



User Manual

Version 1.5

©2006-2010 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, 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 exrTrader plug in for LightWave 3D and documentation for exrTrader.

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

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 exrTrader

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-appealable 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

© 2006-2010 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 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 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.

This plugin uses 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

Mark Adler madler@alumni.caltech.edu

This plugin uses 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

Michael Wolf

Product Marketing

Dagmar Bornemann

This piece of software is dedicated to Zareh and Mike.

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

Special kudos to Phil for proof reading the manual!

We would also like to thank NewTek-Europe 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
info@db-w.com

Table of Contents

Trademark and copyright information.....	iv
Acknowledgements.....	vi

Introduction	1
Buffers & Channels – a Basic Introduction.....	1
Compatibility.....	2
Features.....	3
Setting up exrTrader	4
Installation.....	4
Windows 32-bit/64-bit.....	4
Mac OS X, UB and CFM.....	4
Included Plugins.....	5
Registration.....	6
exrTrader – OpenEXR Layered Export	7
Presets.....	8
Save as Preset.....	8
Delete Preset.....	8
Load Settings from File.....	8
Save Settings To File.....	8
Custom directories for presets.....	8
Render Mode.....	9
VIPER and Save Buffers.....	9
VIPER only (save in Screamer.net).....	9
Save Buffers Only.....	9
VIPER Display Gamma.....	9
Toggle VIPER.....	9
Post-apply Display Gamma.....	9
Output File.....	10
Use Render Path.....	10
Format.....	10
Single File Naming.....	10
Output_Layer_###.exr.....	11
Layer/Output_###.exr.....	11
Layer/Output_Layer_###.exr.....	11
Layer_Output_###.exr.....	11
Custom_###.exr.....	11
Custom Single Image Name.....	11
Examples.....	11
Compression.....	12
Metadata.....	13
Embed Preview.....	13
Scale / Width / Height.....	13
Comment.....	13
Author.....	13
Buffers.....	14
Buffer.....	14

Enabled.....	14
Description.....	14
Quick Buffer Enable.....	14
Individual Settings for Buffers.....	15
Layer Name.....	15
Saving.....	15
Processing.....	17
Channels.....	18
Review Settings.....	19
About exrTrader.....	19

exrTrader and VIPER **20**

Preview Options.....	20
Pixel Inspector.....	21

Basic OpenEXR Loader and Saver Plugins **22**

OpenEXR Settings.....	22
Compression.....	22
Pixel Format.....	22

Special Plugins **23**

OpenEXR Dummy Saver.....	23
exrTrader.....	23
OpenEXR Settings.....	23

Appendices **24**

Available buffers.....	24
Network Rendering.....	31
Third party LightWave 3D plugins with exrTrader support.....	31
OpenEXR support in third party applications.....	31
LScript.....	32
Recommended Reading.....	32
Supported Metadata.....	33
Basic Metadata.....	33
Maker Metadata.....	35

Introduction

Quoted from www.openexr.com:

OpenEXR is a high dynamic-range (HDR) image file format developed by Industrial Light & Magic for use in computer imaging applications.

OpenEXR's features include:

- *Higher dynamic range and color precision than existing 8- and 10-bit image file formats.*
- *Support for 16-bit floating-point, 32-bit floating-point, and 32-bit integer pixels. The 16-bit floating-point format, called "half", is compatible with the half data type in NVIDIA's Cg graphics language and is supported natively on their new GeForce FX and Quadro FX 3D graphics solutions.*
- *Multiple lossless image compression algorithms. Some of the included codecs can achieve 2:1 lossless compression ratios on images with film grain.*
- *Extensibility. New compression codecs and image types can easily be added by extending the C++ classes included in the OpenEXR software distribution. New image attributes (strings, vectors, integers, etc.) can be added to OpenEXR image headers without affecting backward compatibility with existing OpenEXR applications.*

exrTrader makes the OpenEXR file format available for users of LightWave 3D, providing basic image loading and saving as well as supporting the advanced features of OpenEXR, such as multiple buffers stored in a single image file.

This manual covers the usage of exrTrader only.¹

We are planning on publishing further tutorials of how to use OpenEXR in various pipelines in the future though; please check www.exrtrader.com for updates.

The capabilities of OpenEXR itself are explained extremely well at www.openexr.com. The "Technical Introduction to OpenEXR", available for download as a PDF, is well worth a read.

Buffers & Channels – a Basic Introduction

Let's start with some definitions:

A **Channel** is a single image component, such as the red channel of an image.

A **Buffer** consist of one or more channels, for example:

Final Render is a buffer consisting of the red, green and glue channels.

Depth is a buffer consisting of one channel: Z

Motion is a buffer with two channels, X and Y (this is velocity in pixels)

OpenEXR is quite a complex, extensible image format, and many compositing applications only support a subset of its features, such as the following basic channels:

R	Red
B	Blue
G	Green
A	Alpha / Mask
Y	Luminosity
Z	Depth

Fusion 5 for example expects the following channel names by default (fortunately this can be changed in the Fusion 5 image loader):

¹ It has come to our attention that some of our customers regard this documentation, including assorted footnotes, as being "entertaining". Let me assure you that this is not the case. We're Germans. If you, for whatever reason, have the urge to take any part of this manual humorously, please re-read the entire license agreement as an exercise to regain your composure.

R, G, B, A, Z, pixelCover, objectID, materialID, U, V, NX, NY, NZ, velX, velY

This is why we decided to include a preset system to cater for the needs of different applications and pipelines.

If the buffers are saved to a single file they are named as follows:

“LayerName.ChannelName”

Either the layer or the channel name may be empty (in that case the “.” is omitted) - but never both of them. You cannot have duplicate channel names within the same file.

The “Review Settings...” option tries to catch as many errors and omissions as possible.

If your application only supports the most basic channels, you can use one of the options to save a buffer to an individual image file.

In that case the Layer Name is omitted from the channel name, but is used as part of the file name. Make sure that the channel names are R,G,B (or Y in the case of a single channel image), otherwise more limited applications that expect the default channel names won't be able to read the image!

Some OpenEXR savvy applications, such as Nuke, can group the channels by their layer name, so they see “LayerName” as a group of channels (or a layer).

The appendix contains a list of applications and their support for OpenEXR.

Compatibility

exrTrader V1.5 is based on OpenEXR 1.6.1

exrTrader is compatible with LightWave3D 7.5 up to LightWave 9.6.1 and includes experimental support for LightWave HC.

It currently runs with the Windows 32bit, Windows 64bit, Mac OSX CFM and Mac OSX UB 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 ports in this manual.

The OS X 64-bit UB of LightWave 9.6.1 is still in beta at the time of this writing and not supported.

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

Please visit <http://www.exrtrader.com> or <http://exrtrader.db-w.com> for more up to date information.

Features

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

- standard LightWave Loader supporting RGBA, does not require a license.
- standard LightWave Savers, supporting all save-able image types except for palletized images, does not require a license.
- a scene type master plugin for setting the image saver options, which are saved in the scene file.
- a dummy saver to enable network render managers to find saved files for success checks.
- an image filter plugin that allows to save arbitrary buffers to a single OpenEXR file that:
 - either uses a custom output file, or the RGB output file defined in the render globals.
 - mimics LightWave's name formatting scheme.
 - supports all compression schemes defined by OpenEXR.
 - supports all output buffers generated by LightWave and exposed through the SDK.
 - supports buffers that may consist of one (i.e. Depth), two (i.e. Motion), three (i.e. RGB) or more channels.
 - offers separate naming options for channels and buffers in the OpenEXR file, this will translate to buffer.channel as defined in the OpenEXR specs. Either may be blank.
 - supports different pixel types per channel.
 - offers the processing of buffers (invert, min, max, scale, offset).
 - supports OpenEXR metadata, including aspect ratio, comments, previews.
 - data window support (for limited region renders).
 - uses VIPER to preview buffers including gamma correction
 - optional un-multiplication by the image alpha
 - advanced preset system with "default" presets and custom preset directories
- network rendering support.
- plugins consist of one file per platform.
- "Q Approved!" and SWELL.
- Support for LightWave 3D metadata as of LightWave 3D version 9.5
- Options to customize the export to match a variety of compositing workflows.
- Additionally save single buffers using any image format supported by LightWave 3D
- Post-Apply display gamma option for better integration into linear pipelines
- File and buffer names may contain variables that are replaced with the current values when saving.
- Support for naming the files if rendered with a stereoscopic camera.
- Automatic creation of a data window within the exr files to speed up loading and filters in compositing applications that support a DoD, This is transparent to most other applications.
- Added a pixel Inspector window for the VIPER based buffer preview.
- Added support for new buffers as exposed by LW HC.
- free point upgrades, free support, free beer².

New Feature

Version 1.4.1

New Feature

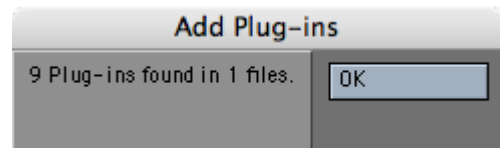
Version 1.5

² O.k., so we're still joking about the beer...

Setting up exrTrader

Installation

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



Windows 32-bit/64-bit

The **exrTrader.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 **exrTrader.p** to LightWave 3D. LightWave will prompt you that it found 9 plugins in one file (LightWave 9.6)³.

Mac OS X, UB and CFM

The OS X versions of exrTrader are included as a DMG (Disk Image). Double click on the icon to mount the disk image.

CFM

The **exrTrader.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 **exrTrader.p** to LightWave 3D.



Universal Binary

You can simply drop the exrTrader.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:

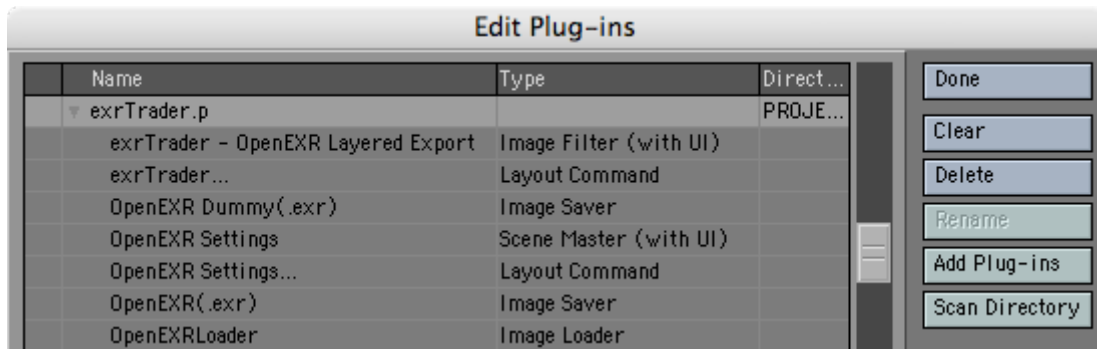
`/Library/Application Support/LightWave3D/Plugins`

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.

³ Actually, there are only 7 plugins, LightWave 3D 9.0 counts the user interfaces for two of these extra though. LightWave 3D 8.0 correctly reports 7 plugins.

Included Plugins



exrTrader is actually a suite of plugins for LightWave 3D. Currently it consists of the following plugins (these are the plugin names as displayed in the Edit Plug-ins window by LightWave):

exrTrader – OpenEXR Layered Export

OpenEXR Settings

exrTraderLoader

OpenEXR(.exr)

OpenEXR Dummy(.exr)

exrTrader...

OpenEXR Settings...

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

The basic loaders and savers are fully functional without registering.

The last two plugins are convenience tools you can assign to a shortcut, or add to a menu, to quickly apply the respective plugins to the current scene (if they aren't applied already) and open their user interface.

Registration



The image shows a Windows-style dialog box titled "Registration for exrTrader V1.0 (Win32)". It contains several input fields and buttons. At the top, there is a label "Purchase your license at:" followed by a text box containing "www.exrtrader.com". To the right of this are "OK" and "Cancel" buttons. Below this, there are five rows of labels and text boxes: "Lightwave Lock ID:" with "11111", "Full Name:" with "Michael Wolf", "Company Name:" with "db&w GbR", "E-Mail:" with "michael.wolf@db-w.com", and "Activation Key:" with an empty box. Below the "Activation Key" box is a button labeled "Activation Key Valid". At the bottom of the dialog, there are two buttons: "Import Activation ..." and "Export Registration ...".

Once you've installed the plugin, apply the **exrTrader – OpenEXR Layered Export** image filter plugin to your scene, and the registration window will automatically open.

Enter your full name, company name (if applicable) and e-mail address. Export the registration and attach it to an e-mail to **registration@exrtrader.com**. We will mail you an activation code within 24 hours after payment is received, depending on the time difference⁴.

You can now either import the keyfile attached to the activation e-mail, or copy and paste the activation key into the registration panel (please make sure that the personal information is identical to the one submitted to us for the registration).

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

The registration file is stored in the same folder where LightWave 3D is installed⁵, the license file is named exrTrader.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.

⁴ Our key generator needs some sleep every now and then ;)

⁵ Please back up your license file if you install a new version of LightWave into the same directory, especially on OSX when installing the UB of LightWave.

exrTrader – OpenEXR Layered Export

Originally Posted by **Exper**

"Parameter 1: use it to change the value of Parameter 1".

This is the flagship plugin. It allows you to export any of the render buffers provided by LightWave into a single, or multiple OpenEXR files.

OpenEXR is a quite complex image file format. The user interface of exrTrader provides full access to the options provided by the image format.

From the top to the bottom it is basically divided into four areas.

The top covers basic workflow options, as well as global options for the exported image files.

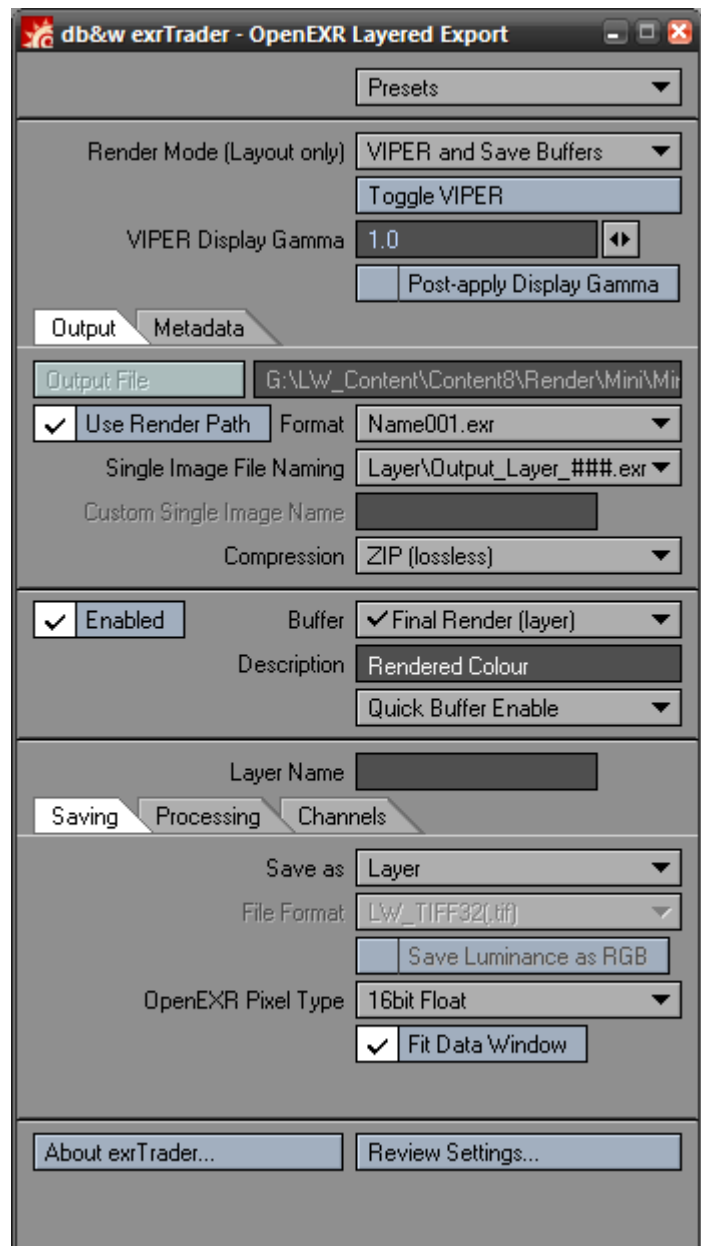
The **Output** tab controls where the images are saved and how they will be named.

The **Metadata** tab contains options for metadata stored with the image files, such as a preview image, or a comment.

The middle section, starting with the **Buffer** pop-up, contains the controls for the individual buffers.

The bottom section includes the **About...** button to quickly check the version number and **Review Settings...** which gives you a quick overview of the applied settings and also highlights potential problems with the applied settings.

The individual controls will be explained in the following sections:

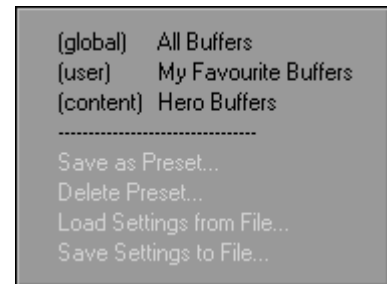


Presets

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

exrTrader 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".



New Feature **global** presets are stored in a directory that is accessible to all users on the current machine (and can be mirrored on a server).
Version 1.4.1

On Windows the location is: %allusersprofile%\Documents\dbw\LightWavePlugins\

On OSX UB it is /Library/Preferences/LightWave3D/SimplePresets/

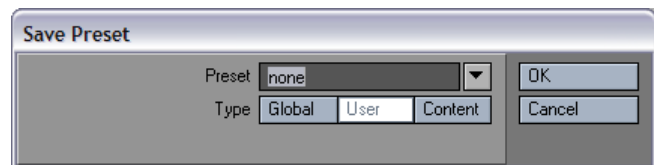
content presets are stored in the current content directory.

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

exrTrader 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.

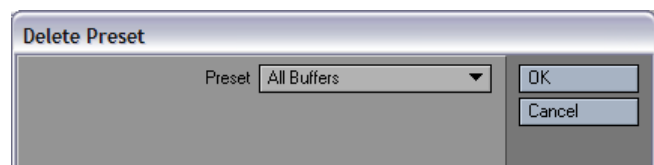
Save as Preset...

This option allows you to save the current exrTrader 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 delete one of the available presets and delete them from the hard drive.



Load Settings from File...

This allows you to load previously saved settings for exrTrader from a file.

Save Settings To File...

This allows you to save the current exrTrader settings to a file.

Custom directories for presets

New Feature It is now possible to define user directories to store the presets in. This is done by placing a text file called "SimplePresetPaths.cfg" either into:
Version 1.5

- The LightWave/Programs directory
- The directory where the LW settings are located
- and/or the content directory.

The files are read in that order, and preset paths defined in an earlier configuration file can be overwritten.

The text file should be formatted as such

```
# this is just a comment, anything that starts with a # is ignored
# first the name of the path as displayed to the user (without spaces)
# followed by the path
network \\db-w\lightwave\presets
local_temp c:\my_temp_presets
```

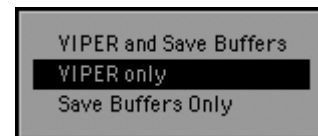
The path name is not case sensitive, it will be capitalized automatically. It may not contain spaces though.

You can also overwrite the existing default paths by using **user**, **content** or **global** as path names.

Render Mode

The Render Mode defines how exrTrader will handle the buffers when rendering in Layout.

exrTrader can display all buffers using VIPER after an initial render with exrTrader applied to the scene, this requires up to 42MB of memory.



VIPER and Save Buffers

This will save the buffers to disk, and also store a scaled down version in memory to be used in conjunction with VIPER.

VIPER only (save in ScreamerNet)

This will only store the buffers in memory but not save them to disk. However, rendering the scene using ScreamerNet will save the images. This is to prevent the overwriting of existing images during test renders.

Save Buffers Only

This option only saves the buffers to disk, but will not store them in memory for VIPER previews.

Note: the **Render Mode** setting is only relevant when rendering in Layout. When rendering using lwsn/ScreamerNet, the network renderer of LightWave 3D, exrTrader will **always** save the buffers, and **never** store them in memory for VIPER⁶.

VIPER Display Gamma

This is the gamma value applied to buffers that are previewed in VIPER. The gamma correction is not applied to the buffers prior to saving them, it is only for preview purposes.

Only buffers with three channels are gamma corrected for display.

Toggle VIPER

New Feature
Version 1.5

This button opens or closes VIPER. Please have a look at the chapter on exrTrader and VIPER to see the possibilities of using VIPER to preview buffers.

Post-apply Display Gamma

⁶ Obviously, since there is no VIPER in the network renderer.

If you check this control then exrTrader will apply the Display Gamma setting to the main RGB buffer after exrTrader has processed and exported the buffers.

This allows you to view a render with gamma correction in the LightWave image viewer, but still save out a linear image.

Output File

This control allows you to select an output file for the buffer saver to write to. Any extension and frame numbers of the selected file will be automatically removed. You can only select an output file if “Use Render Path” is not activated.

exrTrader also supports special variables in the path name that will be replaced as the image is being saved.

The currently supported variables are:

%scene% Is replaced with the scene name (without the .lws extension)

%camera% The name of the camera used to render the current image

%layer% The name of the current layer

New Feature

Version 1.4.1

%eye% If the current camera is rendering in stereo mode, this will be either **L** or **R** depending on which eye the currently saved frame represents.

%dot% This is just replaced with a single “.” It allows you to add a . at the end of the file name in the Output File Control (where it would normally be stripped with any file name extension).

You can use the **Review Settings...** option to verify if the file names are created as you'd expect them to be.

Use Render Path

If you select this option, exrTrader will use the output file defined in the Lightwave3D render globals.

This option is quite important if used across a render farm in conjunction with the OpenEXR Dummy Saver, described later in this manual.

Format

This settings defines how exrTrader will append the frame number and file extension to the Output File. These settings are identical to the settings in the Lightwave3D render globals.

```
Name001
Name001.exr
Name0001
Name0001.exr
Name_001
Name_001.exr
Name_0001
Name_0001.exr
```

Single File Naming

exrTrader by default saves all buffers into a single OpenEXR image file. You may optionally save single buffers into a separate image file.

This option defines how these separate image files will be named.

Output is the file name, as defined by the Output File control, or the render global in Lightwave3D if “Use Render Path” is active.

Layer is the layer name of the selected buffer, as described below in the documentation.

is the frame number, the amount of digits is defined by the Format option.

```
Output_Layer_###.exr
Layer\Output_###.exr
Layer\Output_Layer_###.exr
Layer_Output_###.exr
Custom_###.exr
```

- _** The underscores are only added if the Format option contains underscores as well.
- .exr** The file extension is added only if the Format option contains the file extension.

Output_Layer_###.exr

This will save the buffer image files into the same directory as the Output File, and append the “Layer” name to the file name.

Layer/Output_###.exr

This will create a subdirectory named after the “Layer” name where the Output File is written, and write the buffer image into that subdirectory.

Layer/Output_Layer_###.exr

This will create a subdirectory named after the “Layer” name where the Output File is written, and write the buffer image into that subdirectory appending the “Layer” name to the file name.

Layer_Output_###.exr

This will save the buffer image files into the same directory as the Output File, and prepend the “Layer” name to the file name.

Custom_###.exr

This will save the buffer image files into the same directory as the Output File, and use the “Custom Single Image Name” as the file name.

Custom Single Image Name

If the Single File Naming option is set to **Custom_###.exr**, the file name for single image files will be constructed from the text entered into this control.

You may also add slashes or colons (depending on the OS platform) to designate sub-directories. These will be automatically created by exrTrader if needed.

The currently supported variables are:

%scene% Is replaced with the scene name (without the .lws extension)

%camera% The name of the camera used to render the current image

%layer% The name of the current layer (as defined in the per buffer **Layer Name** option)

%eye% If the current camera is rendering in stereo mode, this will be either **L** or **R** depending on which eye the currently saved frame represents.

New Feature

Version 1.4.1

Examples

To hopefully make this a bit easier to understand, here is an example of how these options work.

Let's assume we're rendering out to “images/Ninja.exr” as the Output File. The Format we're using is “Name_0001.exr”.

We want to save the Reflection buffer as a separate image file, the “Layer” name of it is “Specular”.

These will be the buffer image file written at frame 60, depending on the Single File Naming option:

<code>Output_Base_###.exr</code>	<code>images/Ninja_Specular_0060.exr</code>
<code>Layer/Output_###.exr</code>	<code>images/Specular/Ninja_0060.exr</code>
<code>Layer/Output_Base_###.exr</code>	<code>images/Specular/Ninja_Specular_0060.exr</code>
<code>Layer_Output_###.exr</code>	<code>images/Specular_Ninja_0060.exr</code>

In all cases, the main file saved will be the same: `images/Ninja_0060.exr`

Compression

All compression options are lossless, except for PXR24, which is lossy when saving 32bit float image data (it cuts off the lower 8bit, effectively cutting the precision to 24bits).

exrTrader V1.1 has been compiled to support OpenEXR Version 1.6, which includes the following compression modes:

- **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) - default**
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)**
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 (as some compositing packages might do in proxy or ROI/DOD mode).
- **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.
- **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.

Uncompressed
RLE (lossless)
ZIPS (lossless)
ZIP (lossless)
PIZ (lossless)
PXR24 (lossy)
B44 (lossy)
B44A (lossy)

These are explained in more detail in the Technical Introduction to OpenEXR, available as a PDF at www.openexr.com.⁷

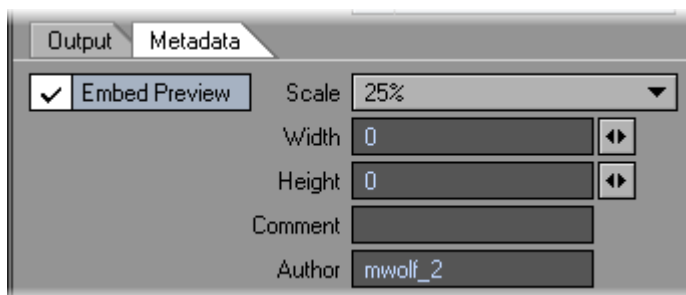
⁷ I admit it, I didn't want to copy and paste the section over from the OpenEXR website. The document is highly recommended to understand the capabilities of OpenEXR.

Metadata

The metadata section of the interface allows you to store some additional information within your images.

Please feel free to contact us at support@db-w.com if you require other metadata to be stored with your OpenEXR files.

In addition to the metadata exposed by the user interface, exrTrader will also save the information listed in the appendix as metadata in your image files:

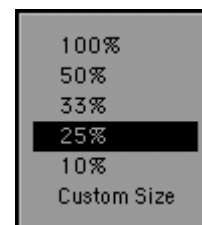


Embed Preview

Select this option to save a small preview image within the OpenEXR image files. This will be saved as 8bit data.

Scale / Width / Height

This option defines the scale of the preview image. You may select any of the pre-defined percentages, or Custom Size to manually enter a **Height** and **Width** for the preview image.



Comment

Here you can enter a comment that will be stored within the OpenEXR files saved by exrTrader

Author

You can use this control to enter the name of the person that created the images rendered with exrTrader in this scene. The name will be stored in the image files as the "owner".

Buffers

This section of the interface allows you to individually select and edit buffers for exporting.

The section above the dividing line allows you to select the available buffers.

The section below the dividing line contains the individual settings for the selected buffer.

Buffer

This pop-up displays all buffers currently available to exrTrader via the Lightwave3D SDK.

Buffers that are enabled for export have a check mark in front of them, Buffers that are written to a single file have the text "(image)" behind them.

Note: We think a couple of important buffers are missing from this list, especially the coverage data, object and surface ids as well as normal and UV data. Unfortunately the current SDK for LightWave3D provides no access to these.

A complete list of buffers is available in the appendices.

Enabled

This check box enables the buffer for saving.

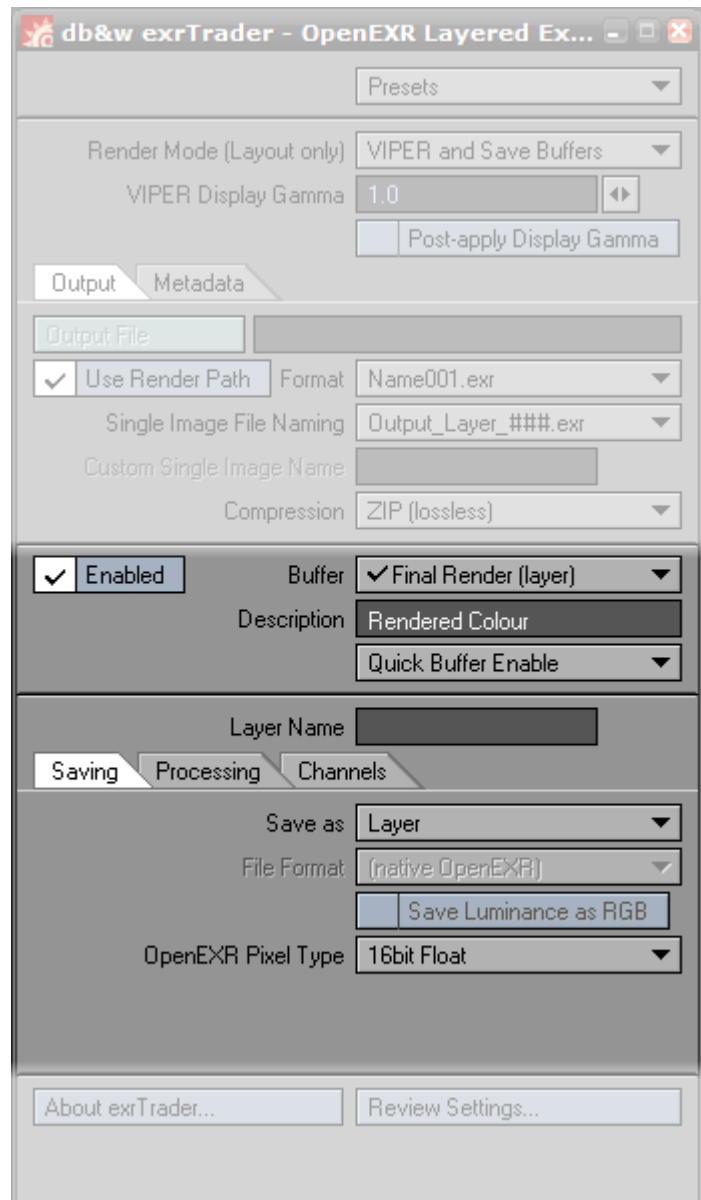
Description

A short description of the currently selected buffer is displayed here.

Quick Buffer Enable

This pop-up allows you to quickly enable/disable some of the more commonly used buffers.

Enable All
Disable All
Toggle
Enable RGB only
Enable RGBA only
Enable RGBAZ only

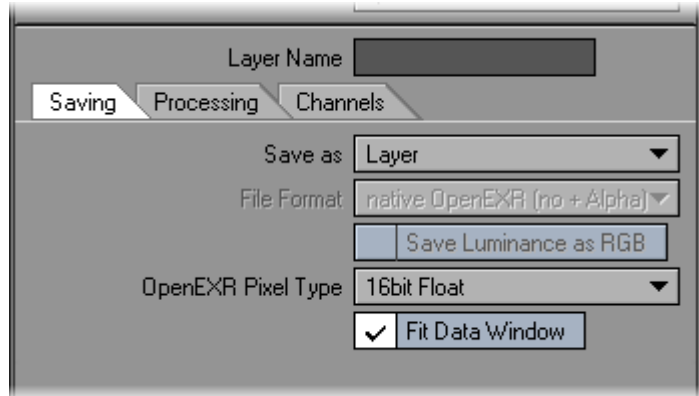


Individual Settings for Buffers

The following settings are per buffer, and define how a buffer will be processed by exrTrader.

These are organized into three tabs, Saving, Processing and Channels

Here you can also name the buffers, specify their pixel type and save them as single image files.



Layer Name

This is the base name for the current buffer.

If the buffer is saved as a channel in the main output file, the channels will include this base name as a prefix, i.e. "Specular.Red".

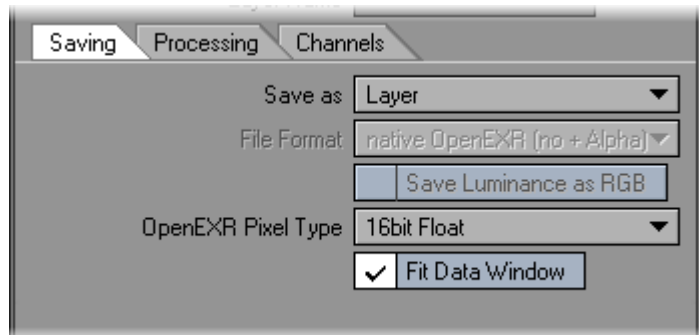
If the buffer is saved as a single image file, the base name will be a part of the image file name as explained in the **Single File Naming** section.

Saving

Save as

Due to popular demand we've extended the ability to save a buffer as a separate image.

A buffer can now be saved as a separate image of any type supported by LightWave 3D, as well as being a part of the main OpenEXR image.

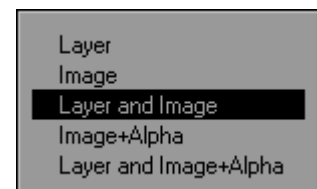


Layer

This option will save the current buffer only as a layer in the main OpenEXR image file.

Layer and Image

This option will save the current buffer as a layer in the current OpenEXR image file and as a separate image file. The file format for the separate file is determined by the **File Format** option.



Layer and Image+Alpha

This option will save the current buffer as a layer in the current OpenEXR image file and as a separate image file including the alpha channel. The file format for the separate file is determined by the **File Format** option.

Image

This option will only save the current buffer as a separate image file. The file format for the separate file is determined by the **File Format** option.

Image+Alpha

This option will only save the current buffer as a separate image file, including the alpha channel. The file format for the separate file is determined by the **File Format** option.

File Format

This pop-up allows you to select the file format used when saving the current buffer as a separate Image (as determined by the **Save as** option).

The **(native OpenEXR)** saver is equivalent to the option **Save as Single Image** in previous releases of exrTrader. It uses the settings for the main OpenEXR file (such as the compression settings) and uses an internal saver as opposed to using the LightWave 3D image plugins.

Please note that the **(native OpenEXR)** saver will not save the additional Alpha channel if you select any of the **Image+Alpha** options.

Save Luminance as RGB

This option is only available when “Save as Single Image File” is activated.

By default exrTrader saves single channel OpenEXR images with a channel names “Y”. As defined in the OpenEXR specifications this is a luminance channel that should be interpreted as greyscale data by reading applications.

Unfortunately this isn't the case, some applications do not handle Y channel only images (or don't handle them well). This option will save the greyscale data as RGB data instead.

This increases the size of the image threefold, but also maximizes compatibility.

OpenEXR Pixel Type

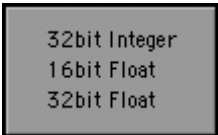
This option allows you to define how the pixels of the current buffer will be saved in the OpenEXR file. OpenEXR allows for all pixel types to co-exist in a single image file.

The Float pixel types allow for HDR buffers to be stored.

Integer is currently of little use with the available buffers. It is designed to be used with precise, discrete values such as object or surface Ids.

We recommend to use 16bit Float for all buffers, except for the Depth buffer which profits greatly from saving it at 32bit Float.

For more colour precision in compositing you can always switch buffers to 32bit float.



32bit Integer
16bit Float
32bit Float

Fit Data Window

New Feature
Version 1.5

This option shrinks the data window of the layer to only encompass pixels that actually contain values within the layer.

OpenEXR images use the data window to define a rectangle that actually contains pixel values. This may be smaller than the actual size of the image.

This is basically like a only saving a limited region of the final image, but computed every time an image is saved.

If this option is off, then the current layer is not used to compute the data window.

If it is on, then the smallest data window for the current layer is computed. If a multi-layer OpenEXR image is saved, then the data windows of all layers as well as the current limited region settings are used to compute an optimal data window for the image.

If the native OpenEXR saver is used to save the current layer as a separate image, then the data window for the current layer is taken into account as well.

All other image savers ignore it.

This option only slows down saving a little, but can speed up loading the saved OpenEXR files in compositing applications a lot.

Fit Data Window is disabled by default as some applications (notably Photoshop and After Effects) don't load images with a data window properly (they only load an image as large as the data window itself).

If you use a compositing application that supports OpenEXR data windows (such as Fusion6, Shake or Nuke) then we recommend you to once enable the data window for all buffers and save the setting as a **default** preset.

Processing

Invert

This option will invert the current buffer.

Minimum / Maximum

These two options allow you to change the range of the data saved in the buffer.

For most buffers, the Minimum and the Maximum define the range of values written out that correspond to the range of 0 .. 1.0 the the buffer as passed by LW (or "raw" buffer).

For the Depth buffer, these are handled differently, the Minimum and Maximum define the distances in metres from the camera that correspond to the 0 .. 1.0 range in the written buffer.

Expressed mathematically:

$output = Minimum + buffer * (Maximum - Minimum)$ for all a buffers except Depth

$output = (buffer - Minimum) / (Maximum - Minimum)$ for the Depth buffer

Offset / Scale

Offset and Scale allow further tweaking of the buffer values.

Offset is just added to the buffer values, Scale scales the range of values in the buffer by the entered amount. The Offset is applied before the Scale.

Gamma Correction

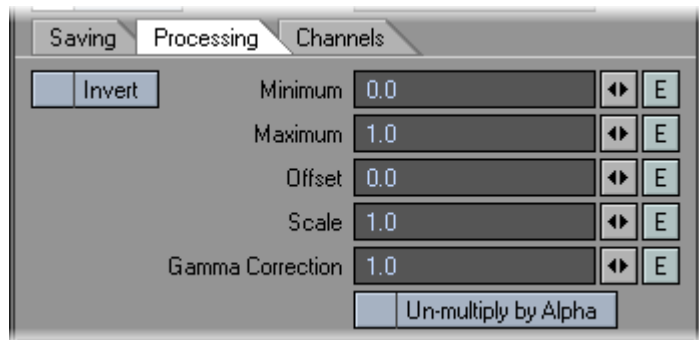
This setting allows you to apply a gamma value to a buffer.

To remove gamma applied to the buffer, enter "1/gamma_value". As an example: $1 / 2.4 = .4167$ removes a gamma of 2.4 from the buffer.

Un-multiply by Alpha

This option un-multiplies the current buffer by the alpha channel. This setting is needed for compositing applications that don't support the additive merging of images.

This option is only available for buffers that contain three channels.



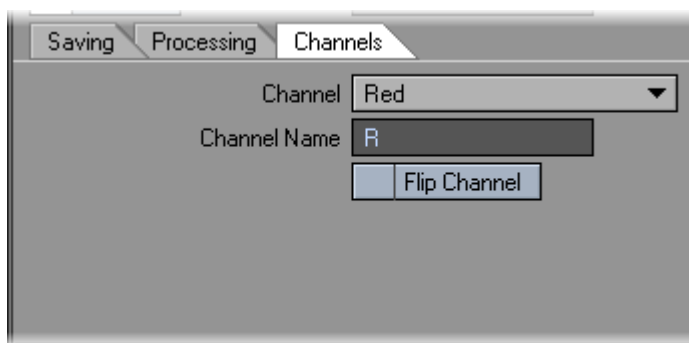
Channels

Channel / Channel Name

New Feature
Version 1.4.1

Since OpenEXR allows for the naming of every channel saved in the file, you can use the **Channel** control to pick a channel, and **Channel Name** to give it a new name.

Since version 1.4.1 channels can be renamed for single channel buffers as well. This was implemented to increase compatibility with Nuke, which expects single channels to be named "A" (mask/alpha) as opposed to "Y" (luminance).



Flip

This is a per channel setting. It will flip the values within the channel around the 0 value. This is different from inverting a channel.

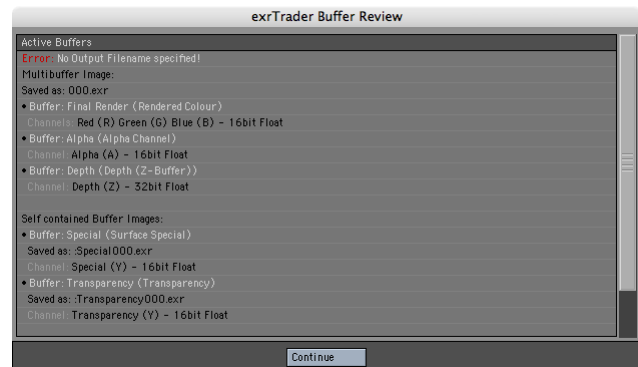
This is designed to flip the X/Y motion channels to match the expectations of different compositing applications.

Review Settings...

This option will show you a window with a summary of all buffers that are being saved out. Any potential errors will be highlighted in red.

Review Settings tries to cover all potential issues when saving an image. It may not succeed in all cases, though.

If it reports no error, and you still have problems saving, please create a preset of the settings and mail them to us at support@db-w.com, we will update Review Settings to catch those issues as well.



About exrTrader...

This shows the about box of exrTrader, including the version you are using.



exrTrader and VIPER

exrTrader is unique in supporting VIPER to preview buffers prior to saving them out to disk. It even previews the effects of the editing settings applied to the buffers.

To use VIPER, make sure the **Render Mode** is set to either “VIPER and Save Buffers” or “VIPER only” and render a single frame of your scene. The buffers stored in memory will use up at most 42 MB of RAM.

You can now open VIPER and select any of the buffers in the exrTrader interface. The title of the VIPER window will read “VIPER : exrTrader Buffers” if it can display the exrTrader Buffers.

Adjustments to the buffer settings will be displayed immediately.

VIPER also provides Preview Options that control how the preview is displayed:



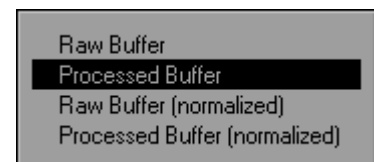
blue_red_ninja.lws by William Vaughan

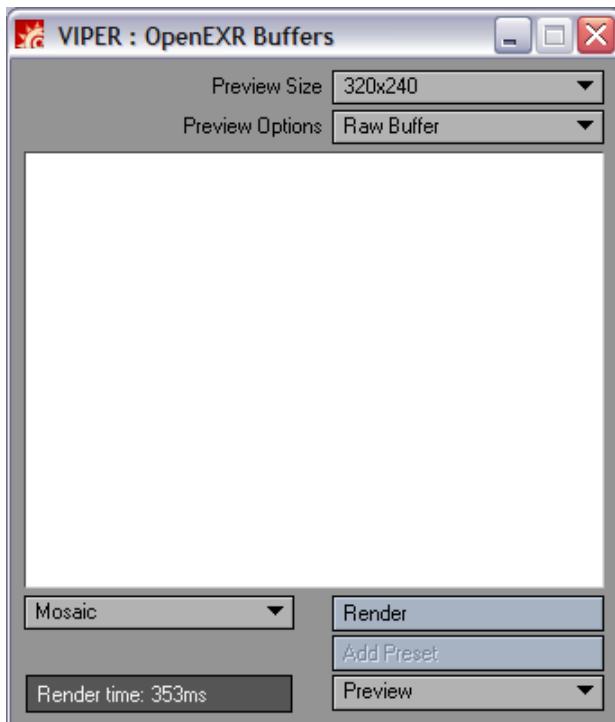
Preview Options

VIPER can either display the **Raw Buffer** (the buffer as rendered by Lightwave3D) or the **Processed Buffer** (as processed by exrTrader using the editing settings, and as finally saved to disk).

These can also be displayed as a **normalized** preview. In this case exrTrader will display all values into the buffer in the range between 0 and 1.0 (black and white), effectively making the complete range of values in a buffer visible.

Below: Raw Buffer display vs. normalized display of the Depth-Buffer of the Ninja Scene.





Pixel Inspector

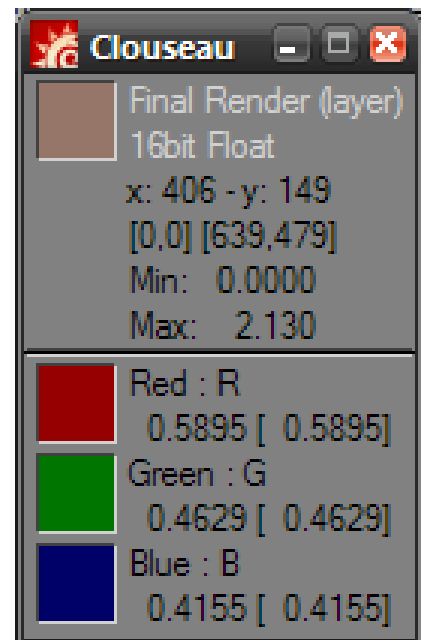
New Feature
Version 1.5

Clicking on the VIPER preview will open the Pixel Inspector as pictured here..

It displays some information about the pixel that has been clicked on in VIPER as well as the current buffer.

The Information displayed is:

- The name of the current layer and in brackets how it is being saved by exrTrader
- The coordinate of the pixel that was clicked on relative to the image size of the final rendered image.
- The computed data window for the current layer in pixel coordinates of the [top left] and [bottom right] corner.
- The minimum and maximum value present in the current buffer (these are the values that are used to normalize the VIPER preview as well).
- Values for the individual channels:
- The name of the channel and the name of the channel as saved in the exr image.
- The floating point value of the channel as displayed (this depends on the current Preview Options settings for VIPER) followed by the unmodified, raw value as generated by LightWave 3D.



The colour swatches display the current pixel or the current channel depending on the Preview Options settings for VIPER.

Please note that the pixel values are read from a scaled down version of the rendered image which is used for the VIPER preview as well. They are thus only approximations of the values as saved in the final images⁸.

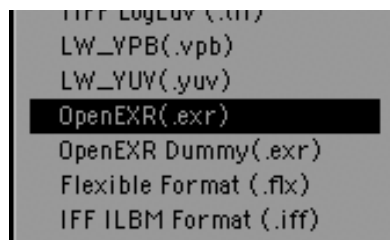
⁸ To phrase it differently: They are not 100% accurate.

Basic OpenEXR Loader and Saver Plugins

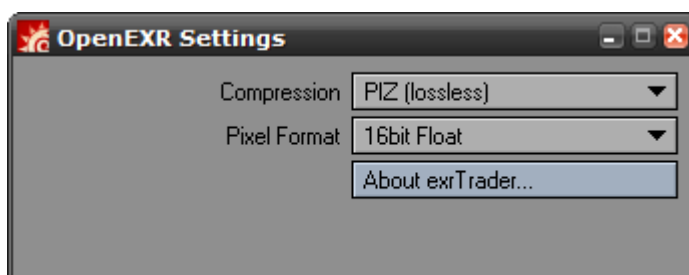
exrTrader includes basic LightWave image loader and saver plugins that allow you to load and save OpenEXR images in any part of LightWave that loads and saves images.

These are included in exrTrader for free, and may be used freely without registering exrTrader.

To change the settings for the OpenEXR Saver, we've included a Layout Master plugin that can be saved with any scene to change the saving options.



OpenEXR Settings



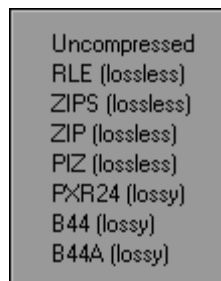
OpenEXR Settings is a Scene Master plugin that gets saved with a Lightwave3D scene.

It allows you to control the settings used by the OpenEXR saver, since a Lightwave3D image saver does not have a way to display an options panel.

The available settings are:

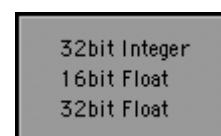
Compression

This defines the compression method used by the saver. Please refer to the previous section on this subject for details.



Pixel Format

The defines the pixel format for saving. Again, please refer to the previous Pixel Format documentation for details.



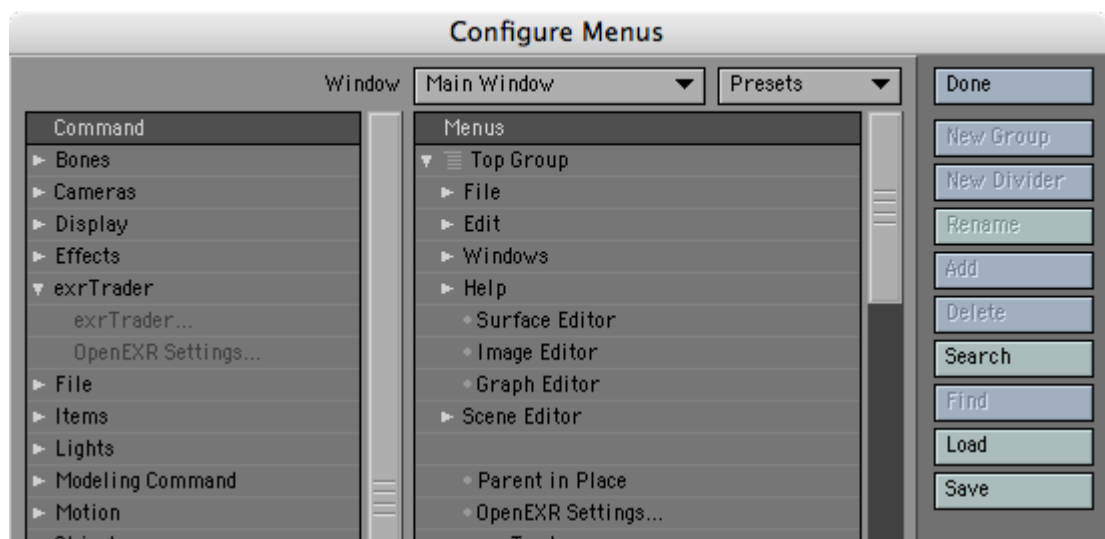
Special Plugins

OpenEXR Dummy Saver

The dummy saver has been designed to be used in conjunction with the exrTrader Buffer saver and third party render controllers that check render jobs for successfully rendered frames.

It doesn't save any image data, but allows a render controller to retrieve the output file path and extension from a Lightwave3D scene. If the scene only saves image data from the exrTrader Layered Export the render controller will still be able to check for rendered frames.

To use it, use the OpenEXR Dummy(.exr) image saver as the RGB Files saver in the render globals, make sure that "Use Render Path" is activated in exrTrader, and that the Format option of exrTrader matches the Format option in the Lightwave3D render globals.



exrTrader...

This is a Generic Layout Plugin that has been designed to be assigned to a short cut, or added to a menu.

It will first check if exrTrader Layered Export is assigned to the current scene as an Image Filter. If not, exrTrader will be added.

Then it will open the user interface of the exrTrader Layered Export attached to the current scene.

OpenEXR Settings...

This is a Generic Layout Plugin that has been designed to be assigned to a short cut, or added to a menu.

It will first check if OpenEXR Settings is assigned to the current scene as a Scene Master. If not, OpenEXR Settings will be added.

Then it will open the user interface of the OpenEXR Settings attached to the current scene.

Appendices

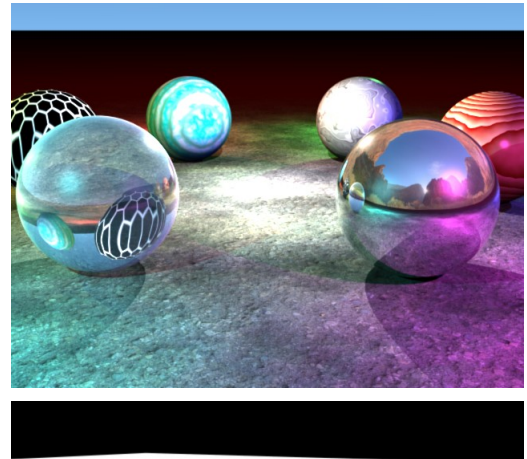
Available buffers

This is a list of all the buffers available in LightWave 3D.

ExrTrader V1.5 also supports the new buffers of LightWave HC as added up to the current released of exrTrader. Since LW HC is still in open beta, support for these buffers is considered to be experimental.

Final Render

This buffer contains the final, rendered image. This is identical to the image save by using the LightWave 3D render globals.



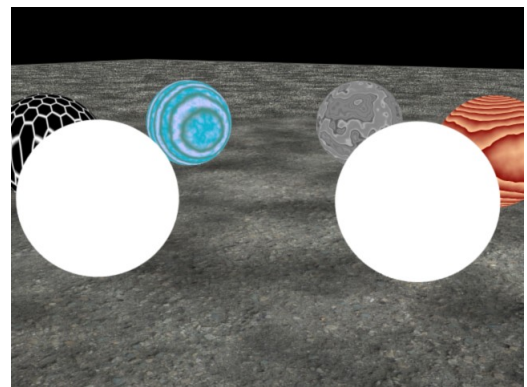
Alpha

This is the same as the alpha channel saved in images using the RGB output in the render globals.



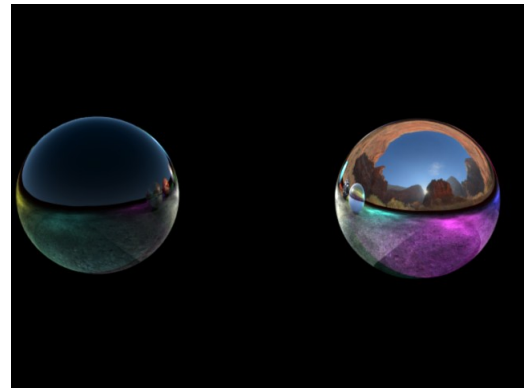
Raw Colour

This is the basic surface colour without any shading applied to it.

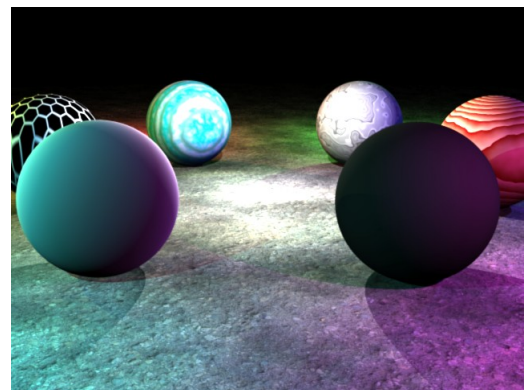


Reflection Colour

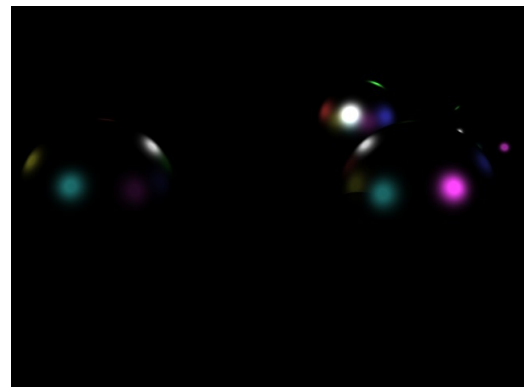
This colour buffers stores the reflections of surfaces.
It could also be referred to as reflection shading.

**Diffuse Colour**

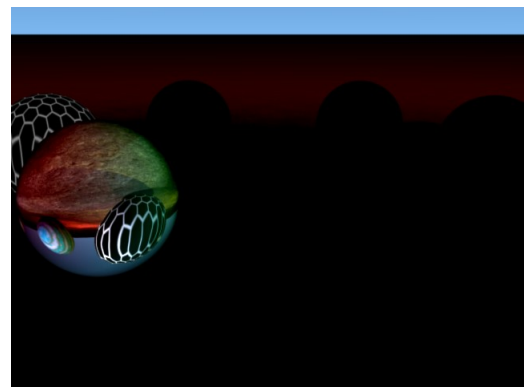
This buffers contains the shading due to the surfaces colour and the diffuse shading.

**Specular Colour**

This buffer stores the shading due to the specularity settings.

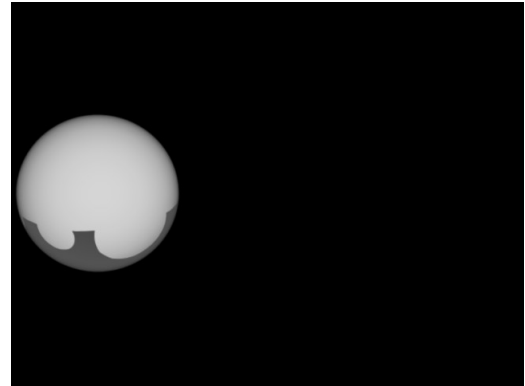
**Refraction Colour (LightWave 3D 9.6+)**

This buffer stored the refractions of surfaces... basically the shading due to transparency.



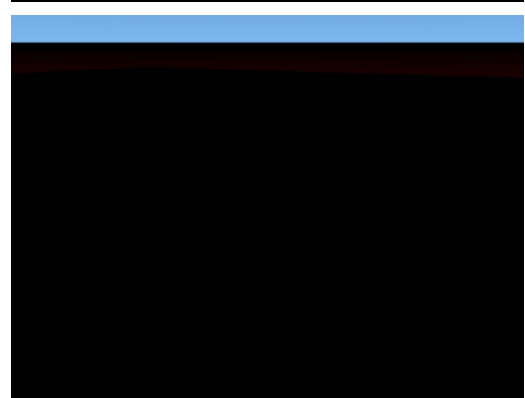
Refraction Alpha (LightWave 3D 9.6+)

This alpha channel represents the transparencies of the pixels stored in the refraction colour buffer.



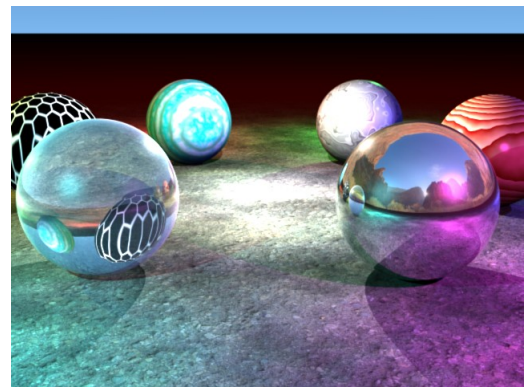
Backdrop Colour

This buffer just stores the backdrop colour.



Pre-Effect Colour

This is the final rendered image before any pixel filter or image filter plugins are applied to it.



Special

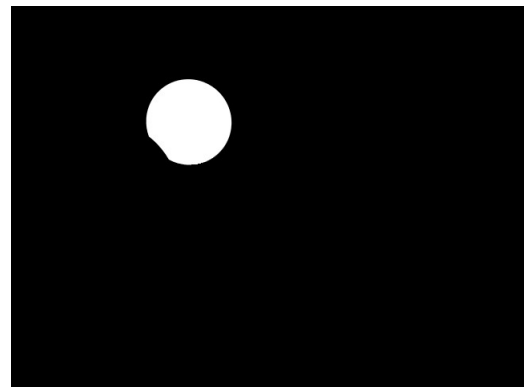
The special buffer is quite ... “special” indeed.

In the surface editor, under the **Advanced** tab, there is a button to edit the **Special Buffers**. LightWave 3D provides 4 of them. These are applied on a surface by surface basis.

A value of 0.0 in a special buffer is equivalent to black, 1.0 is white (values can go beyond those limit as they are stored in float/HDR buffers).

The tricky bit is the fact that image filters can only read one of these special buffers. Which one depends on the location of the image filter plugin in the list of applied image filters in the **Processing** tab of the **Effects** panel in LightWave 3D.

The first image filter in the list will only be able to read special buffer #1, the second image filter can only read special buffer #2, the third one buffer #3 and the fourth one only buffer #4. All image filters beyond



that only have access to special buffer #4.

Practically this means that exrTrader will need to be applied four times as an image filter to the list to be able to access all four special buffers. In this case we'd recommend using the first instance of exrTrader in the list to save all the other buffers as well, and only use the slots 2-4 to save the special buffers – to separate image files.

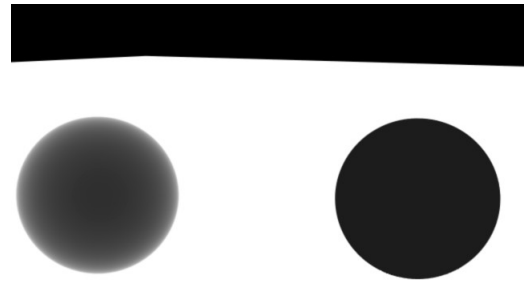
Luminosity

This greyscale buffer stores the Luminosity value of a surface.



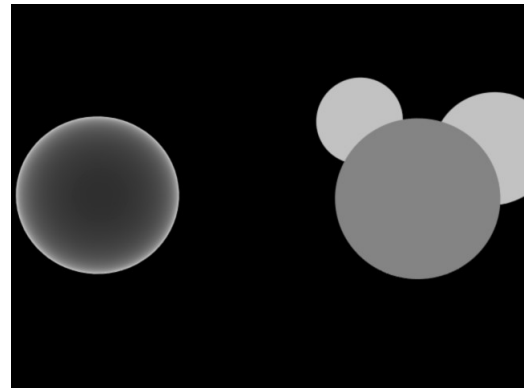
Diffuse

This greyscale buffer stores the diffuse value of the surfaces, as defined in the surface editor.



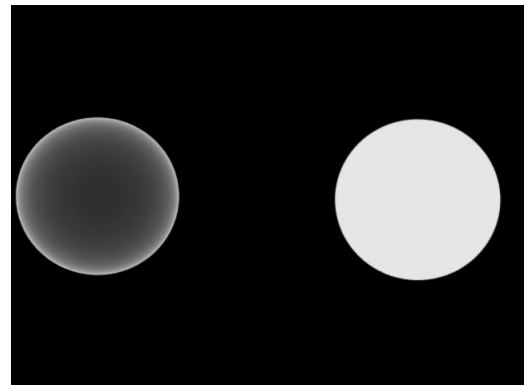
Specularity

This greyscale buffer stores the specularity value of the surfaces, as defined in the surface editor.



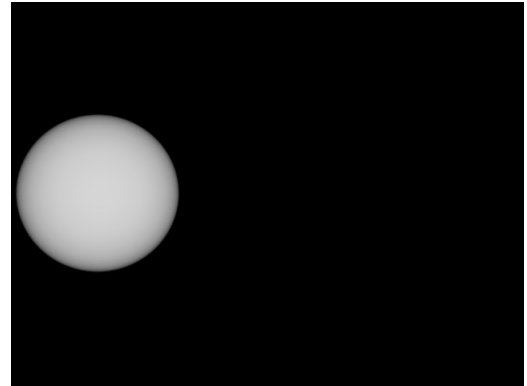
Reflectivity

This greyscale buffer stores the reflectivity value of the surfaces, as defined in the surface editor.



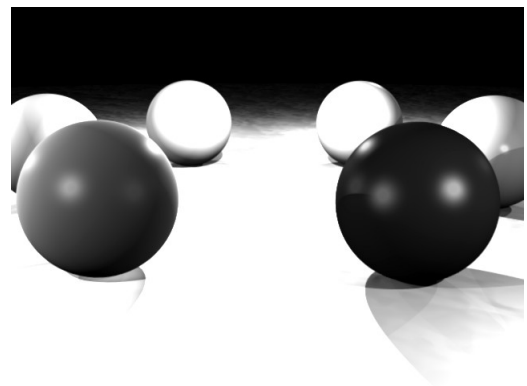
Transparency

This greyscale buffer stores the transparency value of the surfaces, as defined in the surface editor.



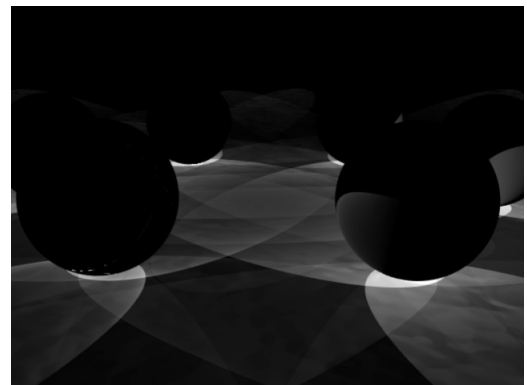
Shading

This buffer contains the shading due to the diffuse and specular values as a greyscale image.



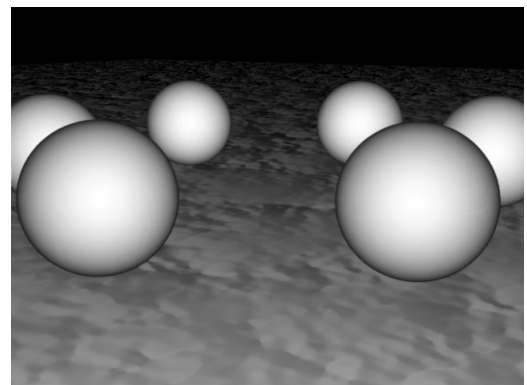
Shadow

This buffer highlights the areas of the image that receive shadows from lights.



Geometry

This buffer stores the orientation of surfaces toward the camera. Facing surfaces are white while surfaces that face away from the camera are black.

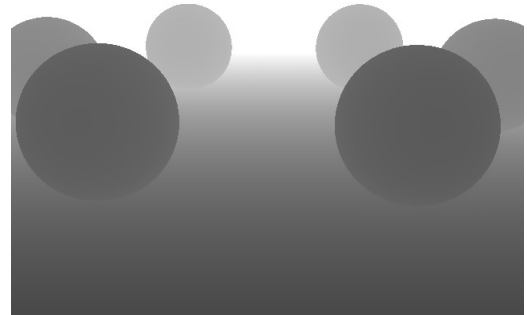


Depth

This buffer stores the distance of a surface from the camera.

Due to the fact that the depth is stored as a proper float image and the pixels represent the distance to the camera in metres, there's a few facts to remember.

The first image to the right shows the normalized depth buffer for the sample render. There isn't much to see because the top of the image is the backdrop. In LightWave the backdrop is always rendered at an infinite distance⁹.



A normalized view looks at the highest and the lowest value in an image and then tries to display anything between them in the visible range from 0.0 to 1.0. In this case the backdrop compressed to be at 1.0, and all other values are compressed respectively. This means that the actual items are relatively close to the camera (at least in relation to the backdrop) and thus displayed as being black (which corresponds of a distance of 0, or something very close to it).



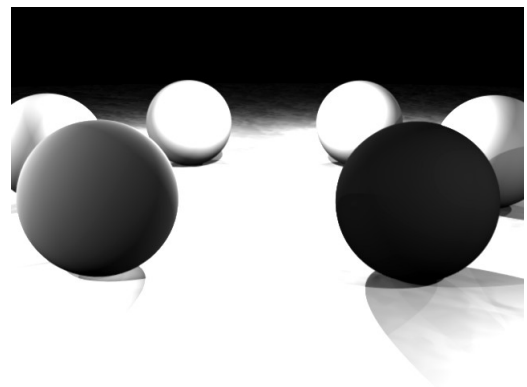
If you intend to composite in an application that handles float images (and the tools that are used to actually work with a float depth buffer) then there should be no need to change anything.

If you intend to save the depth buffer as a low dynamic range image then you can use the Minimum and Maximum settings to define a range which should be normalized into the visible range.

The second image illustrates this, it is the same depth buffer as in the first image, but the Maximum has been set to 10.0.

DiffShade

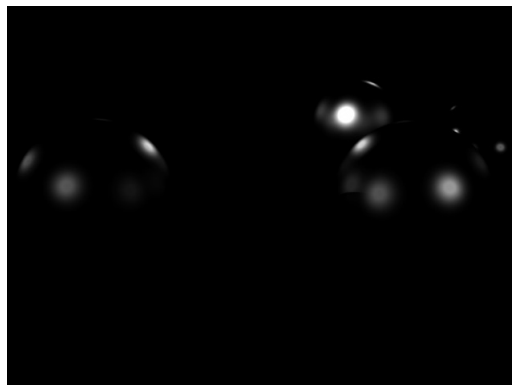
This buffer is quite similar to the Shading buffer. It stores the diffuse shading of surfaces due to their Diffuse property and the lighting.



⁹ Well, not really infinite... but very, very, very far away,

SpecShade

This buffer is quite similar to the Shading buffer or the previous Diffshade buffer. It stores the specular shading of surfaces due to their Specular property and the lighting.



Motion

This buffer stores the motion of a pixel during the current frame in screen space. The motion is encoded in the R and B channel.

Since LightWave 3D creates float channels, the values represent the movement in pixels. This implies that the values may actually be negative as well, depending on the direction of movement.

If the compositing package can deal with floating point motion buffers this is the best way to export them.

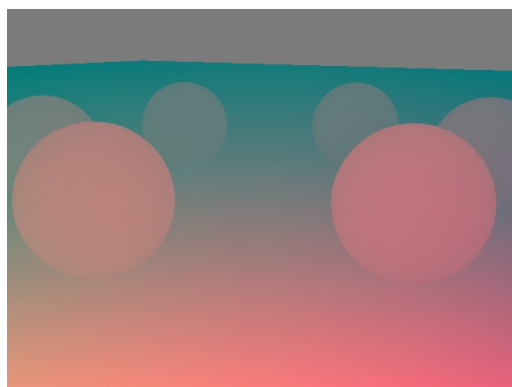
Some plugins require the motion vectors to be normalized into values from 0.0 to 1.0, where 0.5 is equivalent to no motion at all.

The **Offset** and **Scale** processing options allow you to change the motion buffer to be acceptable by such plugins¹⁰ - as a downside one needs to estimate how far the any of the pixels actually travel (in pixels) - or use an arbitrary value, such as the largest dimension of the image in pixels.

As an example, using 1920 as the largest distance an pixel could travel:

The **Offset** would need to be 1920 - to basically push the negative values into the positive range.

The **Scale** would need to be $1 / (1920 * 2)$ (roughly 0.00026) - to compress the values into a range from 0 to 1.0.



¹⁰ However, you may be able to edit it in your compositing package prior to sending it to those plugins as well. This would be a little less dangerous and more flexible.

Network Rendering

exrTrader 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.

The OpenEXR Dummy Saver has been designed to allow render controllers to check for saved frames from rendered scenes, even if only the exrTrader Layered Export is actually writing frames.

Third party LightWave 3D plugins with exrTrader support

The following plugins for LightWave 3D support exrTrader natively:

ButterFlyNetRender ButterFlyNetRender is a network render controller.

<http://www.liquiddreamsolutions.com/>

Janus Janus is a multi-pass rendering solution for LightWave3D. It allows you to quickly layer your objects and render them separately.

<http://janus.faulknermano.com/>

PassPort PassPort is a powerful tool that allows you to have one set of beauty objects and one scene, and generate unlimited* outputs with all your data passes and scene layers, all with minimal set-up.

<http://www.lwpassport.com/>

OpenEXR support in third party applications

Here is a short list of third party applications, mainly compositing applications, and a summary of their ability to read OpenEXR:

Digital Fusion 4 Reads RGBA only.

Fusion 5-6 Reads RGBA and "deep" channels, R, G, B, A, Z, pixelCover, objectID, materialID, U, V, NX, NY, NZ, velX, velY
Any channel of the file may be assigned to any of the deep channels used by Fusion 5, a channel may only be assigned once in version 5.01 or earlier of Fusion.

Shake 4 As far as we're aware it only reads RGBA, but allows the user to define which channels of the source file to use

After Effects 7-CS3 Basic RGBA support
Advanced support for OpenEXR is available from
<http://www.fnordware.com/ProEXR/>

After Effects CS4 Includes advanced support for OpenEXR due to the inclusion of plugins from
<http://www.fnordware.com/>

Combustion 4 Basic RGBA support

Toxic Only handles RGBA buffers, but allows the user to pick which OpenEXR buffers are loaded into the RGBA channels.

Nuke Extended OpenEXR support, reads up to 64 arbitrary channels of an OpenEXR file, supports layers/grouped channels.
Single channel buffers should be saved using "A" as the channel name, Nuke will then interpret them as masks and read them properly. You can use the layer name to differentiate between different buffers.

New Feature

Version 1.4.1

PhotoShop CS4 Uses the same loader as After Effects 7, so basic RGBA support only.
Advanced support for OpenEXR is available from
<http://www.fnordware.com/ProEXR/>

LScript

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

Generic_Open_exrTrader – Applies the exrTrader Image Filter (if needed) and opens the user interface

Generic_Open_OpenEXRSettings - Applies the OpenEXR Settings Master Handler (if needed) and opens the user interface

Recommended Reading

The OpenEXR website at www.openexr.com is packed with information about the OpenEXR image file format.

We recommend downloading the “Technical Introduction to OpenEXR” PDF and at least having a quick look at it. We believe it explains some of the reasoning behind the design of exrTrader, and also give a great introduction to the capabilities of the image file format.

Supported Metadata

LightWave 3D can read and write metadata embedded in images as of version 9.5

exrTrader supports all of the metadata provided by LightWave 3D when using the exrTrader image loader and saver, the exrTrader image filter supports a selection of metadata.

The metadata written by the exrTrader image saver is provided by LightWave 3D. The metadata written by the image filter is generated by exrTrader.

The following two tables list the metadata tags as documented in the LightWave 3D SDK, their content and the equivalent attribute as saved and loaded by exrTrader.

The metadata supported by LightWave mirrors EXIF tags which have been designed for digital photography. Tags that don't apply to OpenEXR images have been omitted.

Basic Metadata

LightWave metadata tag	Description	OpenEXR Attribute	Written by exrTrader image filter
LWIMPAR_ASPECT	x / y Pixel Aspect.	native	✓
LWIMPAR_PIXELWIDTH	Actual (scanned)Pixel Width in (mm).	xDensity	✓
LWIMPAR_FRAMESPERSECOND	Number Of Frames Per Second.	FramesPerSecond	✓
LWIMPAR_BLACKPOINT	Black Point Of Layer.		
LWIMPAR_WHITEPOINT	White Point Of Layer.	WhiteLuminance	✓
LWIMPAR_GAMMA	Linearity Of RGB Color.	always 1 ¹¹	
LWIMPAR_PIXELXDIMENSION	Valid image width.	native	✓
LWIMPAR_PIXELYDIMENSION	Valid image height.	native	✓
LWIMPAR_EXPOSURETIME	Exposure time (reciprocal of shutter speed). Unit is seconds.	ExpTime	✓
LWIMPAR_FNUMBER	The actual F-number(F-stop) of lens when the image was taken.	Aperture	✓

¹¹ According to the specs, OpenEXR stores linear images only-

LightWave metadata tag	Description	OpenEXR Attribute	Written by exrTrader image filter
LWIMPAR_ISOSPEEDRATINGS	CCD sensitivity equivalent to Ag-Hr film speedrate. (int)	IsoSpeed	✓
LWIMPAR_DATETIMEORIGINAL	Date/Time of original image taken. This value should not be modified by user program.	CapDate	✓
LWIMPAR_FOCALLENGTH	Focal length of lens used to take image. Unit is millimeter.	FocalLength	✓
LWIMPAR_MAKERNOTE	Maker dependent internal data. Some of maker such as Olympus/Nikon/Sanyo etc. uses IFD format for this area.	See table of Maker Metadata	✓
LWIMPAR_USERCOMMENT	Stores user comment.	Comment	✓ editable
LWIMPAR_APERTUREVALUE	The actual aperture value of lens when the image was taken. To convert this value to ordinary F- number (F-stop), calculate this value's power of root 2 (=1.4142). For example, if value is '5', F-number is $1.4142^5 = F5.6$.	Aperture	✓
LWIMPAR_SUBJECTDISTANCE	Distance to focus point, unit is meter.	Focus	✓
LWIMPAR_MAKE	Manufacturer of recording equipment.	Make	✓
LWIMPAR_MODEL	Model name or model number of recording equipment.	Model	✓
LWIMPAR_SOFTWARE	Name and version of the software used.	Software	✓
n/a	Name of the computer that has written the image ¹²	ComputerName	✓
n/a	Timezone offset for the capture date ¹³	UtcOffset	✓
n/a	Username	Owner	✓

¹² Not available on OSX

¹³ Unfortunately also not available on OSX

Maker Metadata

LightWave metadata tag	Description	OpenEXR Attribute	Written by exrTrader image filter
LWMN_COMPANYNAME	Company Name	LW.Company	✓
LWMN_FRAMENUMBER	Current frame number	LW.FrameNumber	✓
LWMN_FIRSTFRAME	First frame number	LW.FirstFrame	✓
LWMN_LASTFRAME	Last frame number	LW.LastFrame	✓
LWMN_RENDERTIME	Render time	LW.Rendertime	✓
LWMN_CAMERANAME	Camera Name	LW.CameraName	✓
LWMN_CAMERATYPE	Camera Type	LW.CameraType	✓
LWMN_ANTIALIASING	Antialiasing	LW.AA	✓
LWMN_LENS	Camera lens name	LW.Lens	
LWMN_PARTICLEBLUR	Particle blur On/Off	LW.ParticleBlur	✓
LWMN_MOTIONBLURTYPE	Motion blur type	LW.MBType	✓
LWMN_MOTIONBLURAMOUNT	Motion blur amount	LW.MBAmount	✓
LWMN_MOTIONBLURPASSES	Motion blur passes.	LW.MBPasses	✓
LWMN_TOTALPOINTS	Total points	LW.TotalPoints	✓
LWMN_TOTALPOLYGONS	Total polygons	LW.TotalPolygons	✓
LWMN_TOTALMEMORY	Total memory	LW.TotalMemory	
LWMN_FRAMESTEP	Frame step	LW.Framestep	✓
LWMN_RECONSTRUCTIONFILTER	Reconstruction filter	LW.AAFilter	✓
LWMN_FIELDRENDERING	Field rendering	LW.FieldRendering	✓
LWMN_RTREFLECTIONS	RT reflections	LW.RTReflections	✓
LWMN_ADAPTIVESAMPLING	Adaptive sampling	LW.AdaptiveSampling	✓
LWMN_STEREORENDER	Stereo render.	LW.Stereo	✓

LightWave metadata tag	Description	OpenEXR Attribute	Written by exrTrader image filter
LWMN_RTREFRACTIONS	RT refractions	LW.RTRefractions	✓
LWMN_RADIOSITY	Radiosity	LW.Radiosity	✓
LWMN_RTTRANSPARENCY	RT transparency	LW.RTTtransparency	✓
LWMN_CAUSTICS	Caustics	LW.Caustics	✓
LWMN_RENDERTHREADS	Render threads	LW.RenderThreads	✓
LWMN_DEPTHOFFIELD	Depth of field	LW.DepthOfField	✓
LWMN_OUTPUTFILES	Output files	LW.OutputFiles	