases we recommend using the Procedural Texture instead of the shader. A few reasons are:

- The procedural texture is more versatile than the shader and can be used on any surface channel
- Unlike the shader the procedural texture is compatible with FPrime
- Procedurals seem to be faster when rendering compared to shaders.

Unfortunately the SDK provided with the current releases for LightWave 3D doesn't allow procedural textures to access UV Maps.

There are also limitations to the accuracy of the Texture Quality setting. As long as the Scale of the texture is correct and the textured object itself has not been scaled within Layout, the Quality Setting will be correct<sup>13</sup>.

A change of the size of the object in Layout will need a readjustment of the Texture Quality setting. The Quality has to be set to roughly the square of the scale factor of the mesh multiplied by 100 to get the percentage. For example, if the object has been scaled to <5.0, 5.0, 5.0> the Texture Quality should be 2500% ( 5 x 5 x 100).

**As a rule of thumb, do not scale an object in Layout that has the infiniMap texture layer applied to its surface.**

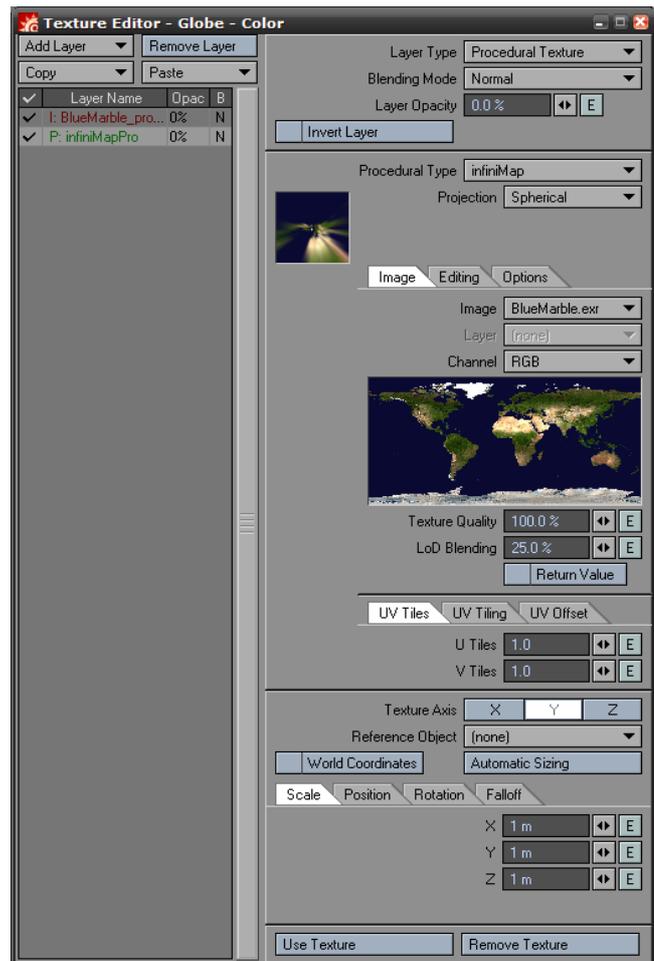
### Return Value

This option is only available on the procedural texture if the texture is applied to a colour texture layer (i.e. The texture layers that modify the colour of a surface).

It basically sends the image data differently to LightWave so the result can be used as the source value in a gradient. You should not have to activate it unless the infiniMap layer does not behave as expected in conjunction with other layers.

If you activate "Return Value" you will be able to enter a colour or percentage (depending on the texture layer type), just as you can with any other procedural texture.

Please note that toggling "Return Value" does not immediately update the user interface. You will need to select another layer and then select infiniMap again for the user interface change to show.



<sup>13</sup> Well, at least very close. Unfortunately the plugin can only estimate the optimum settings.

## infiniMap Shader

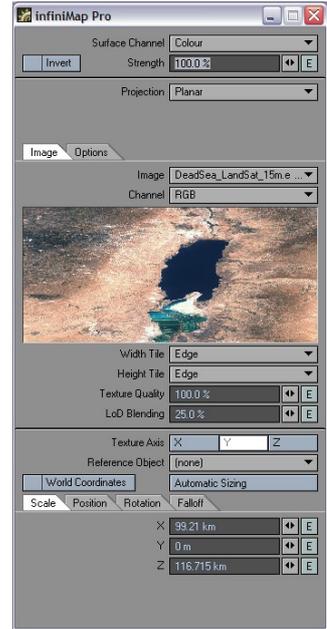
The infiniMap shader provides some additional functionality over the procedural texture, mainly UV mapping support which is unfortunately not possible within a procedural texture.

The shader can also estimate a better quality setting for the image. A value of 100% should be the optimum setting in any case.

Basically all user interface elements starting with the Texture Axis and below have been designed to work exactly the same way as their counterparts in the LightWave 3D image texture layers. Please refer to your LightWave 3D manual for more information on those options.

infiniMap also supports applying the image to other surface channels. The controls for that appear at the top of the interface.

The affected channel will also be displayed as the plugin description in the shader plugins list of the surface editor.



### Surface Channel

This will open a pop-up with all surface channels that can be affected by the shader.

### Invert

Just like the LightWave texture layers, you may invert the effect of the shader.

### Strength

The strength setting allows you to mix the channel value generated by infiniMap on top of the channel value defined by the surface settings.



## infiniMap Node

The infiniMap Node provides the same image mapping functionality as the shader, including support for UV maps.

All relevant inputs are exposed and evaluated once per shaded spot, allowing for a fine control of the parameters.

The outputs are derived from the selected Layer and Channel.

### Colour

This output applies the Opacity and the Blending mode to merge the infiniMap image on top of the Bg Colour.

If the infiniMap image contains an alpha channel, this will be applied while merging it onto the Bg Colour. (Similar to how image editing applications merge a layer with transparencies on top of other layers).

This behaviour is identical to the LightWave 3D image node.

### RGB

This output makes that raw RGB values of the image available.

Unlike the Colour output, no blending is being done.

### Alpha

This output provides the raw Alpha channel of the loaded image, if one exists in the infiniMap image.

### Normal

This output is the normal to be shaded by LightWave 3D.

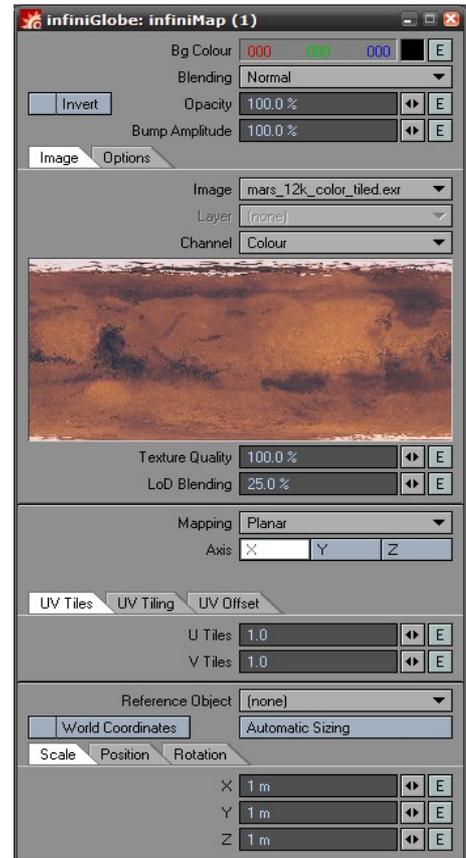
It is derived from the original surface normal being modified by the image as a bump map. Basically it is the Bump output subtracted from the original surface normal.

The Bump Amplitude affects this output.

### Bump

This is the change of the surface normal when interpreting the infiniMap image as a bump map.

This is affected by the Bump Amplitude.



## infiniMap Extended Node

The infiniMap Extended Node is designed for images that contain more than one layer (only OpenEXR images currently support layers).

The extended node works like the normal infiniMap Node with the exception of not allowing the image to be blended onto a background colour.

If an image contains more than one layer (Currently only OpenEXR images support layers) it can output all layers at once.

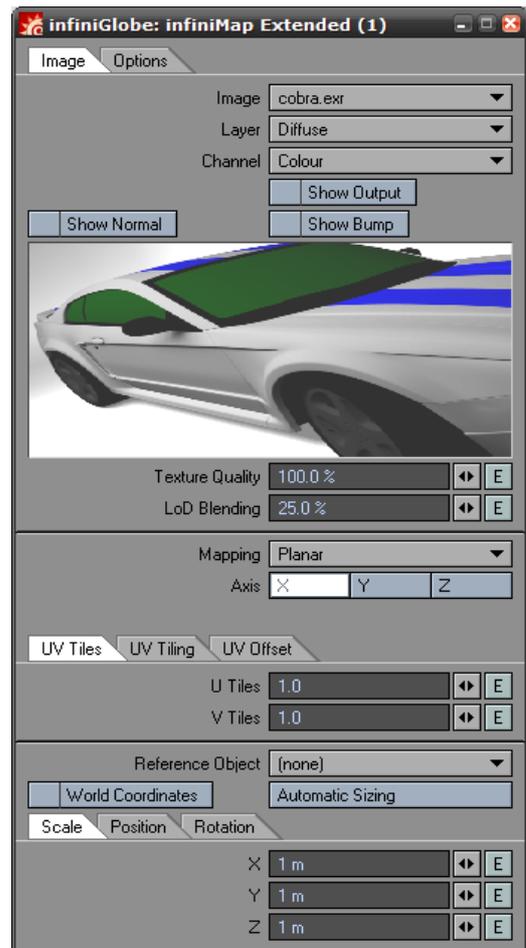
Instead of using multiple nodes to extract layered data from a single image, one can thus use only this node to extract all layers (provided that they all use the same projection).

The number of outputs and their names depend on the loaded image.

By default it exposes all Layers as well as the Bump/Normal output for the first layer.

By selecting the different layers/channels in the user interface and changing the settings of **Show Output**, **Show Normal** and **Show Bump** one can change which channels are actually exposed.

To create an image that can be used with the Extended node we recommend using ProEXR for Photoshop® to export a layered OpenEXR image. This will then be converted by infiniMap into a tiled OpenEXR image, maintaining the layers.



## infiniMap Normal Mapper Node

**New Feature**  
**Version 1.6**

The Normal Mapper Node has been designed for image maps that contain normals.

Normal maps basically store a direction vector. That's three components, X,Y and Z, each ranging from -1 to 1.

They are used to bend the surface normal when shading, resulting in a relief like change of the shading that is very similar to the effects of a bump map<sup>14</sup>.

### Type of Normal Map

There are two types of normal maps supported by this node. This setting changes the way that the normals stored in the image are applied to the current surface when rendering.

#### Object Space

Object space normals are normals stored in relation to the object. These are rarely used.

The advantage of an object space normal map is the fact that it does not require a projection. The downside is that it is a lot harder to evaluate on deformed meshes.

Basically, the XYZ components of the normal map relate to the XYZ coordinates of the mesh itself.

#### Tangent Space

Tangent space normals are the most common type of normal maps. These store the normal in relation to the current spot that is being shaded.

Tangent space normals require a mapping projection (commonly using a UV map) but handle deformations of the mesh well and are easy to evaluate.

In this case the XYZ components in the normal map are in relation to a (temporary) XYZ coordinate system that is valid for the currently shaded spot only. Z is equivalent to the original surface normal, X and Y are orthogonal to it.

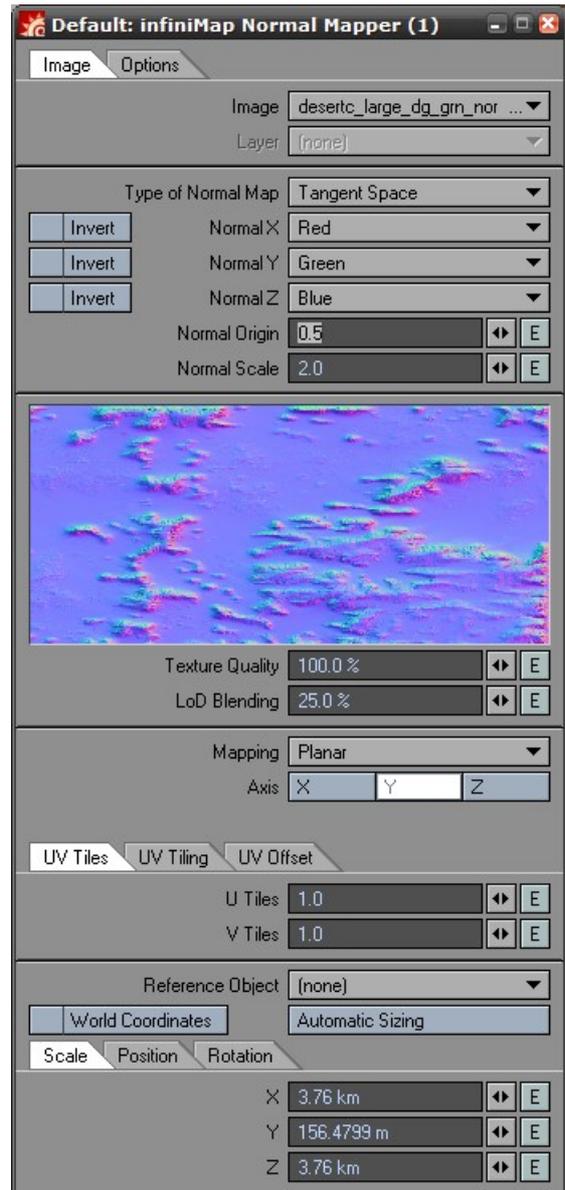
The other two components are derived from the mapping projection.

### Normal X, Normal Y, Normal Z

These controls allow you to assign any of the channels present in the current infinimap image to be used as X, Y or Z. You can also selectively invert the directions on a per channel basis.

### Normal Origin, Normal Scale

Since most image formats don't handle negative numbers, those values are stored in a range from 0 to 1 (or 0 to 255 in 8-bit pixel terms).



<sup>14</sup> A bump map is more or less a simplified variant of a normal map. When rendering, normals are derived from the bump map.

In this case 0.5 (or 128) corresponds to 0, anything lower is negative and anything above positive.

Some applications also store normal maps using a file format that supports negative numbers (such as OpenEXR), in this case a -1 in the normal map is a -1 in the file.

To cater for those, we've added the **Normal Origin** and **Normal Scale** options to bring what is stored in the image into the range from -1.0 to 1.0

For a normal map stored in the range 0..1, the default settings bring the values into the -1..1 range.

Basically:  $\text{value} = (\text{pixel} - \text{Normal Origin}) * \text{Normal Scale}$

For a proper float image with negative values, the settings would be 0 for the **Normal Origin** and 1 for the **Normal Scale**.

## infiniMap Browser

The infinimap Browser is a small generic plugin that you can keep open while you work in Layout. It allows you to have a quick glance at all infinimap images currently in use by the scene.

You can permanently keep this window open while working.

It also displays some of the options available in the shader, texture layer or nodes Options tab, allowing you, for example, to quickly change the quality settings for FPrime renders.

These options are explained in more detail in the Common User Interface Elements section.



## infiniMap Converter

The Converter will convert any image that LightWave can load using plugins<sup>15</sup> into an ECW/JPEG 2000 image, ready to be used within infinimap.

Due to licensing restrictions the Converter is limited to a maximum source image size of 500MB<sup>16</sup> when converting to ECW / JPEG 2000 images. InfiniMap Pro can read much larger images created in other applications though.

Converting to OpenEXR images is not limited in size.

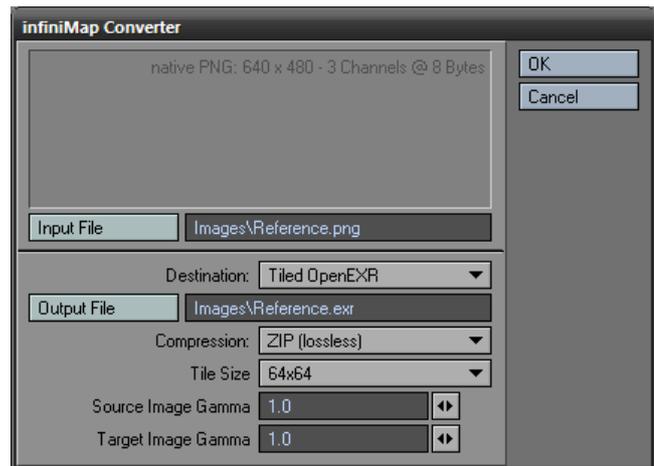
This converter panel will show up when you attempt to load an image that is not native to infinimap into one of the infinimap plugins.

Licensed users also have access to a standalone plugin batch converter, covered in the next chapter.

Depending on the context of the conversion and the license of infinimap, some of the controls may be missing.

The conversion settings are stored between sessions in the content directory (this allows for content/project specific conversion settings).

Here is a short description of the controls:



## Image Information

This area just displays some information about the source image, mainly the channels used (as returned by LightWave)

## Source

The converter can use two different types of source images.

### From File

You can load an image from a file. This can be a ECW, JPEG 2000 or OpenEXR image. The converter also supports loading any type of image that LightWave 3D provides a plugin for (this doesn't include Targa, FLX and IFF, which are native to LightWave 3D).

PNG images that are not interlaced are also loaded and converted natively by infinimap, reducing the memory usage tremendously compared to using the native LightWave 3D PNG loader plugin.

If you load an image that uses the LightWave 3D plugins, infinimap will need to create a temporary file first. This file can be rather huge (it is uncompressed) and stored in the default temporary directory of the computer. The file is deleted after conversion.

The conversion itself will use as little memory as possible, converting one scanline after the other. This conversion process is multi-threaded.

<sup>15</sup> Unfortunately this excludes TGA, FLX and IFF which are both loaded natively by LightWave 3D

<sup>16</sup> The 500MB refer to the raw, uncompressed size of the image. This can be computed using the following formula:  
width \* height \* number of channels \* depth per channel in bytes.

For example, an image that is 1920x1080 pixels large, contains 4 colour channels (i.e. Red, Green, Blue, Alpha) using 8 byte each (for 256 different shades per channel) would be:

$1920 * 1080 * 4 * 1 = 6220800$  bytes, which is 5.9 MB (divide twice by 1024 to convert from bytes to Megabytes).

Fortunately infinimap does that math and will warn you if the 500MB limit is exceeded.

If you pick an OpenEXR source image file, all layers and channels will be preserved.

### Loaded LightWave Image

You can also convert an image that is currently loaded into LightWave 3D. If you select this option you can pick one of the loaded images as a source for conversion.

### Input File

Here you can select a file to use as the source image if the Source is set to **From File**.

### Image

This pop-up allows you to pick an image loaded into LightWave 3D as a conversion source if the Source is set to **Loaded LightWave Image**.

### Destination

infiniMap can either work with ECW / JPEG 2000 images or tiled OpenEXR images natively. Only the Pro version supports ECW / JPEG 2000.

This option allows you to select the kind of output image and will then display the relevant options.

#### ECW / JPEG 2000 (Pro / Le only)

The file name extension of the output file determines if this option writes out a ECW or a JPEG 2000 image.

Call your file <filename>.jp2 to convert to a JPEG 2000 image.

Call your file <filename>.ecw to convert to an ECW image.

JPEG 2000 files are a standardized, cross-vendor image format that can be read by other applications as well. It can also supports lossless compression.

ECW on the other hand is an image format developed by ERMapper and is mainly designed for geodata. The compression is always lossy. However, it is slightly faster to compress and decompress compared to JPEG 2000.

#### Compression Ratio 1:

This slider allows you to set the compression ratio for the converted image. A higher value will producer smaller images. Since the compression is lossy, a higher value will also display more compression artefacts. A value between 5 and 20 is a good trade off between final file size and image quality.

JPEG 2000 supports lossless compression. Set the Compression Ratio to 1:1 to compress without loss.

#### Tiled OpenEXR

Tiled OpenEXR images are a special flavour of OpenEXR images. Basically the image gets split up into small tiles that can be read independently of each other. The tiles are also stored in different resolutions.

OpenEXR supports multiple channels and layers per image file, as well as 32bit or 16bit float pixels (HDR).

If you select an OpenEXR image as the source for the conversion it will be converted with identical channels and layers (but tiled).

## Compression

OpenEXR offers a variety of compression modes, we recommend ZIP.

- **Uncompressed**  
Doesn't compress the data at all, recommended for highest speeds (if the hard drive is fast) - but also uses the most storage space.
- **RLE (lossless)**  
Compresses the difference between adjacent pixels. Fast to compress and decompress, but in general only works well for images with large areas of solid colour.
- **ZIPS (lossless)**  
This uses a ZIP type compression scheme to compress one scanline at a time. Slow to compress, fast to decompress and also offers a fairly high compression ratio.
- **ZIP (lossless) - default**  
Just like ZIPS, except that it compresses 16 scanlines in one go. This results in a slightly better compression compared to ZIPS, but also slows down reading single scanlines from an image.  
We recommend ZIP for non-grainy images (or images with little grain).
- **PIZ (lossless)**  
This is a wavelet based compression scheme that has a compression ratio that is comparable to ZIP(S), but is faster to compress ... on the other hand it is slower to decompress.  
PIZ is recommended for grainy images.
- **PXR24 (lossy)**  
A wavelet based compression scheme similar to PIZ, 32bit float numbers are cut off to 24bit, losing 8 bits of precision.
- **B44 (lossy)**  
A compression scheme designed for the real-time playback of OpenEXR images with a constant compression ratio.
- **B44A (lossy)**  
Just like B44, but solid areas (such as alpha channels) have a better compression ratio.

These are explained in more detail in the *Technical Introduction to OpenEXR*, available as a PDF at [www.openexr.com](http://www.openexr.com).<sup>17</sup>

## Tile Size

This determines the size of the tiles in the tiled OpenEXR images.

Larger tiles allow the compression to be more effective, reducing the final image size. They are also slightly faster to load and decompress.

Smaller tiles allow infinimap to use memory more effectively while rendering.

We recommend a size of 32x32 or 64x64.

**New Feature**  
Version 1.6

## Source Image Gamma / Target Image Gamma

These allow the user to define the gamma of the source image as well as the target image to either convert from one gamma to another or to slightly increase the accuracy of the tiles generated by the converter (if the source image has a gamma applied to it).

There are basically three usage scenarios:

- If you don't want to change a thing, just keep both at 1.0 when converting (this is also as fast as the conversions were previously).

<sup>17</sup> I admit it, I didn't want to copy and paste the entire section over from the OpenEXR website. The document is highly recommended to understand the capabilities of OpenEXR.

- If you want the target EXR to be linear then enter a source image gamma (usually 2.2 if you're dealing with "normal" sRGB images). This will slow down the conversion process.
- If you want the target EXR to still have the gamma applied, but you also want a slightly better scaling for the tiles in the target exr, then enter the same gamma both for the source and the target image (again, usually 2.2).

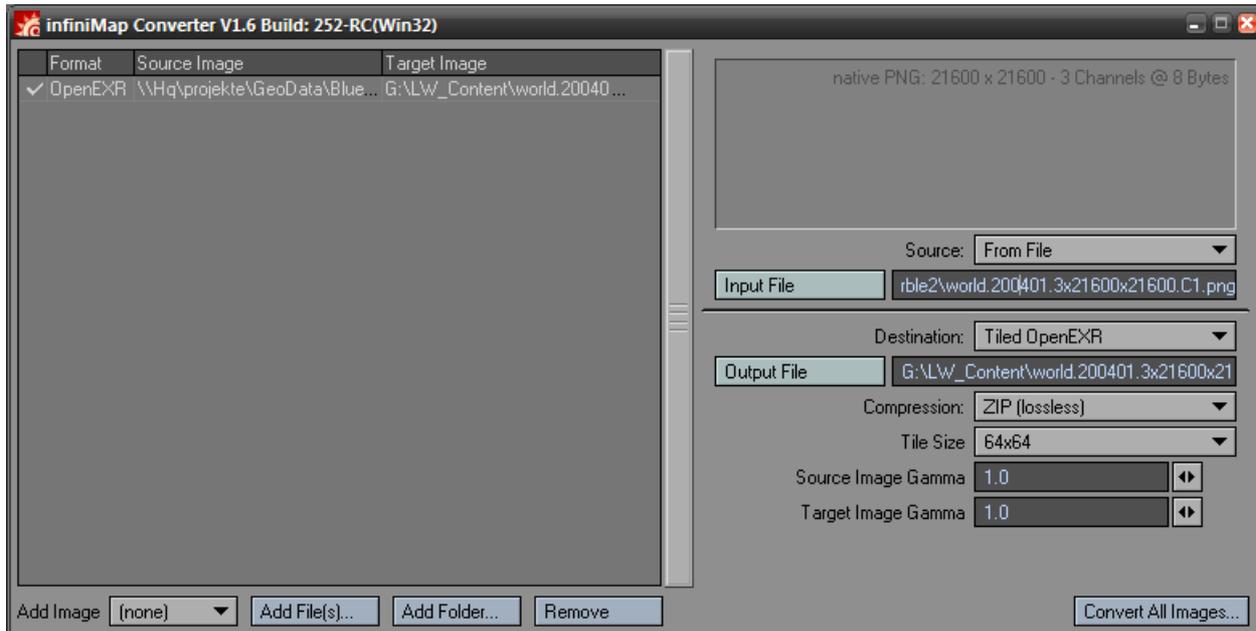
## **Output File**

This opens a file requester allowing you to pick an output file.

## **Convert Image**

This will finally convert the image. Depending on the image size and CPU speed this may take a couple of minutes. A progress window is displayed during the conversion, and infinimap will also display some statistics after the conversion has completed.

## infiniMap Converter (standalone plugin, Batch)



The batch converter is only available for licensed versions of infiniMap (either infiniMap Exr or infiniMap Pro).

Unlicensed versions of infiniMap will display the converter as described in the previous chapter.

The left hand side of the converter shows the list of conversion tasks. The right hand side shows the conversion settings for the task selected on the left. These options are identical to the normal converter as described in the previous chapter and are described there.

The task list is stored in the content directory between sessions of the batch converter. The task list also remains if you close and restart LightWave 3D.

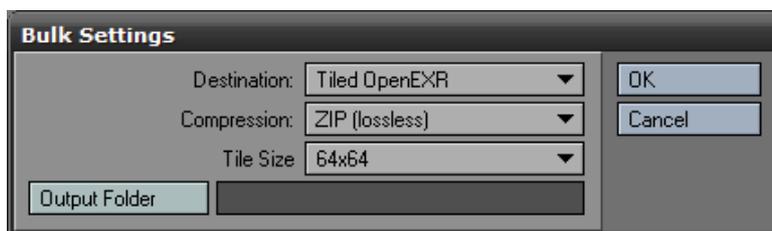
The buttons at the bottom of the batch converter allow you to:

### Add Image

Add any image that is currently loaded into LightWave 3D as a new conversion task.

### Add Files(s)...

This opens a file selector to pick one or more images (using the shift or ctrl key) and adds the respective conversion tasks. If you select more than one image you will be prompted for the conversion settings to be used for all tasks.



The settings correspond to the respective settings in the converter.

If you leave the Output Folder blank, the converted images will be saved in the same directory as the selected files.

### Add Folder...

This will add a complete folder of images to the task list. Other than that this option behaves just like **Add Files...**

## Remove

This will remove the currently selected entry from the list of processing tasks.

## Convert All Images...

This will process all conversion tasks.

You will be prompted with a window displaying further options:

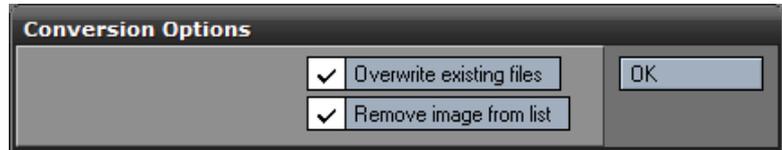
### Overwrite existing files (activated by default)

This will allow the converter to overwrite existing files automatically.

If you turn this off and infinimap would overwrite an existing file you will be prompted to either allow overwriting the file or not.

### Remove image from list (activated by default)

This will remove successful conversion tasks from the batch list.



## Appendices

### JPEG 2000 and ECW conversion tools

In addition to the Converter shipping with infiniMap there are numerous other tools to convert images to the JPEG 2000 or the ECW image file format:

- **Free ECW JPEG 2000 Compressor** by ERMapper ([www.ermapper.com](http://www.ermapper.com))  
This is one of the best compression tools out there, it is however limited to a raw image size of 500MB and only supports TIF and JPEG as source images.  
ERMMapper also have a free ECW/JPEG 2000 viewer for download that is quite nice and handles huge images beautifully.
- **Photoshop® CS2** by Adobe ([www.adobe.com](http://www.adobe.com))  
Photoshop® now supports JPEG 2000 images as well, so you can save your images in JPEG 2000 from within Photoshop® CS2
- **j2k** by Fnord software ([www.fnordware.com/j2k/](http://www.fnordware.com/j2k/))  
A free compression/decompression library for Photoshop® and After Effects®, both Mac and Windows are supported.
- **JasPer** by Michael D. Adams (<http://www.ece.uvic.ca/~mdadams/jasper/>)  
This is a free implementation of a JPEG 2000 compressor/decompressor, source code included. It allows for command line compression of images, there are some graphical front-ends available though (most notable, GeoJasPer at <http://www.dimin.net/software/utis.html> ).
- **LuraWave** by LuraTech ([www.luratech.com](http://www.luratech.com))  
A commercial compressor available either as a stand-alone tool or as a Photoshop® plugin.
- **JPEG Toolbox V2** by Morgan Multimedia ([www.morgan-multimedia.com](http://www.morgan-multimedia.com))
- **JPEG 2000 / A3D Compressor** from Anything 3D ([www.anything3d.com](http://www.anything3d.com))

Most of these compression tools will choke on images that are larger than your main memory. Also TIFF images beyond 2GB are a problem for most conversion tools (since the TIFF standard is only a 32bit standard).

For large scale compression we recommend **ERMMapper** by ERMapper ([www.ermapper.com](http://www.ermapper.com)) or **Manifold® System** by manifold.net (<http://www.manifold.net>) for the budget conscious.

A full list of tools is also available in the support section at [www.db-w.com](http://www.db-w.com)

### OpenEXR related tools

We recommend ProEXR ( <http://www.fnordware.com/ProEXR/> ), there you will also find some simple command line tools to display OpenEXR images or to extract detailed information.

### Tips & Tricks

Well, only one so far:

- To speed up the Surface Editor, increase the size of the preview sphere visible at the top of the Surface Editor, or set the 'Refresh Rate' to 'Manual' in the Surface Editor options.

## Network Rendering

infiniMap is ScreamerNet compatible and allows for network renders. You can install the plugin on any render node and it will render as expected. Here are a couple of pointers to make network rendering with infiniMap a painless experience.

- Make sure infiniMap is properly installed in the plugin configuration file.
- Use content directories.  
infiniMap saves and loads images in relation to the content directory if possible.
- To make maximum use of the image cache used by infiniMap, configure your render controller to render successive frames on the same render node. In Spider for example, you can configure the nodes to render any number of frames at a time, instead of distributing the frames to be rendered one at a time to the render nodes.  
Other render controllers may have similar options.

## Known Issues

Unfortunately some things just never work the way they should. We tried to squash every bug we encountered, however sometimes it wasn't possible due to circumstances outside of our control. LightWave 3D itself has a couple of holes within the SDK, and the interaction with FPrime can be troublesome at times as well.

Please note that newer releases of LightWave 3D or infiniMap might have solved some of these issues. Please check [www.infinimap.com](http://www.infinimap.com) regularly for updates.

Here is a short list of known issues and a collection of workarounds that should make your life easier in critical situations:

- Pasting procedural texture layers may crash LightWave if FPrime is rendering at the same time. It is recommended that you "Pause" FPrime before pasting a texture layer and then "Unpause" to commence rendering after the paste operation.
- If LightWave slows down drastically if you have the Surface Editor open, try to increase the size of the preview sphere and/or change the update mode to Manual.

## LScript

The following commands can be issued from LScript or as a LightWave command.

### **Generic\_infiniMap\_Browser**

Opens the infiniMap Browser

### **Generic\_infiniMap\_Converter**

Opens the infiniMap Converter

## Glossary of Used Terms

### **SDK (software development kit)**

Basically a set of tools and documents to allow the creation to plugins. In the case of LightWave3D the SDK provided by NewTek.

### **JPEG 2000**

Wavelet based successor to the JPEG standard for compressed images. It allows for lossless compression, more than 8 bits per channel and has a variety of enhanced features.

### **ECW**

an advanced wavelet based compressed image file format developed by ER Mapper. Now replaced by JPEG 2000.

### **OpenEXR**

A HDR image format designed by ILM for post-production pipelines that has found a wide acceptance in the industry.

### **OpenGL preview layer**

A term used by infiniMap for a texture layer within a surface that serves no other purpose but to display an approximation of an infiniMap image in the OpenGL real-time preview of LightWave3D.

Basically, it is an image layer with the opacity set to 0%, which will still display in OpenGL but not during renders.

Please note that GLSL needs to be turned off in the LightWave 3D OpenGL options for the preview layer to be visible in the viewports.

## FAQ

### **I can't create an OpenGL proxy layer from the Shader**

Unfortunately, for this to work from the shader you will have to manually activate the texture layering on the surface colour.

Click on the [T] button to the right of the surface base colour once to activate it. Now the infiniMap shader can create an OpenGL proxy layer for that surface.

### **I've created an OpenGL proxy layer, but I don't see it in the texture layer window (it does show in OpenGL though)**

While the layer has actually been created, LightWave3D will not display the layer in the texture list until you close and re-open the texture list.

### **No image preview in the Modeler when working on the procedural texture**

Due to a limitation imposed by LightWave 3D, the image preview is not available in the procedural texture component of infiniMap 3D when running in Modeler.

### **I change the preview resolution, but the Preview / VIPER doesn't update**

Sometimes LightWave3D doesn't register the fact that a texture value has been changed. Switching the preview resolution is one of those cases. If it happens, please refresh the preview or VIPER manually.

---

## Alphabetical Index

**C**

Channel.....2p., 6pp., 15pp., 19, 30

Converter..... 1pp., 7p., 22, 24, 26pp.

    Batch..... 1, 3, vii, 22, 26p.

**E**

ECW..... 1pp., 7pp., 22p., 28, 30

ERMapper..... 23, 28

**F**

FPrime.....2, 7p., 11, 15, 21, 29

**J**

JPEG 2000..... 1pp., 7pp., 22p., 28, 30

**L**

Layer.....2p., 5pp., 9p., 15, 17p., 21, 29p.

Layout.....2, 5, 15, 21

LoD Blending..... 6, 11p.

LScript..... vii, 29

**M**

Modeler..... 8, 30

**O**

OpenEXR..... 1pp., v, 7, 9, 18, 20, 22pp., 28, 30

OpenGL..... 2, 9p., 30

Options..... 6, 8, 21

**P**

Photoshop..... 18, 28

Projection..... 3, 6p., 13, 18p.

    Cylindrical..... 6

    Front..... 6, 28

    Planar..... 3, 6, 13

    Spherical..... 6, 13

    UV..... 3, 6p., 13, 15pp., 19

**S**

ScreamerNet..... 2, 29

**T**

Texture Quality..... 6, 11p., 15

**V**

VIPER..... 11, 30